ACME Certificate and Account Provider

The Automated Certificate Management Environment (ACME) is an evolving standard for the automation of a domain-validated certificate authority. Clients register themselves on an authority using a private key and contact information, and answer challenges for domains that they own by supplying response data issued by the authority via either HTTP or DNS. Via this process, they prove that they own the domains in question, and can then request certificates for them via the CA. No part of this process requires user interaction, a traditional blocker in obtaining a domain validated certificate.

Currently the major ACME CA is Let's Encrypt (https://letsencrypt.org), but the ACME support in Terraform can be configured to use any ACME CA, including an internal one that is set up using Boulder (https://github.com/letsencrypt/boulder), or another CA that implements the ACME standard with Let's Encrypt's divergences (https://github.com/letsencrypt/boulder/blob/master/docs/acme-divergences.md).

For more detail on the ACME process, see here (https://letsencrypt.org/how-it-works/). For the ACME spec, click here (https://ietf-wg-acme.github.io/acme/draft-ietf-acme-acme.html). Note that as mentioned in the last paragraph, the ACME provider may diverge (https://github.com/letsencrypt/boulder/blob/master/docs/acme-divergences.md) from the current ACME spec to account for the real-world divergences that are made by CAs such as Let's Encrypt.

NOTE: The upstream version of the ACME provider supports ACME v2 only. For ACME v1 endpoints, version 0.6.0 is required, which can be found here (https://github.com/vancluever/terraform-provider-acme/releases/tag/v0.6.0). Note that this version is a 3rd party plugin (/docs/configuration/providers.html#third-party-plugins) and needs to be installed as such.

Basic Example

The following example can be used to create an account using the `acme_registration` (/docs/providers/acme/r/registration.html) resource, and a certificate using the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource. The initial private key is created using the `tls_private_key` (/docs/providers/tls/r/private_key.html) resource, but can be supplied via other means. DNS validation is performed by using Amazon Route 53 (https://aws.amazon.com/route53/), for which appropriate credentials are assumed to be in your environment.

NOTE: The directory URLs in all examples in this provider reference Let's Encrypt's staging server endpoint. For production use, change the directory URLs to the production endpoints, which can be found here (https://letsencrypt.org/docs/acme-protocol-updates/).

```
provider "acme" {
  server_url = "https://acme-staging-v02.api.letsencrypt.org/directory"
}

resource "tls_private_key" "private_key" {
  algorithm = "RSA"
}

resource "acme_registration" "reg" {
  account_key_pem = "${tls_private_key.private_key.private_key_pem}"
  email_address   = "nobody@example.com"
}

resource "acme_certificate" "certificate" {
  account_key_pem           = "${acme_registration.reg.account_key_pem}"
  common_name               = "www.example.com"
  subject_alternative_names = ["www2.example.com"]

  dns_challenge {
    provider = "route53"
  }
}
```

Argument Reference

The following arguments are required:

- `server_url` - (Required) The URL to the ACME endpoint's directory.

Note that the account key is not a provider-level config value at this time to allow the management of accounts and certificates within the same provider.

acme_certificate DNS Challenge Providers

This subsection documents all of the DNS challenge providers that can be used with the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource.

For complete information on how to use these providers with the `acme_certifiate` resource, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

Refer to a specific provider on the left sidebar for more details.

Using Variable Files for Provider Arguments

Most provider arguments can be suffixed with `_FILE` to specify that you wish to store that value in a local file. This can be useful if local storage for these values is desired over configuration as variables or within the environment.

See the example (/docs/providers/acme/r/certificate.html#using-variable-files-for-provider-arguments) in the `acme_certificate` resource for more details.

Relation to Terraform provider configuration

The DNS provider configurations specified in the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource are separate from any that you supply in a corresponding provider whose functionality overlaps with the certificate's DNS providers. This ensures that there are no hard dependencies between any of these providers and the ACME provider, but it is important to note so that configuration is supplied correctly.

As an example, if you specify manual configuration for the AWS provider (/docs/providers/aws/index.html) via the `provider` (/docs/configuration/providers.html) block instead of the environment, you will still need to supply the configuration explicitly in the `config` block of the `dns_challenge` (/docs/providers/acme/r/certificate.html#dns_challenge) argument.

Note that some of Terraform's providers have environment variable settings that overlap with the settings here, generally depending on whether or not these variables are supported by the corresponding provider's SDK.

We alias certain provider environment variables so the same settings can be supplied to both ACME and the respective native cloud provider. For specific details, see the page for the provider in question.

NOTE: The following documentation is auto-generated from the ACME provider's API library lego (https://go-acme.github.io/lego/). Some sections may refer to lego directly - in most cases, these sections apply to the Terraform provider as well.

Joohoi's ACME-DNS DNS Challenge Provider

The `acme-dns` DNS challenge provider can be used to perform DNS challenges for the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource with Joohoi's ACME-DNS (https://github.com/joohoi/acme-dns).

