Provider panos

PAN-OS® is the operating system for Palo Alto Networks® NGFWs and Panorama™. The panos provider allows you to manage various aspects of a firewall's or a Panorama's config, such as data interfaces and security policies.

Use the navigation to the left to read about the available Panorama and NGFW resources.

Versioning

The panos provider has support for PAN-OS 6.1 - 9.0.

Some resources may contain variables that are only applicable for newer versions of PAN-OS. If this is the case, then make sure to use conditionals (https://www.terraform.io/docs/configuration/expressions.html#conditional-expressions) along with the `panos_system_info` data source to only set these variables when the version of PAN-OS is appropriate.

One such resource is `panos_ethernet_interface` and the `ipv4_mss_adjust` parameter. Doing the following is one way to correctly configure this parameter only when it's applicable:

```
data "panos_system_info" "config" {}

data "panos_ethernet_interface" "eth1" {
    name = "ethernet1/1"
    vsys = "vsys1"
    mode = "layer3"
    adjust_tcp_mss = true
    ipv4_mss_adjust = "${data.panos_system_info.config.version_major >= 8 ? 42 : 0}"
    # ...
}
```

Commits

As of right now, Terraform does not provide native support for commits, so commits are handled out-of-band. Please use the following for commits:

```
package main

import (
    "encoding/json"
    "flag"
    "log"
    "os"

    "github.com/PaloAltoNetworks/pango"
)

type Credentials struct {
    Hostname string `json:"hostname"`
    Username string `json:"username"`
    Password string `json:"password"`
    ApiKey string `json:"api_key"`
```

```
    Protocol string `json:"protocol"`
    Port uint `json:"port"`
    Timeout int `json:"timeout"`
}

func getCredentials(configFile, hostname, username, password, apiKey string) (Credentials) {
    var (
        config Credentials
        val string
        ok bool
    )

    // Auth from the config file.
    if configFile != "" {
        fd, err := os.Open(configFile)
        if err != nil {
            log.Fatalf("ERROR: %s", err)
        }
        defer fd.Close()

        dec := json.NewDecoder(fd)
        err = dec.Decode(&config)
        if err != nil {
            log.Fatalf("ERROR: %s", err)
        }
    }

    // Auth from env variables.
    if val, ok = os.LookupEnv("PANOS_HOSTNAME"); ok {
        config.Hostname = val
    }
    if val, ok = os.LookupEnv("PANOS_USERNAME"); ok {
        config.Username = val
    }
    if val, ok = os.LookupEnv("PANOS_PASSWORD"); ok {
        config.Password = val
    }
    if val, ok = os.LookupEnv("PANOS_API_KEY"); ok {
        config.ApiKey = val
    }

    // Auth from CLI args.
    if hostname != "" {
        config.Hostname = hostname
    }
    if username != "" {
        config.Username = username
    }
    if password != "" {
        config.Password = password
    }
    if apiKey != "" {
        config.ApiKey = apiKey
    }

    if config.Hostname == "" {
        log.Fatalf("ERROR: No hostname specified")
    } else if config.Username == "" && config.ApiKey == "" {
        log.Fatalf("ERROR: No username specified")
    } else if config.Password == "" && config.ApiKey == "" {
```

```
    } else if config.Password ==    && config.ApiKey ==    {
        log.Fatalf("ERROR: No password specified")
    }

    return config
}

func main() {
    var (
        err error
        configFile, hostname, username, password, apiKey string
        job uint
    )

    log.SetFlags(log.Ldate | log.Ltime | log.Lmicroseconds)

    flag.StringVar(&configFile, "config", "", "JSON config file with panos connection info")
    flag.StringVar(&hostname, "host", "", "PAN-OS hostname")
    flag.StringVar(&username, "user", "", "PAN-OS username")
    flag.StringVar(&password, "pass", "", "PAN-OS password")
    flag.StringVar(&apiKey, "key", "", "PAN-OS API key")
    flag.Parse()

    config := getCredentials(configFile, hostname, username, password, apiKey)

    fw := &pango.Firewall{Client: pango.Client{
        Hostname: config.Hostname,
        Username: config.Username,
        Password: config.Password,
        ApiKey: config.ApiKey,
        Protocol: config.Protocol,
        Port: config.Port,
        Timeout: config.Timeout,
        Logging: pango.LogOp | pango.LogAction,
    }}
    if err = fw.Initialize(); err != nil {
        log.Fatalf("Failed: %s", err)
    }

    job, err = fw.Commit(flag.Arg(0), nil, true, true, false, true)
    if err != nil {
        log.Fatalf("Error in commit: %s", err)
    } else if job == 0 {
        log.Printf("No commit needed")
    } else {
        log.Printf("Committed config successfully")
    }
}
```

Compile the above, put it somewhere in your `$PATH` (such as `$HOME/bin`), then invoke it after `terraform apply` and `terraform destroy`:

```
$ go get github.com/PaloAltoNetworks/pango
$ go build commit.go
$ mv commit ~/bin
$ terraform apply && commit -config fwauth.json 'My commit comment'
```

The first trailing CLI arg is assumed to be the commit comment. If there is no CLI arg present then no commit comment is given to PAN-OS.

The authentication credentials can be given multiple ways, and if all are present then this is the order, from lowest to highest priority:

1. JSON authentication credential file
2. Environment variables
3. CLI arguments (WARNING: this is insecure)

See the argument reference section below for more info on the JSON config file and supported environment variables.

PAN-OS API Key

API connections to PAN-OS require an API key (https://www.paloaltonetworks.com/documentation/71/pan-os/xml-api/get-started-with-the-pan-os-xml-api/get-your-api-key). If you do not provide the API key to the panos provider, then the API key is generated before every single API call. Thus, some slight speed gains can be realized in the panos provider by specifying the API key instead of the username/password combo. The following may be used to generate the API key:

```
package main

import (
    "fmt"
    "os"

    "github.com/PaloAltoNetworks/pango"
)

func main() {
    var (
        hostname, username, password string
        ok bool
    )

    if hostname, ok = os.LookupEnv("PANOS_HOSTNAME"); !ok {
        os.Stderr.WriteString("PANOS_HOSTNAME must be set\n")
        return
    }
    if username, ok = os.LookupEnv("PANOS_USERNAME"); !ok {
        os.Stderr.WriteString("PANOS_USERNAME must be set\n")
        return
    }
    if password, ok = os.LookupEnv("PANOS_PASSWORD"); !ok {
        os.Stderr.WriteString("PANOS_PASSWORD must be set\n")
        return
    }

    fw := &pango.Firewall{Client: pango.Client{
        Hostname: hostname,
        Username: username,
        Password: password,
        Logging: pango.LogQuiet,
    }}
    if err := fw.Initialize(); err != nil {
        os.Stderr.WriteString(fmt.Sprintf("Failed initialize: %s\n", err))
        return
    }
    os.Stdout.WriteString(fmt.Sprintf("%s\n", fw.ApiKey))
}
```

Then execute it like this:

```
$ go get github.com/PaloAltoNetworks/pango
$ go run make_api_key.go
```

The API key is output to stdout, but you can redirect this to a file using normal shell redirection if desired:

```
$ go run make_api_key.go > my_api_key.txt
```

Connection information for the above is expected to be set as environment variables:

- PANOS_HOSTNAME

- PANOS_USERNAME
- PANOS_PASSWORD

AWS / GCP Considerations

There are a few types (https://aws.amazon.com/marketplace/seller-profile?id=0ed48363-5064-4d47-b41b-a53f7c937314) of PAN-OS VMs available to bring up in AWS. Both these VMs as well as the ones that can be deployed in Google Cloud Platform are different in that the `admin` password is unset, but it has an SSH key associated with it. As the panos Terraform provider package authenticates via username/password, an initialization step of configuring a password using the given SSH key is required. Right now, this initialization step requires manual intervention; the user must download this SSH key, at which point the following may be used to automate this initialization:

```
package main

import (
    "fmt"
    "io"
    "io/ioutil"
    "os"
    "regexp"
    "strings"
    "time"

    "golang.org/x/crypto/ssh"
)

// Various prompts.
var (
    P1 *regexp.Regexp
    P2 *regexp.Regexp
    P3 *regexp.Regexp
)

func init() {
    P1 = regexp.MustCompile(`[a-zA-Z][a-zA-Z0-9\._\-]+@[a-zA-Z][a-zA-Z0-9\._\-]+> `)
    P2 = regexp.MustCompile(`[a-zA-Z][a-zA-Z0-9\._\-]+@[a-zA-Z][a-zA-Z0-9\._\-]+# `)
    P3 = regexp.MustCompile(`(Enter|Confirm) password\s+:\s+?`)
}

// Globals to handle I/O.
var (
    stdin io.Writer
    stdout io.Reader
    buf [65 * 1024]byte
)

// ReadTo reads from stdout until the desired prompt is encountered.
func ReadTo(prompt *regexp.Regexp) (string, error) {
    var i int

    for {
        n, err := stdout.Read(buf[i:])
        if n > 0 {
            os.Stdout.Write(buf[i:i + n])
```

```
        }
        if err != nil {
            return "", err
        }
        i += n
        if prompt.Find(buf[:i]) != nil {
            return string(buf[:i]), nil
        }
    }
}

// Perform user initialization.
func panosInit() error {
    var err error

    // Load environment variables.
    hostname := os.Getenv("PANOS_HOSTNAME")
    username := os.Getenv("PANOS_USERNAME")
    password := os.Getenv("PANOS_PASSWORD")

    // Sanity check input.
    if len(os.Args) == 1 || os.Args[1] == "-h" || os.Args[1] == "--help" || hostname == "" || username =
= "" || password == "" {
        u := []string{
            fmt.Sprintf("Usage: %s <key_file>", os.Args[0]),
            "",
            "This will connect to a PAN-OS NGFW and perform initial config:",
            "",
            " * Adds the user as a superuser (if not the admin user)",
            " * Sets the user's password",
            " * Commit",
            "",
            "The following environment variables are required:",
            "",
            " * PANOS_HOSTNAME",
            " * PANOS_USERNAME",
            " * PANOS_PASSWORD",
        }
        for i := range u {
            fmt.Printf("%s\n", u[i])
        }
        os.Exit(0)
    }

    // Read in the ssh key file.
    data, err := ioutil.ReadFile(os.Args[1])
    if err != nil {
        return fmt.Errorf("Failed to read SSH key file %q: %s", os.Args[1], err)
    }

    signer, err := ssh.ParsePrivateKey(data)
    if err != nil {
        return fmt.Errorf("Failed to parse private key: %s", err)
    }

    useSshKey := ssh.PublicKeys(signer)

    // Configure and open the ssh connection.
    config := &ssh.ClientConfig{
        User: "admin",
```

```
        User: "admin",
        Auth: []ssh.AuthMethod{
            useSshKey,
        },
        HostKeyCallback: ssh.InsecureIgnoreHostKey(),
    }

    client, err := ssh.Dial("tcp", fmt.Sprintf("%s:22", hostname), config)
    if err != nil {
        return fmt.Errorf("Failed dial: %s", err)
    }
    defer client.Close()

    session, err := client.NewSession()
    if err != nil {
        return fmt.Errorf("Failed to create session: %s", err)
    }
    defer session.Close()

    modes := ssh.TerminalModes{
        ssh.ECHO: 0,
        ssh.TTY_OP_ISPEED: 14400,
        ssh.TTY_OP_OSPEED: 14400,
    }

    if err = session.RequestPty("vt100", 80, 80, modes); err != nil {
        return fmt.Errorf("pty request failed: %s", err)
    }

    // Get input/output pipes for the ssh connection.
    stdin, err = session.StdinPipe()
    if err != nil {
        return fmt.Errorf("setup stdin err: %s", err)
    }

    stdout, err = session.StdoutPipe()
    if err != nil {
        return fmt.Errorf("setup stdout err: %s", err)
    }

    // Invoke a shell on the remote host.
    if err = session.Start("/bin/sh"); err != nil {
        return fmt.Errorf("failed session.Start: %s", err)
    }

    // Perform initial config.
    ok := true
    commands := []struct{
        Send string
        Expect *regexp.Regexp
        Validation string
        OmitIfAdmin bool
    }{
        {"", P1, "", false},
        {"set cli pager off", P1, "", false},
        {"show system info", P1, "", false},
        {"configure", P2, "", false},
        {fmt.Sprintf("set mgt-config users %s permissions role-based superuser yes", username), P2, "", t
rue},
        {fmt.Sprintf("set mgt-config users %s password", username), P3, "", false},
```

```
        {password, P3, "", false},
        {password, P2, "", false},
        {"commit description 'initial config'", P2, "Configuration committed successfully", false},
        {"exit", P1, "", false},
        {"exit", nil, "", false},
    }

    for _, cmd := range commands {
        if cmd.OmitIfAdmin && username == "admin" {
            continue
        }
        if cmd.Send != "" {
            stdin.Write([]byte(cmd.Send + "\n"))
        }
        if cmd.Expect != nil {
            out, err := ReadTo(cmd.Expect)
            if err != nil {
                return fmt.Errorf("Error in %q: %s", cmd.Send, err)
            }
            if cmd.Validation != "" {
                ok = ok && strings.Contains(out, cmd.Validation)
            }
            // Delay slightly before sending passwords.
            if cmd.Expect == P3 {
                time.Sleep(1 * time.Second)
            }
        } else {
            fmt.Printf("exit\n")
            session.Wait()
        }
    }

    // Completed successfully.
    return nil
}

func main() {
    if err := panosInit(); err != nil {
        fmt.Printf("\nFailed initial config: %s\n", err)
        os.Exit(1)
    }
    fmt.Printf("\nConfig initialization successful")
}
```

Compile the above, put it somewhere in your `$PATH` (such as `$HOME/bin`), then invoke it after the device is accessible in AWS:

```
$ go get golang.org/x/crypto/ssh
$ go build panos_init.go
$ mv panos_init ~/bin
$ panos_init my_ssh_key.pem
```

The API key is expected to be given as the first param, while the hostname is retrieved from the following environment variable:

- `PANOS_HOSTNAME`

The username and password are expected to be in the following environment variables:

- `PANOS_USERNAME`
- `PANOS_PASSWORD`

If `PANOS_USERNAME` is set to `admin`, then the above will skip the step that creates the account, as the `admin` account already exists.

Importing Resources

Many resources support being imported. Any resource that supports `terraform import` will have a "Import Name" section in the documentation. The variables given in this section directly match up with the resource params you would specify in your plan file. Thus if you were importing an ethernet interface whose import name is `<vsys>:<name>`, your import name would be something like `vsys1:ethernet1/1`.

Of special note is the Panorama resources. The templated resources often have both the template and the template stack in the resource name, however only one of these can ever be present. Thus, the one that isn't being used should just be an empty string. For example, if you were trying to import a Panorama IPv4 static route whose import name is `<template>:<template_stack>:<virtual_router>:<name>` that resides in a template, your import name would be something like `myTemplate::myVirtualRouter:myStaticRouteName`.

Example Provider Usage

```
# Configure the panos provider
provider "panos" {
    hostname = "127.0.0.1"
    username = "admin"
    password = "secret"
}

# Add a new zone to the firewall
resource "panos_zone" "zone1" {
    # ...
}
```

Argument Reference

The following arguments are supported:

- `hostname` - (Optional) This is the hostname / IP address of the firewall. It must be provided, but can also be defined via the `PANOS_HOSTNAME` environment variable.
- `username` - (Optional) The username to authenticate to the firewall as. It must be provided, but can also be defined via the `PANOS_USERNAME` environment variable.
- `password` - (Optional) The password for the given username. It must be provided, but can also be defined via the `PANOS_PASSWORD` environment variable.

- `api_key` - (Optional) The API key for the firewall. If this is given, then the `username` and `password` settings are ignored. This can also be defined via the `PANOS_API_KEY` environment variable.
- `protocol` - (Optional) The communication protocol. This can be set to either `https` or `http`. If left unspecified, this defaults to `https`.
- `port` - (Optional) If the port number is non-standard for the desired protocol, then the port number to use.
- `timeout` - (Optional) The timeout for all communications with the firewall. If left unspecified, this will be set to 10 seconds.
- `logging` - (Optional) List of logging options for the provider's connection to the API. If this is unspecified, then it defaults to `["action", "uid"]`.
- `json_config_file` - (Optional) The path to a JSON configuration file that contains any number of the provider's parameters. If specified, the params present act as a last resort for any other provider param that has not been specified yet.

The list of strings supported for `logging` are as follows:

- `quiet` - Disables logging. This is ignored, however, if other logging flags are present.
- `action` - Log `set / edit / delete`.
- `query` - Log `get`.
- `op` - Log `op`.
- `uid` - Log user-id envocations.
- `xpath` - Log the XPATH associated with various actions.
- `send` - Log the raw request sent to the device. This is probably only useful in development of the provider itself.
- `receive` - Log the raw response sent back from the device. This is probably only useful in development of the provider itself.

Support

This template/solution are released under an as-is, best effort, support policy. These scripts should be seen as community supported and Palo Alto Networks will contribute our expertise as and when possible. We do not provide technical support or help in using or troubleshooting the components of the project through our normal support options such as Palo Alto Networks support teams, or ASC (Authorized Support Centers) partners and backline support options. The underlying product used (the VM-Series firewall) by the scripts or templates are still supported, but the support is only for the product functionality and not for help in deploying or using the template or script itself. Unless explicitly tagged, all projects or work posted in our GitHub repository (at https://github.com/PaloAltoNetworks (https://github.com/PaloAltoNetworks)) or sites other than our official Downloads page on https://support.paloaltonetworks.com (https://support.paloaltonetworks.com) are provided under the best effort policy.

panos_dhcp_interface_info

Use this data source to retrieve DHCP client information about the given firewall interface.

Example Usage

```
data "panos_dhcp_interface_info" "example" {
    interface = "ethernet1/1"
}

output "eth1_ip" {
    value = "${data.panos_dhcp_interface_info.example.ip}"
}
```

Attribute Reference

The following attributes are present:

- `interface` - (Required) The data interface to get DHCP information for.

These attributes are exported once the data source refreshes:

- `state` - The interface's state.
- `ip` - DHCP IP address.
- `gateway` - The default gateway assigned.
- `server` - The DHCP server IP
- `server_id` - DHCP server ID
- `primary_dns` - Primary DNS server
- `secondary_dns` - Secondary DNS server
- `primary_wins` - Primary WINS server
- `secondary_wins` - Secondary WINS
- `primary_nis` - Primary NIS
- `secondary_nis` - Secondary NIS
- `primary_ntp` - Primary NTP
- `secondary_ntp` - Secondary NTP
- `pop3_server` - POP3 Server
- `smtp_server` - SMTP Server
- `dns_suffix` - DNS Suffix

panos_plugin_info

Use this data source to retrieve "show system info" from the NGFW or Panorama.

All contents of "show system info" are saved to the `info` variable. In addition, the version number of PAN-OS encountered is saved to multiple fields for ease of access.

Example Usage

```
data "panos_panorama_plugin" "example" {}
```

Attribute Reference

The following attributes are present:

- `installed` - A list of installed plugins.
- `total` - (int) Total number of plugins, installed or not.
- `details` - A list of maps (see below).

The following attributes are present in each `details` entry:

- `name` - The name.
- `version` - The version number.
- `release_date` - Release date.
- `release_note_url` - Release note URL.
- `package_file` - The package file.
- `size` - The size.
- `platform` - Platform.
- `installed` - If the package is installed or not.
- `downloaded` - If the package is downloaded or not.

panos_system_info

Use this data source to retrieve "show system info" from the NGFW or Panorama.

All contents of "show system info" are saved to the `info` variable. In addition, the version number of PAN-OS encountered is saved to multiple fields for ease of access.

Example Usage

```
data "panos_system_info" "example" {}
```

Attribute Reference

The following attributes are present:

- `info` - a map containing the contents of `show system info`.
- `version_major` - Major version number.
- `version_minor` - Minor version number.
- `version_patch` - Patch version number.

panos_address_group

This resource allows you to add/update/delete address groups.

Address groups are either statically defined or dynamically defined, so only `static_addresses` or `dynamic_match` should be defined within a given address group.

Import Name

```
<vsys>:<name>
```

Example Usage

```
# Static group
resource "panos_address_group" "example1" {
    name = "static ntp grp"
    description = "My NTP servers"
    static_addresses = ["ntp1", "ntp2", "ntp3"]
}

# Dynamic group
resource "panos_address_group" "example2" {
    name = "dynamic grp"
    description = "My internal NTP servers"
    dynamic_match = "'internal' and 'ntp'"
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The address group's name.
- `vsys` - (Optional) The vsys to put the address group into (default: `vsys1`).
- `static_addresses` - (Optional) The address objects to include in this statically defined address group.
- `dynamic_match` - (Optional) The IP tags to include in this DAG.
- `description` - (Optional) The address group's description.
- `tags` - (Optional) List of administrative tags.

panos_address_object

This resource allows you to add/update/delete address objects.

Import Name

```
<vsys>:<name>
```

Example Usage

```
resource "panos_address_object" "example" {
    name = "localnet"
    value = "192.168.80.0/24"
    description = "The 192.168.80 network"
    tags = ["internal", "dmz"]
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The address object's name.
- `vsys` - (Optional) The vsys to put the address object into (default: `vsys1`).
- `type` - (Optional) The type of address object. This can be `ip-netmask` (default), `ip-range`, `fqdn`, or `ip-wildcard` (PAN-OS 9.0+).
- `value` - (Required) The address object's value. This can take various forms depending on what type of address object this is, but can be something like `192.168.80.150` or `192.168.80.0/24`.
- `description` - (Optional) The address object's description.
- `tags` - (Optional) List of administrative tags.

panos_administrative_tag

This resource allows you to add/update/delete administrative tags.

Import Name

```
<vsys>:<name>
```

Example Usage

```
resource "panos_administrative_tag" "example" {
    name = "tag1"
    vsys = "vsys2"
    color = "color5"
    comment = "Internal resources"
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The administrative tag's name.
- `vsys` - (Optional) The vsys to put the administrative tag into (default: `vsys1`).
- `color` - (Optional) The tag's color. This should be either an empty string (no color) or a string such as `color1` or `color15`. Note that for maximum portability, you should limit color usage to `color16`, which was available in PAN-OS 6.1. PAN-OS 8.1's colors go up to `color42`. The value `color18` is reserved internally by PAN-OS and thus not available for use.
- `comment` - (Optional) The administrative tag's description.

panos_aggregate_interface

This resource allows you to add/update/delete aggregate ethernet interfaces.

Import Name

```
<vsys>:<name>
```

Example Usage

```
resource "panos_aggregate_interface" "example" {
    vsys = "vsys1"
    name = "ae5"
    mode = "layer3"
    static_ips = ["10.1.1.1/24"]
    comment = "Configured for internal traffic"
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The interface's name.
- `vsys` - (Required) The vsys that will use this interface. This should be something like `vsys1` or `vsys3`.
- `mode` - (Required) The interface mode. Valid values are `layer3` (default), `layer2`, `virtual-wire`, `ha`, or `decrypt-mirror`.
- `netflow_profile` - (Optional) The netflow profile.
- `mtu` - (Optional) The MTU.
- `adjust_tcp_mss` - (Optional) Adjust TCP MSS (default: false).
- `ipv4_mss_adjust` - (Optional) The IPv4 MSS adjust value.
- `ipv6_mss_adjust` - (Optional) The IPv6 MSS adjust value.
- `enable_untagged_subinterface` - (Optional, bool) Set to `true` to enable untagged subinterfaces.
- `static_ips` - (Optional) List of static IPv4 addresses.
- `ipv6_enabled` - (Optional, bool) Set to `true` to enable IPv6.
- `ipv6_interface_id` - (Optional) The IPv6 interface ID.
- `management_profile` - (Optional) The management profile.

- `enable_dhcp` - (Optional, bool) Set to `true` to enable DHCP.
- `create_dhcp_default_route` - (Optional) Set to `true` to create a DHCP default route.
- `dhcp_default_route_metric` - (Optional) The metric for the DHCP default route.
- `comment` - (Optional) The interface comment.
- `decrypt_forward` - (Optional, bool, PAN-OS 8.1+) Set to `true` to enable decrypt forward.
- `dhcp_send_hostname_enable` - (Optional, PAN-OS 9.0+) For DHCP layer3 interfaces: enable sending the firewall or a custom hostname to DHCP server
- `dhcp_send_hostname_value` - (Optional, PAN-OS 9.0+) For DHCP layer3 interfaces: the interface hostname. Leaving this unspecified with `dhcp_send_hostname_enable` set means to send the system hostname.

panos_application_group

This resource allows you to add/update/delete application groups.

Import Name

```
<vsys>:<name>
```

Example Usage

```
resource "panos_application_group" "example" {
    name = "myApp"
    applications = [
        "app1",
        "app2",
    ]
}
```

Argument Reference

The following arguments are supported:

- `vsys` - (Optional) The group's vsys (default: `vsys1`).
- `name` - (Required) The group's name.
- `applications` - (Optional) List of applications in this group.

panos_application_object

This resource allows you to add/update/delete application objects.

Import Name

```
<vsys>:<name>
```

Example Usage

```
resource "panos_application_object" "example" {
    name = "myApp"
    description = "made by terraform"
    category = "media"
    subcategory = "gaming"
    technology = "browser-based"
    defaults {
        port {
            ports = [
                "udp/dynamic",
            ]
        }
    }
    risk = 4
    scanning {
        viruses = true
    }
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The object's name.
- `vsys` - (Optional) The object's vsys (default: `vsys1`).
- `defaults` - (Optional) The application's defaults spec (defined below). To have a "defaults" of `None`, omit this section.
- `category` - (Required) The category.
- `subcategory` - (Required) The subcategory.
- `technology` - (Required) The technology.
- `description` - (Optional) The object's description.
- `timeout_settings` - (Optional) The timeout spec (defined below).

- `risk` - (Optional, int) The risk (default: 1).
- `parent_app` - (Optional) The parent application.
- `able_to_file_transfer` - (Optional, bool) Able to file transfer.
- `excessive_bandwidth` - (Optional, bool) Excessive bandwidth use.
- `tunnels_other_applications` - (Optional, bool) This application tunnels other apps.
- `has_known_vulnerability` - (Optional, bool) Has known vulnerabilities.
- `used_by_malware` - (Optional, bool) App is used by malware.
- `evasive_behavior` - (Optional, bool) App is evasive.
- `pervasive_use` - (Optional, bool) App is pervasive.
- `prone_to_misuse` - (Optional, bool) Prone to misuse.
- `continue_scanning_for_other_applications` - (Optional, bool) Continue scanning for other applications.
- `scanning` - The scanning spec (defined below).
- `alg_disable_capability` - (Optional) The alg disable capability.
- `no_app_id_caching` - (Optional, bool) No appid caching.

`defaults` supports the following arguments:

- `port` - (Optional) The port spec (defined below)
- `ip_protocol` - (Optional) The ip protocol spec (defined below)
- `icmp` - (Optional) The ICMP spec (defined below)
- `icmp6` - (Optional) The ICMP6 spec (defined below)

`defaults.port` supports the following arguments:

- `ports` - (Required) List of ports.

`defaults.ip_protocol` supports the following arguments:

- `value` - (Required, int) The IP protocol value.

`defaults.icmp` supports the following arguments:

- `type` - (Required, int) The type.
- `code` - (Optional, int) The code.

`defaults.icmp6` supports the following arguments:

- `type` - (Required, int) The type.
- `code` - (Optional, int) The code.

`timeout_settings` supports the following arguments:

- `timeout` - (Optional, int) The timeout.

- `tcp_timeout` - (Optional, int) TCP timeout.
- `udp_timeout` - (Optional, int) UDP timeout.
- `tcp_half_closed` - (Optional, int) TCP half closed timeout.
- `tcp_time_wait` - (Optional, int) TCP time wait timeout.

`scanning` supports the following arguments:

- `file_types` - (Optional, bool) File type scanning.
- `viruses` - (Optional, bool) Virus scanning.
- `data_patterns` - (Optional, bool) Data pattern scanning.

panos_application_signature

This resource allows you to add/update/delete application signatures.

Import Name

```
<vsys>:<application_object>:<name>
```

Example Usage

```
resource "panos_application_signature" "example" {
    application_object = panos_application_object.myapp.name
    comment = "made by terraform"
    ordered_match = true
    and_condition {
        or_condition {
            pattern_match {
                context = "http-req-headers"
                pattern = "somepattern"
                qualifiers = {
                    "http-method": "COPY",
                    "req-hdr-type": "HOST",
                }
            }
        }
        or_condition {
            greater_than {
                // X.400-message size
                context = "cotp-req-x420-message-size"
                value = "123456"
            }
        }
        or_condition {
            less_than {
                // X.400-message size
                context = "cotp-req-x420-message-size"
                value = "42"
            }
        }
    }
    and_condition {
        or_condition {
            equal_to {
                context = "unknown-req-tcp"
                position = "first-4bytes"
                mask = "0Xff112345"
                value = "0X11bb33dd"
            }
        }
    }
}
```

```
resource "panos_application_object" "myapp" {
    name = "myApp"
    description = "made by terraform"
    category = "media"
    subcategory = "gaming"
    technology = "browser-based"
    defaults {
        port {
            ports = [
                "udp/dynamic",
            ]
        }
    }
    risk = 4
    scanning {
        viruses = true
    }
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The signature's name.
- `vsys` - (Optional) The signature's vsys (default: `vsys1`).
- `application_object` - (Required) The applciation object for this signature.
- `comment` - (Optional) The description.
- `scope` - (Optional) The signature's scope. Valid values are `transaction` (default) or `session`.
- `ordered_match` - (Optional, bool) Set to `false` to disable ordered matching (default: `true`).
- `and_condition` - (Optional) The and condition spec (defined below).

`and_condition` supports the following arguments:

- `name` - (Computed) And condition name, this is computed and cannot be configured.
- `or_condition` - (Required) The or condition spec (defined below).

`and_condition.or_condition` supports the following arguments:

- `name` - (Computed) Or condition name, this is computed and cannot be configured.
- `pattern_match` - (Optional) The pattern match spec (defined below).
- `greater_than` - (Optional) The greater than spec (defined below).
- `less_than` - (Optional) the less than spec (defined below).
- `equal_to` - (Optional) The equal to spec (defined below).

`and_condition.or_condition.pattern_match` supports the following arguments:

- `context` - (Required) The context.
- `pattern` - (Required) The pattern.
- `qualifiers` - (Optional, map) The qualifiers.

`and_condition.or_condition.greater_than` supports the following arguments:

- `context` - (Required) The context.
- `value` - (Required) The value.
- `qualifiers` - (Optional, map) The qualifiers.

`and_condition.or_condition.less_than` supports the following arguments:

- `context` - (Required) The context.
- `value` - (Required) The value.
- `qualifiers` - (Optional, map) The qualifiers.

`and_condition.or_condition.equal_to` supports the following arguments:

- `context` - (Required) The context.
- `value` - (Required) The value.
- `position` - (Optional) The position.
- `mask` - (Optional) The mask.

panos_bfd_profile.

This resource allows you to add/update/delete BFD profiles.

Note: This resource is only applicable for PAN-OS 7.1+.

Import Name

```
<name>
```

Example Usage

```
resource "panos_bfd_profile" "example" {
    name = "myBfdProfile"
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The BBFD profile's name.
- `mode` - (Optional) BFD operation mode. Valid values are `active` (default) or `passive`.
- `minimum_tx_interval` - (Optional, int) Desired minimum TX interval in ms. Default is `1000`.
- `minimum_rx_interval` - (Optional, int) Required minimum RX interval in ms. Default is `1000`.
- `detection_multiplier` - (Optional, int) Multiplier sent to remote system. Default is `3`.
- `hold_time` - (Optional, int) Delay transmission and reception of control packets in ms.
- `minimum_rx_ttl` - (Optional, int) Minimum accepted ttl on received BFD packet.

panos_bgp_aggregate_advertise_filter

This resource allows you to add/update/delete a route advertise filter for a BGP address aggregation rule.

Import Name

```
<virtual_router>:<bgp_aggregate>:<name>
```

Example Usage

```
resource "panos_bgp_aggregate_advertise_filter" "example" {
    virtual_router = "${panos_bgp_aggregate.ag.virtual_router}"
    bgp_aggregate = "${panos_bgp_aggregate.ag.name}"
    name = "my advertise filter"
    as_path_regex = "*42*"
    med = "443"
    address_prefix {
        prefix = "10.1.1.0/24"
        exact = true
    }
    address_prefix {
        prefix = "10.1.2.0/24"
    }
}

resource "panos_bgp_aggregate" "ag" {
    virtual_router = "${panos_bgp.conf.virtual_router}"
    name = "addyAgg1"
    prefix = "192.168.1.0/24"
}

resource "panos_bgp" "conf" {
    virtual_router = "${panos_virtual_router.rtr.name}"
    router_id = "5.5.5.5"
    as_number = "42"
}

resource "panos_virtual_router" "rtr" {
    name = "my virtual router"
}
```

Argument Reference

The following arguments are supported:

- `virtual_router` - (Required) The virtual router to add this filter to.

- `bgp_aggregate` - (Required) The BGP address aggregation rule.
- `name` - (Required) The name.
- `enable` - (Optional, bool) Enable or not (default: `true`).
- `as_path_regex` - (Optional) AS path to match.
- `community_regex` - (Optional) Community to match.
- `extended_community_regex` - (Optional) Extended community to match.
- `med` - (Optional) Match MED.
- `route_table` - (Optional, PAN-OS 8.0+) Route table to match rule. Valid values are `unicast`, `multicast`, or `both`.
- `address_prefix` - (Optional, repeatable) Matching address prefix definition (see below).
- `next_hops` - (Optional) List of next hop attributes.
- `from_peers` - (Optional) List of peers that advertised the route entry.

Each `address_prefix` section offers the following params:

- `prefix` - (Required) Address prefix.
- `exact` - (Optional, bool) Match exact prefix length.

panos_bgp_aggregate

This resource allows you to add/update/delete BGP address aggregation rules.

Import Name

```
<virtual_router>:<name>
```

Example Usage

```
resource "panos_bgp_aggregate" "example" {
    virtual_router = "${panos_bgp.conf.virtual_router}"
    name = "myAggRule"
    prefix = "192.168.1.0/24"
    weight = 17
}

resource "panos_bgp" "conf" {
    virtual_router = "${panos_virtual_router.vr.name}"
    router_id = "1.2.3.4"
    as_number = 443
}

resource "panos_virtual_router" "vr" {
    name = "my vr"
}
```

Argument Reference

The following arguments are supported:

- `virtual_router` - (Required) The virtual router to put the rule into.
- `name` - (Required) The rule name.
- `prefix` - (Required) Aggregating address prefix.
- `enable` - (Optional, bool) Enable this rule (default: `true`)
- `as_set` - (Optional, bool) Generate AS-set attribute.
- `summary` - (Optional, bool) Summarize route.
- `local_preference` - (Optional) New local preference value.
- `med` - (Optional) New MED value.
- `weight` - (Optional, int) New weight value.

- `next_hop` - (Optional) Next hop address.
- `origin` - (Optional) New route origin. Valid values are `incomplete` (default), `igp`, or `egp`.
- `as_path_limit` - (Optional, int) Add AS path limit attribute if it does not exist.
- `as_path_type` - (Optional) AS path update options. Valid values are `none` (default) or `prepend`.
- `as_path_value` - (Optional) For `as_path_type` of `prepend`, the value to prepend.
- `community_type` - (Optional) Community update options. Valid values are `none` (default), `remove-all`, `remove-regex`, `append`, or `overwrite`.
- `community_value` - (Optional) If `community_type` is `remove-regex`, `append`, or `overwrite`, the value associated with that setting. For the `append` and `overwrite` types specifically, valid values are `no-export`, `no-advertise`, `local-as`, or `nopeer`.
- `extended_community_type` - (Optional) Extended community update options. Valid values are `none` (default), `remove-all`, `remove-regex`, `append`, or `overwrite`.
- `extended_community_vaule` - (Optional) If `extended_community_type` is `remove-regex`, `append`, or `overwrite`, the value associated with that setting.

panos_bgp_aggregate_suppress_filter

This resource allows you to add/update/delete a route suppression filter for a BGP address aggregation rule.

Import Name

```
<virtual_router>:<bgp_aggregate>:<name>
```

Example Usage

```
resource "panos_bgp_aggregate_suppress_filter" "example" {
    virtual_router = "${panos_bgp_aggregate.ag.virtual_router}"
    bgp_aggregate = "${panos_bgp_aggregate.ag.name}"
    name = "my suppression filter"
    as_path_regex = "*42*"
    med = "443"
    address_prefix {
        prefix = "10.1.1.0/24"
        exact = true
    }
    address_prefix {
        prefix = "10.1.2.0/24"
    }
}

resource "panos_bgp_aggregate" "ag" {
    virtual_router = "${panos_bgp.conf.virtual_router}"
    name = "addyAgg1"
    prefix = "192.168.1.0/24"
}

resource "panos_bgp" "conf" {
    virtual_router = "${panos_virtual_router.rtr.name}"
    router_id = "5.5.5.5"
    as_number = "42"
}

resource "panos_virtual_router" "rtr" {
    name = "my virtual router"
}
```

Argument Reference

The following arguments are supported:

- `virtual_router` - (Required) The virtual router to add this filter to.

- `bgp_aggregate` - (Required) The BGP address aggregation rule.
- `name` - (Required) The name.
- `enable` - (Optional, bool) Enable or not (default: `true`).
- `as_path_regex` - (Optional) AS path to match.
- `community_regex` - (Optional) Community to match.
- `extended_community_regex` - (Optional) Extended community to match.
- `med` - (Optional) Match MED.
- `route_table` - (Optional, PAN-OS 8.0+) Route table to match rule. Valid values are `unicast`, `multicast`, or `both`.
- `address_prefix` - (Optional, repeatable) Matching address prefix definition (see below).
- `next_hops` - (Optional) List of next hop attributes.
- `from_peers` - (Optional) List of peers that advertised the route entry.

Each `address_prefix` section offers the following params:

- `prefix` - (Required) Address prefix.
- `exact` - (Optional, bool) Match exact prefix length.

panos_bgp_auth_profile

This resource allows you to add/update/delete a BGP auth profile.

Example Usage