For complete information on how to use this provider with the `acme_certifiate` resource, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

Example

```
resource "acme_certificate" "certificate" {
  ...

  dns_challenge {
    provider = "acme-dns"
  }
}
```

Argument Reference

The following arguments can be either passed as environment variables, or directly through the `config` block in the `dns_challenge` (/docs/providers/acme/r/certificate.html#dns_challenge) argument in the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource. For more details, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

In addition, arguments can also be stored in a local file, with the path supplied by supplying the argument with the `_FILE` suffix. See here (/docs/providers/acme/r/certificate.html#using-variable-files-for-provider-arguments) for more information.

- `ACME_DNS_API_BASE` - The ACME-DNS API address.
- `ACME_DNS_STORAGE_PATH` - The ACME-DNS JSON account data file. A per-domain account will be registered/persisted to this file and used for TXT updates..

NOTE: The following documentation is auto-generated from the ACME provider's API library lego (https://go-acme.github.io/lego/). Some sections may refer to lego directly - in most cases, these sections apply to the Terraform provider as well.

Alibaba Cloud DNS DNS Challenge Provider

The `alidns` DNS challenge provider can be used to perform DNS challenges for the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource with Alibaba Cloud DNS (https://www.alibabacloud.com/product/dns).

For complete information on how to use this provider with the `acme_certifiate` resource, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

Example

```
resource "acme_certificate" "certificate" {
  ...

  dns_challenge {
    provider = "alidns"
  }
}
```

Argument Reference

The following arguments can be either passed as environment variables, or directly through the `config` block in the `dns_challenge` (/docs/providers/acme/r/certificate.html#dns_challenge) argument in the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource. For more details, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

In addition, arguments can also be stored in a local file, with the path supplied by supplying the argument with the `_FILE` suffix. See here (/docs/providers/acme/r/certificate.html#using-variable-files-for-provider-arguments) for more information.

- `ALICLOUD_ACCESS_KEY` - Access key ID.
- `ALICLOUD_SECRET_KEY` - Access Key secret.
- `ALICLOUD_HTTP_TIMEOUT` - API request timeout.
- `ALICLOUD_POLLING_INTERVAL` - Time between DNS propagation check.
- `ALICLOUD_PROPAGATION_TIMEOUT` - Maximum waiting time for DNS propagation.
- `ALICLOUD_TTL` - The TTL of the TXT record used for the DNS challenge.

NOTE: The following documentation is auto-generated from the ACME provider's API library lego (https://go-acme.github.io/lego/). Some sections may refer to lego directly - in most cases, these sections apply to the Terraform provider as well.

Aurora DNS DNS Challenge Provider

The `auroradns` DNS challenge provider can be used to perform DNS challenges for the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource with Aurora DNS (https://www.pcextreme.com/aurora/dns).

For complete information on how to use this provider with the `acme_certifiate` resource, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

Example

```
resource "acme_certificate" "certificate" {
  ...

  dns_challenge {
    provider = "auroradns"
  }
}
```

Argument Reference

The following arguments can be either passed as environment variables, or directly through the `config` block in the `dns_challenge` (/docs/providers/acme/r/certificate.html#dns_challenge) argument in the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource. For more details, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

In addition, arguments can also be stored in a local file, with the path supplied by supplying the argument with the `_FILE` suffix. See here (/docs/providers/acme/r/certificate.html#using-variable-files-for-provider-arguments) for more information.

- `AURORA_ENDPOINT` - API endpoint URL.
- `AURORA_KEY` - User API key.
- `AURORA_USER_ID` - User ID.
- `AURORA_POLLING_INTERVAL` - Time between DNS propagation check.
- `AURORA_PROPAGATION_TIMEOUT` - Maximum waiting time for DNS propagation.
- `AURORA_TTL` - The TTL of the TXT record used for the DNS challenge.

NOTE: The following documentation is auto-generated from the ACME provider's API library lego (https://go-acme.github.io/lego/). Some sections may refer to lego directly - in most cases, these sections apply to the Terraform provider as well.

Azure DNS Challenge Provider

The `azure` DNS challenge provider can be used to perform DNS challenges for the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource with Azure (https://azure.microsoft.com/services/dns/).

For complete information on how to use this provider with the `acme_certifiate` resource, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

Example

```
resource "acme_certificate" "certificate" {
  ...

  dns_challenge {
    provider = "azure"
  }
}
```

Argument Reference

The following arguments can be either passed as environment variables, or directly through the `config` block in the `dns_challenge` (/docs/providers/acme/r/certificate.html#dns_challenge) argument in the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource. For more details, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

In addition, arguments can also be stored in a local file, with the path supplied by supplying the argument with the `_FILE` suffix. See here (/docs/providers/acme/r/certificate.html#using-variable-files-for-provider-arguments) for more information.

- `AZURE_CLIENT_ID` - Client ID.
- `AZURE_CLIENT_SECRET` - Client secret.
- `AZURE_RESOURCE_GROUP` - Resource group.
- `AZURE_SUBSCRIPTION_ID` - Subscription ID.
- `AZURE_TENANT_ID` - Tenant ID.
- `instance metadata service` - If the credentials are **not** set via the environment, then it will attempt to get a bearer token via the instance metadata service (https://docs.microsoft.com/en-us/azure/virtual-machines/windows/instance-metadata-service)..

- `AZURE_METADATA_ENDPOINT` - Metadata Service endpoint URL.
- `AZURE_POLLING_INTERVAL` - Time between DNS propagation check.
- `AZURE_PROPAGATION_TIMEOUT` - Maximum waiting time for DNS propagation.
- `AZURE_TTL` - The TTL of the TXT record used for the DNS challenge.

The following variables are **Terraform-specific** aliases for the above configuration values:

- `ARM_CLIENT_ID` - alias for `AZURE_CLIENT_ID`.
- `ARM_CLIENT_SECRET` - alias for `AZURE_CLIENT_SECRET`.
- `ARM_RESOURCE_GROUP` - alias for `AZURE_RESOURCE_GROUP`.
- `ARM_SUBSCRIPTION_ID` - alias for `AZURE_SUBSCRIPTION_ID`.
- `ARM_TENANT_ID` - alias for `AZURE_TENANT_ID`.

NOTE: The following documentation is auto-generated from the ACME provider's API library lego (https://go-acme.github.io/lego/). Some sections may refer to lego directly - in most cases, these sections apply to the Terraform provider as well.

Bindman DNS Challenge Provider

The `bindman` DNS challenge provider can be used to perform DNS challenges for the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource with Bindman (https://github.com/labbsr0x/bindman-dns-webhook).

For complete information on how to use this provider with the `acme_certifiate` resource, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

Example

```
resource "acme_certificate" "certificate" {
  ...

  dns_challenge {
    provider = "bindman"
  }
}
```

Argument Reference

The following arguments can be either passed as environment variables, or directly through the `config` block in the `dns_challenge` (/docs/providers/acme/r/certificate.html#dns_challenge) argument in the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource. For more details, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

In addition, arguments can also be stored in a local file, with the path supplied by supplying the argument with the `_FILE` suffix. See here (/docs/providers/acme/r/certificate.html#using-variable-files-for-provider-arguments) for more information.

- `BINDMAN_MANAGER_ADDRESS` - The server URL, should have scheme, hostname, and port (if required) of the Bindman-DNS Manager server.
- `BINDMAN_HTTP_TIMEOUT` - API request timeout.
- `BINDMAN_POLLING_INTERVAL` - Time between DNS propagation check.
- `BINDMAN_PROPAGATION_TIMEOUT` - Maximum waiting time for DNS propagation.

NOTE: The following documentation is auto-generated from the ACME provider's API library lego (https://go-acme.github.io/lego/). Some sections may refer to lego directly - in most cases, these sections apply to the Terraform provider as well.

Bluecat DNS Challenge Provider

The `bluecat` DNS challenge provider can be used to perform DNS challenges for the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource with Bluecat (https://www.bluecatnetworks.com).

For complete information on how to use this provider with the `acme_certifiate` resource, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

Example

```
resource "acme_certificate" "certificate" {
  ...

  dns_challenge {
    provider = "bluecat"
  }
}
```

Argument Reference

The following arguments can be either passed as environment variables, or directly through the `config` block in the `dns_challenge` (/docs/providers/acme/r/certificate.html#dns_challenge) argument in the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource. For more details, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

In addition, arguments can also be stored in a local file, with the path supplied by supplying the argument with the `_FILE` suffix. See here (/docs/providers/acme/r/certificate.html#using-variable-files-for-provider-arguments) for more information.

- `BLUECAT_CONFIG_NAME` - Configuration name.
- `BLUECAT_DNS_VIEW` - External DNS View Name.
- `BLUECAT_PASSWORD` - API password.
- `BLUECAT_SERVER_URL` - The server URL, should have scheme, hostname, and port (if required) of the authoritative Bluecat BAM serve.
- `BLUECAT_USER_NAME` - API username.
- `BLUECAT_HTTP_TIMEOUT` - API request timeout.
- `BLUECAT_POLLING_INTERVAL` - Time between DNS propagation check.

- `BLUECAT_PROPAGATION_TIMEOUT` - Maximum waiting time for DNS propagation.
- `BLUECAT_TTL` - The TTL of the TXT record used for the DNS challenge.

NOTE: The following documentation is auto-generated from the ACME provider's API library lego (https://go-acme.github.io/lego/). Some sections may refer to lego directly - in most cases, these sections apply to the Terraform provider as well.

Cloudflare DNS Challenge Provider

The `cloudflare` DNS challenge provider can be used to perform DNS challenges for the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource with Cloudflare (https://www.cloudflare.com/dns/).

For complete information on how to use this provider with the `acme_certifiate` resource, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

Example

```
resource "acme_certificate" "certificate" {
  ...

  dns_challenge {
    provider = "cloudflare"
  }
}
```

Argument Reference

The following arguments can be either passed as environment variables, or directly through the `config` block in the `dns_challenge` (/docs/providers/acme/r/certificate.html#dns_challenge) argument in the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource. For more details, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

In addition, arguments can also be stored in a local file, with the path supplied by supplying the argument with the `_FILE` suffix. See here (/docs/providers/acme/r/certificate.html#using-variable-files-for-provider-arguments) for more information.