```
resource "panos_bgp_auth_profile" "example" {
    virtual_router = "${panos_bgp.conf.virtual_router}"
    name = "prof1"
    secret = "secret"
}

resource "panos_bgp" "conf" {
    virtual_router = "${panos_virtual_router.rtr.name}"
    router_id = "5.5.5.5"
    as_number = "42"
}

resource "panos_virtual_router" "rtr" {
    name = "my virtual router"
}
```

Argument Reference

The following arguments are supported:

- `virtual_router` - (Required) The virtual router to add this BGP auth profile to.
- `name` - (Required) The name.
- `secret` - (Optional) Shared secret for the TCP MD5 authentication.

panos_bgp_conditional_adv_advertise_filter

This resource allows you to add/update/delete an advertise filter for a BGP conditional advertisement.

Note: A BGP conditional advertisement is valid only if there is at least one non-exist filter and one advertise filter attached. This filter must be paired with the other in order for the configuration to be valid.

Import Name

```
<virtual_router>:<bgp_conditional_adv>:<name>
```

Example Usage

```
data "panos_system_info" "x" {}

resource "panos_bgp_conditional_adv_advertise_filter" "example" {
    virtual_router = "${panos_bgp.conf.virtual_router}"
    bgp_conditional_adv = "${panos_bgp_conditional_adv.ca.name}"
    name = "af"
    route_table = "${data.panos_system_info.x.version_major >= 8 ? "unicast" : ""}"
    address_prefixes = ["192.168.1.0/24"]
}

resource "panos_bgp_conditional_adv" "ca" {
    virtual_router = "${panos_bgp.conf.virtual_router}"
    name = "example"
}

resource "panos_bgp_conditional_adv_non_exist_filter" "af" {
    virtual_router = "${panos_bgp.conf.virtual_router}"
    bgp_conditional_adv = "${panos_bgp_conditional_adv.ca.name}"
    name = "nef"
    route_table = "${data.panos_system_info.x.version_major >= 8 ? "unicast" : ""}"
    address_prefixes = ["192.168.2.0/24"]
}

resource "panos_bgp" "conf" {
    virtual_router = "${panos_virtual_router.rtr.name}"
    router_id = "5.5.5.5"
    as_number = "42"
}

resource "panos_virtual_router" "rtr" {
    name = "my virtual router"
}
```

Argument Reference

The following arguments are supported:

- `virtual_router` - (Required) The virtual router to add this filter to.
- `bgp_conditional_adv` - (Required) The BGP conditional advertisement to add this filter to.
- `name` - (Required) The name.
- `enable` - (Optional, bool) Enable or not (default: `true`).
- `as_path_regex` - (Optional) AS path to match.
- `community_regex` - (Optional) Community to match.
- `extended_community_regex` - (Optional) Extended community to match.
- `med` - (Optional) Match MED.
- `route_table` - (Optional, PAN-OS 8.0+) Route table to match rule. Valid values are `unicast`, `multicast`, or `both`. As of PAN-OS 8.1, there doesn't seem to be a way to configure this in the GUI, it is always set to `unicast`. Thus, if you're running this resource against PAN-OS 8.0+, the appropriate thing to do is set this value to `unicast` as well to match the GUI functionality.
- `address_prefixes` - (Optional) List of matching address prefixes.
- `next_hops` - (Optional) List of next hop attributes.
- `from_peers` - (Optional) List of peers that advertised the route entry.

panos_bgp_conditional_adv

This resource allows you to add/update/delete a BGP conditional advertisement.

Note: In the PAN-OS GUI, this resource cannot be created without also creating at least one non-exist filter and one advertise filter. The API behaves a little differently: you can create the conditional advertisement itself, but the API will start throwing errors if you try to update it and there is not at least one non-exist filter and one advertise filter. In order for a conditional advertisement to be valid, you must specify at least one non-exist and one advertise filter.

Import Name

```
<virtual_router>:<name>
```

Example Usage

```
data "panos_system_info" "x" {}

resource "panos_bgp_conditional_adv" "example" {
    virtual_router = "${panos_bgp.conf.virtual_router}"
    name = "example"
    enable = false
}

resource "panos_bgp_conditional_adv_non_exist_filter" "nef" {
    virtual_router = "${panos_bgp.conf.virtual_router}"
    bgp_conditional_adv = "${panos_bgp_conditional_adv.example.name}"
    route_table = "${data.panos_system_info.x.version_major >= 8 ? "unicast" : ""}"
    name = "nef"
    address_prefixes = ["192.168.1.0/24"]
}

resource "panos_bgp_conditional_adv_advertise_filter" "af" {
    virtual_router = "${panos_bgp.conf.virtual_router}"
    bgp_conditional_adv = "${panos_bgp_conditional_adv.example.name}"
    route_table = "${data.panos_system_info.x.version_major >= 8 ? "unicast" : ""}"
    name = "af"
    address_prefixes = ["192.168.2.0/24"]
}

resource "panos_bgp" "conf" {
    virtual_router = "${panos_virtual_router.rtr.name}"
    router_id = "5.5.5.5"
    as_number = "42"
}

resource "panos_virtual_router" "rtr" {
    name = "my virtual router"
}
```

Argument Reference

The following arguments are supported:

- `virtual_router` - (Required) The virtual router to add this BGP conditional advertisement to.
- `name` - (Required) The name.
- `enable` - (Optional, bool) Enable or not (default: `true`).
- `used_by` - (Optional) List of BGP peer groups that use this rule.

panos_bgp_conditional_adv_non_exist_filter

This resource allows you to add/update/delete a non-exist filter for a BGP conditional advertisement.

Note: A BGP conditional advertisement is valid only if there is at least one non-exist filter and one advertise filter attached. This filter must be paired with the other in order for the configuration to be valid.

Import Name

```
<virtual_router>:<bgp_conditional_adv>:<name>
```

Example Usage

```
data "panos_system_info" "x" {}

resource "panos_bgp_conditional_adv_non_exist_filter" "example" {
    virtual_router = "${panos_bgp.conf.virtual_router}"
    bgp_conditional_adv = "${panos_bgp_conditional_adv.ca.name}"
    name = "nef"
    route_table = "${data.panos_system_info.x.version_major >= 8 ? "unicast" : ""}"
    address_prefixes = ["192.168.1.0/24"]
}

resource "panos_bgp_conditional_adv" "ca" {
    virtual_router = "${panos_bgp.conf.virtual_router}"
    name = "example"
}

resource "panos_bgp_conditional_adv_advertise_filter" "af" {
    virtual_router = "${panos_bgp.conf.virtual_router}"
    bgp_conditional_adv = "${panos_bgp_conditional_adv.ca.name}"
    name = "af"
    route_table = "${data.panos_system_info.x.version_major >= 8 ? "unicast" : ""}"
    address_prefixes = ["192.168.2.0/24"]
}

resource "panos_bgp" "conf" {
    virtual_router = "${panos_virtual_router.rtr.name}"
    router_id = "5.5.5.5"
    as_number = "42"
}

resource "panos_virtual_router" "rtr" {
    name = "my virtual router"
}
```

Argument Reference

The following arguments are supported:

- `virtual_router` - (Required) The virtual router to add this filter to.
- `bgp_conditional_adv` - (Required) The BGP conditional advertisement to add this filter to.
- `name` - (Required) The name.
- `enable` - (Optional, bool) Enable or not (default: `true`).
- `as_path_regex` - (Optional) AS path to match.
- `community_regex` - (Optional) Community to match.
- `extended_community_regex` - (Optional) Extended community to match.
- `med` - (Optional) Match MED.
- `route_table` - (Optional, PAN-OS 8.0+) Route table to match rule. Valid values are `unicast`, `multicast`, or `both`. As of PAN-OS 8.1, there doesn't seem to be a way to configure this in the GUI, it is always set to `unicast`. Thus, if you're running this resource against PAN-OS 8.0+, the appropriate thing to do is set this value to `unicast` as well to match the GUI functionality.
- `address_prefixes` - (Optional) List of matching address prefixes.
- `next_hops` - (Optional) List of next hop attributes.
- `from_peers` - (Optional) List of peers that advertised the route entry.

panos_bgp_dampening_profile

This resource allows you to add/update/delete a BGP dampening profile.

Import Name

```
<virtual_router>:<name>
```

Example Usage

```
resource "panos_bgp_dampening_profile" "example" {
    virtual_router = "${panos_bgp.conf.virtual_router}"
    name = "myDampeningProfile"
}

resource "panos_bgp" "conf" {
    virtual_router = "${panos_virtual_router.rtr.name}"
    router_id = "5.5.5.5"
    as_number = "42"
}

resource "panos_virtual_router" "rtr" {
    name = "my virtual router"
}
```

Argument Reference

The following arguments are supported:

- `virtual_router` - (Required) The virtual router to add this BGP dampening profile to.
- `name` - (Required) The name.
- `enable` - (Optional, bool) Enable or not (default: `true`).
- `cutoff` - (Optional, float) Cutoff threshold value (default: `1.25`).
- `reuse` - (Optional, float) Reuse threshold value (default: `0.5`).
- `max_hold_time` - (Optional, int) Maximum hold-down time, in seconds (default: `900`).
- `decay_half_life_reachable` - (Optional, int) Decay half-life while reachable, in seconds (default: `300`).
- `decay_half_life_unreachable` - (Optional, int) Decay half-life while unreachable, in seconds (default: `900`).

panos_bgp_export_rule_group

This resource allows you to add/update/delete BGP export rule groups.

This resource manages clusters of export rules in a virtual router, enforcing both the contents of individual rules as well as their ordering. Rules are defined in a `rule` config block.

Although you cannot modify non-group export rules with this resource, the `position_keyword` and `position_reference` parameters allow you to reference some other export rule that already exists, using it as a means to ensure some rough placement within the ruleset as a whole.

Best Practices

As is to be expected, if you are separating your deployment across multiple plan files, make sure that at most only one plan specifies any given absolute positioning keyword such as "top" or "directly below", otherwise they'll keep shoving each other out of the way indefinitely.

Best practices are to specify one group as `top` (if you need it), one group as `bottom`, then all other groups should be `above` the first rule of the bottom group. You do it this way because rules will natually be added at the tail end of the ruleset, so they will always be `after` the first group, but what you want is for them to be `before` the last group's rules.

Example Usage

```
resource "panos_bgp_export_rule_group" "example" {
    virtual_router = "${panos_bgp.conf.virtual_router}"
    rule {
        name = "first"
        match_as_path_regex = "*foo*"
        match_address_prefix {
            prefix = "192.168.1.0/24"
        }
        match_address_prefix {
            prefix = "192.168.2.0/24"
            exact = true
        }
        match_route_table = "${data.panos_system_info.x.version_major >= 8 ? "unicast" : ""}"
        local_preference = "42"
        med = "43"
        weight = 44
        origin = "incomplete"
    }
    rule {
        name = "second"
        match_as_path_regex = "*bar*"
        action = "deny"
        match_route_table = "${data.panos_system_info.x.version_major >= 8 ? "unicast" : ""}"
    }
}

data "panos_system_info" "x" {}

resource "panos_bgp" "conf" {
    virtual_router = "${panos_virtual_router.vr.name}"
    router_id = "1.2.3.4"
    as_number = 443
}

resource "panos_virtual_router" "vr" {
    name = "my vr"
}
```

Argument Reference

The following arguments are supported:

- `virtual_router` - (Required) The virtual router to put the rule into.
- `position_keyword` - (Optional) A positioning keyword for this group. This can be `before`, `directly before`, `after`, `directly after`, `top`, `bottom`, or left empty (the default) to have no particular placement. This param works in combination with the `position_reference` param.
- `position_reference` - (Optional) Required if `position_keyword` is one of the "above" or "below" variants, this is the name of a non-group rule to use as a reference to place this group.
- `rule` - The export rule definition (see below). The export rule ordering will match how they appear in the terraform plan file.

The following arguments are valid for each `rule` section:

- `name` - (Required) The security rule name.
- `enable` - (Optional, bool) Enable this export rule (default: `true`)
- `used_by` - (Optional) List of peer groups.
- `match_as_path_regex` - (Optional) AS path to match.
- `match_community_regex` - (Optional) Community to match.
- `match_extended_community_regex` - (Optional) Extended community to match.
- `match_med` - (Optional) Match MED.
- `match_route_table` - (Optional, PAN-OS 8.0+) Route table to match rule. Valid values are `unicast`, `multicast`, or `both`. As of PAN-OS 8.1, there doesn't seem to be a way to configure this in the GUI, it is always set to `unicast`. Thus, if you're running this resource against PAN-OS 8.0+, the appropriate thing to do is set this value to `unicast` as well to match the GUI functionality.
- `match_address_prefix` - (Optional, repeatable) Matching address prefix definition (see below). below for the params for this section.
- `match_next_hops` - (Optional) List of next hop attributes.
- `match_from_peers` - (Optional) List of peers that advertised the route entry.
- `action` - (Optional) Rule action. Valid values are `allow` (default) or `deny`.
- `dampening` - (Optional) Route flap dampening profile.
- `local_preference` - (Optional) New local preference value.
- `med` - (Optional) New MED value.
- `weight` - (Optional, int) New weight value.
- `next_hop` - (Optional) Next hop address.
- `origin` - (Optional) New route origin. Valid values are `igp`, `egp`, or `incomplete`.
- `as_path_limit` - (Optional, int) Add AS path limit attribute if it does not exist.
- `as_path_type` - (Optional) AS path update options. Valid values are `none`, `remove`, `prepend`, or `remove-and-prepend`.
- `as_path_value` - (Optional) If `as_path_type` is `prepend` or `remove-and-prepend`, the value to prepend.
- `community_type` - (Optional) Community update options. Valid values are `none`, `remove-all`, `remove-regex`, `append`, or `overwrite`.
- `community_value` - (Optional) If `community_type` is `remove-regex`, `append`, or `overwrite`, the value associated with that setting. For the `append` and `overwrite` types specifically, valid values for `community_value` are `no-export`, `no-advertise`, `local-as`, or `nopeer`.
- `extended_community_type` - (Optional) Extended community update options. Valid values are `none`, `remove-all`, `remove-regex`, `append`, or `overwrite`.

- `extended_community_vaule` - (Optional) If `extended_community_type` is `remove-regex`, `append`, or `overwrite`, the value associated with that setting.

Each `match_address_prefix` section offers the following params:

- `prefix` - (Required) Address prefix.
- `exact` - (Optional, bool) Match exact prefix length.

panos_bgp

This resource allows you to add/update/delete a virtual router's BGP configuration.

Important Note: When it comes to BGP configuration, PAN-OS requires that BGP itself first be configured before you can add other BGP sub-config, such as dampening profiles or peer groups. Since every BGP resource must reference a virtual router, the key to accomplishing this is by pointing the `virtual_router` param for each BGP sub-config to `panos_bgp.foo.virtual_router` instead of `panos_virtual_router.bar.name`.

Import Name

```
<virtual_router>
```

Example Usage

```
resource "panos_bgp" "example" {
    virtual_router = "${panos_virtual_router.rtr.name}"
    router_id = "5.5.5.5"
    as_number = "42"
}

resource "panos_virtual_router" "rtr" {
    name = "my virtual router"
}
```

Argument Reference

The following arguments are supported:

- `virtual_router` - (Required) The virtual router to add this BGP configuration to.
- `enable` - (Optional, bool) Enable BGP or not (default: `true`).
- `router_id` - (Optional) Router ID of this BGP instance.
- `as_number` - (Optional) Local AS number.
- `bfd_profile` - (Optional, PAN-OS 7.1+) BFD configuration.
- `reject_default_route` - (Optional, bool) Do not learn default route from BGP (default: `true`).
- `install_route` - (Optional, bool) Populate BGP learned route to global route table.
- `aggregate_med` - (Optional, bool) Aggregate route only if they have same MED attributes (default: `true`).
- `default_local_preference` - (Optional) Default local preference (default: `"100"`).

- `as_format` - (Optional) AS format. Valid values are `"2-byte"` (default) or `"4-byte"`.
- `always_compare_med` - (Optional, bool) Always compare MEDs.
- `deterministic_med_comparison` - (Optional, bool) Deterministic MED comparison (default: `true`).
- `ecmp_multi_as` - (Optional, bool) Support multiple AS in ECMP.
- `enforce_first_as` - (Optional, bool) Enforce First AS for EBGP (default: `true`).
- `enable_graceful_restart` - (Optional, bool) Enable graceful restart (default: `true`).
- `stale_route_time` - (Optional, int) Time to remove stale routes after peer restart, in seconds (default: `120`).
- `local_restart_time` - (Optional, int) Local restart time to advertise to peer, in seconds (default: `120`).
- `max_peer_restart_time` - (Optional, int) Maximum of peer restart time accepted, in seconds (default: `120`).
- `reflector_cluster_id` - (Optional) Route reflector cluster ID.
- `confederation_member_as` - (Optional) Confederation requires member-AS number.
- `allow_redistribute_default_route` - (Optional, bool) Allow redistribute default route to BGP.

panos_bgp_import_rule_group

This resource allows you to add/update/delete BGP import rule groups.

This resource manages clusters of import rules in a virtual router, enforcing both the contents of individual rules as well as their ordering. Rules are defined in a `rule` config block.

Although you cannot modify non-group import rules with this resource, the `position_keyword` and `position_reference` parameters allow you to reference some other import rule that already exists, using it as a means to ensure some rough placement within the ruleset as a whole.

Best Practices

As is to be expected, if you are separating your deployment across multiple plan files, make sure that at most only one plan specifies any given absolute positioning keyword such as "top" or "directly below", otherwise they'll keep shoving each other out of the way indefinitely.

Best practices are to specify one group as `top` (if you need it), one group as `bottom`, then all other groups should be `above` the first rule of the bottom group. You do it this way because rules will natually be added at the tail end of the ruleset, so they will always be `after` the first group, but what you want is for them to be `before` the last group's rules.

Example Usage

```
resource "panos_bgp_import_rule_group" "example" {
    virtual_router = "${panos_bgp.conf.virtual_router}"
    rule {
        name = "first"
        match_as_path_regex = "*foo*"
        match_address_prefix {
            prefix = "192.168.1.0/24"
        }
        match_address_prefix {
            prefix = "192.168.2.0/24"
            exact = true
        }
        match_route_table = "${data.panos_system_info.x.version_major >= 8 ? "unicast" : ""}"
        local_preference = "42"
        med = "43"
        weight = 44
        origin = "incomplete"
    }
    rule {
        name = "second"
        match_as_path_regex = "*bar*"
        action = "deny"
        match_route_table = "${data.panos_system_info.x.version_major >= 8 ? "unicast" : ""}"
    }
}

data "panos_system_info" "x" {}

resource "panos_bgp" "conf" {
    virtual_router = "${panos_virtual_router.vr.name}"
    router_id = "1.2.3.4"
    as_number = 443
}

resource "panos_virtual_router" "vr" {
    name = "my vr"
}
```

Argument Reference

The following arguments are supported:

- `virtual_router` - (Required) The virtual router to put the rule into.
- `position_keyword` - (Optional) A positioning keyword for this group. This can be `before`, `directly before`, `after`, `directly after`, `top`, `bottom`, or left empty (the default) to have no particular placement. This param works in combination with the `position_reference` param.
- `position_reference` - (Optional) Required if `position_keyword` is one of the "above" or "below" variants, this is the name of a non-group rule to use as a reference to place this group.
- `rule` - The import rule definition (see below). The import rule ordering will match how they appear in the terraform plan file.

The following arguments are valid for each `rule` section:

- `name` - (Required) The security rule name.
- `enable` - (Optional, bool) Enable this import rule (default: `true`)
- `used_by` - (Optional) List of peer groups.
- `match_as_path_regex` - (Optional) AS path to match.
- `match_community_regex` - (Optional) Community to match.
- `match_extended_community_regex` - (Optional) Extended community to match.
- `match_med` - (Optional) Match MED.
- `match_route_table` - (Optional, PAN-OS 8.0+) Route table to match rule. Valid values are `unicast`, `multicast`, or `both`. As of PAN-OS 8.1, there doesn't seem to be a way to configure this in the GUI, it is always set to `unicast`. Thus, if you're running this resource against PAN-OS 8.0+, the appropriate thing to do is set this value to `unicast` as well to match the GUI functionality.
- `match_address_prefix` - (Optional, repeatable) Matching address prefix definition (see below). below for the params for this section.
- `match_next_hops` - (Optional) List of next hop attributes.
- `match_from_peers` - (Optional) List of peers that advertised the route entry.
- `action` - (Optional) Rule action. Valid values are `allow` (default) or `deny`.
- `dampening` - (Optional) Route flap dampening profile.
- `local_preference` - (Optional) New local preference value.
- `med` - (Optional) New MED value.
- `weight` - (Optional, int) New weight value.
- `next_hop` - (Optional) Next hop address.
- `origin` - (Optional) New route origin. Valid values are `igp`, `egp`, or `incomplete`.
- `as_path_limit` - (Optional, int) Add AS path limit attribute if it does not exist.
- `as_path_type` - (Optional) AS path update options. Valid values are `none` or `remove`.
- `community_type` - (Optional) Community update options. Valid values are `none`, `remove-all`, `remove-regex`, `append`, or `overwrite`.
- `community_value` - (Optional) If `community_type` is `remove-regex`, `append`, or `overwrite`, the value associated with that setting. For the `append` and `overwrite` types specifically, valid values for `community_value` are `no-export`, `no-advertise`, `local-as`, or `nopeer`.
- `extended_community_type` - (Optional) Extended community update options. Valid values are `none`, `remove-all`, `remove-regex`, `append`, or `overwrite`.
- `extended_community_vaule` - (Optional) If `extended_community_type` is `remove-regex`, `append`, or `overwrite`, the value associated with that setting.

Each `match_address_prefix` section offers the following params:

- `prefix` - (Required) Address prefix.
- `exact` - (Optional, bool) Match exact prefix length.

panos_bgp_peer_group

This resource allows you to add/update/delete a BGP peer group.

Import Name

```
<virtual_router>:<name>
```

Example Usage

```
resource "panos_bgp_peer_group" "example" {
    virtual_router = "${panos_bgp.conf.virtual_router}"
    name = "myName"
}

resource "panos_bgp" "conf" {
    virtual_router = "${panos_virtual_router.rtr.name}"
    router_id = "5.5.5.5"
    as_number = "42"
}

resource "panos_virtual_router" "rtr" {
    name = "my virtual router"
}
```

Argument Reference

The following arguments are supported:

- `virtual_router` - (Required) The virtual router to add this BGP peer group to.
- `name` - (Required) The name.
- `enable` - (Optional, bool) Enable or not (default: `true`).
- `aggregated_confed_as_path` - (Optional, bool) The peers understand aggregated confederation AS path (default: `true`).
- `soft_reset_with_stored_info` - (Optional, bool) Soft reset with stored info.
- `type` - (Optional) Peer group type. Valid options are `ebgp` (default), `ebgp-confed`, `ibgp`, or `ibgp-confed`.
- `export_next_hop` - (Optional) Export next hop. Valid values are `original`, `use-self`, or `resolve`.
- `import_next_hop` - (Optional) Import next hop. Valid values are `original`, `use-peer`, or the empty string.
- `remove_private_as` - (Optional, bool) Remove private AS when exporting route. Only available for `type=ebgp`.

panos_bgp_peer

This resource allows you to add/update/delete a BGP peer.

Import Name

```
<virtual_router>:<bgp_peer_group>:<name>
```

Example Usage

```
data "panos_system_info" "x" {}

// Peer definition that will work starting from PAN-OS 6.1.
resource "panos_bgp_peer" "example" {
    virtual_router = "${panos_bgp.conf.virtual_router}"
    bgp_peer_group = "${panos_bgp_peer_group.pg.name}"
    name = "peer1"
    peer_as = "${panos_bgp.conf.as_number}"
    local_address_interface = "${panos_ethernet_interface.e.name}"
    local_address_ip = "${panos_ethernet_interface.e.static_ips.0}"
    peer_address_ip = "5.6.7.8"
    max_prefixes = "unlimited"
    bfd_profile = "${
        data.panos_system_info.x.version_major >= 7 ?
            data.panos_system_info.x.version_minor >= 1 ? "None" : ""
        : ""
    }"
    address_family_type = "${data.panos_system_info.x.version_major >= 8 ? "ipv4" : ""}"
    reflector_client = "${data.panos_system_info.x.version_major >= 8 ? "non-client" : ""}"
    min_route_advertisement_interval = "${
        data.panos_system_info.x.version_major >= 8 ?
            data.panos_system_info.x.version_minor >= 1 ? 30 : 0
        : 0
    }"
}

resource "panos_bgp_peer_group" "pg" {
    virtual_router = "${panos_bgp.conf.virtual_router}"
    name = "myName"
    type = "ibgp"
}

resource "panos_bgp" "conf" {
    virtual_router = "${panos_virtual_router.rtr.name}"
    router_id = "5.5.5.5"
    as_number = "42"
}

resource "panos_virtual_router" "rtr" {
    name = "my virtual router"
    interfaces = ["${panos_ethernet_interface.e.name}"]
}

resource "panos_ethernet_interface" "e" {
    name = "ethernet1/5"
    mode = "layer3"
    vsys = "vsys1"
    static_ips = ["192.168.1.1/24"]
}
```

Argument Reference

The following arguments are supported:

- `virtual_router` - (Required) The virtual router to add this BGP peer to.

- `bgp_peer_group` - (Required) The BGP peer group to put this peer into.
- `name` - (Required) The name.
- `enable` - (Optional, bool) Enable or not (default: `true`).
- `peer_as` - (Optional) Peer AS number.
- `local_address_interface` - (Required) Interface to accept BGP session.
- `local_address_ip` - (Optional) Specify exact IP address if interface has multiple addresses.
- `peer_address_ip` - (Required) Peer IP address configuration.
- `reflector_client` - (Optional) This peer is reflector client. Valid values are `non-client`, `client`, or `meshed-client`.
- `peering_type` - (Optional) Peering type that affects NOPEER community value handling. Valid values are `unspecified` (default) or `bilateral`.
- `max_prefixes` - (Optional) Maximum of prefixes to receive from the peer. This can be a number such as `"5000"` (default) or `unlimited`.
- `auth_profile` - (Optional) Auth profile.
- `keep_alive_interval` - (Optional, int) Keep alive interval, in seconds (default: `30`).
- `multi_hop` - (Optional, int) IP TTL value used for sending BGP packet.
- `open_delay_time` - (Optional, int) Open delay time, in seconds.
- `hold_time` - (Optional, int) Hold time, in seconds.
- `idle_hold_time` - (Optional, int) Idle hold time, in seconds.
- `allow_incoming_connections` - (Optional, bool) Allow incoming connections (default: `true`).
- `incoming_connections_remote_port` - (Optional, int) Restrict remote port for incoming BGP connections.
- `allow_outgoing_connections` - (Optional, bool) Allow outgoing connections (default: `true`).
- `outgoing_connections_local_port` - (Optional, int) Use specific local port for outgoing BGP connections.
- `bfd_profile` - (Optional, PAN-OS 7.1+) BFD profile. This can be a specific BFD profile name, `None` (disables BFD), or `Inherit-vr-global-setting`.
- `enable_mp_bgp` - (Optional, bool, PAN-OS 8.0+) Enable MP BGP.
- `address_family_type` - (Optional, PAN-OS 8.0+) Set the AFI for this peer. Valid values are `ipv4` or `ipv6`.
- `subsequent_address_family_unicast` - (Optional, bool, PAN-OS 8.0+) Enable unicast subsequent address family for this peer.
- `subsequent_address_family_multicast` - (Optional, bool, PAN-OS 8.0+) Enable multicast subsequent address family for this peer.
- `enable_sender_side_loop_detection` - (Optional, bool, PAN-OS 8.0+) Enable sender side loop detection.
- `min_route_advertisement_interval` - (Optional, int, PAN-OS 8.1+) Minimum route advertisement interval, in seconds.

panos_bgp_redist_rule

This resource allows you to add/update/delete a BGP redistribution rule.

Import Name

```
<virtual_router>:<name>
```

Example Usage

```
resource "panos_bgp_redist_rule" "example" {
    virtual_router = "${panos_bgp.conf.virtual_router}"
    route_table = "${data.panos_system_info.x.version_major >= 8 ? "unicast" : ""}"
    name = "192.168.1.0/24"
    set_med = "42"
}

data "panos_system_info" "x" {}

resource "panos_bgp" "conf" {
    virtual_router = "${panos_virtual_router.rtr.name}"
    router_id = "5.5.5.5"
    as_number = "42"
}

resource "panos_virtual_router" "rtr" {
    name = "my virtual router"
}
```

Argument Reference

The following arguments are supported:

- `virtual_router` - (Required) The virtual router to add this BGP redist rule to.
- `name` - (Required) A subnet or a redistribution profile.
- `enable` - (Optional, bool) Enable this rule or not (default: `true`).
- `address_family` - (Optional) The address family. Valid values are `ipv4` (default) or `ipv6`.
- `route_table` - (Optional, PAN-OS 8.0+) Route table to match rule. Valid values are `unicast`, `multicast`, or `both`. As of PAN-OS 8.1, there doesn't seem to be a way to configure this in the GUI, it is always set to `unicast`. Thus, if you're running this resource against PAN-OS 8.0+, the appropriate thing to do is set this value to `unicast` as well to match the GUI functionality.
- `metric` - (Optional, int) Metric value.

- `set_origin` - (Optional) Add the origin path attribute. Valid values are `incomplete` (default), `igp`, or `egp`.
- `set_med` - (Optional) Add the MULTI_EXIT_DISC path attribute.
- `set_local_preference` - (Optional) Add the LOCAL_PREF path attribute.
- `set_as_path_limit` - (Optional, int) Add the AS_PATHLIMIT path attribute.
- `set_communities` - (Optional) List of COMMUNITY path attributes to add.
- `set_extended_communities` - (Optional) List of EXTENDED COMMUNITY path attributes to add.

panos_dag_tags

This resource allows you to add and remove dynamic address group tags.

The `ip` field should be unique in the `panos_dag_tags` block, and there should only be one `panos_dag_tags` block defined in a given plan.

Note - Tags are only removed during `terraform destroy`. Updating an applied terraform plan to have alternative tags will leave behind the old tags from the previously published plan(s).

Example Usage

```
resource "panos_dag_tags" "example" {
    vsys = "vsys1"
    register {
        ip = "10.1.1.1"
        tags = ["tag1", "tag2"]
    }
    register {
        ip = "10.1.1.2"
        tags = ["tag3"]
    }
}
```

Argument Reference

The following arguments are supported:

- `vsys` - (Optional) The vsys to put the DAG tags in (default: `vsys1`).
- `register` - (Required) A set that includes `ip`, the IP address to be tagged and `tags`, a list of tags to associate with the given IP.

panos_edl

This resource allows you to add/update/delete external dynamic lists (EDL).

Setting `repeat_at`

The acceptable PAN-OS values for the `repeat_at` field is a combination of the version of PAN-OS that you're running against and the setting of the `repeat` parameter.

The following shorthand is used:

- `N/A` - `repeat_at` should not be set
- `minute` - A two character minute string (e.g. - `07` or `59`)
- `24hr hour` - A two character hour string in 24hr notation (e.g. - `09` or `15`)
- `24hr time` - A five character hour/minute string in 24hr notation (e.g. - `09:00` or `23:59`)

Here are the valid settings for `repeat_at` given your desired `repeat` value and the version of PAN-OS you're running against:

- PAN-OS 6.1 - 7.0
  - `hourly` - minute
  - `daily`, `weekly`, `monthly` - 24hr time
- PAN-OS 7.1+
  - `every five minutes`, `hourly` - N/A
  - `daily`, `weekly`, `monthly` - 24hr hour

Import Name

```
<vsys>:<name>
```

Example Usage

```
resource "panos_edl" "example" {
    name = "example"
    type = "ip"
    description = "my edl"
    source = "https://example.com"
    repeat = "every five minutes"
    exceptions = ["10.1.1.1", "10.1.1.2"]
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The object's name
- `vsys` - (Optional) The vsys to put the object into (default: `vsys1`)
- `type` - (Optional) The type of EDL. This can be `ip` (the default; and the only valid value for PAN-OS 6.1 - 7.0), `domain`, `url`, or `predefined` (PAN-OS 8.0+)
- `description` - (Optional) The object's description.
- `source` - (Optional) The EDL source URL
- `certificate_profile` - (Optional) Profile for authenticating client certificates
- `username` - (Optional) EDL username
- `password` - (Optional) EDL password
- `repeat` - (Optional) How often to retrieve the EDL. This can be `hourly` (the default), `daily`, `weekly`, `monthly`, or `every five minutes` (valid for PAN-OS 7.1+)
- `repeat_at` - (Optional) The time at which to retrieve the EDL. Please refer to the section above for how to set this value properly.
- `repeat_day_of_week` - (Optional) If `repeat` is `weekly`, then this should be set to the desired day of the week. Valid values are `sunday`, `monday`, `tuesday`, `wednesday`, `thursday`, `friday`, `saturday`, and `sunday`
- `repeat_day_of_month` - (Optional, int) If `repeat` is `monthly`, then this should be set to the desired day of the month.
- `exceptions` - (Optional, list) Provide a list of exception entries.

panos_email_server_profile

This resource allows you to add/update/delete email server profiles.

Import Name

```
<vsys>:<name>
```

Example Usage

```
resource "panos_email_server_profile" "example" {
    name = "myProfile"
    threat_format = "$serial $severity"
    email_server {
        name = "my-server"
        display_name = "foobar"
        from_email = "source@example.com"
        to_email = "alerts@example.com"
        email_gateway = "mail.example.com"
    }
}
```

Argument Reference

The following arguments are supported:

- `vsys` - (Optional) The vsys (default: `shared`).
- `name` - (Required) The group's name.
- `config_format` - (Optional) Config format.
- `system_format` - (Optional) System format.
- `threat_format` - (Optional) Threat format.
- `traffic_format` - (Optional) Traffic format.
- `hip_match_format` - (Optional) HIP match format.
- `url_format` - (Optional) URL format.
- `data_format` - (Optional) Data format.
- `wildfire_format` - (Optional) Wildfire format.
- `tunnel_format` - (Optional) Tunnel format.

- `user_id_format` - (Optional) UserID format.
- `gtp_format` - (Optional) GTP format.
- `auth_format` - (Optional) Auth format.
- `sctp_format` - (Optional) SCTP format.
- `iptag_format` - (Optional) IP tag format.
- `escaped_characters` - (Optional) The escaped characters (as a string).
- `escape_character` - (Optional) The escape character.
- `email_server` - (Required, repeatable) The server spec (defined below).

`email_server` supports the following arguments:

- `name` - (Required) Server name.
- `display_name` - (Optional) The display name.
- `from_email` - (Required) From email address.
- `to_email` - (Required) To email address.
- `also_to_email` - (Optional) Secondary to email address.
- `email_gateway` - (Required) The email server.

panos_ethernet_interface

This resource allows you to add/update/delete ethernet interfaces.