- `CF_API_EMAIL` - Account email.
- `CF_API_KEY` - API key.
- `CF_DNS_API_TOKEN` - API token with DNS:Edit permission (since v3.1.0).
- `CF_ZONE_API_TOKEN` - API token with Zone:Read permission (since v3.1.0).
- `CLOUDFLARE_API_KEY` - Alias to CF_API_KEY.
- `CLOUDFLARE_DNS_API_TOKEN` - Alias to CF_DNS_API_TOKEN.
- `CLOUDFLARE_EMAIL` - Alias to CF_API_EMAIL.
- `CLOUDFLARE_ZONE_API_TOKEN` - Alias to CF_ZONE_API_TOKEN.

- `CLOUDFLARE_HTTP_TIMEOUT` - API request timeout.
- `CLOUDFLARE_POLLING_INTERVAL` - Time between DNS propagation check.
- `CLOUDFLARE_PROPAGATION_TIMEOUT` - Maximum waiting time for DNS propagation.
- `CLOUDFLARE_TTL` - The TTL of the TXT record used for the DNS challenge.

Description

You may use `CF_API_EMAIL` and `CF_API_KEY` to authenticate, or `CF_DNS_API_TOKEN`, or `CF_DNS_API_TOKEN` and `CF_ZONE_API_TOKEN`.

API keys

If using API keys (`CF_API_EMAIL` and `CF_API_KEY`), the Global API Key needs to be used, not the Origin CA Key.

Please be aware, that this in principle allows Lego to read and change *everything* related to this account.

API tokens

With API tokens (`CF_DNS_API_TOKEN`, and optionally `CF_ZONE_API_TOKEN`), very specific access can be granted to your resources at Cloudflare. See this Cloudflare announcement (https://blog.cloudflare.com/api-tokens-general-availability/) for details.

The main resources Lego cares for are the DNS entries for your Zones. It also need to resolve a domain name to an internal Zone ID in order to manipulate DNS entries.

Hence, you should create an API token with the following permissions:

- Zone / Zone / Read
- Zone / DNS / Edit

You also need to scope the access to all your domains for this to work. Then pass the API token as `CF_DNS_API_TOKEN` to Lego.

Alternatively, if you prefer a more strict set of privileges, you can split the access tokens:

- Create one with *Zone / Zone / Read* permissions and scope it to all your zones. This is needed to resolve domain names to Zone IDs and can be shared among multiple Lego installations. Pass this API token as `CF_ZONE_API_TOKEN` to Lego.
- Create another API token with *Zone / DNS / Edit* permissions and set the scope to the domains you want to manage with a single Lego installation. Pass this token as `CF_DNS_API_TOKEN` to Lego.
- Repeat the previous step for each host you want to run Lego on.

This "paranoid" setup is mainly interesting for users who manage many zones/domains with a single Cloudflare account. It follows the principle of least privilege and limits the possible damage, should one of the hosts become compromised.

NOTE: The following documentation is auto-generated from the ACME provider's API library lego (https://go-acme.github.io/lego/). Some sections may refer to lego directly - in most cases, these sections apply to the Terraform provider as well.

ClouDNS DNS Challenge Provider

The `cloudns` DNS challenge provider can be used to perform DNS challenges for the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource with ClouDNS (https://www.cloudns.net).

For complete information on how to use this provider with the `acme_certifiate` resource, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

Example

```
resource "acme_certificate" "certificate" {
  ...

  dns_challenge {
    provider = "cloudns"
  }
}
```

Argument Reference

The following arguments can be either passed as environment variables, or directly through the `config` block in the `dns_challenge` (/docs/providers/acme/r/certificate.html#dns_challenge) argument in the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource. For more details, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

In addition, arguments can also be stored in a local file, with the path supplied by supplying the argument with the `_FILE` suffix. See here (/docs/providers/acme/r/certificate.html#using-variable-files-for-provider-arguments) for more information.

- `CLOUDNS_AUTH_ID` - The API user ID.
- `CLOUDNS_AUTH_PASSWORD` - The password for API user ID.
- `CLOUDNS_HTTP_TIMEOUT` - API request timeout.
- `CLOUDNS_POLLING_INTERVAL` - Time between DNS propagation check.
- `CLOUDNS_PROPAGATION_TIMEOUT` - Maximum waiting time for DNS propagation.
- `CLOUDNS_TTL` - The TTL of the TXT record used for the DNS challenge.

NOTE: The following documentation is auto-generated from the ACME provider's API library lego (https://go-acme.github.io/lego/). Some sections may refer to lego directly - in most cases, these sections apply to the Terraform provider as well.

CloudXNS DNS Challenge Provider

The `cloudxns` DNS challenge provider can be used to perform DNS challenges for the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource with CloudXNS (https://www.cloudxns.net/).

For complete information on how to use this provider with the `acme_certifiate` resource, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

Example

```
resource "acme_certificate" "certificate" {
  ...

  dns_challenge {
    provider = "cloudxns"
  }
}
```

Argument Reference

The following arguments can be either passed as environment variables, or directly through the `config` block in the `dns_challenge` (/docs/providers/acme/r/certificate.html#dns_challenge) argument in the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource. For more details, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

In addition, arguments can also be stored in a local file, with the path supplied by supplying the argument with the `_FILE` suffix. See here (/docs/providers/acme/r/certificate.html#using-variable-files-for-provider-arguments) for more information.

- `CLOUDXNS_API_KEY` - The API key.
- `CLOUDXNS_SECRET_KEY` - THe API secret key.
- `CLOUDXNS_HTTP_TIMEOUT` - API request timeout.
- `CLOUDXNS_POLLING_INTERVAL` - Time between DNS propagation check.
- `CLOUDXNS_PROPAGATION_TIMEOUT` - Maximum waiting time for DNS propagation.
- `CLOUDXNS_TTL` - The TTL of the TXT record used for the DNS challenge.

NOTE: The following documentation is auto-generated from the ACME provider's API library lego (https://go-acme.github.io/lego/). Some sections may refer to lego directly - in most cases, these sections apply to the Terraform provider as well.

ConoHa DNS Challenge Provider

The `conoha` DNS challenge provider can be used to perform DNS challenges for the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource with ConoHa (https://www.conoha.jp/).

For complete information on how to use this provider with the `acme_certifiate` resource, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

Example

```
resource "acme_certificate" "certificate" {
  ...

  dns_challenge {
    provider = "conoha"
  }
}
```

Argument Reference

The following arguments can be either passed as environment variables, or directly through the `config` block in the `dns_challenge` (/docs/providers/acme/r/certificate.html#dns_challenge) argument in the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource. For more details, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

In addition, arguments can also be stored in a local file, with the path supplied by supplying the argument with the `_FILE` suffix. See here (/docs/providers/acme/r/certificate.html#using-variable-files-for-provider-arguments) for more information.

- `CONOHA_API_PASSWORD` - The API password.
- `CONOHA_API_USERNAME` - The API username.
- `CONOHA_TENANT_ID` - Tenant ID.
- `CONOHA_HTTP_TIMEOUT` - API request timeout.
- `CONOHA_POLLING_INTERVAL` - Time between DNS propagation check.
- `CONOHA_PROPAGATION_TIMEOUT` - Maximum waiting time for DNS propagation.
- `CONOHA_REGION` - The region.
- `CONOHA_TTL` - The TTL of the TXT record used for the DNS challenge.

NOTE: The following documentation is auto-generated from the ACME provider's API library lego (https://go-acme.github.io/lego/). Some sections may refer to lego directly - in most cases, these sections apply to the Terraform provider as well.

Designate DNSaaS for Openstack DNS Challenge Provider

The `designate` DNS challenge provider can be used to perform DNS challenges for the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource with Designate DNSaaS for Openstack (https://docs.openstack.org/designate/latest/).

For complete information on how to use this provider with the `acme_certifiate` resource, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

Example

```
resource "acme_certificate" "certificate" {
  ...

  dns_challenge {
    provider = "designate"
  }
}
```

Argument Reference

The following arguments can be either passed as environment variables, or directly through the `config` block in the `dns_challenge` (/docs/providers/acme/r/certificate.html#dns_challenge) argument in the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource. For more details, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

In addition, arguments can also be stored in a local file, with the path supplied by supplying the argument with the `_FILE` suffix. See here (/docs/providers/acme/r/certificate.html#using-variable-files-for-provider-arguments) for more information.

- `OS_AUTH_URL` - Identity endpoint URL.
- `OS_PASSWORD` - Password.
- `OS_PROJECT_NAME` - Project name.
- `OS_REGION_NAME` - Region name.
- `OS_TENANT_NAME` - Tenant name (deprecated see OS_PROJECT_NAME and OS_PROJECT_ID).
- `OS_USERNAME` - Username.
- `DESIGNATE_POLLING_INTERVAL` - Time between DNS propagation check.

- `DESIGNATE_PROPAGATION_TIMEOUT` - Maximum waiting time for DNS propagation.
- `DESIGNATE_TTL` - The TTL of the TXT record used for the DNS challenge.
- `OS_PROJECT_ID` - Project ID.

NOTE: The following documentation is auto-generated from the ACME provider's API library lego (https://go-acme.github.io/lego/). Some sections may refer to lego directly - in most cases, these sections apply to the Terraform provider as well.

Digital Ocean DNS Challenge Provider

The `digitalocean` DNS challenge provider can be used to perform DNS challenges for the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource with Digital Ocean (https://www.digitalocean.com/docs/networking/dns/).

For complete information on how to use this provider with the `acme_certifiate` resource, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

Example

```
resource "acme_certificate" "certificate" {
  ...

  dns_challenge {
    provider = "digitalocean"
  }
}
```

Argument Reference

The following arguments can be either passed as environment variables, or directly through the `config` block in the `dns_challenge` (/docs/providers/acme/r/certificate.html#dns_challenge) argument in the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource. For more details, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

In addition, arguments can also be stored in a local file, with the path supplied by supplying the argument with the `_FILE` suffix. See here (/docs/providers/acme/r/certificate.html#using-variable-files-for-provider-arguments) for more information.