Import Name

```
<vsys>:<name>
```

Example Usage

```
# Configure a bare-bones ethernet interface.
resource "panos_ethernet_interface" "example1" {
    name = "ethernet1/3"
    vsys = "vsys1"
    mode = "layer3"
    static_ips = ["10.1.1.1/24"]
    comment = "Configured for internal traffic"
}

# Configure a DHCP ethernet interface for vsys1 to use.
resource "panos_ethernet_interface" "example2" {
    name = "ethernet1/4"
    vsys = "vsys1"
    mode = "layer3"
    enable_dhcp = true
    create_dhcp_default_route = true
    dhcp_default_route_metric = 10
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The ethernet interface's name. This should be something like `ethernet1/X`.
- `vsys` - (Required) The vsys that will use this interface. This should be something like `vsys1` or `vsys3`.
- `mode` - (Required) The interface mode. This can be any of the following values: `layer3`, `layer2`, `virtual-wire`, `tap`, `ha`, `decrypt-mirror`, or `aggregate-group`.
- `static_ips` - (Optional) List of static IPv4 addresses to set for this data interface.
- `enable_dhcp` - (Optional) Set to `true` to enable DHCP on this interface.
- `create_dhcp_default_route` - (Optional) Set to `true` to create a DHCP default route.
- `dhcp_default_route_metric` - (Optional) The metric for the DHCP default route.

- `ipv6_enabled` - (Optional) Set to `true` to enable IPv6.
- `management_profile` - (Optional) The management profile.
- `mtu` - (Optional) The MTU.
- `adjust_tcp_mss` - (Optional) Adjust TCP MSS (default: false).
- `netflow_profile` - (Optional) The netflow profile.
- `lldp_enabled` - (Optional) Enable LLDP (default: false).
- `lldp_profile` - (Optional) LLDP profile.
- `link_speed` - (Optional) Link speed. This can be any of the following: `10`, `100`, `1000`, or `auto`.
- `link_duplex` - (Optional) Link duplex setting. This can be `full`, `half`, or `auto`.
- `link_state` - (Optional) The link state. This can be `up`, `down`, or `auto`.
- `aggregate_group` - (Optional) The aggregate group (applicable for physical firewalls only).
- `comment` - (Optional) The interface comment.
- `ipv4_mss_adjust` - (Optional, PAN-OS 7.1+) The IPv4 MSS adjust value.
- `ipv6_mss_adjust` - (Optional, PAN-OS 7.1+) The IPv6 MSS adjust value.
- `decrypt_forward` - (Optional, PAN-OS 8.1+) Enable decrypt forwarding.
- `rx_policing_rate` - (Optional, PAN-OS 8.1+) Receive policing rate in Mbps.
- `tx_policing_rate` - (Optional, PAN-OS 8.1+) Transmit policing rate in Mbps.
- `dhcp_send_hostname_enable` - (Optional, PAN-OS 9.0+) For DHCP layer3 interfaces: enable sending the firewall or a custom hostname to DHCP server
- `dhcp_send_hostname_value` - (Optional, PAN-OS 9.0+) For DHCP layer3 interfaces: the interface hostname. Leaving this unspecified with `dhcp_send_hostname_enable` set means to send the system hostname.

panos_general_settings

This resource allows you to update the general device settings, such as DNS or the hostname.

All params are optional for this resource. If any options are not specified, then whatever is already configured on the firewall is left as-is. The general device settings will always exist on the firewall, so `terraform destroy` does not remove config from the firewall.

Example Usage

```
resource "panos_general_settings" "example" {
    hostname = "ngfw220"
    dns_primary = "10.5.1.10"
    ntp_primary = "10.5.1.10"
    ntp_primary_auth_type = "none"
}
```

Argument Reference

The following arguments are supported:

- `hostname` - Firewall hostname.
- `timezone` - The timezone (e.g. - `US/Pacific`).
- `domain` - The domain.
- `update_server` - The update server (Default: `updates.paloaltonetworks.com`).
- `verify_update_server` - Verify update server identity (Default: `true`).
- `proxy_server` - (1.5+) Specify a proxy server.
- `proxy_port` - (int, 1.5+) Proxy's port number.
- `proxy_username` - (1.5+) Proxy's username.
- `proxy_password` - (1.5+) Proxy's password.
- `dns_primary` - Primary DNS server.
- `dns_secondary` - Secondary DNS server.
- `ntp_primary_address` - Primary NTP server.
- `ntp_primary_auth_type` - Primary NTP auth type. This can be `none` , `autokey` , or `symmetric-key` .
- `ntp_primary_key_id` - Primary NTP `symmetric-key` key ID.
- `ntp_primary_algorithm` - Primary NTP `symmetric-key` algorithm. This can be `sha1` or `md5` .
- `ntp_primary_auth_key` - Primary NTP `symmetric-key` auth key. This is the SHA1 hash if the algorithm is `sha1` , or

the md5sum if the algorithm is `md5`.

- `ntp_secondary_address` - Secondary NTP server.
- `ntp_secondary_auth_type` - Secondary NTP auth type. This can be `none`, `autokey`, or `symmetric-key`.
- `ntp_secondary_key_id` - Secondary NTP `symmetric-key` key ID.
- `ntp_secondary_algorithm` - Secondary NTP `symmetric-key` algorithm. This can be `sha1` or `md5`.
- `ntp_secondary_auth_key` - Secondary NTP `symmetric-key` auth key. This is the SHA1 hash if the algorithm is `sha1`, or the md5sum if the algorithm is `md5`.

panos_gre_tunnel

This resource allows you to add/update/delete GRE tunnels.

Minimum PAN-OS version: 9.0

Import Name

```
<name>
```

Example Usage

```
resource "panos_gre_tunnel" "example" {
    name = "myGreTunnel"
    interface = panos_ethernet_interface.ei.name
    local_address_value = panos_ethernet_interface.ei.static_ips.0
    peer_address = "192.168.1.1"
    tunnel_interface = panos_tunnel_interface.ti.name
    ttl = 42
}

resource "panos_ethernet_interface" "ei" {
    name = "ethernet1/1"
    vsys = "vsys1"
    mode = "layer3"
    static_ips = ["10.1.1.1/24"]
}

resource "panos_tunnel_interface" "ti" {
    name = "tunnel.7"
    vsys = "vsys1"
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The GRE tunnel name.
- `interface` - (Required) Interface to terminate tunnel.
- `local_address_type` - (Optional) Type of local address. Valid values are `ip` (default) or `floating-ip`.
- `local_address_value` - (Required) IP address value.
- `peer_address` - (Required) Peer IP address.
- `tunnel_interface` - (Required) Tunnel interface to apply the GRE tunnel to.

- `ttl` - (Optional, int) Time to live.
- `copy_tos` - (Optional, bool) Copy IP TOS bits from inner packet to GRE packet.
- `enable_keep_alive` - (Optional, bool) Enable tunnel monitoring.
- `keep_alive_interval` - (Optional, int) Keep alive interval.
- `keep_alive_retry` - (Optional, int) Keep alive retry.
- `keep_alive_hold_timer` - (Optional, int) Keep alive hold timer.
- `disabled` - (Optional, bool) Disable the GRE tunnel.

panos_http_server_profile

This resource allows you to add/update/delete HTTP server profiles.

Minimum PAN-OS version: 7.1

Example Usage

```
resource "panos_http_server_profile" "example" {
    name = "myProfile"
    url_format {
        name = "my url format"
        uri_format = "/api/incident/url"
        headers = {
            "Content-Type": "text/plain",
        }
    }
    http_server {
        name = "myServer"
        address = "siem.example.com"
    }
}
```

Argument Reference

The following arguments are supported:

- `vsys` - (Optional) The vsys (default: `shared`).
- `name` - (Required) The group's name.
- `tag_registration` - (Optional, bool) Perform tag registration.
- `config_format` - (Optional) A format folder spec for config (defined below).
- `system_format` - (Optional) A format folder spec for system (defined below).
- `threat_format` - (Optional) A format folder spec for threat (defined below).
- `traffic_format` - (Optional) A format folder spec for traffic (defined below).
- `hip_match_format` - (Optional) A format folder spec for HIP match (defined below).
- `url_format` - (Optional) A format folder spec for url (defined below).
- `data_format` - (Optional) A format folder spec for data (defined below).
- `wildfire_format` - (Optional) A format folder spec for wildfire (defined below).
- `tunnel_format` - (Optional) A format folder spec for tunnel (defined below).
- `user_id_format` - (Optional) A format folder spec for user ID (defined below).

- `gtp_format` - (Optional) A format folder spec for gtp (defined below).
- `auth_format` - (Optional) A format folder spec for auth (defined below).
- `sctp_format` - (Optional, PAN-OS 8.1+) A format folder spec for sctp (defined below).
- `iptag_format` - (Optional, PAN-OS 9.0+) A format folder spec for iptag (defined below).
- `http_server` - (Optional, repeatable) A server spec (defined below).

All format folders support the following arguments:

- `name` - (Optional) The name.
- `uri_format` - (Optional) The URI format.
- `payload` - (Optional) The payload.
- `headers` - (Optional, map) A map of HTTP headers and their values.
- `params` - (Optional, map) A map of HTTP params and their values.

`http_server` supports the following arguments:

- `name` - (Required) The server name.
- `address` - (Required) The server address.
- `protocol` - (Optional) The protocol. Valid values are `HTTPS` (default) or `HTTP`.
- `port` - (Optional, int) The port number (default: 443).
- `http_method` - (Optional) The HTTP method (default: `POST`).
- `username` - (Optional) The username.
- `password` - (Optional) The password.
- `tls_version` - (Optional) The TLS version.
- `certificate_profile` - (Optional) The certificate profile.

panos_ike_crypto_profile

This resource allows you to add/update/delete IKE crypto profiles.

Import Name

```
<name>
```

Example Usage

```
resource "panos_ike_crypto_profile" "example" {
    name = "example"
    dh_groups = ["group1", "group2"]
    authentications = ["md5", "sha1"]
    encryptions = ["des"]
    lifetime_value = 8
    authentication_multiple = 3
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The object's name
- `dh_groups` - (Required, list) List of DH Group entries. Values should have a prefix if `group`.
- `authentications` - (Required, list) List of authentication types. This c
- `encryptions` - (Required, list) List of encryption types. Valid values are `des`, `3des`, `aes-128-cbc`, `aes-192-cbc`, and `aes-256-cbc`.
- `lifetime_type` - (Optional) The lifetime type. Valid values are `seconds`, `minutes`, `hours` (the default), and `days`.
- `lifetime_value` - (Optional, int) The lifetime value.
- `authentication_multiple` - (Optional, PAN-OS 7.0+, int) IKEv2 SA reauthentication interval equals authetication-multiple * rekey-lifetime; 0 means reauthentication is disabled.

panos_ike_gateway

This resource allows you to add/update/delete IKE gateways.

Example Usage

```
resource "panos_ike_gateway" "example" {
    name = "example"
    peer_ip_type = "dynamic"
    interface = "loopback.42"
    pre_shared_key = "secret"
    local_id_type = "ipaddr"
    local_id_value = "10.1.1.1"
    peer_id_type = "ipaddr"
    peer_id_value = "10.5.1.1"
    ikev1_crypto_profile = "myIkeProfile"
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The object's name
- `version` - (Optional, PAN-OS 7.0+) The IKE gateway version. Valid values are `ikev1`, (the default), `ikev2`, or `ikev2-preferred`. For PAN-OS 6.1, only `ikev1` is acceptable.
- `enable_ipv6` - (Optional, PAN-OS 7.0+, bool) Enable IPv6 or not.
- `disabled` - (Optional, PAN-OS 7.0+, bool) Set to `true` to disable.
- `peer_ip_type` - (Optional) The peer IP type. Valid values are `ip`, `dynamic`, and `fqdn` (PANOS 8.1+).
- `peer_ip_value` - (Optional) The peer IP value.
- `interface` - (Required) The interface.
- `local_ip_address_type` - (Optional) The local IP address type. Valid values for this are `ip`, `floating-ip`, or an empty string (the default) which is `None`.
- `local_ip_address_value` - (Optional) The IP address if `local_ip_address_type` is set to `ip`.
- `auth_type` - (Optional) The auth type. Valid values are `pre-shared-key` (the default), or `certificate`.
- `pre_shared_key` - (Optional) The pre-shared key value.
- `local_id_type` - (Optional) The local ID type. Valid values are `ipaddr`, `fqdn`, `ufqdn`, `keyid`, or `dn`.
- `local_id_value` - (Optional) The local ID value.
- `peer_id_type` - (Optional) The peer ID type. Valid values are `ipaddr`, `fqdn`, `ufqdn`, `keyid`, or `dn`.

- `peer_id_value` - (Optional) The peer ID value.
- `peer_id_check` - (Optional) Enable peer ID wildcard match for certificate authentication. Valid values are `exact` or `wildcard`.
- `local_cert` - (Optional) The local certificate name.
- `cert_enable_hash_and_url` - (Optional, PAN-OS 7.0+, bool) Set to `true` to use hash-and-url for local certificate.
- `cert_base_url` - (Optional) The host and directory part of URL for local certificates.
- `cert_use_management_as_source` - (Optional, PAN-OS 7.0+, bool) Set to `true` to use management interface IP as source to retrieve http certificates
- `cert_permit_payload_mismatch` - (Optional, bool) Set to `true` to permit peer identification and certificate payload identification mismatch.
- `cert_profile` - (Optional) Profile for certificate valdiation during IKE negotiation.
- `cert_enable_strict_validation` - (Optional, bool) Set to `true` to enable strict validation of peer's extended key use.
- `enable_passive_mode` - (Optional, bool) Set to `true` to enable passive mode (responder only).
- `enable_nat_traversal` - (Optional, bool) Set to `true` to enable NAT traversal.
- `nat_traversal_keep_alive` - (Optional, int) Sending interval for NAT keep-alive packets (in seconds). For versions 6.1 - 8.1, this param, if specified, should be a multiple of 10 between 10 and 3600 to be valid.
- `nat_traversal_enable_udp_checksum` - (Optional, bool) Set to `true` to enable NAT traversal UDP checksum.
- `enable_fragmentation` - (Optional, bool) Set to `true` to enable fragmentation.
- `ikev1_exchange_mode` - (Optional) The IKEv1 exchange mode.
- `ikev1_crypto_profile` - (Optional) IKEv1 crypto profile.
- `enable_dead_peer_detection` - (Optional, bool) Set to `true` to enable dead peer detection.
- `dead_peer_detection_interval` - (Optional, int) The dead peer detection interval.
- `dead_peer_detection_retry` - (Optional, int) Number of retries before disconnection.
- `ikev2_crypto_profile` - (Optional, PAN-OS 7.0+) IKEv2 crypto profile.
- `ikev2_cookie_validation` - (Optional, PAN-OS 7.0+) Set to `true` to require cookie.
- `enable_liveness_check` - (Optional, , PAN-OS 7.0+bool) Set to `true` to enable sending empty information liveness check message.
- `liveness_check_interval` - (Optional, , PAN-OS 7.0+int) Delay interval before sending probing packets (in seconds).

panos_ipsec_crypto_profile

This resource allows you to add/update/delete IPSec crypto profiles.

Import Name

```
<name>
```

Example Usage

```
resource "panos_ipsec_crypto_profile" "example" {
    name = "example"
    authentications = ["md5", "sha384"]
    encryptions = ["des", "aes-128-cbc"]
    dh_group = "group14"
    lifetime_type = "hours"
    lifetime_value = 4
    lifesize_type = "mb"
    lifesize_value = 1
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The object's name
- `protocol` - (Optional) The protocol. Valid values are `esp` (the default) or `ah`
- `authentications` - (Required, list) - List of authentication types.
- `encryptions` - (Required, list) - List of encryption types. Valid values are `des`, `3des`, `aes-128-cbc`, `aes-192-cbc`, `aes-256-cbc`, `aes-128-gcm`, `aes-256-gcm`, and `null`. Note that the "gcm" values are only available in PAN-OS 7.0+.
- `dh_group` - (Optional) The DH group value. Valid values should start with the string `group`.
- `lifetime_type` - (Optional) The lifetime type. Valid values are `seconds`, `minutes`, `hours` (the default), or `days`.
- `lifetime_value` - (Optional, int) The lifetime value.
- `lifesize_type` - (Optional) The lifesize type. Valid values are `kb`, `mb`, `gb`, or `tb`.
- `lifesize_value` - (Optional, int) the lifesize value.

panos_ipsec_tunnel

This resource allows you to add/update/delete IPSec tunnels.

A large number of params have prefixes:

- `ak` - Auto key
- `mk` - Manual key
- `gps` - GlobalProtect Satellite

Example Usage

```
resource "panos_ipsec_tunnel" "example" {
    name = "example"
    tunnel_interface = "tunnel.7"
    anti_replay = true
    ak_ike_gateway = "myIkeGateway"
    ak_ipsec_crypto_profile = "myIkeProfile"
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The object's name
- `tunnel_interface` - (Required) The tunnel interface.
- `anti_replay` - (Optional, bool) Set to `true` to enable Anti-Replay check on this tunnel.
- `enable_ipv6` - (Optional, PAN-OS 7.0+, bool) Set to `true` to enable IPv6.
- `copy_tos` - (Optional, bool) Set to `true` to copy IP TOS bits from inner packet to IPSec packet (not recommended).
- `copy_flow_label` - (Optional, PAN-OS 7.0+, bool) Set to `true` to copy IPv6 flow label for 6in6 tunnel from inner packet to IPSec packet (not recommended).
- `disabled` - (Optional, PAN-OS 7.0+, bool) Set to `true` to disable this IPSec tunnel.
- `type` - (Optional) The type. Valid values are `auto-key` (the default), `manual-key`, or `global-protect-satellite`.
- `ak_ike_gateway` - (Optional) IKE gateway name.
- `ak_ipsec_crypto_profile` - (Optional) IPSec crypto profile name.
- `mk_local_spi` - (Optional) Outbound SPI, hex format.
- `mk_remote_spi` - (Optional) Inbound SPI, hex format.
- `mk_local_address_ip` - (Optional) Specify exact IP address if interface has multiple addresses.

- `mk_local_address_floating_ip` - (Optional) Floating IP address in HA Active-Active configuration.
- `mk_protocol` - (Optional) Manual key protocol. Valid valies are `esp` or `ah`.
- `mk_auth_type` - (Optional) Authentication algorithm. Valid values are `md5`, `sha1`, `sha256`, `sha384`, `sha512`, or `none`.
- `mk_auth_key` - (Optional) The auth key for the given auth type.
- `mk_esp_encryption_type` - (Optional) The encryption algorithm. Valid values are `des`, `3des`, `aes-128-cbc`, `aes-192-cbc`, `aes-256-cbc`, or `null`.
- `mk_esp_encryption_key` - (Optional) The encryption key.
- `gps_interface` - (Optional) Interface to communicate with portal.
- `gps_portal_address` - (Optional) GlobalProtect portal address.
- `gps_prefer_ipv6` - (Optional, PAN-OS 8.0+, bool) Prefer to register the portal in IPv6. Only applicable to FQDN portal-address.
- `gps_interface_ip_ipv4` - (Optional) specify exact IP address if interface has multiple addresses (IPv4).
- `gps_interface_ip_ipv6` - (Optional, PAN-OS 8.0+) specify exact IP address if interface has multiple addresses (IPv6).
- `gps_interface_floating_ip_ipv4` - (Optional, PAN-OS 7.0+) Floating IPv4 address in HA Active-Active configuration.
- `gps_interface_floating_ip_ipv6` - (Optional, PAN-OS 8.0+) Floating IPv6 address in HA Active-Active configuration.
- `gps_publish_connected_routes` - (Optional, bool) Set to `true` to to publish connected and static routes.
- `gps_publish_routes` - (Optional, list) Specify list of routes to publish to Global Protect Gateway.
- `gps_local_certificate` - (Optional) GlobalProtect satellite certificate file name.
- `gps_certificate_profile` - (Optional) Profile for authenticating GlobalProtect gateway certificates.
- `enable_tunnel_monitor` - (Optional, bool) Enable tunnel monitoring on this tunnel.
- `tunnel_monitor_destination_ip` - (Optional) Destination IP to send ICMP probe.
- `tunnel_monitor_source_ip` - (Optional) Source IP to send ICMP probe
- `tunnel_monitor_profile` - (Optional) Tunnel monitor profile.
- `tunnel_monitor_proxy_id` - (Optional, PAN-OS 7.0+) Which proxy-id (or proxy-id-v6) the monitoring traffic will use.

panos_ipsec_tunnel_proxy_id_ipv4

This resource allows you to add/update/delete IPSec tunnel proxy IDs to a parent auto key IPSec tunnel.

Import Name

```
<ipsec_tunnel>:<name>
```

Example Usage

```
resource "panos_ipsec_tunnel_proxy_id_ipv4" "example" {
    ipsec_tunnel = "myIpsecTunnel"
    name = "example"
    local = "10.1.1.1"
    remote = "10.2.1.1"
    protocol_any = true
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The object's name
- `ipsec_tunnel` - (Required) The auto key IPSec tunnel to attach this proxy ID to.
- `local` - (Optional) IP subnet or IP address represents local network.
- `remote` - (Optional) IP subnet or IP address represents remote network.
- `protocol_any` - (Optional, bool) Set to `true` for any IP protocol.
- `protocol_number` - (Optional, int) IP protocol number.
- `protocol_tcp_local` - (Optional, int) Local TCP port number.
- `protocol_tcp_remote` - (Optional, int) Remote TCP port number.
- `protocol_udp_local` - (Optional, int) Local UDP port number.
- `protocol_udp_remote` - (Optional, int) Remote UDP port number.

panos_layer2_subinterface

This resource allows you to add/update/delete layer2 subinterfaces.

Import Name

```
<interface_type>:<parent_interface>:<parent_mode>:<vsys>:<name>
```

Example Usage

```
resource "panos_layer2_subinterface" "example" {
    parent_interface = panos_ethernet_interface.e.name
    parent_mode = panos_ethernet_interface.e.mode
    vsys = "vsys1"
    name = "ethernet1/5.5"
    tag = 5
}

resource "panos_ethernet_interface" "e" {
    name = "ethernet1/5"
    vsys = "vsys1"
    mode = "layer2"
}
```

Argument Reference

The following arguments are supported:

- `interface_type` - (Optional) The interface type. Valid values are `ethernet` (default) or `aggregate-ethernet`.
- `parent_interface` - (Required) The name of the parent interface.
- `parent_mode` - (Optional) The parent's mode. Valid values are `layer2` (default) or `virtual-wire`.
- `vsys` - (Required) The vsys that will use this interface. This should be something like `vsys1` or `vsys3`.
- `name` - (Required) The interface's name.
- `tag` - (Optional, int) The interface's tag.
- `netflow_profile` - (Optional) The netflow profile.
- `comment` - (Optional) The interface comment.

panos_layer3_subinterface

This resource allows you to add/update/delete layer3 subinterfaces.

Import Name

```
<interface_type>:<parent_interface>:<vsys>:<name>
```

Example Usage

```
resource "panos_layer3_subinterface" "example" {
    parent_interface = panos_ethernet_interface.e.name
    vsys = "vsys1"
    name = "ethernet1/5.5"
    tag = 5
    static_ips = ["10.1.1.1/24"]
    comment = "Configured for internal traffic"
}

resource "panos_ethernet_interface" "e" {
    name = "ethernet1/5"
    vsys = "vsys1"
    mode = "layer3"
}
```

Argument Reference

The following arguments are supported:

- `interface_type` - (Optional) The interface type. Valid values are `ethernet` (default) or `aggregate-ethernet`.
- `parent_interface` - (Required) The name of the parent interface.
- `vsys` - (Required) The vsys that will use this interface. This should be something like `vsys1` or `vsys3`.
- `name` - (Required) The interface's name.
- `tag` - (Optional, int) The interface's tag.
- `static_ips` - (Optional) List of static IPv4 addresses.
- `ipv6_enabled` - (Optional, bool) Set to `true` to enable IPv6.
- `ipv6_interface_id` - (Optional) The IPv6 interface ID.
- `management_profile` - (Optional) The management profile.
- `mtu` - (Optional) The MTU.

- `adjust_tcp_mss` - (Optional) Adjust TCP MSS (default: false).
- `ipv4_mss_adjust` - (Optional) The IPv4 MSS adjust value.
- `ipv6_mss_adjust` - (Optional) The IPv6 MSS adjust value.
- `netflow_profile` - (Optional) The netflow profile.
- `enable_dhcp` - (Optional, bool) Set to `true` to enable DHCP.
- `create_dhcp_default_route` - (Optional) Set to `true` to create a DHCP default route.
- `dhcp_default_route_metric` - (Optional) The metric for the DHCP default route.
- `comment` - (Optional) The interface comment.
- `decrypt_forward` - (Optional, bool, PAN-OS 8.1+) Set to `true` to enable decrypt forward.
- `dhcp_send_hostname_enable` - (Optional, PAN-OS 9.0+) For DHCP layer3 interfaces: enable sending the firewall or a custom hostname to DHCP server
- `dhcp_send_hostname_value` - (Optional, PAN-OS 9.0+) For DHCP layer3 interfaces: the interface hostname. Leaving this unspecified with `dhcp_send_hostname_enable` set means to send the system hostname.

panos_license_api_key

This resource manages the licensing API key, which is necessary to delicense the PAN-OS firewall.

This resource's `retain_key` param is a Terraform side configuration only. In order for the firewall to delicense itself, the licensing API key must be present. This means that either the `panos_licensing` resource must use `depends_on` and depend on this resource, or you must set the `retain_key` param to `true`. As there is no harm in leaving the licensing API key on the PAN-OS firewall, it is recommended that `retain_key` be set to `true`.

Example Usage

```
resource "panos_license_api_key" "example" {
    key = "secret"
    retain_key = true
}
```

Argument Reference

The following arguments are supported:

- `key` - (Required) The licensing API key.
- `retain_key` - (Optional) Set to `true` to retain the licensing API key even after the deletion of this resource (recommended).

panos_licensing

This resource manages the licenses installed on the PAN-OS firewall.

Installing the standard auth code for the standard PAN-OS license key for the firewall causes the firewall to reboot. Thus it is recommended that you use this resource in a separate step of your overall firewall provisioning, as using this resource will cause the firewall to be temporarily inaccessible.

Example Usage

```
resource "panos_licensing" "example" {
    auth_codes = ["code1", "code2"]
}
```

Argument Reference

The following arguments are supported:

- `auth_codes` - (Required) The list of auth codes to install.
- `delicense` - (Optional, bool) Leave as `true` if you want to delicense the firewall when this resource is removed, otherwise set to `false` to prevent firewall delicensing. Delicensing requires that the licensing API key has been installed.
- `mode` - (Optional) For `delicense` of `true`, the type of delicensing to perform. Right now, only `auto` is supported (no manual delicensing).

Attribute Reference

The following attributes are available after read operations:

- `licenses` - List of licenses.

Licenses have the following attributes:

- `feature` - The feature name.
- `description` - License description.
- `serial` - The serial number.
- `issued` - When the license was issued.
- `expires` - When the license expires.
- `expired` - If the license has expired or not.
- `auth_code` - Associated auth code (if applicable).

panos_log_forwarding_profile

This resource allows you to add/update/delete log forwarding profiles.

Minimum PAN-OS version: 8.0

Import Name

```
<vsys>:<name>
```

Example Usage

```
resource "panos_log_forwarding_profile" "example" {
    name = "myProfile"
    description = "made by Terraform"
    match_list {
        name = "myMatchList"
        log_type = "url"
        http_server_profiles = [
            panos_http_server_profile.h1.name,
            panos_http_server_profile.h2.name,
        ]
        action {
            name = "tagging int"
            tagging_integration {
                timeout = 5
                local_registration {
                    tags = [
                        panos_administrative_tag.t.name,
                    ]
                }
            }
        }
        action {
            name = "azure int"
            azure_integration { }
        }
    }
}

resource "panos_http_server_profile" "h1" {
    name = "h1"
    http_server {
        name = "h1"
        address = "h1.example.com"
    }
}

resource "panos_http_server_profile" "h2" {
    name = "h2"
    http_server {
        name = "h2"
        address = "h2.example.com"
    }
}

resource "panos_administrative_tag" "t" {
    name = "myTag"
    color = "color12"
}
```

Argument Reference

The following arguments are supported:

- `vsys` - (Optional) The vsys (default: `vsys1`).

- `name` - (Required) The group's name.
- `description` - (Optional) The description.
- `enhanced_logging` - (Optional, bool, PAN-OS 8.1+) Set to `true` to enable enhanced logging.
- `match_list` - (Optional, repeatable) The match list spec (defined below).

`match_list` supports the following arguments:

- `name` - (Required) The name.
- `description` - (Optional) The description.
- `log_type` - (Optional) The log type. Valid values are `traffic` (default), `threat`, `wildfire`, `url`, `data`, `gtp`, `tunnel`, `auth`, or `sctp`.
- `filter` - (Optional) The filter (default: `All Logs`).
- `send_to_panorama` - (Optional, bool) Set to `true` to send to Panorama.
- `snmptrap_server_profiles` - (Optional) List of SNMP server profiles.
- `email_server_profiles` - (Optional) List of email server profiles.
- `syslog_server_profiles` - (Optional) List of syslog server profiles.
- `http_server_profiles` - (Optional) List of http server profiles.
- `action` - (Optional, repeatable) Match list action spec (defined below).

`match_list.action` supports the following arguments:

- `name` - (Required) The name.
- `azure_integration` - (Optional) The Azure integration spec (defined below). Mutually exclusive with `tagging_integration`.
- `tagging_integration` - (Optional) The tagging integration spec (defined below). Mutually exclusive with `azure_integration`.

`match_list.action.azure_integration` supports the following arguments:

- `azure_integration` - (Optional, bool) This param defaults to `true` and should not be changed.

`match_list.action.tagging_integration` supports the following arguments:

- `action` - (Optional) The action. Valid values are `add-tag` (default) or `remove-tag`.
- `target` - (Optional) The target. Valid values are `source-address` (default) or `destination-address`.
- `timeout` - (Optional, int) The timeout.
- `local_registration` - (Optional) The local registration spec (defined below). Only one of `local_registration`, `remote_registration`, or `panorama_registration` should be defined.
- `remote_registration` - (Optional) The remote registration spec (defined below). Only one of `local_registration`, `remote_registration`, or `panorama_registration` should be defined.
- `panorama_registration` - (Optional) The panorama registration spec (defined below). Only one of

`local_registration`, `remote_registration`, or `panorama_registration` should be defined.

`match_list.action.tagging_integration.local_registration` supports the following arguments:

- `tags` - (Required) List of administrative tags.

`match_list.action.tagging_integration.remote_registration` supports the following arguments:

- `tags` - (Required) List of administrative tags.
- `http_profile` - (Required) The HTTP profile.

`match_list.action.tagging_integration.panorama_registration` supports the following arguments:

- `tags` - (Required) List of administrative tags.

panos_loopback_interface

This resource allows you to add/update/delete loopback interfaces.

Import Name

```
<vsys>:<name>
```

Example Usage

```
resource "panos_loopback_interface" "example1" {
    name = "loopback.2"
    comment = "my loopback interface"
    static_ips = ["10.1.1.1"]
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The interface's name. This must start with `loopback.`.
- `vsys` - (Optional) The vsys that will use this interface (default: `vsys1`).
- `comment` - (Optional) The interface comment.
- `netflow_profile` - (Optional) The netflow profile.
- `static_ips` - (Optional) List of static IPv4 addresses to set for this data interface.
- `management_profile` - (Optional) The management profile.
- `mtu` - (Optional) The MTU.
- `adjust_tcp_mss` - (Optional, bool) Adjust TCP MSS (default: false).
- `ipv4_mss_adjust` - (Optional, PAN-OS 8.0+) The IPv4 MSS adjust value.
- `ipv6_mss_adjust` - (Optional, PAN-OS 8.0+) The IPv6 MSS adjust value.

panos_management_profile

This resource allows you to add/update/delete interface management profiles.

Import Name

```
<name>
```

Example Usage

```
resource "panos_management_profile" "example" {
    name = "allow ping"
    ping = true
    permitted_ips = ["10.1.1.0/24", "192.168.80.0/24"]
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The management profile's name.
- `ping` - (Optional) Allow ping.
- `telnet` - (Optional) Allow telnet.
- `ssh` - (Optional) Allow SSH.
- `http` - (Optional) Allow HTTP.
- `http_ocsp` - (Optional) Allow HTTP OCSP.
- `https` - (Optional) Allow HTTPS.
- `snmp` - (Optional) Allow SNMP.
- `response_pages` - (Optional) Allow response pages.
- `userid_service` - (Optional) Allow User ID service.
- `userid_syslog_listener_ssl` - (Optional) Allow User ID syslog listener for SSL.
- `userid_syslog_listener_udp` - (Optional) Allow User ID syslog listener for UDP.
- `permitted_ips` - (Optional) The list of permitted IP addresses or address ranges for this management profile.

panos_monitor_profile

This resource allows you to add/update/delete monitor profiles.

Minimum PAN-OS version: 7.1

Import Name

```
<name>
```

Example Usage

```
resource "panos_monitor_profile" "example" {
    name = "myProfile"
    interval = 5
    threshold = 3
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The monitor profile name.
- `interval` - (Optional, int) The probing interval in seconds.
- `threshold` - (Optional, int) The number of failed probes to determine that the tunnel is down.
- `action` - (Optional) Action triggered when tunnel's status changes. Valid values are `wait-recover` (default) or `fail-over`.

panos_nat_rule_group

This resource allows you to add/update/delete a group of NAT rules.

This resource manages clusters of NAT rules in a single vsys, enforcing both the contents of individual rules as well as their ordering. Rules are defined in a `rule` config block.

Although you cannot modify non-group NAT rules with this resource, the `position_keyword` and `position_reference` parameters allow you to reference some other NAT rule that already exists, using it as a means to ensure some rough placement within the ruleset as a whole.

Best Practices

As is to be expected, if you are separating your deployment across multiple plan files, make sure that at most only one plan specifies any given absolute positioning keyword such as "top" or "directly below", otherwise they'll keep shoving each other out of the way indefinitely.

Best practices are to specify one group as `top` (if you need it), one group as `bottom` (if needed), then all other groups should be `above` the first rule of the bottom group. You do it this way because rules will natually be added at the tail end of the rulebase, so they will always be `after` the first group, but what you want is for them to be `before` the last group's rules.

Example Usage

```
resource "panos_nat_rule_group" "bot" {
    rule {
        name = "second"
        original_packet {
            source_zones = ["${panos_zone.z2.name}"]
            destination_zone = "${panos_zone.z3.name}"
            destination_interface = "${panos_ethernet_interface.x.name}"
            source_addresses = ["any"]
            destination_addresses = ["any"]
        }
        translated_packet {
            source {}
            destination {
                static_translation {
                    address = "10.2.3.1"
                    port = 5678
                }
            }
        }
    }
    rule {
        name = "third"
        original_packet {
            source_zones = ["${panos_zone.z3.name}"]
            destination_zone = "${panos_zone.z2.name}"
            destination_interface = "${panos_ethernet_interface.x.name}"
            source_addresses = ["any"]
```

```
            destination_addresses = ["any"]
        }
        translated_packet {
            source {
                static_ip {
                    translated_address = "192.168.1.5"
                    bi_directional = true
                }
            }
            destination {}
        }
    }
}

resource "panos_nat_rule_group" "top" {
    position_keyword = "directly before"
    position_reference = "${panos_nat_rule_group.bot.rule.0.name}"
    rule {
        name = "first"
        original_packet {
            source_zones = ["${panos_zone.z1.name}"]
            destination_zone = "${panos_zone.z1.name}"
            destination_interface = "${panos_ethernet_interface.x.name}"
            source_addresses = ["any"]
            destination_addresses = ["any"]
        }
        translated_packet {
            source {
                dynamic_ip_and_port {
                    interface_address {
                        interface = "${panos_ethernet_interface.x.name}"
                        ip_address = "${panos_ethernet_interface.x.static_ips.0}"
                    }
                }
            }
            destination {
                static_translation {
                    address = "10.1.1.1"
                    port = 1234
                }
            }
        }
    }
}

resource "panos_ethernet_interface" "x" {
    name = "ethernet1/6"
    mode = "layer3"
    vsys = "vsys1"
    static_ips = ["10.5.5.1/24"]
}

resource "panos_zone" "z1" {
    name = "z1"
    mode = "layer3"
}

resource "panos_zone" "z2" {
    name = "z2"
```

```
    mode = "layer3"
}

resource "panos_zone" "z3" {
    name = "z3"
    mode = "layer3"
}
```

Argument Reference

The following arguments are supported:

- `vsys` - (Optional) The vsys to put the NAT rule group into (default: `vsys1`).
- `position_keyword` - (Optional) A positioning keyword for this group. This can be `before`, `directly before`, `after`, `directly after`, `top`, `bottom`, or left empty (the default) to have no particular placement. This param works in combination with the `position_reference` param.
- `position_reference` - (Optional) Required if `position_keyword` is one of the "above" or "below" variants, this is the name of a non-group rule to use as a reference to place this group.
- `rule` - (Repeatable) The rule definition (see below). The rule ordering will match how they appear in the terraform plan file.

Each `rule` defined supports the following arguments:

- `name` - (Required) The NAT rule's name.
- `description` - (Optional) The description.
- `type` - (Optional). NAT type. This can be `ipv4` (default), `nat64`, or `nptv6`.
- `tags` - (Optional) List of administrative tags.
- `disabled` - (Optional) Set to `true` to disable this rule.
- `original_packet` - (Required) The original packet specification (see below).
- `translated_packet` - (Required) The translated packet spec (see below).

`original_packet` supports the following arguments:

- `source_zones` - (Required) The list of source zone(s).
- `destination_zone` - (Required) The destination zone.
- `destination_interface` - (Optional) Egress interface from route lookup (default: `any`).
- `service` - (Optional) Service (default: `any`).
- `source_addresses` - (Required) List of source address(es).
- `destination_addresses` - (Required) List of destination address(es).

`translated_packet` supports the following arguments:

- `source` - (Required) The source spec (see below). Leave this empty for a destination NAT of "none".

- `destination` - (Required) The destination spec (see below). Leave this empty for a destination NAT of "none".

`translated_packet.source` supports the following arguments:

- `dynamic_ip_and_port` - (Optional) Dynamic IP and port source translation spec (see below).
- `dynamic_ip` - (Optional) Dynamic IP source translation spec (see below).
- `static_ip` - (Optional) Static IP source translation spec (see below).

`translated_packet.source.dynamic_ip_and_port` supports the following arguments:

- `translated_address` - (Optional) Translated address source translation type spec (see below).
- `interface_address` - (Optional) Interface address source translation type spec (see below).

`translated_packet.source.dynamic_ip_and_port.translated_address` supports the following arguments:

- `translated_addresses` - (Required) List of translated addresses.

`translated_packet.source.dynamic_ip_and_port.interface_address` supports the following arguments:

- `interface` - (Required) The interface.
- `ip_address` - (Optional) The IP address.

`translated_packet.source.dynamic_ip` supports the following arguments:

- `translated_addresses` - (Optional) The list of translated addresses.
- `fallback` - (Optional) The fallback spec (see below). Leaving this empty or omiting it means a fallback of "None".

`translated_packet.source.dynamic_ip.fallback` supports the following arguments:

- `translated_address` - (Optional) The translated address fallback spec (see below).
- `interface_address` - (Optional) The interface address fallback spec (see below).

`translated_packet.source.dynamic_ip.fallback.translated_address` supports the following arguments:

- `translated_addresses` - (Optional) List of source address translation fallback translated addresses.

`translated_packet.source.dynamic_ip.fallback.interface_address` supports the following arguments:

- `interface` - (Required) Source address translation fallback interface.
- `type` - (Optional) Type of interface fallback. Valid values are `ip` (default) or `floating`.
- `ip_address` - (Optional) IP address of the fallback interface.

`translated_packet.source.static_ip` supports the following arguments:

- `translated_address` - (Required) The statically translated source address.
- `bi_directional` - (Optional, bool) Set to `true` to enable bi-directional source address translation.

`translated_packet.destination` supports the following arguments:

- `static_translation` - (Optional) Specifies a static destination NAT (see below).
- `dynamic_translation` - (Optional, PAN-OS 8.1+) Specify a dynamic destination NAT (see below).

- `static` - (**DEPRECATED**, Optional) Specifies a static destination NAT (see below). This was deprecated in provider version 1.6 in favor of `static_translation` instead.
- `dynamic` - (**DEPRECATED**, Optional, PAN-OS 8.1+) Specify a dynamic destination NAT (see below). If you are using Terraform 0.12+, you cannot use this param as it conflicts with the new dynamic (https://www.terraform.io/docs/configuration/expressions.html#dynamic-blocks) block.

`translated_packet.destination.static` and `translated_packet.destination.static_translation` support the following arguments:

- `address` - (Required) Destination address translation address.
- `port` - (Optional, int) Destination address translation port number.

`translated_packet.destination.dynamic` and `translated_packet.destination.dynamic_translation` support the following arguments:

- `address` - (Required) Destination address translation address.
- `port` - (Optional, int) Destination address translation port number.
- `distribution` - (Optional, PAN-OS 8.1+) Distribution algorithm for destination address pool. The PAN-OS 8.1 GUI doesn't seem to set this anywhere, but this is added here for completeness' sake. The GUI sets this to `round-robin` currently.

panos_nat_rule

This resource allows you to add/update/delete NAT rules.

Note: This resource has been deprecated. Please use `panos_nat_rule_group` instead.

Note: `panos_nat_policy` is known as `panos_nat_rule`.

The prefix `sat` stands for "Source Address Translation" while the prefix "dat" stands for "Destination Address Translation". The order of the params in this resource and their naming matches how the params are presented in the GUI. Thus, having a GUI window open while creating your resource definition will simplify the process.

Note that while many of the params for this resource are optional in an absolute sense, depending on what type of NAT you wish to configure, certain params may become necessary to correctly configure the NAT rule.

Example Usage

```
resource "panos_nat_rule" "example" {
    name = "my nat rule"
    source_zones = ["zone1"]
    destination_zone = "zone2"
    to_interface = "ethernet1/3"
    source_addresses = ["any"]
    destination_addresses = ["any"]
    sat_type = "none"
    dat_type = "static"
    dat_address = "my dat address object"
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The NAT rule's name.
- `vsys` - (Optional) The vsys to put the NAT rule into (default: `vsys1`).
- `rulebase` - (Optional, Deprecated) The rulebase. For firewalls, there is only the `rulebase` value (default), but on Panorama, there is also `pre-rulebase` and `post-rulebase`.
- `description` - (Optional) The description.
- `type` - (Optional). NAT type. This can be `ipv4` (default), `nat64`, or `nptv6`.
- `source_zones` - (Required) The list of source zone(s).
- `destination_zone` - (Required) The destination zone.

- `to_interface` - (Optional) Egress interface from route lookup (default: `any`).
- `service` - (Optional) Service (default: `any`).
- `source_addresses` - (Required) List of source address(es).
- `destination_addresses` - (Required) List of destination address(es).
- `sat_type` - (Optional) Type of source address translation. This can be `none` (default), `dynamic-ip-and-port`, `dynamic-ip`, or `static-ip`.
- `sat_address_type` - (Optional) Source address translation address type. This can be `interface-address` or `translated-address`.
- `sat_translated_addresses` - (Optional) Source address translation list of translated addresses.
- `sat_interface` - (Optional) Source address translation interface.
- `sat_ip_address` - (Optional) Source address translation IP address.
- `sat_fallback_type` - (Optional) Source address translation fallback type. This can be `none`, `interface-address`, or `translated-address`.
- `sat_fallback_translated_addresses` - (Optional) Source address translation list of fallback translated addresses.
- `sat_fallback_interface` - (Optional) Source address translation fallback interface.
- `sat_fallback_ip_type` - (Optional) Source address translation fallback IP type. This can be `ip` or `floating`.
- `sat_fallback_ip_address` - (Optional) The source address translation fallback IP address.
- `sat_static_translated_address` - (Optional) The statically translated source address.
- `sat_static_bi_directional` - (Optional) Set to `true` to enable bi-directional source address translation.
- `dat_type` - (Optional) Destination address translation type. This should be either `static` or `dynamic`. The `dynamic` option is only available on PAN-OS 8.1+.
- `dat_address` - (Optional) Destination address translation's address. Requires `dat_type` be set to "static" or "dynamic".
- `dat_port` - (Optional) Destination address translation's port number. Requires `dat_type` be set to "static" or "dynamic".
- `dat_dynamic_distribution` - (Optional, PAN-OS 8.1+) Distribution algorithm for destination address pool. The PAN-OS 8.1 GUI doesn't seem to set this anywhere, but this is added here for completeness' sake. Requires `dat_type` of "dynamic".
- `disabled` - (Optional) Set to `true` to disable this rule.
- `tags` - (Optional) List of administrative tags.

panos_panorama_address_group

This resource allows you to add/update/delete Panorama address groups.

Address groups are either statically defined or dynamically defined, so only `static_addresses` or `dynamic_match` should be defined within a given address group.

Import Name

```
<device_group>:<name>
```

Example Usage

```
# Static group
resource "panos_panorama_address_group" "example1" {
    name = "static ntp grp"
    description = "My NTP servers"
    static_addresses = ["ntp1", "ntp2", "ntp3"]
}

# Dynamic group
resource "panos_panorama_address_group" "example2" {
    name = "dynamic grp"
    description = "My internal NTP servers"
    dynamic_match = "'internal' and 'ntp'"
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The address group's name.
- `device_group` - (Optional) The device group to put the address group into (default: `shared`).
- `static_addresses` - (Optional) The address objects to include in this statically defined address group.
- `dynamic_match` - (Optional) The IP tags to include in this DAG.
- `description` - (Optional) The address group's description.
- `tags` - (Optional) List of administrative tags.

panos_panorama_address_object

This resource allows you to add/update/delete address objects on Panorama.

Import Name

```
<device_group>:<name>
```

Example Usage

```
resource "panos_panorama_address_object" "example" {
    name = "localnet"
    value = "192.168.80.0/24"
    description = "The 192.168.80 network"
    tags = ["internal", "dmz"]
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The address object's name.
- `device_group` - (Optional) The device group to put the address object into (default: `shared`).
- `type` - (Optional) The type of address object. This can be `ip-netmask` (default), `ip-range`, `fqdn`, or `ip-wildcard` (PAN-OS 9.0+).
- `value` - (Required) The address object's value. This can take various forms depending on what type of address object this is, but can be something like `192.168.80.150` or `192.168.80.0/24`.
- `description` - (Optional) The address object's description.
- `tags` - (Optional) List of administrative tags.

panos_panorama_administrative_tag

This resource allows you to add/update/delete Panorama administrative tags.

Import Name

```
<device_group>:<name>
```

Example Usage

```
resource "panos_panorama_administrative_tag" "example" {
    name = "tag1"
    color = "color5"
    comment = "Internal resources"
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The administrative tag's name.
- `device_group` - (Optional) The device group to put the administrative tag into (default: `shared`).
- `color` - (Optional) The tag's color. This should be either an empty string (no color) or a string such as `color1` or `color15`. Note that for maximum portability, you should limit color usage to `color16`, which was available in PAN-OS 6.1. PAN-OS 8.1's colors go up to `color42`. The value `color18` is reserved internally by PAN-OS and thus not available for use.
- `comment` - (Optional) The administrative tag's description.

panos_panorama_aggregate_interface

This resource allows you to add/update/delete Panorama aggregate ethernet interfaces.

Import Name

```
<template>::<vsys>:<name>
```

Example Usage

```
resource "panos_panorama_aggregate_interface" "example" {
    template = "myTemplate"
    vsys = "vsys1"
    name = "ae5"
    mode = "layer3"
    static_ips = ["10.1.1.1/24"]
    comment = "Configured for internal traffic"
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The interface's name.
- `templaet` - (Required) The template where the interface should be.
- `vsys` - (Required) The vsys that will use this interface. This should be something like `vsys1` or `vsys3`.
- `mode` - (Required) The interface mode. Valid values are `layer3` (default), `layer2`, `virtual-wire`, `ha`, or `decrypt-mirror`.
- `netflow_profile` - (Optional) The netflow profile.
- `mtu` - (Optional) The MTU.
- `adjust_tcp_mss` - (Optional) Adjust TCP MSS (default: false).
- `ipv4_mss_adjust` - (Optional) The IPv4 MSS adjust value.
- `ipv6_mss_adjust` - (Optional) The IPv6 MSS adjust value.
- `enable_untagged_subinterface` - (Optional, bool) Set to `true` to enable untagged subinterfaces.
- `static_ips` - (Optional) List of static IPv4 addresses.
- `ipv6_enabled` - (Optional, bool) Set to `true` to enable IPv6.

- `ipv6_interface_id` - (Optional) The IPv6 interface ID.
- `management_profile` - (Optional) The management profile.
- `enable_dhcp` - (Optional, bool) Set to `true` to enable DHCP.
- `create_dhcp_default_route` - (Optional) Set to `true` to create a DHCP default route.
- `dhcp_default_route_metric` - (Optional) The metric for the DHCP default route.
- `comment` - (Optional) The interface comment.
- `decrypt_forward` - (Optional, bool, PAN-OS 8.1+) Set to `true` to enable decrypt forward.
- `dhcp_send_hostname_enable` - (Optional, PAN-OS 9.0+) For DHCP layer3 interfaces: enable sending the firewall or a custom hostname to DHCP server
- `dhcp_send_hostname_value` - (Optional, PAN-OS 9.0+) For DHCP layer3 interfaces: the interface hostname. Leaving this unspecified with `dhcp_send_hostname_enable` set means to send the system hostname.

panos_panorama_application_group

This resource allows you to add/update/delete Panorama application groups.

Import Name

```
<device_group>:<name>
```

Example Usage

```
resource "panos_panorama_application_group" "example" {
    name = "myApp"
    applications = [
        "app1",
        "app2",
    ]
}
```

Argument Reference

The following arguments are supported:

- `device_group` - (Optional) The group's device group (default: `shared`).
- `name` - (Required) The group's name.
- `applications` - (Optional) List of applications in this group.

panos_panorama_application_object

This resource allows you to add/update/delete Panorama application objects.

Import Name

```
<device_group>:<name>
```

Example Usage

```
resource "panos_panorama_application_object" "example" {
    name = "myApp"
    description = "made by terraform"
    category = "media"
    subcategory = "gaming"
    technology = "browser-based"
    defaults {
        port {
            ports = [
                "udp/dynamic",
            ]
        }
    }
    risk = 4
    scanning {
        viruses = true
    }
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The object's name.
- `device_group` - (Optional) The device group (default: `shared`)
- `defaults` - (Optional) The application's defaults spec (defined below). To have a "defaults" of `None` , omit this section.
- `category` - (Required) The category.
- `subcategory` - (Required) The subcategory.
- `technology` - (Required) The technology.
- `description` - (Optional) The object's description.
- `timeout_settings` - (Optional) The timeout spec (defined below).

- `risk` - (Optional, int) The risk (default: 1).
- `parent_app` - (Optional) The parent application.
- `able_to_file_transfer` - (Optional, bool) Able to file transfer.
- `excessive_bandwidth` - (Optional, bool) Excessive bandwidth use.
- `tunnels_other_applications` - (Optional, bool) This application tunnels other apps.
- `has_known_vulnerability` - (Optional, bool) Has known vulnerabilities.
- `used_by_malware` - (Optional, bool) App is used by malware.
- `evasive_behavior` - (Optional, bool) App is evasive.
- `pervasive_use` - (Optional, bool) App is pervasive.
- `prone_to_misuse` - (Optional, bool) Prone to misuse.
- `continue_scanning_for_other_applications` - (Optional, bool) Continue scanning for other applications.
- `scanning` - The scanning spec (defined below).
- `alg_disable_capability` - (Optional) The alg disable capability.
- `no_app_id_caching` - (Optional, bool) No appid caching.

`defaults` supports the following arguments:

- `port` - (Optional) The port spec (defined below)
- `ip_protocol` - (Optional) The ip protocol spec (defined below)
- `icmp` - (Optional) The ICMP spec (defined below)
- `icmp6` - (Optional) The ICMP6 spec (defined below)

`defaults.port` supports the following arguments:

- `ports` - (Required) List of ports.

`defaults.ip_protocol` supports the following arguments:

- `value` - (Required, int) The IP protocol value.

`defaults.icmp` supports the following arguments:

- `type` - (Required, int) The type.
- `code` - (Optional, int) The code.

`defaults.icmp6` supports the following arguments:

- `type` - (Required, int) The type.
- `code` - (Optional, int) The code.

`timeout_settings` supports the following arguments:

- `timeout` - (Optional, int) The timeout.

- `tcp_timeout` - (Optional, int) TCP timeout.
- `udp_timeout` - (Optional, int) UDP timeout.
- `tcp_half_closed` - (Optional, int) TCP half closed timeout.
- `tcp_time_wait` - (Optional, int) TCP time wait timeout.

`scanning` supports the following arguments:

- `file_types` - (Optional, bool) File type scanning.
- `viruses` - (Optional, bool) Virus scanning.
- `data_patterns` - (Optional, bool) Data pattern scanning.

panos_panorama_application_signature

This resource allows you to add/update/delete Panorama application signatures.

Import Name

```
<device_group>:<application_object>:<name>
```

Example Usage

```
resource "panos_panorama_application_signature" "example" {
    application_object = panos_panorama_application_object.myapp.name
    comment = "made by terraform"
    ordered_match = true
    and_condition {
        or_condition {
            pattern_match {
                context = "http-req-headers"
                pattern = "somepattern"
                qualifiers = {
                    "http-method": "COPY",
                    "req-hdr-type": "HOST",
                }
            }
        }
        or_condition {
            greater_than {
                // X.400-message size
                context = "cotp-req-x420-message-size"
                value = "123456"
            }
        }
        or_condition {
            less_than {
                // X.400-message size
                context = "cotp-req-x420-message-size"
                value = "42"
            }
        }
    }
    and_condition {
        or_condition {
            equal_to {
                context = "unknown-req-tcp"
                position = "first-4bytes"
                mask = "0Xff112345"
                value = "0X11bb33dd"
            }
        }
    }
}
```

```
resource "panos_panorama_application_object" "myapp" {
    name = "myApp"
    description = "made by terraform"
    category = "media"
    subcategory = "gaming"
    technology = "browser-based"
    defaults {
        port {
            ports = [
                "udp/dynamic",
            ]
        }
    }
    risk = 4
    scanning {
        viruses = true
    }
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The signature's name.
- `device_group` - (Optional) The device group (default: `shared`)
- `application_object` - (Required) The applciation object for this signature.
- `comment` - (Optional) The description.
- `scope` - (Optional) The signature's scope. Valid values are `transaction` (default) or `session`.
- `ordered_match` - (Optional, bool) Set to `false` to disable ordered matching (default: `true`).
- `and_condition` - (Optional) The and condition spec (defined below).

`and_condition` supports the following arguments:

- `name` - (Computed) And condition name, this is computed and cannot be configured.
- `or_condition` - (Required) The or condition spec (defined below).

`and_condition.or_condition` supports the following arguments:

- `name` - (Computed) Or condition name, this is computed and cannot be configured.
- `pattern_match` - (Optional) The pattern match spec (defined below).
- `greater_than` - (Optional) The greater than spec (defined below).
- `less_than` - (Optional) the less than spec (defined below).
- `equal_to` - (Optional) The equal to spec (defined below).

`and_condition.or_condition.pattern_match` supports the following arguments:

- `context` - (Required) The context.
- `pattern` - (Required) The pattern.
- `qualifiers` - (Optional, map) The qualifiers.

`and_condition.or_condition.greater_than` supports the following arguments:

- `context` - (Required) The context.
- `value` - (Required) The value.
- `qualifiers` - (Optional, map) The qualifiers.

`and_condition.or_condition.less_than` supports the following arguments:

- `context` - (Required) The context.
- `value` - (Required) The value.
- `qualifiers` - (Optional, map) The qualifiers.

`and_condition.or_condition.equal_to` supports the following arguments:

- `context` - (Required) The context.
- `value` - (Required) The value.
- `position` - (Optional) The position.
- `mask` - (Optional) The mask.

panos_panorama_bfd_profile.

This resource allows you to add/update/delete BFD profiles on Panorama.

Note: This resource is only applicable for PAN-OS 7.1+.

Import Name

```
<template>:<template_stack>:<name>
```

Example Usage

```
resource "panos_panorama_bfd_profile" "example" {
    template = "${panos_panorama_template.t.name}"
    name = "myBfdProfile"
}

resource "panos_panorama_template" "t" {
    name = "myTemplate"
}
```

Argument Reference

One and only one of the following must be specified:

- `template` - The template name.
- `template_stack` - The template stack name.

The following arguments are supported:

- `name` - (Required) The BBFD profile's name.
- `mode` - (Optional) BFD operation mode. Valid values are `active` (default) or `passive`.
- `minimum_tx_interval` - (Optional, int) Desired minimum TX interval in ms. Default is `1000`.
- `minimum_rx_interval` - (Optional, int) Required minimum RX interval in ms. Default is `1000`.
- `detection_multiplier` - (Optional, int) Multiplier sent to remote system. Default is `3`.
- `hold_time` - (Optional, int) Delay transmission and reception of control packets in ms.
- `minimum_rx_ttl` - (Optional, int) Minimum accepted ttl on received BFD packet.

panos_panorama_bgp_aggregate_advertise_filter

This resource allows you to add/update/delete a Panorama route advertise filter for a BGP address aggregation rule.

Import Name

```
<template>:<template_stack>:<virtual_router>:<bgp_aggregate>:<name>
```

Example Usage

```
resource "panos_panorama_bgp_aggregate_advertise_filter" "example" {
    template = "${panos_panorama_template.t.name}"
    virtual_router = "${panos_panorama_bgp_aggregate.ag.virtual_router}"
    bgp_aggregate = "${panos_panorama_bgp_aggregate.ag.name}"
    name = "my advertise filter"
    as_path_regex = "*42*"
    med = "443"
    address_prefix {
        prefix = "10.1.1.0/24"
        exact = true
    }
    address_prefix {
        prefix = "10.1.2.0/24"
    }
}

resource "panos_panorama_template" "t" {
    name = "my template"
}

resource "panos_panorama_bgp_aggregate" "ag" {
    template = "${panos_panorama_template.t.name}"
    virtual_router = "${panos_panorama_bgp.conf.virtual_router}"
    name = "addyAgg1"
    prefix = "192.168.1.0/24"
}

resource "panos_panorama_bgp" "conf" {
    template = "${panos_panorama_template.t.name}"
    virtual_router = "${panos_panorama_virtual_router.rtr.name}"
    router_id = "5.5.5.5"
    as_number = "42"
}

resource "panos_panorama_virtual_router" "rtr" {
    template = "${panos_panorama_template.t.name}"
    name = "my virtual router"
}
```

Argument Reference

One and only one of the following must be specified:

- `template` - The template name.
- `template_stack` - The template stack name.

The following arguments are supported:

- `virtual_router` - (Required) The virtual router to add this filter to.
- `bgp_aggregate` - (Required) The BGP address aggregation rule.
- `name` - (Required) The name.
- `enable` - (Optional, bool) Enable or not (default: `true`).
- `as_path_regex` - (Optional) AS path to match.
- `community_regex` - (Optional) Community to match.
- `extended_community_regex` - (Optional) Extended community to match.
- `med` - (Optional) Match MED.
- `route_table` - (Optional, PAN-OS 8.0+) Route table to match rule. Valid values are `unicast`, `multicast`, or `both`.
- `address_prefix` - (Optional, repeatable) Matching address prefix definition (see below).
- `next_hops` - (Optional) List of next hop attributes.
- `from_peers` - (Optional) List of peers that advertised the route entry.

Each `address_prefix` section offers the following params:

- `prefix` - (Required) Address prefix.
- `exact` - (Optional, bool) Match exact prefix length.

panos_panorama_bgp_aggregate

This resource allows you to add/update/delete Panorama BGP address aggregation rules.

Import Name

```
<template>:<template_stack>:<virtual_router>:<name>
```

Example Usage

```
resource "panos_panorama_bgp_aggregate" "example" {
    template = "${panos_panorama_template.t.name}"
    virtual_router = "${panos_panorama_bgp.conf.virtual_router}"
    name = "myAggRule"
    prefix = "192.168.1.0/24"
    weight = 17
}

resource "panos_panorama_templaet" "t" {
    name = "myTemplate"
}

resource "panos_panorama_bgp" "conf" {
    template = "${panos_panorama_template.t.name}"
    virtual_router = "${panos_panorama_virtual_router.vr.name}"
    router_id = "1.2.3.4"
    as_number = 443
}

resource "panos_panorama_virtual_router" "vr" {
    template = "${panos_panorama_template.t.name}"
    name = "my vr"
}
```

Argument Reference

One and only one of the following must be specified:

- `template` - The template name.
- `template_stack` - The template stack name.

The following arguments are supported:

- `virtual_router` - (Required) The virtual router to put the rule into.
- `name` - (Required) The rule name.

- `prefix` - (Required) Aggregating address prefix.
- `enable` - (Optional, bool) Enable this rule (default: `true`)
- `as_set` - (Optional, bool) Generate AS-set attribute.
- `summary` - (Optional, bool) Summarize route.
- `local_preference` - (Optional) New local preference value.
- `med` - (Optional) New MED value.
- `weight` - (Optional, int) New weight value.
- `next_hop` - (Optional) Next hop address.
- `origin` - (Optional) New route origin. Valid values are `incomplete` (default), `igp`, or `egp`.
- `as_path_limit` - (Optional, int) Add AS path limit attribute if it does not exist.
- `as_path_type` - (Optional) AS path update options. Valid values are `none` (default) or `prepend`.
- `as_path_value` - (Optional) For `as_path_type` of `prepend`, the value to prepend.
- `community_type` - (Optional) Community update options. Valid values are `none` (default), `remove-all`, `remove-regex`, `append`, or `overwrite`.
- `community_value` - (Optional) If `community_type` is `remove-regex`, `append`, or `overwrite`, the value associated with that setting. For the `append` and `overwrite` types specifically, valid values are `no-export`, `no-advertise`, `local-as`, or `nopeer`.
- `extended_community_type` - (Optional) Extended community update options. Valid values are `none` (default), `remove-all`, `remove-regex`, `append`, or `overwrite`.
- `extended_community_vaule` - (Optional) If `extended_community_type` is `remove-regex`, `append`, or `overwrite`, the value associated with that setting.

panos_panorama_bgp_aggregate_suppress_filter

This resource allows you to add/update/delete a Panorama route suppression filter for a BGP address aggregation rule.

Import Name

```
<template>:<template_stack>:<virtual_router>:<bgp_aggregate>:<name>
```

Example Usage

```
resource "panos_panorama_bgp_aggregate_suppress_filter" "example" {
    template = "${panos_panorama_template.t.name}"
    virtual_router = "${panos_panorama_bgp_aggregate.ag.virtual_router}"
    bgp_aggregate = "${panos_panorama_bgp_aggregate.ag.name}"
    name = "my suppression filter"
    as_path_regex = "*42*"
    med = "443"
    address_prefix {
        prefix = "10.1.1.0/24"
        exact = true
    }
    address_prefix {
        prefix = "10.1.2.0/24"
    }
}

resource "panos_panorama_template" "t" {
    name = "my template"
}

resource "panos_panorama_bgp_aggregate" "ag" {
    template = "${panos_panorama_template.t.name}"
    virtual_router = "${panos_panorama_bgp.conf.virtual_router}"
    name = "addyAgg1"
    prefix = "192.168.1.0/24"
}

resource "panos_panorama_bgp" "conf" {
    template = "${panos_panorama_template.t.name}"
    virtual_router = "${panos_panorama_virtual_router.rtr.name}"
    router_id = "5.5.5.5"
    as_number = "42"
}

resource "panos_panorama_virtual_router" "rtr" {
    template = "${panos_panorama_template.t.name}"
    name = "my virtual router"
}
```

Argument Reference

One and only one of the following must be specified:

- `template` - The template name.
- `template_stack` - The template stack name.

The following arguments are supported:

- `virtual_router` - (Required) The virtual router to add this filter to.
- `bgp_aggregate` - (Required) The BGP address aggregation rule.
- `name` - (Required) The name.
- `enable` - (Optional, bool) Enable or not (default: `true`).
- `as_path_regex` - (Optional) AS path to match.
- `community_regex` - (Optional) Community to match.
- `extended_community_regex` - (Optional) Extended community to match.
- `med` - (Optional) Match MED.
- `route_table` - (Optional, PAN-OS 8.0+) Route table to match rule. Valid values are `unicast`, `multicast`, or `both`.
- `address_prefix` - (Optional, repeatable) Matching address prefix definition (see below).
- `next_hops` - (Optional) List of next hop attributes.
- `from_peers` - (Optional) List of peers that advertised the route entry.

Each `address_prefix` section offers the following params:

- `prefix` - (Required) Address prefix.
- `exact` - (Optional, bool) Match exact prefix length.

panos_panorama_bgp_auth_profile

This resource allows you to add/update/delete a Panorama BGP auth profile.

Example Usage

```
resource "panos_panorama_bgp_auth_profile" "example" {
    template = "${panos_panorama_template.t.name}"
    virtual_router = "${panos_panorama_bgp.conf.virtual_router}"
    name = "prof1"
    secret = "secret"
}

resource "panos_panorama_bgp" "conf" {
    template = "${panos_panorama_template.t.name}"
    virtual_router = "${panos_panorama_virtual_router.rtr.name}"
    router_id = "5.5.5.5"
    as_number = "42"
}

resource "panos_panorama_virtual_router" "rtr" {
    template = "${panos_panorama_template.t.name}"
    name = "my virtual router"
}

resource "panos_panorama_template" "t" {
    name = "my template"
}
```

Argument Reference

The following arguments are supported:

- `virtual_router` - (Required) The virtual router to add this BGP auth profile to.
- `name` - (Required) The name.
- `secret` - (Optional) Shared secret for the TCP MD5 authentication.

panos_panorama_bgp_conditional_adv_advertise_filter

This resource allows you to add/update/delete a Panorama advertise filter for a BGP conditional advertisement.

Note: A BGP conditional advertisement is valid only if there is at least one non-exist filter and one advertise filter attached. This filter must be paired with the other in order for the configuration to be valid.

Import Name

```
<template>:<template_stack>:<virtual_router>:<bgp_conditional_adv>:<name>
```

Example Usage

```
data "panos_system_info" "x" {}

resource "panos_panorama_bgp_conditional_adv_advertise_filter" "example" {
    template = "${panos_panorama_template.t.name}"
    virtual_router = "${panos_panorama_bgp.conf.virtual_router}"
    bgp_conditional_adv = "${panos_panorama_bgp_conditional_adv.ca.name}"
    name = "af"
    route_table = "${data.panos_system_info.x.version_major >= 8 ? "unicast" : ""}"
    address_prefixes = ["192.168.1.0/24"]
}

resource "panos_panorama_template" "t" {
    name = "myTemplate"
}

resource "panos_panorama_bgp_conditional_adv" "ca" {
    template = "${panos_panorama_template.t.name}"
    virtual_router = "${panos_panorama_bgp.conf.virtual_router}"
    name = "example"
}

resource "panos_panorama_bgp_conditional_adv_non_exist_filter" "af" {
    template = "${panos_panorama_template.t.name}"
    virtual_router = "${panos_panorama_bgp.conf.virtual_router}"
    bgp_conditional_adv = "${panos_panorama_bgp_conditional_adv.ca.name}"
    name = "nef"
    route_table = "${data.panos_system_info.x.version_major >= 8 ? "unicast" : ""}"
    address_prefixes = ["192.168.2.0/24"]
}

resource "panos_panorama_bgp" "conf" {
    template = "${panos_panorama_template.t.name}"
    virtual_router = "${panos_panorama_virtual_router.rtr.name}"
    router_id = "5.5.5.5"
    as_number = "42"
}

resource "panos_panorama_virtual_router" "rtr" {
    template = "${panos_panorama_template.t.name}"
    name = "my virtual router"
}
```

Argument Reference

One and only one of the following must be specified:

- `template` - The template name.
- `template_stack` - The template stack name.

The following arguments are supported:

- `virtual_router` - (Required) The virtual router to add this filter to.
- `bgp_conditional_adv` - (Required) The BGP conditional advertisement to add this filter to.

- `name` - (Required) The name.
- `enable` - (Optional, bool) Enable or not (default: `true`).
- `as_path_regex` - (Optional) AS path to match.
- `community_regex` - (Optional) Community to match.
- `extended_community_regex` - (Optional) Extended community to match.
- `med` - (Optional) Match MED.
- `route_table` - (Optional, PAN-OS 8.0+) Route table to match rule. Valid values are `unicast`, `multicast`, or `both`. As of PAN-OS 8.1, there doesn't seem to be a way to configure this in the GUI, it is always set to `unicast`. Thus, if you're running this resource against PAN-OS 8.0+, the appropriate thing to do is set this value to `unicast` as well to match the GUI functionality.
- `address_prefixes` - (Optional) List of matching address prefixes.
- `next_hops` - (Optional) List of next hop attributes.
- `from_peers` - (Optional) List of peers that advertised the route entry.

panos_bgp_conditional_adv

This resource allows you to add/update/delete a Panorama BGP conditional advertisement.

Note: In the PAN-OS GUI, this resource cannot be created without also creating at least one non-exist filter and one advertise filter. The API behaves a little differently: you can create the conditional advertisement itself, but the API will start throwing errors if you try to update it and there is not at least one non-exist filter and one advertise filter. In order for a conditional advertisement to be valid, you must specify at least one non-exist and one advertise filter.

Import Name

```
<template>:<template_stack>:<virtual_router>:<name>
```

Example Usage

```
data "panos_system_info" "x" {}

resource "panos_panorama_bgp_conditional_adv" "example" {
    template = "${panos_panorama_template.t.name}"
    virtual_router = "${panos_panorama_bgp.conf.virtual_router}"
    name = "example"
    enable = false
}

resource "panos_panorama_template" "t" {
    name = "myTemplate"
}

resource "panos_panorama_bgp_conditional_adv_non_exist_filter" "nef" {
    template = "${panos_panorama_template.t.name}"
    virtual_router = "${panos_panorama_bgp.conf.virtual_router}"
    bgp_conditional_adv = "${panos_panorama_bgp_conditional_adv.example.name}"
    route_table = "${data.panos_system_info.x.version_major >= 8 ? "unicast" : ""}"
    name = "nef"
    address_prefixes = ["192.168.1.0/24"]
}

resource "panos_panorama_bgp_conditional_adv_advertise_filter" "af" {
    template = "${panos_panorama_template.t.name}"
    virtual_router = "${panos_panorama_bgp.conf.virtual_router}"
    bgp_conditional_adv = "${panos_panorama_bgp_conditional_adv.example.name}"
    route_table = "${data.panos_system_info.x.version_major >= 8 ? "unicast" : ""}"
    name = "af"
    address_prefixes = ["192.168.2.0/24"]
}

resource "panos_panorama_bgp" "conf" {
    template = "${panos_panorama_template.t.name}"
    virtual_router = "${panos_panorama_virtual_router.rtr.name}"
    router_id = "5.5.5.5"
    as_number = "42"
}

resource "panos_panorama_virtual_router" "rtr" {
    template = "${panos_panorama_template.t.name}"
    name = "my virtual router"
}
```

Argument Reference

One and only one of the following must be specified:

- `template` - The template name.
- `template_stack` - The template stack name.

The following arguments are supported:

- `virtual_router` - (Required) The virtual router to add this BGP conditional advertisement to.

- `name` - (Required) The name.
- `enable` - (Optional, bool) Enable or not (default: `true`).
- `used_by` - (Optional) List of BGP peer groups that use this rule.

panos_panorama_bgp_conditional_adv_non_exist_filter

This resource allows you to add/update/delete a Panorama non-exist filter for a BGP conditional advertisement.

Note: A BGP conditional advertisement is valid only if there is at least one non-exist filter and one advertise filter attached. This filter must be paired with the other in order for the configuration to be valid.

Import Name

```
<template>:<template_stack>:<virtual_router>:<bgp_conditional_adv>:<name>
```

Example Usage

```
data "panos_system_info" "x" {}

resource "panos_panorama_bgp_conditional_adv_non_exist_filter" "example" {
    template = "${panos_panorama_template.t.name}"
    virtual_router = "${panos_panorama_bgp.conf.virtual_router}"
    bgp_conditional_adv = "${panos_panorama_bgp_conditional_adv.ca.name}"
    name = "nef"
    route_table = "${data.panos_system_info.x.version_major >= 8 ? "unicast" : ""}"
    address_prefixes = ["192.168.1.0/24"]
}

resource "panos_panorama_template" "t" {
    name = "myTemplate"
}

resource "panos_panorama_bgp_conditional_adv" "ca" {
    template = "${panos_panorama_template.t.name}"
    virtual_router = "${panos_panorama_bgp.conf.virtual_router}"
    name = "example"
}

resource "panos_panorama_bgp_conditional_adv_advertise_filter" "af" {
    template = "${panos_panorama_template.t.name}"
    virtual_router = "${panos_panorama_bgp.conf.virtual_router}"
    bgp_conditional_adv = "${panos_panorama_bgp_conditional_adv.ca.name}"
    name = "af"
    route_table = "${data.panos_system_info.x.version_major >= 8 ? "unicast" : ""}"
    address_prefixes = ["192.168.2.0/24"]
}

resource "panos_panorama_bgp" "conf" {
    template = "${panos_panorama_template.t.name}"
    virtual_router = "${panos_panorama_virtual_router.rtr.name}"
    router_id = "5.5.5.5"
    as_number = "42"
}

resource "panos_panorama_virtual_router" "rtr" {
    template = "${panos_panorama_template.t.name}"
    name = "my virtual router"
}
```

Argument Reference

One and only one of the following must be specified:

- `template` - The template name.
- `template_stack` - The template stack name.

The following arguments are supported:

- `virtual_router` - (Required) The virtual router to add this filter to.
- `bgp_conditional_adv` - (Required) The BGP conditional advertisement to add this filter to.

- `name` - (Required) The name.
- `enable` - (Optional, bool) Enable or not (default: `true`).
- `as_path_regex` - (Optional) AS path to match.
- `community_regex` - (Optional) Community to match.
- `extended_community_regex` - (Optional) Extended community to match.
- `med` - (Optional) Match MED.
- `route_table` - (Optional, PAN-OS 8.0+) Route table to match rule. Valid values are `unicast`, `multicast`, or `both`. As of PAN-OS 8.1, there doesn't seem to be a way to configure this in the GUI, it is always set to `unicast`. Thus, if you're running this resource against PAN-OS 8.0+, the appropriate thing to do is set this value to `unicast` as well to match the GUI functionality.
- `address_prefixes` - (Optional) List of matching address prefixes.
- `next_hops` - (Optional) List of next hop attributes.
- `from_peers` - (Optional) List of peers that advertised the route entry.

panos_panorama_bgp_dampening_profile

This resource allows you to add/update/delete a Panorama BGP dampening profile.

Import Name

```
<template>:<template_stack>:<virtual_router>:<name>
```

Example Usage

```
resource "panos_panorama_bgp_dampening_profile" "example" {
    template = "${panos_panorama_template.t.name}"
    virtual_router = "${panos_panorama_bgp.conf.virtual_router}"
    name = "myDampeningProfile"
}

resource "panos_panorama_bgp" "conf" {
    template = "${panos_panorama_template.t.name}"
    virtual_router = "${panos_panorama_virtual_router.rtr.name}"
    router_id = "5.5.5.5"
    as_number = "42"
}

resource "panos_panorama_virtual_router" "rtr" {
    template = "${panos_panorama_template.t.name}"
    name = "my virtual router"
}

resource "panos_panorama_template" "t" {
    name = "myTemplate"
}
```

Argument Reference

One and only one of the following must be specified:

- `template` - The template name.
- `template_stack` - The template stack name.

The following arguments are supported:

- `virtual_router` - (Required) The virtual router to add this BGP dampening profile to.
- `name` - (Required) The name.
- `enable` - (Optional, bool) Enable or not (default: `true`).

- `cutoff` - (Optional, float) Cutoff threshold value (default: `1.25`).
- `reuse` - (Optional, float) Reuse threshold value (default: `0.5`).
- `max_hold_time` - (Optional, int) Maximum hold-down time, in seconds (default: `900`).
- `decay_half_life_reachable` - (Optional, int) Decay half-life while reachable, in seconds (default: `300`).
- `decay_half_life_unreachable` - (Optional, int) Decay half-life while unreachable, in seconds (default: `900`).

panos_panorama_bgp_export_rule_group

This resource allows you to add/update/delete Panorama BGP export rule groups.

This resource manages clusters of export rules in a virtual router, enforcing both the contents of individual rules as well as their ordering. Rules are defined in a `rule` config block.

Although you cannot modify non-group export rules with this resource, the `position_keyword` and `position_reference` parameters allow you to reference some other export rule that already exists, using it as a means to ensure some rough placement within the ruleset as a whole.

Best Practices

As is to be expected, if you are separating your deployment across multiple plan files, make sure that at most only one plan specifies any given absolute positioning keyword such as "top" or "directly below", otherwise they'll keep shoving each other out of the way indefinitely.

Best practices are to specify one group as `top` (if you need it), one group as `bottom`, then all other groups should be `above` the first rule of the bottom group. You do it this way because rules will natually be added at the tail end of the ruleset, so they will always be `after` the first group, but what you want is for them to be `before` the last group's rules.

Example Usage