- `DO_AUTH_TOKEN` - Authentication token.
- `DO_HTTP_TIMEOUT` - API request timeout.
- `DO_POLLING_INTERVAL` - Time between DNS propagation check.
- `DO_PROPAGATION_TIMEOUT` - Maximum waiting time for DNS propagation.
- `DO_TTL` - The TTL of the TXT record used for the DNS challenge.

NOTE: The following documentation is auto-generated from the ACME provider's API library lego (https://go-acme.github.io/lego/). Some sections may refer to lego directly - in most cases, these sections apply to the Terraform provider as well.

DNSimple DNS Challenge Provider

The `dnsimple` DNS challenge provider can be used to perform DNS challenges for the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource with DNSimple (https://dnsimple.com/).

For complete information on how to use this provider with the `acme_certifiate` resource, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

Example

```
resource "acme_certificate" "certificate" {
  ...

  dns_challenge {
    provider = "dnsimple"
  }
}
```

Argument Reference

The following arguments can be either passed as environment variables, or directly through the `config` block in the `dns_challenge` (/docs/providers/acme/r/certificate.html#dns_challenge) argument in the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource. For more details, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

In addition, arguments can also be stored in a local file, with the path supplied by supplying the argument with the `_FILE` suffix. See here (/docs/providers/acme/r/certificate.html#using-variable-files-for-provider-arguments) for more information.

- `DNSIMPLE_BASE_URL` - API endpoint URL.
- `DNSIMPLE_OAUTH_TOKEN` - OAuth token.
- `DNSIMPLE_POLLING_INTERVAL` - Time between DNS propagation check.
- `DNSIMPLE_PROPAGATION_TIMEOUT` - Maximum waiting time for DNS propagation.
- `DNSIMPLE_TTL` - The TTL of the TXT record used for the DNS challenge.

NOTE: The following documentation is auto-generated from the ACME provider's API library lego (https://go-acme.github.io/lego/). Some sections may refer to lego directly - in most cases, these sections apply to the Terraform provider as well.

DNS Made Easy DNS Challenge Provider

The `dnsmadeeasy` DNS challenge provider can be used to perform DNS challenges for the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource with DNS Made Easy (https://dnsmadeeasy.com/).

For complete information on how to use this provider with the `acme_certifiate` resource, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

Example

```
resource "acme_certificate" "certificate" {
  ...

  dns_challenge {
    provider = "dnsmadeeasy"
  }
}
```

Argument Reference

The following arguments can be either passed as environment variables, or directly through the `config` block in the `dns_challenge` (/docs/providers/acme/r/certificate.html#dns_challenge) argument in the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource. For more details, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

In addition, arguments can also be stored in a local file, with the path supplied by supplying the argument with the `_FILE` suffix. See here (/docs/providers/acme/r/certificate.html#using-variable-files-for-provider-arguments) for more information.

- `DNSMADEEASY_API_KEY` - The API key.
- `DNSMADEEASY_API_SECRET` - The API Secret key.
- `DNSMADEEASY_HTTP_TIMEOUT` - API request timeout.
- `DNSMADEEASY_POLLING_INTERVAL` - Time between DNS propagation check.
- `DNSMADEEASY_PROPAGATION_TIMEOUT` - Maximum waiting time for DNS propagation.
- `DNSMADEEASY_SANDBOX` - Activate the sandbox (boolean).
- `DNSMADEEASY_TTL` - The TTL of the TXT record used for the DNS challenge.

NOTE: The following documentation is auto-generated from the ACME provider's API library lego (https://go-acme.github.io/lego/). Some sections may refer to lego directly - in most cases, these sections apply to the Terraform provider as well.

DNSPod DNS Challenge Provider

The `dnspod` DNS challenge provider can be used to perform DNS challenges for the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource with DNSPod (http://www.dnspod.com/).

For complete information on how to use this provider with the `acme_certifiate` resource, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

Example

```
resource "acme_certificate" "certificate" {
  ...

  dns_challenge {
    provider = "dnspod"
  }
}
```

Argument Reference

The following arguments can be either passed as environment variables, or directly through the `config` block in the `dns_challenge` (/docs/providers/acme/r/certificate.html#dns_challenge) argument in the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource. For more details, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

In addition, arguments can also be stored in a local file, with the path supplied by supplying the argument with the `_FILE` suffix. See here (/docs/providers/acme/r/certificate.html#using-variable-files-for-provider-arguments) for more information.

- `DNSPOD_API_KEY` - The user token.
- `DNSPOD_HTTP_TIMEOUT` - API request timeout.
- `DNSPOD_POLLING_INTERVAL` - Time between DNS propagation check.
- `DNSPOD_PROPAGATION_TIMEOUT` - Maximum waiting time for DNS propagation.
- `DNSPOD_TTL` - The TTL of the TXT record used for the DNS challenge.

NOTE: The following documentation is auto-generated from the ACME provider's API library lego (https://go-acme.github.io/lego/). Some sections may refer to lego directly - in most cases, these sections apply to the Terraform provider as well.