```
resource "panos_panorama_bgp_export_rule_group" "example" {
    template = "${panos_panorama_template.t.name}"
    virtual_router = "${panos_panorama_bgp.conf.virtual_router}"
    rule {
        name = "first"
        match_as_path_regex = "*foo*"
        match_address_prefix {
            prefix = "192.168.1.0/24"
        }
        match_address_prefix {
            prefix = "192.168.2.0/24"
            exact = true
        }
        match_route_table = "${data.panos_system_info.x.version_major >= 8 ? "unicast" : ""}"
        local_preference = "42"
        med = "43"
        weight = 44
        origin = "incomplete"
    }
    rule {
        name = "second"
        match_as_path_regex = "*bar*"
        action = "deny"
        match_route_table = "${data.panos_system_info.x.version_major >= 8 ? "unicast" : ""}"
    }
}

data "panos_system_info" "x" {}

resource "panos_panorama_bgp" "conf" {
    template = "${panos_panorama_template.t.name}"
    virtual_router = "${panos_panorama_virtual_router.vr.name}"
    router_id = "1.2.3.4"
    as_number = 443
}

resource "panos_panorama_virtual_router" "vr" {
    template = "${panos_panorama_template.t.name}"
    name = "my vr"
}

resource "panos_panorama_template" t" {
    name = "myTemplate"
}
```

Argument Reference

One and only one of the following must be specified:

- `template` - The template name.
- `template_stack` - The template stack name.

The following arguments are supported:

- `virtual_router` - (Required) The virtual router to put the rule into.
- `position_keyword` - (Optional) A positioning keyword for this group. This can be `before`, `directly before`, `after`, `directly after`, `top`, `bottom`, or left empty (the default) to have no particular placement. This param works in combination with the `position_reference` param.
- `position_reference` - (Optional) Required if `position_keyword` is one of the "above" or "below" variants, this is the name of a non-group rule to use as a reference to place this group.
- `rule` - The export rule definition (see below). The export rule ordering will match how they appear in the terraform plan file.

The following arguments are valid for each `rule` section:

- `name` - (Required) The security rule name.
- `enable` - (Optional, bool) Enable this export rule (default: `true`)
- `used_by` - (Optional) List of auth profiles.
- `match_as_path_regex` - (Optional) AS path to match.
- `match_community_regex` - (Optional) Community to match.
- `match_extended_community_regex` - (Optional) Extended community to match.
- `match_med` - (Optional) Match MED.
- `match_route_table` - (Optional, PAN-OS 8.0+) Route table to match rule. Valid values are `unicast`, `multicast`, or `both`. As of PAN-OS 8.1, there doesn't seem to be a way to configure this in the GUI, it is always set to `unicast`. Thus, if you're running this resource against PAN-OS 8.0+, the appropriate thing to do is set this value to `unicast` as well to match the GUI functionality.
- `match_address_prefix` - (Optional, repeatable) Matching address prefix definition (see below). below for the params for this section.
- `match_next_hops` - (Optional) List of next hop attributes.
- `match_from_peers` - (Optional) List of peers that advertised the route entry.
- `action` - (Optional) Rule action. Valid values are `allow` (default) or `deny`.
- `dampening` - (Optional) Route flap dampening profile.
- `local_preference` - (Optional) New local preference value.
- `med` - (Optional) New MED value.
- `weight` - (Optional, int) New weight value.
- `next_hop` - (Optional) Next hop address.
- `origin` - (Optional) New route origin. Valid values are `igp`, `egp`, or `incomplete`.
- `as_path_limit` - (Optional, int) Add AS path limit attribute if it does not exist.
- `as_path_type` - (Optional) AS path update options. Valid values are `none`, `remove`, `prepend` or `remove-and-prepend`.

- `as_path_value` - (Optional) If `as_path_type` is `prepend` or `remove-and-prepend`, the value to prepend.
- `community_type` - (Optional) Community update options. Valid values are `none`, `remove-all`, `remove-regex`, `append`, or `overwrite`.
- `community_value` - (Optional) If `community_type` is `remove-regex`, `append`, or `overwrite`, the value associated with that setting. For the `append` and `overwrite` types specifically, valid values for `community_value` are `no-export`, `no-advertise`, `local-as`, or `nopeer`.
- `extended_community_type` - (Optional) Extended community update options. Valid values are `none`, `remove-all`, `remove-regex`, `append`, or `overwrite`.
- `extended_community_vaule` - (Optional) If `extended_community_type` is `remove-regex`, `append`, or `overwrite`, the value associated with that setting.

Each `match_address_prefix` section offers the following params:

- `prefix` - (Required) Address prefix.
- `exact` - (Optional, bool) Match exact prefix length.

panos_panorama_bgp

This resource allows you to add/update/delete a Panorama virtual router BGP configuration.

Important Note: When it comes to BGP configuration, PAN-OS requires that BGP itself first be configured before you can add other BGP sub-config, such as dampening profiles or peer groups. Since every BGP resource must reference a virtual router, the key to accomplishing this is by pointing the `virtual_router` param for each BGP sub-config to `panos_panorama_bgp.foo.virtual_router` instead of `panos_panorama_virtual_router.bar.name`.

Import Name

```
<template>:<template_stack>:<virtual_router>
```

Example Usage

```
resource "panos_panorama_bgp" "example" {
    template = "${panos_panorama_template.t.name}"
    virtual_router = "${panos_panorama_virtual_router.rtr.name}"
    router_id = "5.5.5.5"
    as_number = "42"
}

resource "panos_panorama_template" "t" {
    name = "myTemplate"
}

resource "panos_panorama_virtual_router" "rtr" {
    template = "${panos_panorama_template.t.name}"
    name = "my virtual router"
}
```

Argument Reference

One and only one of the following must be specified:

- `template` - The template name.
- `template_stack` - The template stack name.

The following arguments are supported:

- `virtual_router` - (Required) The virtual router to add this BGP configuration to.
- `enable` - (Optional, bool) Enable BGP or not (default: `true`).
- `router_id` - (Optional) Router ID of this BGP instance.

- `as_number` - (Optional) Local AS number.
- `bfd_profile` - (Optional, PAN-OS 7.1+) BFD configuration.
- `reject_default_route` - (Optional, bool) Do not learn default route from BGP (default: `true`).
- `install_route` - (Optional, bool) Populate BGP learned route to global route table.
- `aggregate_med` - (Optional, bool) Aggregate route only if they have same MED attributes (default: `true`).
- `default_local_preference` - (Optional) Default local preference (default: `"100"`).
- `as_format` - (Optional) AS format. Valid values are `"2-byte"` (default) or `"4-byte"`.
- `always_compare_med` - (Optional, bool) Always compare MEDs.
- `deterministic_med_comparison` - (Optional, bool) Deterministic MED comparison (default: `true`).
- `ecmp_multi_as` - (Optional, bool) Support multiple AS in ECMP.
- `enforce_first_as` - (Optional, bool) Enforce First AS for EBGP (default: `true`).
- `enable_graceful_restart` - (Optional, bool) Enable graceful restart (default: `true`).
- `stale_route_time` - (Optional, int) Time to remove stale routes after peer restart, in seconds (default: `120`).
- `local_restart_time` - (Optional, int) Local restart time to advertise to peer, in seconds (default: `120`).
- `max_peer_restart_time` - (Optional, int) Maximum of peer restart time accepted, in seconds (default: `120`).
- `reflector_cluster_id` - (Optional) Route reflector cluster ID.
- `confederation_member_as` - (Optional) Confederation requires member-AS number.
- `allow_redistribute_default_route` - (Optional, bool) Allow redistribute default route to BGP.

panos_panorama_bgp_import_rule_group

This resource allows you to add/update/delete Panorama BGP import rule groups.

This resource manages clusters of import rules in a virtual router, enforcing both the contents of individual rules as well as their ordering. Rules are defined in a `rule` config block.

Although you cannot modify non-group import rules with this resource, the `position_keyword` and `position_reference` parameters allow you to reference some other import rule that already exists, using it as a means to ensure some rough placement within the ruleset as a whole.

Best Practices

As is to be expected, if you are separating your deployment across multiple plan files, make sure that at most only one plan specifies any given absolute positioning keyword such as "top" or "directly below", otherwise they'll keep shoving each other out of the way indefinitely.

Best practices are to specify one group as `top` (if you need it), one group as `bottom`, then all other groups should be `above` the first rule of the bottom group. You do it this way because rules will natually be added at the tail end of the ruleset, so they will always be `after` the first group, but what you want is for them to be `before` the last group's rules.

Example Usage

```
resource "panos_panorama_bgp_import_rule_group" "example" {
    template = "${panos_panorama_template.t.name}"
    virtual_router = "${panos_panorama_bgp.conf.virtual_router}"
    rule {
        name = "first"
        match_as_path_regex = "*foo*"
        match_address_prefix {
            prefix = "192.168.1.0/24"
        }
        match_address_prefix {
            prefix = "192.168.2.0/24"
            exact = true
        }
        match_route_table = "${data.panos_system_info.x.version_major >= 8 ? "unicast" : ""}"
        local_preference = "42"
        med = "43"
        weight = 44
        origin = "incomplete"
    }
    rule {
        name = "second"
        match_as_path_regex = "*bar*"
        action = "deny"
        match_route_table = "${data.panos_system_info.x.version_major >= 8 ? "unicast" : ""}"
    }
}

data "panos_system_info" "x" {}

resource "panos_panorama_bgp" "conf" {
    template = "${panos_panorama_template.t.name}"
    virtual_router = "${panos_panorama_virtual_router.vr.name}"
    router_id = "1.2.3.4"
    as_number = 443
}

resource "panos_panorama_virtual_router" "vr" {
    template = "${panos_panorama_template.t.name}"
    name = "my vr"
}

resource "panos_panorama_template" t" {
    name = "myTemplate"
}
```

Argument Reference

One and only one of the following must be specified:

- `template` - The template name.
- `template_stack` - The template stack name.

The following arguments are supported:

- `virtual_router` - (Required) The virtual router to put the rule into.
- `position_keyword` - (Optional) A positioning keyword for this group. This can be `before`, `directly before`, `after`, `directly after`, `top`, `bottom`, or left empty (the default) to have no particular placement. This param works in combination with the `position_reference` param.
- `position_reference` - (Optional) Required if `position_keyword` is one of the "above" or "below" variants, this is the name of a non-group rule to use as a reference to place this group.
- `rule` - The import rule definition (see below). The import rule ordering will match how they appear in the terraform plan file.

The following arguments are valid for each `rule` section:

- `name` - (Required) The security rule name.
- `enable` - (Optional, bool) Enable this import rule (default: `true`)
- `used_by` - (Optional) List of auth profiles.
- `match_as_path_regex` - (Optional) AS path to match.
- `match_community_regex` - (Optional) Community to match.
- `match_extended_community_regex` - (Optional) Extended community to match.
- `match_med` - (Optional) Match MED.
- `match_route_table` - (Optional, PAN-OS 8.0+) Route table to match rule. Valid values are `unicast`, `multicast`, or `both`. As of PAN-OS 8.1, there doesn't seem to be a way to configure this in the GUI, it is always set to `unicast`. Thus, if you're running this resource against PAN-OS 8.0+, the appropriate thing to do is set this value to `unicast` as well to match the GUI functionality.
- `match_address_prefix` - (Optional, repeatable) Matching address prefix definition (see below). below for the params for this section.
- `match_next_hops` - (Optional) List of next hop attributes.
- `match_from_peers` - (Optional) List of peers that advertised the route entry.
- `action` - (Optional) Rule action. Valid values are `allow` (default) or `deny`.
- `dampening` - (Optional) Route flap dampening profile.
- `local_preference` - (Optional) New local preference value.
- `med` - (Optional) New MED value.
- `weight` - (Optional, int) New weight value.
- `next_hop` - (Optional) Next hop address.
- `origin` - (Optional) New route origin. Valid values are `igp`, `egp`, or `incomplete`.
- `as_path_limit` - (Optional, int) Add AS path limit attribute if it does not exist.
- `as_path_type` - (Optional) AS path update options. Valid values are `none` or `remove`.
- `community_type` - (Optional) Community update options. Valid values are `none`, `remove-all`, `remove-regex`,

`append`, or `overwrite`.

- `community_value` - (Optional) If `community_type` is `remove-regex`, `append`, or `overwrite`, the value associated with that setting. For the `append` and `overwrite` types specifically, valid values for `community_value` are `no-export`, `no-advertise`, `local-as`, or `nopeer`.
- `extended_community_type` - (Optional) Extended community update options. Valid values are `none`, `remove-all`, `remove-regex`, `append`, or `overwrite`.
- `extended_community_vaule` - (Optional) If `extended_community_type` is `remove-regex`, `append`, or `overwrite`, the value associated with that setting.

Each `match_address_prefix` section offers the following params:

- `prefix` - (Required) Address prefix.
- `exact` - (Optional, bool) Match exact prefix length.

panos_panorama_bgp_peer_group

This resource allows you to add/update/delete a Panorama BGP peer group.

Import Name

```
<template>:<template_stack>:<virtual_router>:<name>
```

Example Usage

```
resource "panos_panorama_bgp_peer_group" "example" {
    template = "${panos_panorama_template.t.name}"
    virtual_router = "${panos_panorama_bgp.conf.virtual_router}"
    name = "myName"
}

resource "panos_panorama_bgp" "conf" {
    template = "${panos_panorama_template.t.name}"
    virtual_router = "${panos_panorama_virtual_router.rtr.name}"
    router_id = "5.5.5.5"
    as_number = "42"
}

resource "panos_panorama_virtual_router" "rtr" {
    template = "${panos_panorama_template.t.name}"
    name = "my virtual router"
}

resource "panos_panorama_template" "t" {
    name = "myTemplate"
}
```

Argument Reference

One and only one of the following must be specified:

- `template` - The template name.
- `template_stack` - The template stack name.

The following arguments are supported:

- `virtual_router` - (Required) The virtual router to add this BGP peer group to.
- `name` - (Required) The name.
- `enable` - (Optional, bool) Enable or not (default: `true`).

- `aggregated_confed_as_path` - (Optional, bool) The peers understand aggregated confederation AS path (default: `true`).
- `soft_reset_with_stored_info` - (Optional, bool) Soft reset with stored info.
- `type` - (Optional) Peer group type. Valid options are `ebgp` (default), `ebgp-confed`, `ibgp`, or `ibgp-confed`.
- `export_next_hop` - (Optional) Export next hop. Valid values are `original`, `use-self`, or `resolve`.
- `import_next_hop` - (Optional) Import next hop. Valid values are `original`, `use-peer`, or the empty string.
- `remove_private_as` - (Optional, bool) Remove private AS when exporting route. Only available for `type=ebgp`.

panos_panorama_bgp_peer

This resource allows you to add/update/delete a Panorama BGP peer.

Import Name

```
<template>:<template_stack>:<virtual_router>:<bgp_peer_group>:<name>
```

Example Usage

```
data "panos_system_info" "x" {}

// Peer definition that will work starting from PAN-OS 6.1.
resource "panos_panorama_bgp_peer" "example" {
    template = "${panos_panorama_template.t.name}"
    virtual_router = "${panos_panorama_bgp.conf.virtual_router}"
    bgp_peer_group = "${panos_panorama_bgp_peer_group.pg.name}"
    name = "peer1"
    peer_as = "${panos_panorama_bgp.conf.as_number}"
    local_address_interface = "${panos_panorama_ethernet_interface.e.name}"
    local_address_ip = "${panos_panorama_ethernet_interface.e.static_ips.0}"
    peer_address_ip = "5.6.7.8"
    max_prefixes = "unlimited"
    bfd_profile = "${
        data.panos_system_info.x.version_major >= 7 ?
            data.panos_system_info.x.version_minor >= 1 ? "None" : ""
        : ""
    }"
    address_family_type = "${data.panos_system_info.x.version_major >= 8 ? "ipv4" : ""}"
    reflector_client = "${data.panos_system_info.x.version_major >= 8 ? "non-client" : ""}"
    min_route_advertisement_interval = "${
        data.panos_system_info.x.version_major >= 8 ?
            data.panos_system_info.x.version_minor >= 1 ? 30 : 0
        : 0
    }"
}

resource "panos_panorama_bgp_peer_group" "pg" {
    template = "${panos_panorama_template.t.name}"
    virtual_router = "${panos_panorama_bgp.conf.virtual_router}"
    name = "myName"
    type = "ibgp"
}

resource "panos_panorama_bgp" "conf" {
    template = "${panos_panorama_template.t.name}"
    virtual_router = "${panos_panorama_virtual_router.rtr.name}"
    router_id = "5.5.5.5"
    as_number = "42"
}

resource "panos_panorama_virtual_router" "rtr" {
    template = "${panos_panorama_template.t.name}"
    name = "my virtual router"
    interfaces = ["${panos_panorama_ethernet_interface.e.name}"]
}

resource "panos_panorama_ethernet_interface" "e" {
    template = "${panos_panorama_template.t.name}"
    name = "ethernet1/5"
    mode = "layer3"
    static_ips = ["192.168.1.1/24"]
}

resource "panos_panorama_template" "t" {
    name = "myTemplate"
}
```

Argument Reference

One and only one of the following must be specified:

- `template` - The template name.
- `template_stack` - The template stack name.

The following arguments are supported:

- `virtual_router` - (Required) The virtual router to add this BGP peer to.
- `bgp_peer_group` - (Required) The BGP peer group to put this peer into.
- `name` - (Required) The name.
- `enable` - (Optional, bool) Enable or not (default: `true`).
- `peer_as` - (Optional) Peer AS number.
- `local_address_interface` - (Required) Interface to accept BGP session.
- `local_address_ip` - (Optional) Specify exact IP address if interface has multiple addresses.
- `peer_address_ip` - (Required) Peer IP address configuration.
- `reflector_client` - (Optional) This peer is reflector client. Valid values are `non-client`, `client`, or `meshed-client`.
- `peering_type` - (Optional) Peering type that affects NOPEER community value handling. Valid values are `unspecified` (default) or `bilateral`.
- `max_prefixes` - (Optional) Maximum of prefixes to receive from the peer. This can be a number such as `"5000"` (default) or `unlimited`.
- `auth_profile` - (Optional) Auth profile.
- `keep_alive_interval` - (Optional, int) Keep alive interval, in seconds (default: `30`).
- `multi_hop` - (Optional, int) IP TTL value used for sending BGP packet.
- `open_delay_time` - (Optional, int) Open delay time, in seconds.
- `hold_time` - (Optional, int) Hold time, in seconds.
- `idle_hold_time` - (Optional, int) Idle hold time, in seconds.
- `allow_incoming_connections` - (Optional, bool) Allow incoming connections (default: `true`).
- `incoming_connections_remote_port` - (Optional, int) Restrict remote port for incoming BGP connections.
- `allow_outgoing_connections` - (Optional, bool) Allow outgoing connections (default: `true`).
- `outgoing_connections_local_port` - (Optional, int) Use specific local port for outgoing BGP connections.
- `bfd_profile` - (Optional, PAN-OS 7.1+) BFD profile. This can be a specific BFD profile name, `None` (disables BFD), or `Inherit-vr-global-setting`.
- `enable_mp_bgp` - (Optional, bool, PAN-OS 8.0+) Enable MP BGP.

- `address_family_type` - (Optional, PAN-OS 8.0+) Set the AFI for this peer. Valid values are `ipv4` or `ipv6`.
- `subsequent_address_family_unicast` - (Optional, bool, PAN-OS 8.0+) Enable unicast subsequent address family for this peer.
- `subsequent_address_family_multicast` - (Optional, bool, PAN-OS 8.0+) Enable multicast subsequent address family for this peer.
- `enable_sender_side_loop_detection` - (Optional, bool, PAN-OS 8.0+) Enable sender side loop detection.
- `min_route_advertisement_interval` - (Optional, int, PAN-OS 8.1+) Minimum route advertisement interval, in seconds.

panos_panorama_bgp_redist_rule

This resource allows you to add/update/delete a Panorama BGP redistribution rule.

Import Name

```
<template>:<template_stack>:<virtual_router>:<name>
```

Example Usage

```
resource "panos_panorama_bgp_redist_rule" "example" {
    template = "${panos_panorama_template.t.name}"
    virtual_router = "${panos_panorama_bgp.conf.virtual_router}"
    route_table = "${data.panos_system_info.x.version_major >= 8 ? "unicast" : ""}"
    name = "192.168.1.0/24"
    set_med = "42"
}

data "panos_system_info" "x" {}

resource "panos_panorama_bgp" "conf" {
    template = "${panos_panorama_template.t.name}"
    virtual_router = "${panos_panorama_virtual_router.rtr.name}"
    router_id = "5.5.5.5"
    as_number = "42"
}

resource "panos_panorama_virtual_router" "rtr" {
    template = "${panos_panorama_template.t.name}"
    name = "my virtual router"
}

resource "panos_panorama_template" "t" {
    name = "myTemplate"
}
```

Argument Reference

One and only one of the following must be specified:

- `template` - The template name.
- `template_stack` - The template stack name.

The following arguments are supported:

- `virtual_router` - (Required) The virtual router to add this BGP redist rule to.

- `name` - (Required) A subnet or a redistribution profile.
- `enable` - (Optional, bool) Enable this rule or not (default: `true`).
- `address_family` - (Optional) The address family. Valid values are `ipv4` (default) or `ipv6`.
- `route_table` - (Optional, PAN-OS 8.0+) Route table to match rule. Valid values are `unicast`, `multicast`, or `both`. As of PAN-OS 8.1, there doesn't seem to be a way to configure this in the GUI, it is always set to `unicast`. Thus, if you're running this resource against PAN-OS 8.0+, the appropriate thing to do is set this value to `unicast` as well to match the GUI functionality.
- `metric` - (Optional, int) Metric value.
- `set_origin` - (Optional) Add the origin path attribute. Valid values are `incomplete` (default), `igp`, or `egp`.
- `set_med` - (Optional) Add the MULTI_EXIT_DISC path attribute.
- `set_local_preference` - (Optional) Add the LOCAL_PREF path attribute.
- `set_as_path_limit` - (Optional, int) Add the AS_PATHLIMIT path attribute.
- `set_communities` - (Optional) List of COMMUNITY path attributes to add.
- `set_extended_communities` - (Optional) List of EXTENDED COMMUNITY path attributes to add.

panos_panorama_device_group_entry

This resource allows you to add/update/delete a specific device in a Panorama device group.

This resource has some overlap with the `panos_panorama_device_group` resource. If you want to use this resource with the other one, then make sure that your `panos_panorama_device_group` spec does not define any `device` blocks, and just stays as "computed".

This is the appropriate resource to use if you have a pre-existing device group in Panorama and don't want Terraform to delete it on `terraform destroy`.

An interesting side effect of the underlying XML API - if the device group does not already exist, then this resource can actually create it. However, since only the single entry for the specific serial number is deleted, then a `terraform destroy` would not remove the device group itself in this situation.

Import Name

```
<device_group>:<serial>
```

Example Usage

```
# Example for a virtual firewall.
resource "panos_panorama_device_group_entry" "example1" {
    device_group = "my device group"
    serial = "00112233"
}

# Example for a physical firewall with multi-vsys enabled.
resource "panos_panorama_device_group_entry" "example2" {
    device_group = "my device group"
    serial = "44556677"
    vsys_list = ["vsys1", "vsys2"]
}
```

Argument Reference

The following arguments are supported:

- `device_group` - (Required) The device group's name.
- `serial` - (Required) The serial number of the firewall.
- `vsys_list` - (Optional) A subset of all available vsys on the firewall that should be in this device group. If the firewall is a virtual firewall, then this parameter should just be omitted.

panos_panorama_device_group

This resource allows you to add/update/delete Panorama device groups.

This resource has some overlap with the `panos_panorama_device_group_entry` resource. If you want to use this resource with the other one, then make sure that your `panos_panorama_device_group` spec does not define any `device` blocks, and just stays as "computed".

This is the appropriate resource to use if `terraform destroy` should delete the device group.

Import Name

```
<name>
```

Example Usage

```
resource "panos_panorama_device_group" "example" {
    name = "my device group"
    description = "description here"
    device {
        serial = "00112233"
    }
    device {
        serial = "44556677"
        vsys_list = ["vsys1", "vsys2"]
    }
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The device group's name.
- `description` - (Optional) The device group's description.
- `device` - The device definition (see below).

The following arguments are valid for each `device` section:

- `serial` - (Required) The serial number of the firewall.
- `vsys_list` - (Optional) A subset of all available vsys on the firewall that should be in this device group. If the firewall is a virtual firewall, then this parameter should just be omitted.

panos_panorama_edl

This resource allows you to add/update/delete Panorama external dynamic lists (EDL).

Setting `repeat_at`

The acceptable PAN-OS values for the `repeat_at` field is a combination of the version of PAN-OS that you're running against and the setting of the `repeat` parameter.

The following shorthand is used:

- `N/A` - `repeat_at` should not be set
- `minute` - A two character minute string (e.g. `07` or `59`)
- `24hr hour` - A two character hour string in 24hr notation (e.g. `09` or `15`)
- `24hr time` - A five character hour/minute string in 24hr notation (e.g. `09:00` or `23:59`)

Here are the valid settings for `repeat_at` given your desired `repeat` value and the version of PAN-OS you're running against:

- PAN-OS 6.1 - 7.0
  - `hourly` - minute
  - `daily`, `weekly`, `monthly` - 24hr time
- PAN-OS 7.1+
  - `every five minutes`, `hourly` - N/A
  - `daily`, `weekly`, `monthly` - 24hr hour

Example Usage

```
resource "panos_panorama_edl" "example" {
    name = "example"
    type = "ip"
    description = "my edl"
    source = "https://example.com"
    repeat = "every five minutes"
    exceptions = ["10.1.1.1", "10.1.1.2"]
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The object's name

- `device_group` - (Optional) The device group (default: `shared`)
- `type` - (Optional) The type of EDL. This can be `ip` (the default; and the only valid value for PAN-OS 6.1 - 7.0), `domain`, `url`, or `predefined` (PAN-OS 8.0+)
- `description` - (Optional) The object's description.
- `source` - (Optional) The EDL source URL
- `certificate_profile` - (Optional) Profile for authenticating client certificates
- `username` - (Optional) EDL username
- `password` - (Optional) EDL password
- `repeat` - (Optional) How often to retrieve the EDL. This can be `hourly` (the default), `daily`, `weekly`, `monthly`, or `every five minutes` (valid for PAN-OS 7.1+)
- `repeat_at` - (Optional) The time at which to retrieve the EDL. Please refer to the section above for how to set this value properly.
- `repeat_day_of_week` - (Optional) If `repeat` is `weekly`, then this should be set to the desired day of the week. Valid values are `sunday`, `monday`, `tuesday`, `wednesday`, `thursday`, `friday`, `saturday`, and `sunday`
- `repeat_day_of_month` - (Optional, int) If `repeat` is `monthly`, then this should be set to the desired day of the month.
- `exceptions` - (Optional, list) Provide a list of exception entries.

panos_panorama_email_server_profile

This resource allows you to add/update/delete Panorama email server profiles.

Import Name

```
<template>:<template_stack>:<vsys>:<device_group>:<name>
```

Example Usage

```
resource "panos_panorama_email_server_profile" "example" {
    name = "myProfile"
    device_group = "shared"
    threat_format = "$serial $severity"
    email_server {
        name = "my-server"
        display_name = "foobar"
        from_email = "source@example.com"
        to_email = "alerts@example.com"
        email_gateway = "mail.example.com"
    }
}
```

Argument Reference

When creating this profile, there are a few options:

- on the Panorama (in the `shared` device group)
- in a vsys in a template
- in a vsys in a template stack

The following arguments are supported:

- `template` - (Optional) The template location. Mutually exclusive with `template_stack` and `device_group`.
- `template_stack` - (Optional) The template stack location. Mutually exclusive with `template` and `device_group`.
- `device_group` - (Optional) The device group location. Mutually exclusive with `template` and `template_stack`.
- `vsys` - (Optional) The vsys. This will likely be `shared`, and it should be defined if you specified either `template` or `template_stack`.
- `name` - (Required) The group's name.
- `config_format` - (Optional) Config format.

- `system_format` - (Optional) System format.
- `threat_format` - (Optional) Threat format.
- `traffic_format` - (Optional) Traffic format.
- `hip_match_format` - (Optional) HIP match format.
- `url_format` - (Optional) URL format.
- `data_format` - (Optional) Data format.
- `wildfire_format` - (Optional) Wildfire format.
- `tunnel_format` - (Optional) Tunnel format.
- `user_id_format` - (Optional) UserID format.
- `gtp_format` - (Optional) GTP format.
- `auth_format` - (Optional) Auth format.
- `sctp_format` - (Optional) SCTP format.
- `iptag_format` - (Optional) IP tag format.
- `escaped_characters` - (Optional) The escaped characters (as a string).
- `escape_character` - (Optional) The escape character.
- `email_server` - (Required, repeatable) The server spec (defined below).

`email_server` supports the following arguments:

- `name` - (Required) Server name.
- `display_name` - (Optional) The display name.
- `from_email` - (Required) From email address.
- `to_email` - (Required) To email address.
- `also_to_email` - (Optional) Secondary to email address.
- `email_gateway` - (Required) The email server.
- `email_server` - (Required, repeatable) The server spec (defined below).

panos_panorama_ethernet_interface

This resource allows you to add/update/delete Panorama ethernet interfaces for templates.

Import Name

```
<template>:<template_stack>:<vsys>:<name>
```

Example Usage

```
# Configure a bare-bones ethernet interface.
resource "panos_panorama_ethernet_interface" "example1" {
    name = "ethernet1/3"
    template = "foo"
    vsys = "vsys1"
    mode = "layer3"
    static_ips = ["10.1.1.1/24"]
    comment = "Configured for internal traffic"
}

# Configure a DHCP ethernet interface for vsys1 to use.
resource "panos_panorama_ethernet_interface" "example2" {
    name = "ethernet1/4"
    template = "bar"
    mode = "layer3"
    enable_dhcp = true
    create_dhcp_default_route = true
    dhcp_default_route_metric = 10
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The ethernet interface's name. This should be something like `ethernet1/X`.
- `template` - (Required) The template name.
- `vsys` - (Optional) The vsys that will use this interface (default: `vsys1`). This should be something like `vsys1` or `vsys3`.
- `mode` - (Required) The interface mode. This can be any of the following values: `layer3`, `layer2`, `virtual-wire`, `tap`, `ha`, `decrypt-mirror`, or `aggregate-group`.
- `static_ips` - (Optional) List of static IPv4 addresses to set for this data interface.
- `enable_dhcp` - (Optional) Set to `true` to enable DHCP on this interface.

- `create_dhcp_default_route` - (Optional) Set to `true` to create a DHCP default route.
- `dhcp_default_route_metric` - (Optional) The metric for the DHCP default route.
- `ipv6_enabled` - (Optional) Set to `true` to enable IPv6.
- `management_profile` - (Optional) The management profile.
- `mtu` - (Optional) The MTU.
- `adjust_tcp_mss` - (Optional) Adjust TCP MSS (default: false).
- `netflow_profile` - (Optional) The netflow profile.
- `lldp_enabled` - (Optional) Enable LLDP (default: false).
- `lldp_profile` - (Optional) LLDP profile.
- `link_speed` - (Optional) Link speed. This can be any of the following: `10`, `100`, `1000`, or `auto`.
- `link_duplex` - (Optional) Link duplex setting. This can be `full`, `half`, or `auto`.
- `link_state` - (Optional) The link state. This can be `up`, `down`, or `auto`.
- `aggregate_group` - (Optional) The aggregate group (applicable for physical firewalls only).
- `comment` - (Optional) The interface comment.
- `ipv4_mss_adjust` - (Optional, PAN-OS 7.1+) The IPv4 MSS adjust value.
- `ipv6_mss_adjust` - (Optional, PAN-OS 7.1+) The IPv6 MSS adjust value.
- `decrypt_forward` - (Optional, PAN-OS 8.1+) Enable decrypt forwarding.
- `rx_policing_rate` - (Optional, PAN-OS 8.1+) Receive policing rate in Mbps.
- `tx_policing_rate` - (Optional, PAN-OS 8.1+) Transmit policing rate in Mbps.
- `dhcp_send_hostname_enable` - (Optional, PAN-OS 9.0+) For DHCP layer3 interfaces: enable sending the firewall or a custom hostname to DHCP server
- `dhcp_send_hostname_value` - (Optional, PAN-OS 9.0+) For DHCP layer3 interfaces: the interface hostname. Leaving this unspecified with `dhcp_send_hostname_enable` set means to send the system hostname.

panos_panorama_gcp_account

This resource allows you to add/update/delete GCP accounts on Panorama.

This resource requires that the GCP plugin be installed.

Import Name

```
<name>
```

Example Usage

```
# A GCP account type (for cluster groups)
resource "panos_panorama_gcp_account" "gcp" {
    name = "myGcpAccount"
    project_id = "gcp-project-123"
    service_account_credential_type = "gcp"
    credential_file = file("gcp-credentials.json")
}

# A GKE account type (for clusters in a group).
resource "panos_panorama_gcp_account" "gke" {
    name = "myGcpAccount"
    project_id = "gcp-project-123"
    service_account_credential_type = "gke"
    credential_file = file("gcp-credentials.json")
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The account's name.
- `description` - (Optional) Account description.
- `project_id` - (Required) The GCP project ID.
- `service_account_credential_type` - (Optional) The service account credential type. Valid values are `gcp` (default) or `gke`.
- `credential_file` - (Required) The contents of a GCP credentials file; use the `file()` function to pass in the credentials file.

panos_panorama_gke_cluster_group

This resource allows you to add/update/delete a GKE cluster group.

This resource requires that the GCP plugin be installed.

Import Name

```
<name>
```

Example Usage

```
resource "panos_panorama_gke_cluster_group" "grp" {
    name = "myCluster"
    gcp_project_credential = panos_panorama_gcp_account.gcp.name
    device_group = panos_panorama_device_group.dg.name
    template_stack = panos_panorama_template_stack.ts.name
}

resource "panos_panorama_device_group" "dg" {
    name = "my device group"
}

resource "panos_panorama_template_stack" "ts" {
    name = "myTemplateStack"
}

resource "panos_panorama_gcp_account" "gcp" {
    name = "myGcpAccount"
    project_id = "gcp-project-123"
    service_account_credential_type = "gcp"
    credential_file = file("gcp-credentials.json")
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The cluster group's name.
- `description` - (Optional) The description.
- `gcp_project_credential` - (Required) The GCP account to use.
- `device_group` - (Required) The device group.
- `template_stack` - (Required) The template stack.

panos_panorama_gke_cluster

This resource allows you to add/update/delete a GKE cluster in a cluster group.

This resource requires that the GCP plugin be installed.

Import Name

```
<gke_cluster_group>:<name>
```

Example Usage

```
resource "panos_panorama_gke_cluster" "cluster" {
    gke_cluster_group = panos_panorama_gke_cluster_group.grp.name
    name = "cluster1"
    gcp_zone = "us-central-1b"
    cluster_credential = panos_panorama_gcp_account.gke.name
}

resource "panos_panorama_gke_cluster_group" "grp" {
    name = "myCluster"
    gcp_project_credential = panos_panorama_gcp_account.gcp.name
    device_group = panos_panorama_device_group.dg.name
    template_stack = panos_panorama_template_stack.ts.name
}

resource "panos_panorama_device_group" "dg" {
    name = "my device group"
}

resource "panos_panorama_template_stack" "ts" {
    name = "myTemplateStack"
}

resource "panos_panorama_gcp_account" "gcp" {
    name = "myGcpAccount"
    project_id = "gcp-project-123"
    service_account_credential_type = "gcp"
    credential_file = file("gcp-credentials.json")
}

resource "panos_panorama_gcp_account" "gke" {
    name = "myGkeAccount"
    project_id = "gcp-project-123"
    service_account_credential_type = "gke"
    credential_file = file("gcp-credentials.json")
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The cluster group's name.
- `gke_cluster_group` - (Required) The cluster group name.
- `gcp_zone` - (Required) The GCP zone.
- `cluster_credential` - (Required) The GKE account to use.

panos_panorama_gre_tunnel

This resource allows you to add/update/delete Panorama GRE tunnels.

Minimum PAN-OS version: 9.0

Import Name

```
<template>::<name>
```

Example Usage

```
resource "panos_panorama_gre_tunnel" "example" {
    template = panos_panorama_template.tmpl.name
    name = "myGreTunnel"
    interface = panos_panorama_ethernet_interface.ei.name
    local_address_value = panos_panorama_ethernet_interface.ei.static_ips.0
    peer_address = "192.168.1.1"
    tunnel_interface = panos_panorama_tunnel_interface.ti.name
    ttl = 42
}

resource "panos_panorama_template" "tmpl" {
    name = "My Template"
}

resource "panos_panorama_ethernet_interface" "ei" {
    template = panos_panorama_template.tmpl.name
    name = "ethernet1/1"
    vsys = "vsys1"
    mode = "layer3"
    static_ips = ["10.1.1.1/24"]
}

resource "panos_panorama_tunnel_interface" "ti" {
    template = panos_panorama_template.tmpl.name
    name = "tunnel.7"
    vsys = "vsys1"
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The GRE tunnel name.
- `template` - (Required) The template name.