Domain Offensive (do.de) DNS Challenge Provider

The `dode` DNS challenge provider can be used to perform DNS challenges for the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource with Domain Offensive (do.de) (https://www.do.de/).

For complete information on how to use this provider with the `acme_certifiate` resource, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

Example

```
resource "acme_certificate" "certificate" {
  ...

  dns_challenge {
    provider = "dode"
  }
}
```

Argument Reference

The following arguments can be either passed as environment variables, or directly through the `config` block in the `dns_challenge` (/docs/providers/acme/r/certificate.html#dns_challenge) argument in the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource. For more details, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

In addition, arguments can also be stored in a local file, with the path supplied by supplying the argument with the `_FILE` suffix. See here (/docs/providers/acme/r/certificate.html#using-variable-files-for-provider-arguments) for more information.

- `DODE_TOKEN` - API token.
- `DODE_HTTP_TIMEOUT` - API request timeout.
- `DODE_POLLING_INTERVAL` - Time between DNS propagation check.
- `DODE_PROPAGATION_TIMEOUT` - Maximum waiting time for DNS propagation.
- `DODE_SEQUENCE_INTERVAL` - Interval between iteration.
- `DODE_TTL` - The TTL of the TXT record used for the DNS challenge.

NOTE: The following documentation is auto-generated from the ACME provider's API library lego (https://go-acme.github.io/lego/). Some sections may refer to lego directly - in most cases, these sections apply to the Terraform provider as well.

DreamHost DNS Challenge Provider

The `dreamhost` DNS challenge provider can be used to perform DNS challenges for the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource with DreamHost (https://www.dreamhost.com).

For complete information on how to use this provider with the `acme_certifiate` resource, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

Example

```
resource "acme_certificate" "certificate" {
  ...

  dns_challenge {
    provider = "dreamhost"
  }
}
```

Argument Reference

The following arguments can be either passed as environment variables, or directly through the `config` block in the `dns_challenge` (/docs/providers/acme/r/certificate.html#dns_challenge) argument in the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource. For more details, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

In addition, arguments can also be stored in a local file, with the path supplied by supplying the argument with the `_FILE` suffix. See here (/docs/providers/acme/r/certificate.html#using-variable-files-for-provider-arguments) for more information.

- `DREAMHOST_API_KEY` - The API key.
- `DREAMHOST_HTTP_TIMEOUT` - API request timeout.
- `DREAMHOST_POLLING_INTERVAL` - Time between DNS propagation check.
- `DREAMHOST_PROPAGATION_TIMEOUT` - Maximum waiting time for DNS propagation.
- `DREAMHOST_TTL` - The TTL of the TXT record used for the DNS challenge.

NOTE: The following documentation is auto-generated from the ACME provider's API library lego (https://go-acme.github.io/lego/). Some sections may refer to lego directly - in most cases, these sections apply to the Terraform provider as well.

Duck DNS DNS Challenge Provider

The `duckdns` DNS challenge provider can be used to perform DNS challenges for the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource with Duck DNS (https://www.duckdns.org/).

For complete information on how to use this provider with the `acme_certifiate` resource, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

Example

```
resource "acme_certificate" "certificate" {
  ...

  dns_challenge {
    provider = "duckdns"
  }
}
```

Argument Reference

The following arguments can be either passed as environment variables, or directly through the `config` block in the `dns_challenge` (/docs/providers/acme/r/certificate.html#dns_challenge) argument in the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource. For more details, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

In addition, arguments can also be stored in a local file, with the path supplied by supplying the argument with the `_FILE` suffix. See here (/docs/providers/acme/r/certificate.html#using-variable-files-for-provider-arguments) for more information.

- `DUCKDNS_TOKEN` - Account token.
- `DUCKDNS_HTTP_TIMEOUT` - API request timeout.
- `DUCKDNS_POLLING_INTERVAL` - Time between DNS propagation check.
- `DUCKDNS_PROPAGATION_TIMEOUT` - Maximum waiting time for DNS propagation.
- `DUCKDNS_SEQUENCE_INTERVAL` - Interval between iteration.
- `DUCKDNS_TTL` - The TTL of the TXT record used for the DNS challenge.

NOTE: The following documentation is auto-generated from the ACME provider's API library lego (https://go-acme.github.io/lego/). Some sections may refer to lego directly - in most cases, these sections apply to the Terraform provider as well.

Dyn DNS Challenge Provider

The `dyn` DNS challenge provider can be used to perform DNS challenges for the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource with Dyn (https://dyn.com/).

For complete information on how to use this provider with the `acme_certifiate` resource, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

Example

```
resource "acme_certificate" "certificate" {
  ...

  dns_challenge {
    provider = "dyn"
  }
}
```

Argument Reference

The following arguments can be either passed as environment variables, or directly through the `config` block in the `dns_challenge` (/docs/providers/acme/r/certificate.html#dns_challenge) argument in the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource. For more details, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

In addition, arguments can also be stored in a local file, with the path supplied by supplying the argument with the `_FILE` suffix. See here (/docs/providers/acme/r/certificate.html#using-variable-files-for-provider-arguments) for more information.