- `interface` - (Required) Interface to terminate tunnel.
- `local_address_type` - (Optional) Type of local address. Valid values are `ip` (default) or `floating-ip` .
- `local_address_value` - (Required) IP address value.
- `peer_address` - (Required) Peer IP address.
- `tunnel_interface` - (Required) Tunnel interface to apply the GRE tunnel to.
- `ttl` - (Optional, int) Time to live.
- `copy_tos` - (Optional, bool) Copy IP TOS bits from inner packet to GRE packet.
- `enable_keep_alive` - (Optional, bool) Enable tunnel monitoring.
- `keep_alive_interval` - (Optional, int) Keep alive interval.
- `keep_alive_retry` - (Optional, int) Keep alive retry.
- `keep_alive_hold_timer` - (Optional, int) Keep alive hold timer.
- `disabled` - (Optional, bool) Disable the GRE tunnel.

panos_panorama_http_server_profile

This resource allows you to add/update/delete Panorama HTTP server profiles.

Minimum PAN-OS version: 7.1

Example Usage

```
resource "panos_panorama_http_server_profile" "example" {
    device_group = "shared"
    name = "myProfile"
    url_format {
        name = "my url format"
        uri_format = "/api/incident/url"
        headers = {
            "Content-Type": "text/plain",
        }
    }
    http_server {
        name = "myServer"
        address = "siem.example.com"
    }
}
```

Argument Reference

When creating this profile, there are a few options:

- on the Panorama (in the `shared` device group)
- in a vsys in a template
- in a vsys in a template stack

The following arguments are supported:

- `template` - (Optional) The template location. Mutually exclusive with `template_stack` and `device_group`.
- `template_stack` - (Optional) The template stack location. Mutually exclusive with `template` and `device_group`.
- `device_group` - (Optional) The device group location. Mutually exclusive with `template` and `template_stack`.
- `vsys` - (Optional) The vsys. This will likely be `shared`, and it should be defined if you specified either `template` or `template_stack`.
- `name` - (Required) The group's name.
- `tag_registration` - (Optional, bool) Perform tag registration.
- `config_format` - (Optional) A format folder spec for config (defined below).
- `system_format` - (Optional) A format folder spec for system (defined below).

- `threat_format` - (Optional) A format folder spec for threat (defined below).
- `traffic_format` - (Optional) A format folder spec for traffic (defined below).
- `hip_match_format` - (Optional) A format folder spec for HIP match (defined below).
- `url_format` - (Optional) A format folder spec for url (defined below).
- `data_format` - (Optional) A format folder spec for data (defined below).
- `wildfire_format` - (Optional) A format folder spec for wildfire (defined below).
- `tunnel_format` - (Optional) A format folder spec for tunnel (defined below).
- `user_id_format` - (Optional) A format folder spec for user ID (defined below).
- `gtp_format` - (Optional) A format folder spec for gtp (defined below).
- `auth_format` - (Optional) A format folder spec for auth (defined below).
- `sctp_format` - (Optional, PAN-OS 8.1+) A format folder spec for sctp (defined below).
- `iptag_format` - (Optional, PAN-OS 9.0+) A format folder spec for iptag (defined below).
- `http_server` - (Optional, repeatable) A server spec (defined below).

All format folders support the following arguments:

- `name` - (Optional) The name.
- `uri_format` - (Optional) The URI format.
- `payload` - (Optional) The payload.
- `headers` - (Optional, map) A map of HTTP headers and their values.
- `params` - (Optional, map) A map of HTTP params and their values.

`http_server` supports the following arguments:

- `name` - (Required) The server name.
- `address` - (Required) The server address.
- `protocol` - (Optional) The protocol. Valid values are `HTTPS` (default) or `HTTP`.
- `port` - (Optional, int) The port number (default: 443).
- `http_method` - (Optional) The HTTP method (default: `POST`).
- `username` - (Optional) The username.
- `password` - (Optional) The password.
- `tls_version` - (Optional) The TLS version.
- `certificate_profile` - (Optional) The certificate profile.

panos_panorama_ike_crypto_profile

This resource allows you to add/update/delete Panorama IKE crypto profiles to a template or template stack.

Import Name

```
<template>:<template_stack>:<name>
```

Example Usage

```
resource "panos_panorama_ike_crypto_profile" "example" {
    name = "example"
    template = "my template"
    dh_groups = ["group1", "group2"]
    authentications = ["md5", "sha1"]
    encryptions = ["des"]
    lifetime_value = 8
    authentication_multiple = 3
}
```

Argument Reference

One and only one of the following must be specified:

- `template` - The template name.
- `template_stack` - The template stack name.

The following arguments are supported:

- `name` - (Required) The object's name
- `dh_groups` - (Required, list) List of DH Group entries. Values should have a prefix if `group`.
- `authentications` - (Required, list) List of authentication types. This c
- `encryptions` - (Required, list) List of encryption types. Valid values are `des`, `3des`, `aes-128-cbc`, `aes-192-cbc`, and `aes-256-cbc`.
- `lifetime_type` - (Optional) The lifetime type. Valid values are `seconds`, `minutes`, `hours` (the default), and `days`.
- `lifetime_value` - (Optional, int) The lifetime value.
- `authentication_multiple` - (Optional, PAN-OS 7.0+, int) IKEv2 SA reauthentication interval equals authetication-multiple * rekey-lifetime; 0 means reauthentication is disabled.

panos_panorama_ike_gateway

This resource allows you to add/update/delete Panorama IKE gateways for both templates and template stacks.

Example Usage

```
resource "panos_panorama_ike_gateway" "example" {
    name = "example"
    template = "my template"
    peer_ip_type = "dynamic"
    interface = "loopback.42"
    pre_shared_key = "secret"
    local_id_type = "ipaddr"
    local_id_value = "10.1.1.1"
    peer_id_type = "ipaddr"
    peer_id_value = "10.5.1.1"
    ikev1_crypto_profile = "myIkeProfile"
}
```

Argument Reference

One and only one of the following must be specified:

- `template` - The template name.
- `template_stack` - The template stack name.

The following arguments are supported:

- `name` - (Required) The object's name
- `version` - (Optional, PAN-OS 7.0+) The IKE gateway version. Valid values are `ikev1`, (the default), `ikev2`, or `ikev2-preferred`. For PAN-OS 6.1, only `ikev1` is acceptable.
- `enable_ipv6` - (Optional, PAN-OS 7.0+, bool) Enable IPv6 or not.
- `disabled` - (Optional, PAN-OS 7.0+, bool) Set to `true` to disable.
- `peer_ip_type` - (Optional) The peer IP type. Valid values are `ip`, `dynamic`, and `fqdn` (PANOS 8.1+).
- `peer_ip_value` - (Optional) The peer IP value.
- `interface` - (Required) The interface.
- `local_ip_address_type` - (Optional) The local IP address type. Valid values for this are `ip`, `floating-ip`, or an empty string (the default) which is `None`.
- `local_ip_address_value` - (Optional) The IP address if `local_ip_address_type` is set to `ip`.
- `auth_type` - (Optional) The auth type. Valid values are `pre-shared-key` (the default), or `certificate`.
- `pre_shared_key` - (Optional) The pre-shared key value.

- `local_id_type` - (Optional) The local ID type. Valid values are `ipaddr`, `fqdn`, `ufqdn`, `keyid`, or `dn`.
- `local_id_value` - (Optional) The local ID value.
- `peer_id_type` - (Optional) The peer ID type. Valid values are `ipaddr`, `fqdn`, `ufqdn`, `keyid`, or `dn`.
- `peer_id_value` - (Optional) The peer ID value.
- `peer_id_check` - (Optional) Enable peer ID wildcard match for certificate authentication. Valid values are `exact` or `wildcard`.
- `local_cert` - (Optional) The local certificate name.
- `cert_enable_hash_and_url` - (Optional, PAN-OS 7.0+, bool) Set to `true` to use hash-and-url for local certificate.
- `cert_base_url` - (Optional) The host and directory part of URL for local certificates.
- `cert_use_management_as_source` - (Optional, PAN-OS 7.0+, bool) Set to `true` to use management interface IP as source to retrieve http certificates
- `cert_permit_payload_mismatch` - (Optional, bool) Set to `true` to permit peer identification and certificate payload identification mismatch.
- `cert_profile` - (Optional) Profile for certificate valdiation during IKE negotiation.
- `cert_enable_strict_validation` - (Optional, bool) Set to `true` to enable strict validation of peer's extended key use.
- `enable_passive_mode` - (Optional, bool) Set to `true` to enable passive mode (responder only).
- `enable_nat_traversal` - (Optional, bool) Set to `true` to enable NAT traversal.
- `nat_traversal_keep_alive` - (Optional, int) Sending interval for NAT keep-alive packets (in seconds). For versions 6.1 - 8.1, this param, if specified, should be a multiple of 10 between 10 and 3600 to be valid.
- `nat_traversal_enable_udp_checksum` - (Optional, bool) Set to `true` to enable NAT traversal UDP checksum.
- `enable_fragmentation` - (Optional, bool) Set to `true` to enable fragmentation.
- `ikev1_exchange_mode` - (Optional) The IKEv1 exchange mode.
- `ikev1_crypto_profile` - (Optional) IKEv1 crypto profile.
- `enable_dead_peer_detection` - (Optional, bool) Set to `true` to enable dead peer detection.
- `dead_peer_detection_interval` - (Optional, int) The dead peer detection interval.
- `dead_peer_detection_retry` - (Optional, int) Number of retries before disconnection.
- `ikev2_crypto_profile` - (Optional, PAN-OS 7.0+) IKEv2 crypto profile.
- `ikev2_cookie_validation` - (Optional, PAN-OS 7.0+) Set to `true` to require cookie.
- `enable_liveness_check` - (Optional, , PAN-OS 7.0+bool) Set to `true` to enable sending empty information liveness check message.
- `liveness_check_interval` - (Optional, , PAN-OS 7.0+int) Delay interval before sending probing packets (in seconds).

panos_panorama_ipsec_crypto_profile

This resource allows you to add/update/delete Panorama IPSec crypto profiles for both templates and template stacks.

Import Name

```
<template>:<template_stack>:<name>
```

Example Usage

```
resource "panos_panorama_ipsec_crypto_profile" "example" {
    name = "example"
    template = "my template"
    authentications = ["md5", "sha384"]
    encryptions = ["des", "aes-128-cbc"]
    dh_group = "group14"
    lifetime_type = "hours"
    lifetime_value = 4
    lifesize_type = "mb"
    lifesize_value = 1
}
```

Argument Reference

One and only one of the following must be specified:

- `template` - The template name.
- `template_stack` - The template stack name.

The following arguments are supported:

- `name` - (Required) The object's name
- `protocol` - (Optional) The protocol. Valid values are `esp` (the default) or `ah`
- `authentications` - (Required, list) - List of authentication types.
- `encryptions` - (Required, list) - List of encryption types. Valid values are `des`, `3des`, `aes-128-cbc`, `aes-192-cbc`, `aes-256-cbc`, `aes-128-gcm`, `aes-256-gcm`, and `null`. Note that the "gcm" values are only available in PAN-OS 7.0+.
- `dh_group` - (Optional) The DH group value. Valid values should start with the string `group`.
- `lifetime_type` - (Optional) The lifetime type. Valid values are `seconds`, `minutes`, `hours` (the default), or `days`.
- `lifetime_value` - (Optional, int) The lifetime value.

- `lifesize_type` - (Optional) The lifesize type. Valid values are `kb`, `mb`, `gb`, or `tb`.
- `lifesize_value` - (Optional, int) the lifesize value.

panos_panorama_ipsec_tunnel

This resource allows you to add/update/delete Panorama IPSec tunnels for templates.

A large number of params have prefixes:

- `ak` - Auto key
- `mk` - Manual key
- `gps` - GlobalProtect Satellite

Example Usage

```
resource "panos_panorama_ipsec_tunnel" "example" {
    name = "example"
    template = "my template"
    tunnel_interface = "tunnel.7"
    anti_replay = true
    ak_ike_gateway = "myIkeGateway"
    ak_ipsec_crypto_profile = "myIkeProfile"
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The object's name
- `template` - (Required) The template name.
- `tunnel_interface` - (Required) The tunnel interface.
- `anti_replay` - (Optional, bool) Set to `true` to enable Anti-Replay check on this tunnel.
- `enable_ipv6` - (Optional, PAN-OS 7.0+, bool) Set to `true` to enable IPv6.
- `copy_tos` - (Optional, bool) Set to `true` to copy IP TOS bits from inner packet to IPSec packet (not recommended).
- `copy_flow_label` - (Optional, PAN-OS 7.0+, bool) Set to `true` to copy IPv6 flow label for 6in6 tunnel from inner packet to IPSec packet (not recommended).
- `disabled` - (Optional, PAN-OS 7.0+, bool) Set to `true` to disable this IPSec tunnel.
- `type` - (Optional) The type. Valid values are `auto-key` (the default), `manual-key`, or `global-protect-satellite`.
- `ak_ike_gateway` - (Optional) IKE gateway name.
- `ak_ipsec_crypto_profile` - (Optional) IPSec crypto profile name.
- `mk_local_spi` - (Optional) Outbound SPI, hex format.

- `mk_remote_spi` - (Optional) Inbound SPI, hex format.
- `mk_local_address_ip` - (Optional) Specify exact IP address if interface has multiple addresses.
- `mk_local_address_floating_ip` - (Optional) Floating IP address in HA Active-Active configuration.
- `mk_protocol` - (Optional) Manual key protocol. Valid valies are `esp` or `ah`.
- `mk_auth_type` - (Optional) Authentication algorithm. Valid values are `md5`, `sha1`, `sha256`, `sha384`, `sha512`, or `none`.
- `mk_auth_key` - (Optional) The auth key for the given auth type.
- `mk_esp_encryption_type` - (Optional) The encryption algorithm. Valid values are `des`, `3des`, `aes-128-cbc`, `aes-192-cbc`, `aes-256-cbc`, or `null`.
- `mk_esp_encryption_key` - (Optional) The encryption key.
- `gps_interface` - (Optional) Interface to communicate with portal.
- `gps_portal_address` - (Optional) GlobalProtect portal address.
- `gps_prefer_ipv6` - (Optional, PAN-OS 8.0+, bool) Prefer to register the portal in IPv6. Only applicable to FQDN portal-address.
- `gps_interface_ip_ipv4` - (Optional) specify exact IP address if interface has multiple addresses (IPv4).
- `gps_interface_ip_ipv6` - (Optional, PAN-OS 8.0+) specify exact IP address if interface has multiple addresses (IPv6).
- `gps_interface_floating_ip_ipv4` - (Optional, PAN-OS 7.0+) Floating IPv4 address in HA Active-Active configuration.
- `gps_interface_floating_ip_ipv6` - (Optional, PAN-OS 8.0+) Floating IPv6 address in HA Active-Active configuration.
- `gps_publish_connected_routes` - (Optional, bool) Set to `true` to to publish connected and static routes.
- `gps_publish_routes` - (Optional, list) Specify list of routes to publish to Global Protect Gateway.
- `gps_local_certificate` - (Optional) GlobalProtect satellite certificate file name.
- `gps_certificate_profile` - (Optional) Profile for authenticating GlobalProtect gateway certificates.
- `enable_tunnel_monitor` - (Optional, bool) Enable tunnel monitoring on this tunnel.
- `tunnel_monitor_destination_ip` - (Optional) Destination IP to send ICMP probe.
- `tunnel_monitor_source_ip` - (Optional) Source IP to send ICMP probe
- `tunnel_monitor_profile` - (Optional) Tunnel monitor profile.
- `tunnel_monitor_proxy_id` - (Optional, PAN-OS 7.0+) Which proxy-id (or proxy-id-v6) the monitoring traffic will use.

panos_panorama_ipsec_tunnel_proxy_id_ipv4

This resource allows you to add/update/delete Panorama IPSec tunnel proxy IDs to a parent auto key IPSec tunnel for templates.

Import Name

```
<template>:<template_stack>:<ipsec_tunnel>:<name>
```

Example Usage

```
resource "panos_panorama_ipsec_tunnel_proxy_id_ipv4" "example" {
    template = "my template"
    ipsec_tunnel = "myIpsecTunnel"
    name = "example"
    local = "10.1.1.1"
    remote = "10.2.1.1"
    protocol_any = true
}
```

Argument Reference

The following arguments are supported:

- `template` - (Required) The template name.
- `name` - (Required) The object's name
- `ipsec_tunnel` - (Required) The auto key IPSec tunnel to attach this proxy ID to.
- `local` - (Optional) IP subnet or IP address represents local network.
- `remote` - (Optional) IP subnet or IP address represents remote network.
- `protocol_any` - (Optional, bool) Set to `true` for any IP protocol.
- `protocol_number` - (Optional, int) IP protocol number.
- `protocol_tcp_local` - (Optional, int) Local TCP port number.
- `protocol_tcp_remote` - (Optional, int) Remote TCP port number.
- `protocol_udp_local` - (Optional, int) Local UDP port number.
- `protocol_udp_remote` - (Optional, int) Remote UDP port number.

panos_panorama_layer2_subinterface

This resource allows you to add/update/delete Panorama layer2 subinterfaces.

Import Name

```
<template>::<interface_type>:<parent_interface>:<parent_mode>:<vsys>:<name>
```

Example Usage

```
resource "panos_panorama_layer2_subinterface" "example" {
    template = panos_panorama_template.tmpl.name
    parent_interface = panos_panorama_ethernet_interface.e.name
    parent_mode = panos_panorama_ethernet_interface.e.mode
    vsys = "vsys1"
    name = "ethernet1/5.5"
    tag = 5
}

resource "panos_panorama_ethernet_interface" "e" {
    template = panos_panorama_template.tmpl.name
    name = "ethernet1/5"
    vsys = "vsys1"
    mode = "layer2"
}

resource "panos_panorama_template" "tmpl" {
    name = "myTemplate"
}
```

Argument Reference

The following arguments are supported:

- `template` - (Required) The template name.
- `interface_type` - (Optional) The interface type. Valid values are `ethernet` (default) or `aggregate-ethernet`.
- `parent_interface` - (Required) The name of the parent interface.
- `parent_mode` - (Optional) The parent's mode. Valid values are `layer2` (default) or `virtual-wire`.
- `vsys` - (Required) The vsys that will use this interface. This should be something like `vsys1` or `vsys3`.
- `name` - (Required) The interface's name.
- `tag` - (Optional, int) The interface's tag.

- `netflow_profile` - (Optional) The netflow profile.
- `comment` - (Optional) The interface comment.

panos_panorama_layer3_subinterface

This resource allows you to add/update/delete Panorama layer3 subinterfaces.

Import Name

```
<template>::<interface_type>:<parent_interface>:<vsys>:<name>
```

Example Usage

```
resource "panos_panorama_layer3_subinterface" "example" {
    template = panos_panorama_template.tmpl.name
    parent_interface = panos_panorama_ethernet_interface.e.name
    vsys = "vsys1"
    name = "ethernet1/5.5"
    tag = 5
    static_ips = ["10.1.1.1/24"]
    comment = "Configured for internal traffic"
}

resource "panos_panorama_ethernet_interface" "e" {
    template = panos_panorama_template.tmpl.name
    name = "ethernet1/5"
    vsys = "vsys1"
    mode = "layer3"
}

resource "panos_panorama_template" "tmpl" {
    name = "myTemplate"
}
```

Argument Reference

The following arguments are supported:

- `template` - (Required) The template name.
- `interface_type` - (Optional) The interface type. Valid values are `ethernet` (default) or `aggregate-ethernet`.
- `parent_interface` - (Required) The name of the parent interface.
- `vsys` - (Required) The vsys that will use this interface. This should be something like `vsys1` or `vsys3`.
- `name` - (Required) The interface's name.
- `tag` - (Optional, int) The interface's tag.
- `static_ips` - (Optional) List of static IPv4 addresses.

- `ipv6_enabled` - (Optional, bool) Set to `true` to enable IPv6.
- `ipv6_interface_id` - (Optional) The IPv6 interface ID.
- `management_profile` - (Optional) The management profile.
- `mtu` - (Optional) The MTU.
- `adjust_tcp_mss` - (Optional) Adjust TCP MSS (default: false).
- `ipv4_mss_adjust` - (Optional) The IPv4 MSS adjust value.
- `ipv6_mss_adjust` - (Optional) The IPv6 MSS adjust value.
- `netflow_profile` - (Optional) The netflow profile.
- `enable_dhcp` - (Optional, bool) Set to `true` to enable DHCP.
- `create_dhcp_default_route` - (Optional) Set to `true` to create a DHCP default route.
- `dhcp_default_route_metric` - (Optional) The metric for the DHCP default route.
- `comment` - (Optional) The interface comment.
- `decrypt_forward` - (Optional, bool, PAN-OS 8.1+) Set to `true` to enable decrypt forward.
- `dhcp_send_hostname_enable` - (Optional, PAN-OS 9.0+) For DHCP layer3 interfaces: enable sending the firewall or a custom hostname to DHCP server
- `dhcp_send_hostname_value` - (Optional, PAN-OS 9.0+) For DHCP layer3 interfaces: the interface hostname. Leaving this unspecified with `dhcp_send_hostname_enable` set means to send the system hostname.

panos_panorama_log_forwarding_profile

This resource allows you to add/update/delete log forwarding profiles.

Minimum PAN-OS version: 8.0

Import Name

```
<device_group>:<name>
```

Example Usage

```
resource "panos_panorama_log_forwarding_profile" "example" {
    name = "myProfile"
    device_group = "shared"
    description = "made by Terraform"
    match_list {
        name = "myMatchList"
        log_type = "url"
        http_server_profiles = [
            "http1",
            "http2",
        ]
        action {
            name = "tagging int"
            tagging_integration {
                timeout = 5
                local_registration {
                    tags = [
                        panos_panorama_administrative_tag.t.name,
                    ]
                }
            }
        }
        action {
            name = "azure int"
            azure_integration { }
        }
    }
}

resource "panos_panorama_administrative_tag" "t" {
    name = "myTag"
    color = "color12"
}
```

Argument Reference

The following arguments are supported:

- `device_group` - (Optional) The device group (default: `shared`).
- `name` - (Required) The group's name.
- `description` - (Optional) The description.
- `enhanced_logging` - (Optional, bool, PAN-OS 8.1+) Set to `true` to enable enhanced logging.
- `match_list` - (Optional, repeatable) The match list spec (defined below).

`match_list` supports the following arguments:

- `name` - (Required) The name.
- `description` - (Optional) The description.
- `log_type` - (Optional) The log type. Valid values are `traffic` (default), `threat`, `wildfire`, `url`, `data`, `gtp`, `tunnel`, `auth`, or `sctp`.
- `filter` - (Optional) The filter (default: `All Logs`).
- `send_to_panorama` - (Optional, bool) Set to `true` to send to Panorama.
- `snmptrap_server_profiles` - (Optional) List of SNMP server profiles.
- `email_server_profiles` - (Optional) List of email server profiles.
- `syslog_server_profiles` - (Optional) List of syslog server profiles.
- `http_server_profiles` - (Optional) List of http server profiles.
- `action` - (Optional, repeatable) Match list action spec (defined below).

`match_list.action` supports the following arguments:

- `name` - (Required) The name.
- `azure_integration` - (Optional) The Azure integration spec (defined below). Mutually exclusive with `tagging_integration`.
- `tagging_integration` - (Optional) The tagging integration spec (defined below). Mutually exclusive with `azure_integration`.

`match_list.action.azure_integration` supports the following arguments:

- `azure_integration` - (Optional, bool) This param defaults to `true` and should not be changed.

`match_list.action.tagging_integration` supports the following arguments:

- `action` - (Optional) The action. Valid values are `add-tag` (default) or `remove-tag`.
- `target` - (Optional) The target. Valid values are `source-address` (default) or `destination-address`.
- `timeout` - (Optional, int) The timeout.
- `local_registration` - (Optional) The local registration spec (defined below). Only one of `local_registration`, `remote_registration`, or `panorama_registration` should be defined.

- `remote_registration` - (Optional) The remote registration spec (defined below). Only one of `local_registration`, `remote_registration`, or `panorama_registration` should be defined.
- `panorama_registration` - (Optional) The panorama registration spec (defined below). Only one of `local_registration`, `remote_registration`, or `panorama_registration` should be defined.

`match_list.action.tagging_integration.local_registration` supports the following arguments:

- `tags` - (Required) List of administrative tags.

`match_list.action.tagging_integration.remote_registration` supports the following arguments:

- `tags` - (Required) List of administrative tags.
- `http_profile` - (Required) The HTTP profile.

`match_list.action.tagging_integration.panorama_registration` supports the following arguments:

- `tags` - (Required) List of administrative tags.

panos_panorama_loopback_interface

This resource allows you to add/update/delete Panorama loopback interfaces for templates.

Import Name

```
<template>:<template_stack>:<vsys>:<name>
```

Example Usage

```
resource "panos_panorama_loopback_interface" "example1" {
    name = "loopback.2"
    template = "myStack"
    comment = "my loopback interface"
    static_ips = ["10.1.1.1"]
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The interface's name. This must start with `loopback.`.
- `template` - (Required) The template name.
- `vsys` - (Optional) The vsys that will use this interface (default: `vsys1`).
- `comment` - (Optional) The interface comment.
- `netflow_profile` - (Optional) The netflow profile.
- `static_ips` - (Optional) List of static IPv4 addresses to set for this data interface.
- `management_profile` - (Optional) The management profile.
- `mtu` - (Optional) The MTU.
- `adjust_tcp_mss` - (Optional, bool) Adjust TCP MSS (default: false).
- `ipv4_mss_adjust` - (Optional, PAN-OS 8.0+) The IPv4 MSS adjust value.
- `ipv6_mss_adjust` - (Optional, PAN-OS 8.0+) The IPv6 MSS adjust value.

panos_panorama_management_profile

This resource allows you to add/update/delete Panorama interface management profiles for both templates and template stacks.

Import Name

```
<template>:<template_stack>:<name>
```

Example Usage

```
resource "panos_panorama_management_profile" "example" {
    name = "allow ping"
    template = "foo"
    ping = true
    permitted_ips = ["10.1.1.0/24", "192.168.80.0/24"]
}
```

Argument Reference

One and only one of the following must be specified:

- `template` - The template name.
- `template_stack` - The template stack name.

The following arguments are supported:

- `name` - (Required) The management profile's name.
- `ping` - (Optional) Allow ping.
- `telnet` - (Optional) Allow telnet.
- `ssh` - (Optional) Allow SSH.
- `http` - (Optional) Allow HTTP.
- `http_ocsp` - (Optional) Allow HTTP OCSP.
- `https` - (Optional) Allow HTTPS.
- `snmp` - (Optional) Allow SNMP.
- `response_pages` - (Optional) Allow response pages.
- `userid_service` - (Optional) Allow User ID service.

- `userid_syslog_listener_ssl` - (Optional) Allow User ID syslog listener for SSL.
- `userid_syslog_listener_udp` - (Optional) Allow User ID syslog listener for UDP.
- `permitted_ips` - (Optional) The list of permitted IP addresses or address ranges for this management profile.

panos_panorama_monitor_profile

This resource allows you to add/update/delete Panorama monitor profiles.

Minimum PAN-OS version: 7.1

Import Name

```
<template>:<template_stack>:<name>
```

Example Usage

```
resource "panos_panorama_monitor_profile" "example" {
    name = "myProfile"
    interval = 5
    threshold = 3
}
```

Argument Reference

One and only one of the following must be specified:

- `template` - The template name.
- `template_stack` - The template stack name.

The following arguments are supported:

- `name` - (Required) The monitor profile name.
- `interval` - (Optional, int) The probing interval in seconds.
- `threshold` - (Optional, int) The number of failed probes to determine that the tunnel is down.
- `action` - (Optional) Action triggered when tunnel's status changes. Valid values are `wait-recover` (default) or `fail-over`.

panos_panorama_nat_rule_group

This resource allows you to add/update/delete a group of Panorama NAT rules.

This resource manages clusters of NAT rules in a single device group, enforcing both the contents of individual rules as well as their ordering. Rules are defined in a `rule` config block.

Although you cannot modify non-group NAT rules with this resource, the `position_keyword` and `position_reference` parameters allow you to reference some other NAT rule that already exists, using it as a means to ensure some rough placement within the ruleset as a whole.

Best Practices

As is to be expected, if you are separating your deployment across multiple plan files, make sure that at most only one plan specifies any given absolute positioning keyword such as "top" or "directly below", otherwise they'll keep shoving each other out of the way indefinitely.

Best practices are to specify one group as `top` (if you need it), one group as `bottom` (if needed), then all other groups should be `above` the first rule of the bottom group. You do it this way because rules will natually be added at the tail end of the rulebase, so they will always be `after` the first group, but what you want is for them to be `before` the last group's rules.

Example Usage

```
resource "panos_panorama_nat_rule_group" "bot" {
    device_group = "${panos_panorama_device_group.dg.name}"
    rule {
        name = "second"
        original_packet {
            source_zones = ["z2"]
            destination_zone = "z3"
            destination_interface = "any"
            source_addresses = ["any"]
            destination_addresses = ["any"]
        }
        translated_packet {
            source {}
            destination {
                static_translation {
                    address = "10.2.3.1"
                    port = 5678
                }
            }
        }
    }
    rule {
        name = "third"
        original_packet {
            source_zones = ["z3"]
            destination_zone = "z2"
            destination_interface = "any"
```

```
            source_addresses = ["any"]
            destination_addresses = ["any"]
        }
        translated_packet {
            source {
                static_ip {
                    translated_address = "192.168.1.5"
                    bi_directional = true
                }
            }
            destination {}
        }
    }
}

resource "panos_panorama_nat_rule_group" "top" {
    device_group = "${panos_panorama_device_group.dg.name}"
    position_keyword = "directly before"
    position_reference = "${panos_panorama_nat_rule_group.bot.rule.0.name}"
    rule {
        name = "first"
        target {
            serial = "123456"
            vsys_list = ["vsys1", "vsys2"]
        }
        original_packet {
            source_zones = ["z1"]
            destination_zone = "z1"
            destination_interface = "any"
            source_addresses = ["any"]
            destination_addresses = ["any"]
        }
        translated_packet {
            source {
                dynamic_ip_and_port {
                    interface_address {
                        interface = "ethernet1/6"
                    }
                }
            }
            destination {
                static_translation {
                    address = "10.1.1.1"
                    port = 1234
                }
            }
        }
    }
}

resource "panos_panorama_device_group" "dg" {
    name = "myDeviceGroup"
}
```

Argument Reference

The following arguments are supported:

- `vsys` - (Optional) The vsys to put the NAT rule group into (default: `vsys1`).
- `device_group` - (Optional) Device group the NAT rules should be put into (default: `shared`).
- `rulebase` - (Optional) The rulebase the NAT rules should be put into. Valid values are `pre-rulebase` (default), `rulebase`, or `post-rulebase`.
- `position_keyword` - (Optional) A positioning keyword for this group. This can be `before`, `directly before`, `after`, `directly after`, `top`, `bottom`, or left empty (the default) to have no particular placement. This param works in combination with the `position_reference` param.
- `position_reference` - (Optional) Required if `position_keyword` is one of the "above" or "below" variants, this is the name of a non-group rule to use as a reference to place this group.
- `rule` - (Repeatable) The rule definition (see below). The rule ordering will match how they appear in the terraform plan file.

Each `rule` defined supports the following arguments:

- `name` - (Required) The NAT rule's name.
- `description` - (Optional) The description.
- `type` - (Optional). NAT type. This can be `ipv4` (default), `nat64`, or `nptv6`.
- `tags` - (Optional) List of administrative tags.
- `disabled` - (Optional) Set to `true` to disable this rule.
- `target` - (Optional, repeatable) A target definition (see below). If there are no target sections, then the rule will apply to every vsys of every device in the device group.
- `negate_target` - (Optional, bool) Instead of applying the rule for the given serial numbers, apply it to everything except them.
- `original_packet` - (Required) The original packet specification (see below).
- `translated_packet` - (Required) The translated packet spec (see below).

`target` supports the following arguments:

- `serial` - (Required) The serial number of the firewall.
- `vsys_list` - (Optional) A subset of all available vsys on the firewall that should be in this device group. If the firewall is a virtual firewall, then this parameter should just be omitted.

`original_packet` supports the following arguments:

- `source_zones` - (Required) The list of source zone(s).
- `destination_zone` - (Required) The destination zone.
- `destination_interface` - (Optional) Egress interface from route lookup (default: `any`).
- `service` - (Optional) Service (default: `any`).
- `source_addresses` - (Required) List of source address(es).

- `destination_addresses` - (Required) List of destination address(es).

`translated_packet` supports the following arguments:

- `source` - (Required) The source spec (see below). Leave this empty for a destination NAT of "none".
- `destination` - (Required) The destination spec (see below). Leave this empty for a destination NAT of "none".

`translated_packet.source` supports the following arguments:

- `dynamic_ip_and_port` - (Optional) Dynamic IP and port source translation spec (see below).
- `dynamic_ip` - (Optional) Dynamic IP source translation spec (see below).
- `static_ip` - (Optional) Static IP source translation spec (see below).

`translated_packet.source.dynamic_ip_and_port` supports the following arguments:

- `translated_address` - (Optional) Translated address source translation type spec (see below).
- `interface_address` - (Optional) Interface address source translation type spec (see below).

`translated_packet.source.dynamic_ip_and_port.translated_address` supports the following arguments:

- `translated_addresses` - (Required) List of translated addresses.

`translated_packet.source.dynamic_ip_and_port.interface_address` supports the following arguments:

- `interface` - (Required) The interface.
- `ip_address` - (Optional) The IP address.

`translated_packet.source.dynamic_ip` supports the following arguments:

- `translated_addresses` - (Optional) The list of translated addresses.
- `fallback` - (Optional) The fallback spec (see below). Leaving this empty or omiting it means a fallback of "None".

`translated_packet.source.dynamic_ip.fallback` supports the following arguments:

- `translated_address` - (Optional) The translated address fallback spec (see below).
- `interface_address` - (Optional) The interface address fallback spec (see below).

`translated_packet.source.dynamic_ip.fallback.translated_address` supports the following arguments:

- `translated_addresses` - (Optional) List of source address translation fallback translated addresses.

`translated_packet.source.dynamic_ip.fallback.interface_address` supports the following arguments:

- `interface` - (Required) Source address translation fallback interface.
- `type` - (Optional) Type of interface fallback. Valid values are `ip` (default) or `floating`.
- `ip_address` - (Optional) IP address of the fallback interface.

`translated_packet.source.static_ip` supports the following arguments:

- `translated_address` - (Required) The statically translated source address.
- `bi_directional` - (Optional, bool) Set to `true` to enable bi-directional source address translation.

`translated_packet.destination` supports the following arguments:

- `static_translation` - (Optional) Specifies a static destination NAT (see below).
- `dynamic_translation` - (Optional, PAN-OS 8.1+) Specify a dynamic destination NAT (see below).
- `static` - (**DEPRECATED**, Optional) Specifies a static destination NAT (see below). This was deprecated in provider version 1.6 in favor of `static_translation` instead.
- `dynamic` - (**DEPRECATED**, Optional, PAN-OS 8.1+) Specify a dynamic destination NAT (see below). If you are using Terraform 0.12+, you cannot use this param as it conflicts with the new dynamic (https://www.terraform.io/docs/configuration/expressions.html#dynamic-blocks) block.

`translated_packet.destination.static` and `translated_packet.destination.static_translation` support the following arguments:

- `address` - (Required) Destination address translation address.
- `port` - (Optional, int) Destination address translation port number.

`translated_packet.destination.dynamic` and `translated_packet.destination.dynamic_translation` support the following arguments:

- `address` - (Required) Destination address translation address.
- `port` - (Optional, int) Destination address translation port number.
- `distribution` - (Optional, PAN-OS 8.1+) Distribution algorithm for destination address pool. The PAN-OS 8.1 GUI doesn't seem to set this anywhere, but this is added here for completeness' sake. The GUI sets this to `round-robin` currently.

panos_panorama_nat_rule

This resource allows you to add/update/delete Panorama NAT rules.

Note: This resource has been deprecated. Please use `panos_panorama_nat_rule_group` instead.

Note: `panos_panorama_nat_policy` is known as `panos_panorama_nat_rule`.

The prefix `sat` stands for "Source Address Translation" while the prefix "dat" stands for "Destination Address Translation". The order of the params in this resource and their naming matches how the params are presented in the GUI. Thus, having a GUI window open while creating your resource definition will simplify the process.

Note that while many of the params for this resource are optional in an absolute sense, depending on what type of NAT you wish to configure, certain params may become necessary to correctly configure the NAT rule.

Example Usage

```
resource "panos_panorama_nat_rule" "example" {
    name = "my nat rule"
    source_zones = ["zone1"]
    destination_zone = "zone2"
    to_interface = "ethernet1/3"
    source_addresses = ["any"]
    destination_addresses = ["any"]
    sat_type = "none"
    dat_type = "static"
    dat_address = "my dat address object"
    target {
        serial = "123456"
        vsys_list = ["vsys1", "vsys2"]
    }
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The NAT rule's name.
- `device_group` - (Optional) The device group to put the NAT rule into (default: `shared`).
- `rulebase` - (Optional) The rulebase. This can be `pre-rulebase` (default), `post-rulebase`, or `rulebase`.
- `description` - (Optional) The description.
- `type` - (Optional). NAT type. This can be `ipv4` (default), `nat64`, or `nptv6`.
- `source_zones` - (Required) The list of source zone(s).

- `destination_zone` - (Required) The destination zone.
- `to_interface` - (Optional) Egress interface from route lookup (default: `any`).
- `service` - (Optional) Service (default: `any`).
- `source_addresses` - (Required) List of source address(es).
- `destination_addresses` - (Required) List of destination address(es).
- `sat_type` - (Optional) Type of source address translation. This can be `none` (default), `dynamic-ip-and-port`, `dynamic-ip`, or `static-ip`.
- `sat_address_type` - (Optional) Source address translation address type. This can be `interface-address` or `translated-address`.
- `sat_translated_addresses` - (Optional) Source address translation list of translated addresses.
- `sat_interface` - (Optional) Source address translation interface.
- `sat_ip_address` - (Optional) Source address translation IP address.
- `sat_fallback_type` - (Optional) Source address translation fallback type. This can be `none`, `interface-address`, or `translated-address`.
- `sat_fallback_translated_addresses` - (Optional) Source address translation list of fallback translated addresses.
- `sat_fallback_interface` - (Optional) Source address translation fallback interface.
- `sat_fallback_ip_type` - (Optional) Source address translation fallback IP type. This can be `ip` or `floating`.
- `sat_fallback_ip_address` - (Optional) The source address translation fallback IP address.
- `sat_static_translated_address` - (Optional) The statically translated source address.
- `sat_static_bi_directional` - (Optional) Set to `true` to enable bi-directional source address translation.
- `dat_type` - (Optional) Destination address translation type. This should be either `static` or `dynamic`. The `dynamic` option is only available on PAN-OS 8.1+.
- `dat_address` - (Optional) Destination address translation's address. Requires `dat_type` be set to "static" or "dynamic".
- `dat_port` - (Optional) Destination address translation's port number. Requires `dat_type` be set to "static" or "dynamic".
- `dat_dynamic_distribution` - (Optional, PAN-OS 8.1+) Distribution algorithm for destination address pool. The PAN-OS 8.1 GUI doesn't seem to set this anywhere, but this is added here for completeness' sake. Requires `dat_type` of "dynamic".
- `disabled` - (Optional) Set to `true` to disable this rule.
- `tags` - (Optional) List of administrative tags.
- `target` - (Optional) A target definition (see below). If there are no target sections, then the rule will apply to every vsys of every device in the device group.
- `negate_target` - (Optional, bool) Instead of applying the rule for the given serial numbers, apply it to everything except them.

The following arguments are valid for each `target` section:

- `serial` - (Required) The serial number of the firewall.
- `vsys_list` - (Optional) A subset of all available vsys on the firewall that should be in this device group. If the firewall is a virtual firewall, then this parameter should just be omitted.

panos_panorama_pbf_rule_group

This resource allows you to add/update/delete Panorama policy based forwarding rule groups.

This resource manages clusters of policy based forwarding rules in a single vsys, enforcing both the contents of individual rules as well as their ordering. Rules are defined in a `rule` config block.

Although you cannot modify non-group PBF rules with this resource, the `position_keyword` and `position_reference` parameters allow you to reference some other PBF rule that already exists, using it as a means to ensure some rough placement within the ruleset as a whole.

Best Practices

As is to be expected, if you are separating your deployment across multiple plan files, make sure that at most only one plan specifies any given absolute positioning keyword such as "top" or "directly below", otherwise they'll keep shoving each other out of the way indefinitely.

Example Usage

```
resource "panos_panorama_pbf_rule_group" "example" {
    device_group = panos_panorama_device_group.grp.name
    position_keyword = "above"
    position_reference = "deny everything else"
    rule {
        name = "my-pbf"
        description = "deployed by terraform"
        source {
            zones = ["zone1"]
            addresses = ["10.50.50.50"]
            users = ["any"]
            negate = true
        }
        destination {
            addresses = ["10.80.80.80"]
            applications = ["any"]
            services = ["application-default"]
        }
        forwarding {
            action = "discard"
        }
    }
}

resource "panos_panorama_device_group" "grp" {
    name = "myDeviceGroup"
    description = "deployed by Terraform"
}
```

Argument Reference

The following arguments are supported:

- `device_group` - (Optional) The device group to put the rules into (default: `shared`).
- `rulebase` - (Optional) The rulebase. This can be `pre-rulebase` (default), `post-rulebase`, or `rulebase`.
- `position_keyword` - (Optional) A positioning keyword for this group. This can be `before`, `directly before`, `after`, `directly after`, `top`, `bottom`, or left empty (the default) to have no particular placement. This param works in combination with the `position_reference` param.
- `position_reference` - (Optional) Required if `position_keyword` is one of the "above" or "below" variants, this is the name of a non-group rule to use as a reference to place this group.
- `rule` - The rule definition (see below). The rule ordering will match how they appear in the terraform plan file.

The following arguments are valid for each `rule` section:

- `name` - (Required) The rule name.
- `description` - (Optional) The rule description.
- `tags` - (Optional) List of tags for this rule.
- `active_active_device_binding` - (Optional) The active-active device binding.
- `schedule` - (Optional) The schedule.
- `disabled` - (Optional, bool) Set to `true` to disable this rule.
- `uuid` - (Optional, computed, PAN-OS 9.0+) The UUID for the rule.
- `source` - (Required) The source spec (defined below).
- `destination` - (Required) The destination spec (defined below).
- `forwarding` - (Required) The forwarding spec (defined below).
- `target` - (Optional) A target definition (see below). If there are no target sections, then the rule will apply to every vsys of every device in the device group.
- `negate_target` - (Optional, bool) Instead of applying the rule for the given serial numbers, apply it to everything except them.

`rule.source` supports the following arguments:

- `zones` - (Optional) If you want a source type of "zone", then define this list with the desired source zones. Mutually exclusive with `rule.interfaces`.
- `interfaces` - (Optional) If you want a source type of "interface", then define this list with the desired source interfaces. Mutually exclusive with `rule.zones`.
- `addresses` - (Required) List of source IP addresses.
- `users` - (Required) List of source users.
- `negate` - (Optional, bool) Set to `true` to negate the source.

`rule.destination` supports the following arguments:

- `addresses` - (Required) The list of destination addresses.
- `application` - (Required) The list of applications.
- `services` - (Required) The list of services.
- `negate` - (Optional, bool) Set to `true` to negate the destination.

`rule.forwarding` supports the following arguments:

- `action` - (Optional) The action to take. Valid values are `forward` (default), `forward-to-vsys`, `discard`, or `no-pbf`.
- `vsys` - (Optional) If `action=forward-to-vsys`, the vsys to forward to.
- `egress_interface` - (Optional) If `action=forward`, the egress interface.
- `next_hop_type` - (Optional) If `action=forward`, the next hop type. Valid values are `ip-address`, `fqdn`, or leaving this empty for a next hop type of None.
- `next_hop_value` - (Optional) If `action=forward` and `next_hop_type` is defined, then the next hop address.
- `monitor` - (Optional) The monitor spec (defined below). If you do not want to enable monitoring, then do not specify a `monitor` config block.
- `symmetric_return` - (Optional) The symmetric return spec (defined below). If you do not want to enforce symmetric

`rule.forwarding.monitor` supports the following arguments:

- `profile` - (Optional) The montior profile to use.
- `ip_address` - (Optional) The monitor IP address.
- `disable_if_unreachable` - (Optional, bool) Set to `true` to disable this rule if nexthop/monitor IP is unreachable.

`rule.forwarding.symmetric_return` supports the following arguments:

- `enable` - (Optional, bool) Set to `true` to enforce symmetric return.
- `addresses` - (Optional) List of next hop addresses.

`rule.target` supports the following arguments:

- `serial` - (Required) The serial number of the firewall.
- `vsys_list` - (Optional) A subset of all available vsys on the firewall that should be in this device group. If the firewall is a virtual firewall, then this parameter should just be omitted.

panos_panorama_redistribution_profile_ipv4

This resource allows you to add/update/delete Panorama IPv4 redistribution profiles on a virtual router.

Import Name

```
<template>:<template_stack>:<virtual_router>:<name>
```

Example Usage

```
resource "panos_panorama_redistribution_profile_ipv4" "example" {
    name = "example"
    template = "${panos_panorama_template.t.name}"
    virtual_router = "${panos_panorama_virtual_router.vr.name}"
    priority = 1
    action = "redist"
    types = ["static"]
    interfaces = ["${panos_panorama_virtual_router.vr.interfaces}"]
}

resource "panos_panorama_virtual_router" "vr" {
    name = "my virtual router"
    template = "${panos_panorama_template.t.name}"
    interfaces = ["ethernet1/2"]
}

resource "panos_panorama_template" "t" {
    name = "myTemplate"
    description = "my template"
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The redistribution profile's name.
- `template` - (Required) The template name.
- `virtual_router` - (Required) The virtual router to add the redistribution profile to.
- `priority` - (Required, int) The priority, integer from 1 to 255.
- `action` - (Optional) The action. Valid values are `redist` (default) or `no-redist`.
- `types` - (Optional) The source types. Valid values are `bgp`, `connect`, `ospf`, `rip`, and `static`.
- `interfaces` - (Optional) Specify candidate routes.

- `destinations` - (Optional) Specify candidate routes' next-hop addresses (subnet match).
- `next_hops` - (Optional) Specify candidate routes' next-hop addresses (subnet match).
- `ospf_path_types` - (Optional) OSPF path types. Valid values are `intra-area`, `inter-area`, `ext-1`, and `ext-2`.
- `ospf_areas` - (Optional) OSPF areas.
- `ospf_tags` - (Optional) OSPF tags.
- `bgp_communities` - (Optional) BGP communities.
- `bgp_extended_communities` - (Optional) BGP extended communities.

panos_panorama_security_policy

This resource allows you to manage the full security posture.

> **Note:** `panos_panorama_security_policies` is known as `panos_panorama_security_policy`.

This resource manages the full set of security rules, enforcing both the contents of individual rules as well as their ordering. Rules are defined in a `rule` config block. As this manages the full set of security rules for a given rulebase, any extraneous rules are removed on `terraform apply`.

> **Note**: This resource will remove any security rule not defined in your plan file.

For each security rule, there are three styles of profile settings:

- `None` (the default)
- `Group`
- `Profiles`

The Profile Setting is implicitly chosen based on what params are configured for the security rule. If you want a Profile Setting of `Group`, then the `group` param should be set to the desired Group Profile. If you want a Profile Setting of `Profiles`, then you will need to specify one or more of the following params:

- `virus`
- `spyware`
- `vulnerability`
- `url_filtering`
- `file_blocking`
- `wildfire_analysis`
- `data_filtering`

If the `group` param and none of the `Profiles` params are specified, then the Profile Setting is set to `None`.

Import Name

```
<device_group>:<rulebase>
```

Example Usage

```
resource "panos_panorama_security_policy" "example" {
    rule {
        name = "allow bizdev to dmz"
        source_zones = ["bizdev"]
        source_addresses = ["any"]
        source_users = ["any"]
        hip_profiles = ["any"]
        destination_zones = ["dmz"]
        destination_addresses = ["any"]
        applications = ["any"]
        services = ["application-default"]
        categories = ["any"]
        action = "allow"
    }
    rule {
        name = "deny sales to eng"
        source_zones = ["sales"]
        source_addresses = ["any"]
        source_users = ["any"]
        hip_profiles = ["any"]
        destination_zones = ["eng"]
        destination_addresses = ["any"]
        applications = ["any"]
        services = ["application-default"]
        categories = ["any"]
        action = "deny"
        target {
            serial = "01234"
        }
        target {
            serial = "56789"
            vsys_list = ["vsys1", "vsys3"]
        }
    }
}
```

Argument Reference

The following arguments are supported:

- `device_group` - (Optional) The device group to put the security policy into (default: `shared`).
- `rulebase` - (Optional) The rulebase. This can be `pre-rulebase` (default), `post-rulebase`, or `rulebase`.
- `rule` - The security rule definition (see below). The security rule ordering will match how they appear in the terraform plan file.

The following arguments are valid for each `rule` section:

- `name` - (Required) The security rule name.
- `type` - (Optional) Rule type. This can be `universal` (default), `interzone`, or `intrazone`.
- `description` - (Optional) The description.

- `tags` - (Optional) List of tags for this security rule.
- `source_zones` - (Required) List of source zones.
- `source_addresses` - (Required) List of source addresses.
- `negate_source` - (Optional, bool) If the source should be negated.
- `source_users` - (Required) List of source users.
- `hip_profiles` - (Required) List of HIP profiles.
- `destination_zones` - (Required) List of destination zones.
- `destination_addresses` - (Required) List of destination addresses.
- `negate_destination` - (Optional, bool) If the destination should be negated.
- `applications` - (Required) List of applications.
- `services` - (Required) List of services.
- `categories` - (Required) List of categories.
- `action` - (Optional) Action for the matched traffic. This can be `allow` (default), `deny`, `drop`, `reset-client`, `reset-server`, or `reset-both`.
- `log_setting` - (Optional) Log forwarding profile.
- `log_start` - (Optional, bool) Log the start of the traffic flow.
- `log_end` - (Optional, bool) Log the end of the traffic flow (default: `true`).
- `disabled` - (Optional, bool) Set to `true` to disable this rule.
- `schedule` - (Optional) The security rule schedule.
- `icmp_unreachable` - (Optional) Set to `true` to enable ICMP unreachable.
- `disable_server_response_inspection` - (Optional) Set to `true` to disable server response inspection.
- `group` - (Optional) Profile Setting: `Group` - The group profile name.
- `virus` - (Optional) Profile Setting: `Profiles` - The antivirus setting.
- `spyware` - (Optional) Profile Setting: `Profiles` - The anti-spyware setting.
- `vulnerability` - (Optional) Profile Setting: `Profiles` - The Vulnerability Protection setting.
- `url_filtering` - (Optional) Profile Setting: `Profiles` - The URL filtering setting.
- `file_blocking` - (Optional) Profile Setting: `Profiles` - The file blocking setting.
- `wildfire_analysis` - (Optional) Profile Setting: `Profiles` - The WildFire Analysis setting.
- `data_filtering` - (Optional) Profile Setting: `Profiles` - The Data Filtering setting.
- `target` - (Optional) A target definition (see below). If there are no target sections, then the rule will apply to every vsys of every device in the device group.
- `negate_target` - (Optional, bool) Instead of applying the rule for the given serial numbers, apply it to everything

except them.

The following arguments are valid for each `target` section:

- `serial` - (Required) The serial number of the firewall.
- `vsys_list` - (Optional) A subset of all available vsys on the firewall that should be in this device group. If the firewall is a virtual firewall, then this parameter should just be omitted.

panos_panorama_security_rule_group

This resource allows you to add/update/delete Panorama security rule groups.

> **Note:** `panos_panorama_security_policy_group` is known as `panos_panorama_security_rule_group`.

This resource manages clusters of security rules in a single device group, enforcing both the contents of individual rules as well as their ordering. Rules are defined in a `rule` config block.

Because this resource only manages what it's told to, it will not manage any rules that may already exist on Panorama. This has implications on the effective security posture of Panorama, but it will allow you to spread your security rules across multiple Terraform state files. If you want to verify that the security rules are only what appears in the plan file, then you should probably be using the panos_panorama_security_policy (/docs/providers/panos/r/panorama_security_policy.html) resource.

Although you cannot modify non-group security rules with this resource, the `position_keyword` and `position_reference` parameters allow you to reference some other security rule that already exists, using it as a means to ensure some rough placement within the ruleset as a whole.

For each security rule, there are three styles of profile settings:

- `None` (the default)
- `Group`
- `Profiles`

The Profile Setting is implicitly chosen based on what params are configured for the security rule. If you want a Profile Setting of `Group`, then the `group` param should be set to the desired Group Profile. If you want a Profile Setting of `Profiles`, then you will need to specify one or more of the following params:

- `virus`
- `spyware`
- `vulnerability`
- `url_filtering`
- `file_blocking`
- `wildfire_analysis`
- `data_filtering`

If the `group` param and none of the `Profiles` params are specified, then the Profile Setting is set to `None`.

Best Practices

As is to be expected, if you are separating your deployment across multiple plan files, make sure that at most only one plan specifies any given absolute positioning keyword such as "top" or "directly below", otherwise they'll keep shoving each other out of the way indefinitely.

Best practices are to specify one group as `top` (if you need it), one group as `bottom` (this is where you have your logging deny rule), then all other groups should be `above` the first rule of the bottom group. You do it this way because rules will natually be added at the tail end of the rulebase, so they will always be `after` the first group, but what you want is for them to be `before` the last group's rules.

Example Usage

```
resource "panos_panorama_security_rule_group" "example" {
    position_keyword = "above"
    position_reference = "deny everything else"
    rule {
        name = "allow bizdev to dmz"
        source_zones = ["bizdev"]
        source_addresses = ["any"]
        source_users = ["any"]
        hip_profiles = ["any"]
        destination_zones = ["dmz"]
        destination_addresses = ["any"]
        applications = ["any"]
        services = ["application-default"]
        categories = ["any"]
        action = "allow"
    }
    rule {
        name = "deny sales to eng"
        source_zones = ["sales"]
        source_addresses = ["any"]
        source_users = ["any"]
        hip_profiles = ["any"]
        destination_zones = ["eng"]
        destination_addresses = ["any"]
        applications = ["any"]
        services = ["application-default"]
        categories = ["any"]
        action = "deny"
        target {
            serial = "01234"
        }
        target {
            serial = "56789"
            vsys_list = ["vsys1", "vsys3"]
        }
    }
}
```

Argument Reference

The following arguments are supported:

- `device_group` - (Optional) The device group to put the security rules into (default: `shared`).
- `rulebase` - (Optional) The rulebase. This can be `pre-rulebase` (default), `post-rulebase`, or `rulebase`.

- `position_keyword` - (Optional) A positioning keyword for this group. This can be `before`, `directly before`, `after`, `directly after`, `top`, `bottom`, or left empty (the default) to have no particular placement. This param works in combination with the `position_reference` param.
- `position_reference` - (Optional) Required if `position_keyword` is one of the "above" or "below" variants, this is the name of a non-group rule to use as a reference to place this group.
- `rule` - The security rule definition (see below). The security rule ordering will match how they appear in the terraform plan file.

The following arguments are valid for each `rule` section:

- `name` - (Required) The security rule name.
- `type` - (Optional) Rule type. This can be `universal` (default), `interzone`, or `intrazone`.
- `description` - (Optional) The description.
- `tags` - (Optional) List of tags for this security rule.
- `source_zones` - (Required) List of source zones.
- `source_addresses` - (Required) List of source addresses.
- `negate_source` - (Optional, bool) If the source should be negated.
- `source_users` - (Required) List of source users.
- `hip_profiles` - (Required) List of HIP profiles.
- `destination_zones` - (Required) List of destination zones.
- `destination_addresses` - (Required) List of destination addresses.
- `negate_destination` - (Optional, bool) If the destination should be negated.
- `applications` - (Required) List of applications.
- `services` - (Required) List of services.
- `categories` - (Required) List of categories.
- `action` - (Optional) Action for the matched traffic. This can be `allow` (default), `deny`, `drop`, `reset-client`, `reset-server`, or `reset-both`.
- `log_setting` - (Optional) Log forwarding profile.
- `log_start` - (Optional, bool) Log the start of the traffic flow.
- `log_end` - (Optional, bool) Log the end of the traffic flow (default: `true`).
- `disabled` - (Optional, bool) Set to `true` to disable this rule.
- `schedule` - (Optional) The security rule schedule.
- `icmp_unreachable` - (Optional) Set to `true` to enable ICMP unreachable.
- `disable_server_response_inspection` - (Optional) Set to `true` to disable server response inspection.
- `group` - (Optional) Profile Setting: `Group` - The group profile name.

- `virus` - (Optional) Profile Setting: `Profiles` - The antivirus setting.
- `spyware` - (Optional) Profile Setting: `Profiles` - The anti-spyware setting.
- `vulnerability` - (Optional) Profile Setting: `Profiles` - The Vulnerability Protection setting.
- `url_filtering` - (Optional) Profile Setting: `Profiles` - The URL filtering setting.
- `file_blocking` - (Optional) Profile Setting: `Profiles` - The file blocking setting.
- `wildfire_analysis` - (Optional) Profile Setting: `Profiles` - The WildFire Analysis setting.
- `data_filtering` - (Optional) Profile Setting: `Profiles` - The Data Filtering setting.
- `target` - (Optional) A target definition (see below). If there are no target sections, then the rule will apply to every vsys of every device in the device group.
- `negate_target` - (Optional, bool) Instead of applying the rule for the given serial numbers, apply it to everything except them.

The following arguments are valid for each `target` section:

- `serial` - (Required) The serial number of the firewall.
- `vsys_list` - (Optional) A subset of all available vsys on the firewall that should be in this device group. If the firewall is a virtual firewall, then this parameter should just be omitted.

panos_panorama_service_group

This resource allows you to add/update/delete Panorama service groups.

Import Name

```
<device_group>:<name>
```

Example Usage

```
resource "panos_panorama_service_group" "example" {
    name = "static ntp grp"
    services = ["svc1", "svc2"]
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The service group's name.
- `device_group` - (Optional) The device group to put the service group into (default: `shared`).
- `services` - (Required) List of services to put in this service group.
- `tags` - (Optional) List of administrative tags.

panos_panorama_service_object

This resource allows you to add/update/delete Panorama service objects.

Import Name

```
<device_group>:<name>
```

Example Usage

```
resource "panos_panorama_service_object" "example" {
    name = "my_service"
    protocol = "tcp"
    description = "My service object"
    source_port = "2000-2049,2051-2099"
    destination_port = "32123"
    tags = ["internal", "dmz"]
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The service object's name.
- `device_group` - (Optional) The device group to put the service object into (default: `shared`).
- `description` - (Optional) The service object's description.
- `protocol` - (Required) The service's protocol. This should be `tcp`, `udp`, or `sctp` (PAN-OS 8.1+).
- `source_port` - (Optional) The source port. This can be a single port number, range (1-65535), or comma separated (80,8080,443).
- `destination_port` - (Required) The destination port. This can be a single port number, range (1-65535), or comma separated (80,8080,443).
- `tags` - (Optional) List of administrative tags.
- `override_session_timeout` - (Optional, bool, PAN-OS 8.1+) Set to `true` to override the default application timeouts.
- `override_timeout` - (Optional, int, PAN-OS 8.1+) The overridden TCP timeout.
- `override_half_closed_timeout` - (Optional, int, PAN-OS 8.1+) The overridden TCP half closed timeout.
- `override_time_wait_timeout` - (Optional, int, PAN-OS 8.1+) The overridden TCP wait time.

panos_panorama_snmptrap_server_profile

This resource allows you to add/update/delete Panorama snmptrap server profiles.

Example Usage

```
resource "panos_panorama_snmptrap_server_profile" "example" {
    device_group = "shared"
    name = "myProfile"
    v2c_server {
        name = "first server"
        manager = "snmp1.example.com"
        community = "public"
    }
}
```

Argument Reference

When creating this profile, there are a few options:

- on the Panorama (in the `shared` device group)
- in a vsys in a template
- in a vsys in a template stack

The following arguments are supported:

- `template` - (Optional) The template location. Mutually exclusive with `template_stack` and `device_group`.
- `template_stack` - (Optional) The template stack location. Mutually exclusive with `template` and `device_group`.
- `device_group` - (Optional) The device group location. Mutually exclusive with `template` and `template_stack`.
- `vsys` - (Optional) The vsys. This will likely be `shared`, and it should be defined if you specified either `template` or `template_stack`.
- `name` - (Required) The group's name.
- `v2c_server` - (Optional, repeatable) A v2c server (defined below).
- `v3_server` - (Optional, repeatable) A v3 server (defined below).

`v2c_server` supports the following arguments:

- `name` - (Required) The server name.
- `manager` - (Required) The hostname.
- `community` - (Required) The SNMP community.

`v3_server` supports the following arguments:

- `name` - (Required) The server name.
- `manager` - (Required) The hostname.
- `user` - (Required) Username.
- `engine_id` - (Optional) The engine ID.
- `auth_password` - (Required) SNMP auth password.
- `priv_password` - (Required) SNMP priv password.

panos_panorama_static_route_ipv4

This resource allows you to add/update/delete Panorama IPv4 static routes on a virtual router for either a template or a template stack.

Import Name

```
<template>:<template_stack>:<virtual_router>:<name>
```

Example Usage

```
resource "panos_panorama_static_route_ipv4" "example" {
    name = "localnet"
    virtual_router = "${panos_panorama_virtual_router.vr1.name}"
    template = "template1"
    destination = "10.1.7.0/32"
    next_hop = "10.1.7.4"
}

resource "panos_panorama_virtual_router" "vr1" {
    name = "my virtual router"
    template = "template1"
}
```

Argument Reference

One and only one of the following must be specified:

- `template` - The template name.
- `template_stack` - The template stack name.

The following arguments are supported:

- `name` - (Required) The static route's name.
- `virtual_router` - (Required) The virtual router to add the static route to.
- `destination` - (Required) Destination IP address / prefix.
- `interface` - (Optional) Interface to use.
- `type` - (Optional) The next hop type. Valid values are `ip-address` (the default), `discard`, `next-vr`, or an empty string for `None`.
- `next_hop` - (Optional) The value for the `type` setting.
- `admin_distance` - (Optional) The admin distance.

- `metric` - (Optional, int) Metric value / path cost (default: `10`).
- `route_table` - (Optional) Target routing table to install the route. Valid values are `unicast` (the default), `no install`, `multicast`, or `both`.
- `bfd_profile` - (Optional, PAN-OS 7.1+) BFD configuration.

panos_panorama_syslog_server_profile

This resource allows you to add/update/delete Panorama syslog server profiles.

Import Name

```
<template>:<template_stack>:<vsys>:<device_group>:<name>
```

Example Usage

```
resource "panos_panorama_syslog_server_profile" "example" {
    device_group = "shared"
    name = "myProfile"
    threat_format = "$serial $severity"
    syslog_server {
        name = "my-server"
        server = "syslog.example.com"
    }
}
```

Argument Reference

When creating this profile, there are a few options:

- on the Panorama (in the `shared` device group)
- in a vsys in a template
- in a vsys in a template stack

The following arguments are supported:

- `template` - (Optional) The template location. Mutually exclusive with `template_stack` and `device_group`.
- `template_stack` - (Optional) The template stack location. Mutually exclusive with `template` and `device_group`.
- `device_group` - (Optional) The device group location. Mutually exclusive with `template` and `template_stack`.
- `vsys` - (Optional) The vsys. This will likely be `shared`, and it should be defined if you specified either `template` or `template_stack`.
- `name` - (Required) The group's name.
- `config_format` - (Optional) Config format.
- `system_format` - (Optional) System format.
- `threat_format` - (Optional) Threat format.

- `traffic_format` - (Optional) Traffic format.
- `hip_match_format` - (Optional) HIP match format.
- `url_format` - (Optional) URL format.
- `data_format` - (Optional) Data format.
- `wildfire_format` - (Optional) Wildfire format.
- `tunnel_format` - (Optional) Tunnel format.
- `user_id_format` - (Optional) UserID format.
- `gtp_format` - (Optional) GTP format.
- `auth_format` - (Optional) Auth format.
- `sctp_format` - (Optional) SCTP format.
- `iptag_format` - (Optional) IP tag format.
- `escaped_characters` - (Optional) The escaped characters (as a string).
- `escape_character` - (Optional) The escape character.
- `syslog_server` - (Required, repeatable) The server spec (defined below).

`syslog_server` supports the following arguments:

- `name` - (Required) Server name.
- `server` - (Required) The hostname.
- `transport` - (Optional) The transport. Valid values are `UDP` (default), `TCP`, or `SSL`.
- `port` - (Optional, int) The port number (default: 514).
- `syslog_format` - (Optional) The syslog format. Valid values are `BSD` (default) or `IETF`.
- `facility` - (Optional) The syslog facility. Valid values are `LOG_USER` (default), `LOG_LOCAL0`, `LOG_LOCAL1`, `LOG_LOCAL2`, `LOG_LOCAL3`, `LOG_LOCAL4`, `LOG_LOCAL5`, `LOG_LOCAL6`, or `LOG_LOCAL7`.

panos_panorama_template_entry

This resource allows you to add/update/delete a specific device in a Panorama template.

This resource has some overlap with the `panos_panorama_template` resource. If you want to use this resource with the other one, then make sure that your `panos_panorama_template` spec does not define any `device` blocks, and just stays as "computed".

This is the appropriate resource to use if you have a pre-existing template in Panorama and don't want Terraform to delete it on `terraform destroy`.

An interesting side effect of the underlying XML API - if the template does not already exist, then this resource can actually create it. However, since only the single entry for the specific serial number is deleted, then a `terraform destroy` would not remove the template itself in this situation.

Import Name

```
<template>:<serial>
```

Example Usage

```
# Example for a virtual firewall.
resource "panos_panorama_template_entry" "example1" {
    template = "my template"
    serial = "00112233"
}

# Example for a physical firewall with multi-vsys enabled.
resource "panos_panorama_template_entry" "example2" {
    template = "my template"
    serial = "44556677"
    vsys_list = ["vsys1", "vsys2"]
}
```

Argument Reference

The following arguments are supported:

- `template` - (Required) The template name.
- `serial` - (Required) The serial number of the firewall.
- `vsys_list` - (Optional) A subset of all available vsys on the firewall that should be in this template. If the firewall is a virtual firewall, then this parameter should just be omitted.

panos_panorama_template

This resource allows you to add/update/delete Panorama templates.

This resource has some overlap with the `panos_panorama_template_entry` resource. If you want to use this resource with the other one, then make sure that your `panos_panorama_template` spec does not define any `device` blocks, and just stays as "computed".

This is the appropriate resource to use if `terraform destroy` should delete the template.

Note - In PAN-OS 8.1, it looks like the `devices` field has been removed. Creating a template stack and specifying devices in the template stack is still present in PAN-OS 8.1.

Import Name

```
<name>
```

Example Usage

```
# This specifies one or more device blocks, so this is applicable only for
# PAN-OS 8.0 and lower.
resource "panos_panorama_template" "example" {
    name = "template1"
    description = "description here"
    device {
        serial = "00112233"
    }
    device {
        serial = "44556677"
        vsys_list = ["vsys1", "vsys2"]
    }
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The template's name.
- `description` - (Optional) The template's description.
- `device` - The device definition (see below).

The following arguments are valid for each `device` section:

- `serial` - (Required) The serial number of the firewall.

- `vsys_list` - (Optional) A subset of all available vsys on the firewall that should be in this template. If the firewall is a virtual firewall, then this parameter should just be omitted.

panos_panorama_template_stack_entry

This resource allows you to add/update/delete a specific device in a Panorama template stack.

This resource has some overlap with the `panos_panorama_template_stack` resource. If you want to use this resource with the other one, then make sure that your `panos_panorama_template_stack` spec does not define the `devices` field.

This is the appropriate resource to use if you have a pre-existing template stack in Panorama and don't want Terraform to delete it on `terraform destroy`.

Import Name

```
<template_stack>:<device>
```

Example Usage

```
resource "panos_panorama_template_stack_entry" "example1" {
    template_stack = "my template stack"
    device = "00112233"
}
```

Argument Reference

The following arguments are supported:

- `template_stack` - (Required) The template name.
- `device` - (Required) The serial number of the device to add.

panos_panorama_template_stack

This resource allows you to add/update/delete Panorama template stacks.

This resource has some overlap with the `panos_panorama_template_stack_entry` resource. If you want to use this resource with the other one, then make sure that your `panos_panorama_template_stack` spec does not define any `device` blocks, and just stays as "computed".

This is the appropriate resource to use if `terraform destroy` should delete the template stack.

Import Name

```
<name>
```

Example Usage

```
resource "panos_panorama_template_stack" "example" {
    name = "myStack"
    description = "description here"
    templates = ["t1", "t2"]
    devices = ["00112233", "44556677"]
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The stack's name.
- `description` - (Optional) The stack's description.
- `default_vsys` - (Optional) The default virtual system template configuration pushed to firewalls with a single virtual system. **Note** - you can only set this if there is at least one template in this stack.
- `templates` - (Optional) List of templates in this stack.
- `devices` - (Optional) List of serial numbers to include in this stack.

panos_panorama_template_variable

This resource allows you to add/update/delete variables for both Panorama templates and template stacks.

Template variables are available in PAN-OS 8.1+.

Import Name

```
<template>:<template_stack>:<name>
```

Example Usage

```
resource "panos_panorama_template_variable" "example" {
    template = "${panos_panorama_template.tmpl1.name}"
    name = "$example"
    type = "ip-address"
    value = "10.1.1.1/24"
}

resource "panos_panorama_template" "tmpl1" {
    name = "MyTemplate"
}
```

Argument Reference

One and only one of the following must be specified:

- `template` - The template name.
- `template_stack` - The template stack name.

The following arguments are supported:

- `name` - (Required) The template's name. This must start with a dollar sign ($).
- `type` - (Optional) The variable type. Valid values are `ip-netmask` (default), `ip-range`, `fqdn`, `group-id`, or `interface`.
- `value` - (Required) The variable value.

panos_panorama_tunnel_interface

This resource allows you to add/update/delete Panorama tunnel interfaces for templates.

Import Name

```
<template>:<template_stack>:<vsys>:<name>
```

Example Usage

```
resource "panos_panorama_tunnel_interface" "example1" {
    name = "tunnel.5"
    template = "foo"
    static_ips = ["10.1.1.1/24"]
    comment = "Configured for internal traffic"
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The interface's name. This must start with `tunnel.`.
- `template` - (Required) The template name.
- `vsys` - (Optional) The vsys that will use this interface (default: `vsys1`).
- `comment` - (Optional) The interface comment.
- `netflow_profile` - (Optional) The netflow profile.
- `static_ips` - (Optional) List of static IPv4 addresses to set for this data interface.
- `management_profile` - (Optional) The management profile.
- `mtu` - (Optional) The MTU.

panos_panorama_virtual_router_entry

This resource allows you to add/update/delete an interface in a Panorama virtual router template.

This resource has some overlap with the `panos_panorama_virtual_router` resource. If you want to use this resource with the other one, then make sure that your `panos_panorama_virtual_router` spec does not define the `interfaces` field.

Import Name

```
<template>:<template_stack>:<virtual_router>:<interface>
```

Example Usage

```
resource "panos_panorama_virtual_router" "vr" {
    template = "my template"
    name = "my vr"
}

resource "panos_panorama_virtual_router_entry" "example" {
    template = "my template"
    virtual_router = "${panos_panorama_virtual_router.vr.name}"
    interface = "ethernet1/5"
}
```

Argument Reference

The following arguments are supported:

- `template` - (Required) The template name.
- `virtual_router` - (Required) The virtual router's name.
- `interface` - (Required) The interface to import into the virtual router.

panos_panorama_virtual_router

This resource allows you to add/update/delete Panorama virtual routers for templates.

Note - The `default` virtual router may be configured with this resource, however it will not be deleted from Panorama. It will only be unexported from the vsys that it is currently imported in, and any interfaces imported into the virtual router will be removed.

This resource has some overlap with the `panos_panorama_virtual_router_entry` resource. If you want to use this resource with the other one, then make sure that your `panos_panorama_virtual_router` spec does not define the `interfaces` field.

Import Name

```
<template>:<template_stack>:<vsys>:<name>
```

Example Usage

```
# Configure a bare-bones ethernet interface.
resource "panos_panorama_virtual_router" "example" {
    name = "my virtual router"
    template = "foo"
    static_dist = 15
    interfaces = ["ethernet1/1", "ethernet1/2"]
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The virtual router's name.
- `template` - (Required) The template name.
- `vsys` - (Required) The vsys that will use this virtual router. This should be something like `vsys1` or `vsys3`.
- `interfaces` - (Optional) List of interfaces that should use this virtual router.
- `static_dist` - (Optional) Admin distance - Static (default: `10`).
- `static_ipv6_dist` - (Optional) Admin distance - Static IPv6 (default: `10`).
- `ospf_int_dist` - (Optional) Admin distance - OSPF Int (default: `30`).
- `ospf_ext_dist` - (Optional) Admin distance - OSPF Ext (default: `110`).
- `ospfv3_int_dist` - (Optional) Admin distance - OSPFv3 Int (default: `30`).

- `ospfv3_ext_dist` - (Optional) Admin distance - OSPFv3 Ext (default: `110`).
- `ibgp_dist` - (Optional) Admin distance - IBGP (default: `200`).
- `ebgp_dist` - (Optional) Admin distance - EBGP (default: `20`).
- `rip_dist` - (Optional) Admin distance - RIP (default: `120`).

panos_panorama_vlan_entry

This resource allows you to add/update/delete an interface in a VLAN on Panorama.

Import Name

```
<template>::<vlan>:<interface>
```

Example Usage

```
resource "panos_panorama_vlan_entry" "example" {
    template = panos_panorama_template.t.name
    vlan = panos_vlan.vlan1.name
    interface = panos_ethernet_interface.e1.name
    mac_addresses = [
        "00:30:48:55:66:77",
        "00:30:48:55:66:88",
    ]
}

resource "panos_panorama_template" "t" {
    name = "my template"
}

resource "panos_panorama_vlan" "vlan1" {
    template = panos_panorama_template.t.name
    name = "myVlan"
}

resource "panos_panorama_ethernet_interface" "e1" {
    template = panos_panorama_template.t.name
    name = "ethernet1/5"
    mode = "layer2"
    vsys = "vsys1"
}
```

Argument Reference

The following arguments are supported:

- `vlan` - (Required) The VLAN's name.
- `template` - (Required) The template name.
- `interface` - (Required) The interface's name.
- `mac_addresses` - (Optional) List of MAC addresses that should go with this entry.

panos_panorama_vlan

This resource allows you to add/update/delete Panorama VLANs.

Import Name

```
<template>::<vsys>:<name>
```

Example Usage

```
resource "panos_panorama_vlan" "example" {
    template = panos_panorama_template.t.name
    name = "myVlan"
    vlan_interface = panos_panorama_vlan_interface.vli.name
}

resource "panos_panorama_vlan_interface" "vli" {
    template = panos_panorama_template.t.name
    name = "vlan.6"
    vsys = "vsys1"
}

resource "panos_panorama_template" "t" {
    name = "myTemplate"
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The object's name.
- `vsys` - (Optional) The vsys to put the object into (default: `vsys1`).
- `template` - (Required) The template name.
- `vlan_interface` - (Optional) The VLAN interface.
- `interfaces` - (Optional, computed) List of layer2 interfaces. You can also leave this blank and also use panos_vlan_entry (/docs/providers/panos/r/vlan_entry.html) for more control.

panos_panorama_vlan_interface

This resource allows you to add/update/delete Panorama VLAN interfaces for templates.

Import Name

```
<template>:<template_stack>:<vsys>:<name>
```

Example Usage

```
resource "panos_panorama_vlan_interface" "example" {
    name = "vlan.17"
    template = "foo"
    mode = "layer3"
    static_ips = ["10.1.1.1/24"]
    comment = "Configured for internal traffic"
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The interface's name. Must start with `vlan.`.
- `template` - (Required) The template name.
- `vsys` - (Optional) The vsys that will use this interface (default: `vsys1`).
- `comment` - (Optional) The interface comment.
- `netflow_profile` - (Optional) The netflow profile.
- `static_ips` - (Optional) List of static IPv4 addresses to set for this data interface.
- `enable_dhcp` - (Optional) Set to `true` to enable DHCP on this interface.
- `create_dhcp_default_route` - (Optional) Set to `true` to create a DHCP default route.
- `dhcp_default_route_metric` - (Optional) The metric for the DHCP default route.
- `management_profile` - (Optional) The management profile.
- `mtu` - (Optional) The MTU.
- `adjust_tcp_mss` - (Optional) Adjust TCP MSS (default: false).
- `ipv4_mss_adjust` - (Optional, PAN-OS 8.0+) The IPv4 MSS adjust value.

- `ipv6_mss_adjust` - (Optional, PAN-OS 8.0+) The IPv6 MSS adjust value.

panos_panorama_zone_entry

This resource allows you to add/update/delete a specific interface in a Panorama zone.

This resource has some overlap with the `panos_panorama_zone` resource. If you want to use this resource with the other one, then make sure that your `panos_panorama_zone` spec does not define the `interfaces` field.

This is the appropriate resource to use if you have a pre-existing zone in Panorama and don't want Terraform to delete it on `terraform destroy`.

Import Name

```
<template>:<template_stack>:<vsys>:<zone>:<mode>:<interface>
```

Example Usage

```
resource "panos_panorama_template" "t" {
    name = "myTemplate"
}

resource "panos_panorama_ethernet_interface" "e5" {
    template = "${panos_panorama_template.t.name}"
    name = "ethernet1/5"
    mode = "layer3"
}

resource "panos_panorama_zone" "z" {
    template = "${panos_panorama_template.t.name}"
    name = "exZone"
    mode = "layer3"
}

resource "panos_panorama_zone_entry" "example" {
    template = "${panos_panorama_template.t.name}"
    zone = "${panos_panorama_zone.z.name}"
    mode = "${panos_panorama_zone.z.mode}"
    interface = "${panos_panorama_ethernet_interface.e5.name}"
}
```

Argument Reference

The following arguments are supported:

- `template` - (Required) The template name.
- `vsys` - (Optional) The vsys (default: `vsys1`).

- `zone` - (Required) The zone's name.
- `mode` - (Optional) The mode. Can be `layer3` (default), `layer2`, `virtual-wire`, `tap`, or `external`.
- `interface` - (Required) The interface's name.

panos_panorama_zone

This resource allows you to add/update/delete zones on Panorama for both templates and template stacks.

This resource has some overlap with the `panos_panorama_zone_entry` resource. If you want to use this resource with the other one, then make sure that your `panos_panorama_zone` spec does not define the `interfaces` field.

Import Name

```
<template>:<template_stack>:<vsys>:<name>
```

Example Usage

```
resource "panos_panorama_zone" "example" {
    name = "myZone"
    template = "${panos_panorama_template.tmpl1.name}"
    mode = "layer3"
    interfaces = ["${panos_panorama_ethernet_interface.e2.name", "${panos_panorama_ethernet_interface.e3.
name}"]
    enable_user_id = true
    exclude_acls = ["192.168.0.0/16"]
}

resource "panos_panorama_template" "tmpl1" {
    name = "MyTemplate"
}

resource "panos_panorama_ethernet_interface" "e2" {
    template = "${panos_panorama_template.tmpl1.name}"
    name = "ethernet1/2"
    mode = "layer3"
}

resource "panos_panorama_ethernet_interface" "e3" {
    template = "${panos_panorama_template.tmpl1.name}"
    name = "ethernet1/3"
    mode = "layer3"
}
```

Argument Reference

One and only one of the following must be specified:

- `template` - The template name.
- `template_stack` - The template stack name.

The following arguments are supported:

- `name` - (Required) The zone's name.
- `vsys` - (Optional) The vsys to put the zone into (default: `vsys1`).
- `mode` - (Required) The zone's mode. This can be `layer3`, `layer2`, `virtual-wire`, `tap`, or `tunnel`.
- `zone_profile` - (Optional) The zone protection profile.
- `log_setting` - (Optional) Log setting.
- `enable_user_id` - (Optional) Boolean to enable user identification.
- `interfaces` - (Optional) List of interfaces to associated with this zone.
- `include_acls` - (Optional) Users from these addresses/subnets will be identified. This can be an address object, an address group, a single IP address, or an IP address subnet.
- `exclude_acls` - (Optional) Users from these addresses/subnets will not be identified. This can be an address object, an address group, a single IP address, or an IP address subnet.

panos_pbf_rule_group

This resource allows you to add/update/delete policy based forwarding rule groups.

This resource manages clusters of policy based forwarding rules in a single vsys, enforcing both the contents of individual rules as well as their ordering. Rules are defined in a `rule` config block.

Although you cannot modify non-group PBF rules with this resource, the `position_keyword` and `position_reference` parameters allow you to reference some other PBF rule that already exists, using it as a means to ensure some rough placement within the ruleset as a whole.

Best Practices

As is to be expected, if you are separating your deployment across multiple plan files, make sure that at most only one plan specifies any given absolute positioning keyword such as "top" or "directly below", otherwise they'll keep shoving each other out of the way indefinitely.

Example Usage

```
resource "panos_pbf_rule_group" "example" {
    position_keyword = "above"
    position_reference = "deny everything else"
    rule {
        name = "my-pbf"
        description = "deployed by terraform"
        source {
            zones = [panos_zone.foo.name]
            addresses = ["10.50.50.50"]
            users = ["any"]
            negate = true
        }
        destination {
            addresses = ["10.80.80.80"]
            applications = ["any"]
            services = ["application-default"]
        }
        forwarding {
            action = "discard"
        }
    }
}

resource "panos_zone" "foo" {
    name = "myZone"
    mode = "layer2"
}
```

Argument Reference

The following arguments are supported:

- `vsys` - (Optional) The vsys to put the rule into (default: `vsys1`).
- `position_keyword` - (Optional) A positioning keyword for this group. This can be `before`, `directly before`, `after`, `directly after`, `top`, `bottom`, or left empty (the default) to have no particular placement. This param works in combination with the `position_reference` param.
- `position_reference` - (Optional) Required if `position_keyword` is one of the "above" or "below" variants, this is the name of a non-group rule to use as a reference to place this group.
- `rule` - The rule definition (see below). The rule ordering will match how they appear in the terraform plan file.

The following arguments are valid for each `rule` section:

- `name` - (Required) The rule name.
- `description` - (Optional) The rule description.
- `tags` - (Optional) List of tags for this rule.
- `active_active_device_binding` - (Optional) The active-active device binding.
- `schedule` - (Optional) The schedule.
- `disabled` - (Optional, bool) Set to `true` to disable this rule.
- `uuid` - (Optional, computed, PAN-OS 9.0+) The UUID for the rule.
- `source` - (Required) The source spec (defined below).
- `destination` - (Required) The destination spec (defined below).
- `forwarding` - (Required) The forwarding spec (defined below).

`rule.source` supports the following arguments:

- `zones` - (Optional) If you want a source type of "zone", then define this list with the desired source zones. Mutually exclusive with `rule.interfaces`.
- `interfaces` - (Optional) If you want a source type of "interface", then define this list with the desired source interfaces. Mutually exclusive with `rule.zones`.
- `addresses` - (Required) List of source IP addresses.
- `users` - (Required) List of source users.
- `negate` - (Optional, bool) Set to `true` to negate the source.

`rule.destination` supports the following arguments:

- `addresses` - (Required) The list of destination addresses.
- `application` - (Required) The list of applications.
- `services` - (Required) The list of services.
- `negate` - (Optional, bool) Set to `true` to negate the destination.

`rule.forwarding` supports the following arguments:

- `action` - (Optional) The action to take. Valid values are `forward` (default), `forward-to-vsys`, `discard`, or `no-pbf`.
- `vsys` - (Optional) If `action=forward-to-vsys`, the vsys to forward to.
- `egress_interface` - (Optional) If `action=forward`, the egress interface.
- `next_hop_type` - (Optional) If `action=forward`, the next hop type. Valid values are `ip-address`, `fqdn`, or leaving this empty for a next hop type of None.
- `next_hop_value` - (Optional) If `action=forward` and `next_hop_type` is defined, then the next hop address.
- `monitor` - (Optional) The monitor spec (defined below). If you do not want to enable monitoring, then do not specify a `monitor` config block.
- `symmetric_return` - (Optional) The symmetric return spec (defined below). If you do not want to enforce symmetric

`rule.forwarding.monitor` supports the following arguments:

- `profile` - (Optional) The montior profile to use.
- `ip_address` - (Optional) The monitor IP address.
- `disable_if_unreachable` - (Optional, bool) Set to `true` to disable this rule if nexthop/monitor IP is unreachable.

`rule.forwarding.symmetric_return` supports the following arguments:

- `enable` - (Optional, bool) Set to `true` to enforce symmetric return.
- `addresses` - (Optional) List of next hop addresses.

panos_redistribution_profile_ipv4

This resource allows you to add/update/delete IPv4 redistribution profiles on a virtual router.

Import Name

```
<virtual_router>:<name>
```

Example Usage

```
resource "panos_redistribution_profile_ipv4" "example" {
    name = "example"
    virtual_router = "${panos_virtual_router.vr.name}"
    priority = 1
    action = "redist"
    types = ["static"]
    interfaces = ["${panos_virtual_router.vr.interfaces}"]
}

resource "panos_virtual_router" "vr" {
    name = "my virtual router"
    interfaces = ["ethernet1/2"]
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The redistribution profile's name.
- `virtual_router` - (Required) The virtual router to add the redistribution profile to.
- `priority` - (Required, int) The priority, integer from 1 to 255.
- `action` - (Optional) The action. Valid values are `redist` (default) or `no-redist`.
- `types` - (Optional) The source types. Valid values are `bgp`, `connect`, `ospf`, `rip`, and `static`.
- `interfaces` - (Optional) Specify candidate routes.
- `destinations` - (Optional) Specify candidate routes' next-hop addresses (subnet match).
- `next_hops` - (Optional) Specify candidate routes' next-hop addresses (subnet match).
- `ospf_path_types` - (Optional) OSPF path types. Valid values are `intra-area`, `inter-area`, `ext-1`, and `ext-2`.
- `ospf_areas` - (Optional) OSPF areas.

- `ospf_tags` - (Optional) OSPF tags.
- `bgp_communities` - (Optional) BGP communities.
- `bgp_extended_communities` - (Optional) BGP extended communities.

panos_security_policy

This resource allows you to manage the full security posture.

> **Note:** `panos_security_policies` is known as `panos_security_policy`.

This resource manages the full set of security rules in a vsys, enforcing both the contents of individual rules as well as their ordering. Rules are defined in a `rule` config block.

> **Note**: This resource will remove any security rule not defined in your plan file.

For each security rule, there are three styles of profile settings:

- `None` (the default)
- `Group`
- `Profiles`

The Profile Setting is implicitly chosen based on what params are configured for the security rule. If you want a Profile Setting of `Group`, then the `group` param should be set to the desired Group Profile. If you want a Profile Setting of `Profiles`, then you will need to specify one or more of the following params:

- `virus`
- `spyware`
- `vulnerability`
- `url_filtering`
- `file_blocking`
- `wildfire_analysis`
- `data_filtering`

If the `group` param and none of the `Profiles` params are specified, then the Profile Setting is set to `None`.

Import Name

```
<vsys>
```

Example Usage

```
resource "panos_security_policy" "example" {
    rule {
        name = "allow bizdev to dmz"
        source_zones = ["bizdev"]
        source_addresses = ["any"]
        source_users = ["any"]
        hip_profiles = ["any"]
        destination_zones = ["dmz"]
        destination_addresses = ["any"]
        applications = ["any"]
        services = ["application-default"]
        categories = ["any"]
        action = "allow"
    }
    rule {
        name = "deny sales to eng"
        source_zones = ["sales"]
        source_addresses = ["any"]
        source_users = ["any"]
        hip_profiles = ["any"]
        destination_zones = ["eng"]
        destination_addresses = ["any"]
        applications = ["any"]
        services = ["application-default"]
        categories = ["any"]
        action = "deny"
    }
}
```

Argument Reference

The following arguments are supported:

- `vsys` - (Optional) The vsys to put the security policy into (default: `vsys1`).
- `rulebase` - (Optional, Deprecated) The rulebase. For firewalls, there is only the `rulebase` value (default), but on Panorama, there is also `pre-rulebase` and `post-rulebase`.
- `rule` - A security rule definition (see below). The security rule ordering will match how they appear in the terraform plan file.

The following arguments are valid for each `rule` section:

- `name` - (Required) The security rule name.
- `type` - (Optional) Rule type. This can be `universal` (default), `interzone`, or `intrazone`.
- `description` - (Optional) The description.
- `tags` - (Optional) List of tags for this security rule.
- `source_zones` - (Required) List of source zones.
- `source_addresses` - (Required) List of source addresses.

- `negate_source` - (Optional, bool) If the source should be negated.
- `source_users` - (Required) List of source users.
- `hip_profiles` - (Required) List of HIP profiles.
- `destination_zones` - (Required) List of destination zones.
- `destination_addresses` - (Required) List of destination addresses.
- `negate_destination` - (Optional, bool) If the destination should be negated.
- `applications` - (Required) List of applications.
- `services` - (Required) List of services.
- `categories` - (Required) List of categories.
- `action` - (Optional) Action for the matched traffic. This can be `allow` (default), `deny`, `drop`, `reset-client`, `reset-server`, or `reset-both`.
- `log_setting` - (Optional) Log forwarding profile.
- `log_start` - (Optional, bool) Log the start of the traffic flow.
- `log_end` - (Optional, bool) Log the end of the traffic flow (default: `true`).
- `disabled` - (Optional, bool) Set to `true` to disable this rule.
- `schedule` - (Optional) The security policy schedule.
- `icmp_unreachable` - (Optional) Set to `true` to enable ICMP unreachable.
- `disable_server_response_inspection` - (Optional) Set to `true` to disable server response inspection.
- `group` - (Optional) Profile Setting: `Group` - The group profile name.
- `virus` - (Optional) Profile Setting: `Profiles` - The antivirus setting.
- `spyware` - (Optional) Profile Setting: `Profiles` - The anti-spyware setting.
- `vulnerability` - (Optional) Profile Setting: `Profiles` - The Vulnerability Protection setting.
- `url_filtering` - (Optional) Profile Setting: `Profiles` - The URL filtering setting.
- `file_blocking` - (Optional) Profile Setting: `Profiles` - The file blocking setting.
- `wildfire_analysis` - (Optional) Profile Setting: `Profiles` - The WildFire Analysis setting.
- `data_filtering` - (Optional) Profile Setting: `Profiles` - The Data Filtering setting.

panos_security_rule_group

This resource allows you to add/update/delete security rule groups.

> **Note:** `panos_security_policy_group` is known as `panos_security_rule_group`.

This resource manages clusters of security rules in a single vsys, enforcing both the contents of individual rules as well as their ordering. Rules are defined in a `rule` config block.

Because this resource only manages what it's told to, it will not manage any rules that may already exist on the firewall. This has implications on the effective security posture of your firewall, but it will allow you to spread your security rules across multiple Terraform state files. If you want to verify that the security rules are only what appears in the plan file, then you should probably be using the panos_security_policy (/docs/providers/panos/r/security_policy.html) resource.

Although you cannot modify non-group security rules with this resource, the `position_keyword` and `position_reference` parameters allow you to reference some other security rule that already exists, using it as a means to ensure some rough placement within the ruleset as a whole.

For each security rule, there are three styles of profile settings:

- `None` (the default)
- `Group`
- `Profiles`

The Profile Setting is implicitly chosen based on what params are configured for the security rule. If you want a Profile Setting of `Group`, then the `group` param should be set to the desired Group Profile. If you want a Profile Setting of `Profiles`, then you will need to specify one or more of the following params:

- `virus`
- `spyware`
- `vulnerability`
- `url_filtering`
- `file_blocking`
- `wildfire_analysis`
- `data_filtering`

If the `group` param and none of the `Profiles` params are specified, then the Profile Setting is set to `None`.

Best Practices

As is to be expected, if you are separating your deployment across multiple plan files, make sure that at most only one plan specifies any given absolute positioning keyword such as "top" or "directly below", otherwise they'll keep shoving each other out of the way indefinitely.

Best practices are to specify one group as `top` (if you need it), one group as `bottom` (this is where you have your logging deny rule), then all other groups should be `above` the first rule of the bottom group. You do it this way because rules will natually be added at the tail end of the rulebase, so they will always be `after` the first group, but what you want is for them to be `before` the last group's rules.

Example Usage

```
resource "panos_security_rule_group" "example" {
    position_keyword = "above"
    position_reference = "deny everything else"
    rule {
        name = "allow bizdev to dmz"
        source_zones = ["bizdev"]
        source_addresses = ["any"]
        source_users = ["any"]
        hip_profiles = ["any"]
        destination_zones = ["dmz"]
        destination_addresses = ["any"]
        applications = ["any"]
        services = ["application-default"]
        categories = ["any"]
        action = "allow"
    }
    rule {
        name = "deny sales to eng"
        source_zones = ["sales"]
        source_addresses = ["any"]
        source_users = ["any"]
        hip_profiles = ["any"]
        destination_zones = ["eng"]
        destination_addresses = ["any"]
        applications = ["any"]
        services = ["application-default"]
        categories = ["any"]
        action = "deny"
    }
}
```

Argument Reference

The following arguments are supported:

- `vsys` - (Optional) The vsys to put the security rule into (default: `vsys1`).
- `position_keyword` - (Optional) A positioning keyword for this group. This can be `before`, `directly before`, `after`, `directly after`, `top`, `bottom`, or left empty (the default) to have no particular placement. This param works in combination with the `position_reference` param.
- `position_reference` - (Optional) Required if `position_keyword` is one of the "above" or "below" variants, this is the name of a non-group rule to use as a reference to place this group.
- `rule` - The security rule definition (see below). The security rule ordering will match how they appear in the terraform

plan file.

The following arguments are valid for each `rule` section:

- `name` - (Required) The security rule name.
- `type` - (Optional) Rule type. This can be `universal` (default), `interzone`, or `intrazone`.
- `description` - (Optional) The description.
- `tags` - (Optional) List of tags for this security rule.
- `source_zones` - (Required) List of source zones.
- `source_addresses` - (Required) List of source addresses.
- `negate_source` - (Optional, bool) If the source should be negated.
- `source_users` - (Required) List of source users.
- `hip_profiles` - (Required) List of HIP profiles.
- `destination_zones` - (Required) List of destination zones.
- `destination_addresses` - (Required) List of destination addresses.
- `negate_destination` - (Optional, bool) If the destination should be negated.
- `applications` - (Required) List of applications.
- `services` - (Required) List of services.
- `categories` - (Required) List of categories.
- `action` - (Optional) Action for the matched traffic. This can be `allow` (default), `deny`, `drop`, `reset-client`, `reset-server`, or `reset-both`.
- `log_setting` - (Optional) Log forwarding profile.
- `log_start` - (Optional, bool) Log the start of the traffic flow.
- `log_end` - (Optional, bool) Log the end of the traffic flow (default: `true`).
- `disabled` - (Optional, bool) Set to `true` to disable this rule.
- `schedule` - (Optional) The security rule schedule.
- `icmp_unreachable` - (Optional) Set to `true` to enable ICMP unreachable.
- `disable_server_response_inspection` - (Optional) Set to `true` to disable server response inspection.
- `group` - (Optional) Profile Setting: `Group` - The group profile name.
- `virus` - (Optional) Profile Setting: `Profiles` - The antivirus setting.
- `spyware` - (Optional) Profile Setting: `Profiles` - The anti-spyware setting.
- `vulnerability` - (Optional) Profile Setting: `Profiles` - The Vulnerability Protection setting.
- `url_filtering` - (Optional) Profile Setting: `Profiles` - The URL filtering setting.

- `file_blocking` - (Optional) Profile Setting: `Profiles` - The file blocking setting.
- `wildfire_analysis` - (Optional) Profile Setting: `Profiles` - The WildFire Analysis setting.
- `data_filtering` - (Optional) Profile Setting: `Profiles` - The Data Filtering setting.

panos_service_group

This resource allows you to add/update/delete service groups.

Import Name

```
<vsys>:<name>
```

Example Usage

```
resource "panos_service_group" "example" {
    name = "static ntp grp"
    services = ["svc1", "svc2"]
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The service group's name.
- `vsys` - (Optional) The vsys to put the service group into (default: `vsys1`).
- `services` - (Required) List of services to put in this service group.
- `tags` - (Optional) List of administrative tags.

panos_service_object

This resource allows you to add/update/delete service objects.

Import Name

```
<vsys>:<name>
```

Example Usage

```
resource "panos_service_object" "example" {
    name = "my_service"
    vsys = "vsys1"
    protocol = "tcp"
    description = "My service object"
    source_port = "2000-2049,2051-2099"
    destination_port = "32123"
    tags = ["internal", "dmz"]
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The service object's name.
- `vsys` - (Optional) The vsys to put the service object into (default: `vsys1`).
- `description` - (Optional) The service object's description.
- `protocol` - (Required) The service's protocol. This should be `tcp`, `udp`, or `sctp` (PAN-OS 8.1+).
- `source_port` - (Optional) The source port. This can be a single port number, range (1-65535), or comma separated (80,8080,443).
- `destination_port` - (Required) The destination port. This can be a single port number, range (1-65535), or comma separated (80,8080,443).
- `tags` - (Optional) List of administrative tags.
- `override_session_timeout` - (Optional, bool, PAN-OS 8.1+) Set to `true` to override the default application timeouts.
- `override_timeout` - (Optional, int, PAN-OS 8.1+) The overridden TCP timeout.
- `override_half_closed_timeout` - (Optional, int, PAN-OS 8.1+) The overridden TCP half closed timeout.

- `override_time_wait_timeout` - (Optional, int, PAN-OS 8.1+) The overridden TCP wait time.

panos_snmptrap_server_profile

This resource allows you to add/update/delete snmptrap server profiles.

Example Usage

```
resource "panos_snmptrap_server_profile" "example" {
    name = "myProfile"
    v2c_server {
        name = "first server"
        manager = "snmp1.example.com"
        community = "public"
    }
}
```

Argument Reference

The following arguments are supported:

- `vsys` - (Optional) The vsys (default: `shared`).
- `name` - (Required) The group's name.
- `v2c_server` - (Optional, repeatable) A v2c server (defined below).
- `v3_server` - (Optional, repeatable) A v3 server (defined below).

`v2c_server` supports the following arguments:

- `name` - (Required) The server name.
- `manager` - (Required) The hostname.
- `community` - (Required) The SNMP community.

`v3_server` supports the following arguments:

- `name` - (Required) The server name.
- `manager` - (Required) The hostname.
- `user` - (Required) Username.
- `engine_id` - (Optional) The engine ID.
- `auth_password` - (Required) SNMP auth password.
- `priv_password` - (Required) SNMP priv password.

panos_static_route_ipv4

This resource allows you to add/update/delete IPv4 static routes on a virtual router.

Import Name

```
<virtual_router>:<name>
```

Example Usage

```
resource "panos_static_route_ipv4" "example" {
    name = "localnet"
    virtual_router = "${panos_virtual_router.vr1.name}"
    destination = "10.1.7.0/32"
    next_hop = "10.1.7.4"
}

resource "panos_virtual_router" "vr1" {
    name = "my virtual router"
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The static route's name.
- `virtual_router` - (Required) The virtual router to add the static route to.
- `destination` - (Required) Destination IP address / prefix.
- `interface` - (Optional) Interface to use.
- `type` - (Optional) The next hop type. Valid values are `ip-address` (the default), `discard`, `next-vr`, or an empty string for `None`.
- `next_hop` - (Optional) The value for the `type` setting.
- `admin_distance` - (Optional) The admin distance.
- `metric` - (Optional, int) Metric value / path cost (default: `10`).
- `route_table` - (Optional) Target routing table to install the route. Valid values are `unicast` (the default), `no install`, `multicast`, or `both`.
- `bfd_profile` - (Optional, PAN-OS 7.1+) BFD configuration.

panos_syslog_server_profile

This resource allows you to add/update/delete syslog server profiles.

Import Name

```
<vsys>:<name>
```

Example Usage

```
resource "panos_syslog_server_profile" "example" {
    name = "myProfile"
    threat_format = "$serial $severity"
    syslog_server {
        name = "my-server"
        server = "syslog.example.com"
    }
}
```

Argument Reference

The following arguments are supported:

- `vsys` - (Optional) The vsys (default: `shared`).
- `name` - (Required) The group's name.
- `config_format` - (Optional) Config format.
- `system_format` - (Optional) System format.
- `threat_format` - (Optional) Threat format.
- `traffic_format` - (Optional) Traffic format.
- `hip_match_format` - (Optional) HIP match format.
- `url_format` - (Optional) URL format.
- `data_format` - (Optional) Data format.
- `wildfire_format` - (Optional) Wildfire format.
- `tunnel_format` - (Optional) Tunnel format.
- `user_id_format` - (Optional) UserID format.
- `gtp_format` - (Optional) GTP format.

- `auth_format` - (Optional) Auth format.
- `sctp_format` - (Optional) SCTP format.
- `iptag_format` - (Optional) IP tag format.
- `escaped_characters` - (Optional) The escaped characters (as a string).
- `escape_character` - (Optional) The escape character.
- `syslog_server` - (Required, repeatable) The server spec (defined below).

`syslog_server` supports the following arguments:

- `name` - (Required) Server name.
- `server` - (Required) The hostname.
- `transport` - (Optional) The transport. Valid values are `UDP` (default), `TCP`, or `SSL`.
- `port` - (Optional, int) The port number (default: 514).
- `syslog_format` - (Optional) The syslog format. Valid values are `BSD` (default) or `IETF`.
- `facility` - (Optional) The syslog facility. Valid values are `LOG_USER` (default), `LOG_LOCAL0`, `LOG_LOCAL1`, `LOG_LOCAL2`, `LOG_LOCAL3`, `LOG_LOCAL4`, `LOG_LOCAL5`, `LOG_LOCAL6`, or `LOG_LOCAL7`.

panos_telemetry

This resource allows you to add/update/delete telemetry sharing.

Join other Palo Alto Networks customers in a global sharing community, helping to raise the bar against the latest attack techniques. Your participation allows us to deliver new threat prevention controls across the attack lifecycle. Choose the type of data you share across applications, threat intelligence, and device health information to improve the fidelity of the protections we deliver. This is an opt-in feature controlled with granular policy, and we encourage you to join the community.

Example Usage

```
resource "panos_telemetry" "example" {
    threat_prevention_reports = true
    threat_prevention_data = true
    threat_prevention_packet_captures = true
}
```

Argument Reference

The following arguments are supported:

- `application_reports` - (Bool, optional) Application reports.
- `threat_prevention_reports` - (Bool, optional) Threat reports.
- `url_reports` - (Bool, optional) URL reports.
- `file_type_identification_reports` - (Bool, optional) File type identification reports.
- `threat_prevention_data` - (Bool, optional) Threat prevention data.
- `threat_prevention_packet_captures` - (Bool, optional) Enable sending packet- captures with threat prevention information. This requires that `threat_prevention_data` also be enabled.
- `product_usage_stats` - (Bool, optional) Health and performance reports.
- `passive_dns_monitoring` - (Bool, optional) Passive DNS monitoring.

panos_tunnel_interface

This resource allows you to add/update/delete tunnel interfaces.

Import Name

```
<vsys>:<name>
```

Example Usage

```
resource "panos_tunnel_interface" "example1" {
    name = "tunnel.5"
    static_ips = ["10.1.1.1/24"]
    comment = "Configured for internal traffic"
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The interface's name. This must start with `tunnel.`.
- `vsys` - (Optional) The vsys that will use this interface (default: `vsys1`).
- `comment` - (Optional) The interface comment.
- `netflow_profile` - (Optional) The netflow profile.
- `static_ips` - (Optional) List of static IPv4 addresses to set for this data interface.
- `management_profile` - (Optional) The management profile.
- `mtu` - (Optional) The MTU.

panos_virtual_router_entry

This resource allows you to add/update/delete an interface in a virtual router.

This resource has some overlap with the `panos_virtual_router` resource. If you want to use this resource with the other one, then make sure that your `panos_virtual_router` spec does not define the `interfaces` field.

Import Name

```
<virtual_router>:<interface>
```

Example Usage

```
resource "panos_virtual_router" "vr" {
    name = "my vr"
}

resource "panos_virtual_router_entry" "example" {
    virtual_router = "${panos_virtual_router.vr.name}"
    interface = "ethernet1/5"
}
```

Argument Reference

The following arguments are supported:

- `virtual_router` - (Required) The virtual router's name.
- `interface` - (Required) The interface to import into the virtual router.

panos_virtual_router

This resource allows you to add/update/delete virtual routers.

Note - The `default` virtual router may be configured with this resource, however it will not be deleted from the firewall. It will only be unexported from the vsys that it is currently imported in, and any interfaces imported into the virtual router will be removed.

This resource has some overlap with the `panos_virtual_router_entry` resource. If you want to use this resource with the other one, then make sure that your `panos_virtual_router` spec does not define the `interfaces` field.

Import Name

```
<vsys>:<name>
```

Example Usage

```
# Configure a bare-bones ethernet interface.
resource "panos_virtual_router" "example" {
    name = "my virtual router"
    static_dist = 15
    interfaces = ["ethernet1/1", "ethernet1/2"]
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The virtual router's name.
- `vsys` - (Required) The vsys that will use this virtual router. This should be something like `vsys1` or `vsys3`.
- `interfaces` - (Optional) List of interfaces that should use this virtual router.
- `static_dist` - (Optional) Admin distance - Static (default: `10`).
- `static_ipv6_dist` - (Optional) Admin distance - Static IPv6 (default: `10`).
- `ospf_int_dist` - (Optional) Admin distance - OSPF Int (default: `30`).
- `ospf_ext_dist` - (Optional) Admin distance - OSPF Ext (default: `110`).
- `ospfv3_int_dist` - (Optional) Admin distance - OSPFv3 Int (default: `30`).
- `ospfv3_ext_dist` - (Optional) Admin distance - OSPFv3 Ext (default: `110`).
- `ibgp_dist` - (Optional) Admin distance - IBGP (default: `200`).

- `ebgp_dist` - (Optional) Admin distance - EBGP (default: `20`).
- `rip_dist` - (Optional) Admin distance - RIP (default: `120`).

panos_vlan_entry

This resource allows you to add/update/delete an interface in a VLAN.

Import Name

```
<vlan>:<interface>
```

Example Usage

```
resource "panos_vlan_entry" "example" {
    vlan = panos_vlan.vlan1.name
    interface = panos_ethernet_interface.e1.name
    mac_addresses = [
        "00:30:48:55:66:77",
        "00:30:48:55:66:88",
    ]
}

resource "panos_vlan" "vlan1" {
    name = "myVlan"
}

resource "panos_ethernet_interface" "e1" {
    name = "ethernet1/5"
    mode = "layer2"
    vsys = "vsys1"
}
```

Argument Reference

The following arguments are supported:

- `vlan` - (Required) The VLAN's name.
- `interface` - (Required) The interface's name.
- `mac_addresses` - (Optional) List of MAC addresses that should go with this entry.

panos_vlan

This resource allows you to add/update/delete VLANs.

Import Name

```
<vsys>:<name>
```

Example Usage

```
resource "panos_vlan" "example" {
    name = "myVlan"
    vlan_interface = panos_vlan_interface.vli.name
}

resource "panos_vlan_interface" "vli" {
    name = "vlan.6"
    vsys = "vsys1"
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The object's name.
- `vsys` - (Optional) The vsys to put the object into (default: `vsys1`).
- `vlan_interface` - (Optional) The VLAN interface.
- `interfaces` - (Optional, computed) List of layer2 interfaces. You can also leave this blank and also use panos_vlan_entry (/docs/providers/panos/r/vlan_entry.html) for more control.

panos_vlan_interface

This resource allows you to add/update/delete vlan interfaces.

Import Name

```
<vsys>:<name>
```

Example Usage

```
resource "panos_vlan_interface" "example" {
    name = "vlan.17"
    vsys = "vsys1"
    mode = "layer3"
    static_ips = ["10.1.1.1/24"]
    comment = "Configured for internal traffic"
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The interface's name. Must start with `vlan.`.
- `vsys` - (Optional) The vsys that will use this interface (default: `vsys1`).
- `comment` - (Optional) The interface comment.
- `netflow_profile` - (Optional) The netflow profile.
- `static_ips` - (Optional) List of static IPv4 addresses to set for this data interface.
- `enable_dhcp` - (Optional) Set to `true` to enable DHCP on this interface.
- `create_dhcp_default_route` - (Optional) Set to `true` to create a DHCP default route.
- `dhcp_default_route_metric` - (Optional) The metric for the DHCP default route.
- `management_profile` - (Optional) The management profile.
- `mtu` - (Optional) The MTU.
- `adjust_tcp_mss` - (Optional) Adjust TCP MSS (default: false).
- `ipv4_mss_adjust` - (Optional, PAN-OS 8.0+) The IPv4 MSS adjust value.
- `ipv6_mss_adjust` - (Optional, PAN-OS 8.0+) The IPv6 MSS adjust value.

panos_zone_entry

This resource allows you to add/update/delete a specific interface in a zone.

This resource has some overlap with the `panos_zone` resource. If you want to use this resource with the other one, then make sure that your `panos_zone` spec does not define the `interfaces` field.

This is the appropriate resource to use if you have a pre-existing zone and don't want Terraform to delete it on `terraform destroy`.

Import Name

```
<vsys>:<zone>:<mode>:<interface>
```

Example Usage

```
resource "panos_ethernet_interface" "e5" {
    name = "ethernet1/5"
    mode = "layer3"
}

resource "panos_zone" "z" {
    name = "exZone"
    mode = "layer3"
}

resource "panos_zone_entry" "example" {
    zone = "${panos_zone.z.name}"
    mode = "${panos_zone.z.mode}"
    interface = "${panos_ethernet_interface.e5.name}"
}
```

Argument Reference

The following arguments are supported:

- `vsys` - (Optional) The vsys (default: `vsys1`).
- `zone` - (Required) The zone's name.
- `mode` - (Optional) The mode. Can be `layer3` (default), `layer2`, `virtual-wire`, `tap`, or `external`.
- `interface` - (Required) The interface's name.

panos_zone

This resource allows you to add/update/delete zones.

This resource has some overlap with the `panos_zone_entry` resource. If you want to use this resource with the other one, then make sure that your `panos_zone` spec does not define the `interfaces` field.

Import Name

```
<vsys>:<name>
```

Example Usage

```
resource "panos_zone" "example" {
    name = "myZone"
    mode = "layer3"
    interfaces = ["${panos_ethernet_interface.e1.name}", "${panos_ethernet_interface.e5.name}"]
    enable_user_id = true
    exclude_acls = ["192.168.0.0/16"]
}

resource "panos_ethernet_interface" "e1" {
    name = "ethernet1/1"
    mode = "layer3"
}

resource "panos_ethernet_interface" "e5" {
    name = "ethernet1/5"
    mode = "layer3"
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The zone's name.
- `vsys` - (Optional) The vsys to put the zone into (default: `vsys1`).
- `mode` - (Required) The zone's mode. This can be `layer3`, `layer2`, `virtual-wire`, `tap`, or `tunnel`.
- `zone_profile` - (Optional) The zone protection profile.
- `log_setting` - (Optional) Log setting.
- `enable_user_id` - (Optional) Boolean to enable user identification.
- `interfaces` - (Optional) List of interfaces to associated with this zone.

- `include_acls` - (Optional) Users from these addresses/subnets will be identified. This can be an address object, an address group, a single IP address, or an IP address subnet.
- `exclude_acls` - (Optional) Users from these addresses/subnets will not be identified. This can be an address object, an address group, a single IP address, or an IP address subnet.