- `DYN_CUSTOMER_NAME` - Customer name.
- `DYN_PASSWORD` - Paswword.
- `DYN_USER_NAME` - User name.
- `DYN_HTTP_TIMEOUT` - API request timeout.
- `DYN_POLLING_INTERVAL` - Time between DNS propagation check.
- `DYN_PROPAGATION_TIMEOUT` - Maximum waiting time for DNS propagation.
- `DYN_TTL` - The TTL of the TXT record used for the DNS challenge.

NOTE: The following documentation is auto-generated from the ACME provider's API library lego (https://go-acme.github.io/lego/). Some sections may refer to lego directly - in most cases, these sections apply to the Terraform provider as well.

EasyDNS DNS Challenge Provider

The `easydns` DNS challenge provider can be used to perform DNS challenges for the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource with EasyDNS (https://easydns.com/).

For complete information on how to use this provider with the `acme_certifiate` resource, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

Example

```
resource "acme_certificate" "certificate" {
  ...

  dns_challenge {
    provider = "easydns"
  }
}
```

Argument Reference

The following arguments can be either passed as environment variables, or directly through the `config` block in the `dns_challenge` (/docs/providers/acme/r/certificate.html#dns_challenge) argument in the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource. For more details, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

In addition, arguments can also be stored in a local file, with the path supplied by supplying the argument with the `_FILE` suffix. See here (/docs/providers/acme/r/certificate.html#using-variable-files-for-provider-arguments) for more information.

- `EASYDNS_KEY` - API Key.
- `EASYDNS_TOKEN` - API Token.
- `EASYDNS_ENDPOINT` - The endpoint URL of the API Server.
- `EASYDNS_HTTP_TIMEOUT` - API request timeout.
- `EASYDNS_POLLING_INTERVAL` - Time between DNS propagation check.
- `EASYDNS_PROPAGATION_TIMEOUT` - Maximum waiting time for DNS propagation.
- `EASYDNS_SEQUENCE_INTERVAL` - Time between sequential requests.
- `EASYDNS_TTL` - The TTL of the TXT record used for the DNS challenge.

To test with the sandbox environment set `EASYDNS_ENDPOINT=https://sandbox.rest.easydns.net`

NOTE: The following documentation is auto-generated from the ACME provider's API library lego (https://go-acme.github.io/lego/). Some sections may refer to lego directly - in most cases, these sections apply to the Terraform provider as well.

External program DNS Challenge Provider

The `exec` DNS challenge provider can be used to perform DNS challenges for the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource with External program.

For complete information on how to use this provider with the `acme_certifiate` resource, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

Example

```
resource "acme_certificate" "certificate" {
  ...

  dns_challenge {
    provider = "exec"
  }
}
```

Base Configuration

Environment Variable Name	Description
`EXEC_MODE`	`RAW` , none
`EXEC_PATH`	The path of the the external program.

Additional Configuration

Environment Variable Name	Description
`EXEC_POLLING_INTERVAL`	Time between DNS propagation check.
`EXEC_PROPAGATION_TIMEOUT`	Maximum waiting time for DNS propagation.

Description

The file name of the external program is specified in the environment variable `EXEC_PATH` .

When it is run by lego, three command-line parameters are passed to it: The action ("present" or "cleanup"), the fully-qualified domain name and the value for the record.

For example, requesting a certificate for the domain 'foo.example.com' can be achieved by calling lego as follows:

```
EXEC_PATH=./update-dns.sh \
    lego --dns exec \
    --domains foo.example.com \
    --email invalid@example.com run
```

It will then call the program './update-dns.sh' with like this:

```
./update-dns.sh "present" "_acme-challenge.foo.example.com." "MsijOYZxqyjGnFGwhjrhfg-Xgbl5r68WPda0J9EgqqI
"
```

The program then needs to make sure the record is inserted. When it returns an error via a non-zero exit code, lego aborts.

When the record is to be removed again, the program is called with the first command-line parameter set to `cleanup` instead of `present`.

If you want to use the raw domain, token, and keyAuth values with your program, you can set `EXEC_MODE=RAW`:

```
EXEC_MODE=RAW \
EXEC_PATH=./update-dns.sh \
    lego --dns exec \
    --domains foo.example.com \
    --email invalid@example.com run
```

It will then call the program `./update-dns.sh` like this:

```
./update-dns.sh "present" "foo.example.com." "--" "some-token" "KxAy-J3NwUmg9ZQuM-gP_Mq1nStaYSaP9tYQs5_-Y
sE.ksT-qywTd8058G-SHHWA3RAN72Pr0yWtPYmmY5UBpQ8"
```

Commands

NOTE: The `--` is because the token MAY start with a `-`, and the called program may try and interpret a `-` as indicating a flag. In the case of urfave, which is commonly used, you can use the `--` delimiter to specify the start of positional arguments, and handle such a string safely.

Present

Mode	Command
default	`myprogram present -- <FQDN> <record>`
`RAW`	`myprogram present -- <domain> <token> <key_auth>`

Cleanup

Mode	Command
default	`myprogram cleanup -- <FQDN> <record>`
RAW	`myprogram cleanup -- <domain> <token> <key_auth>`

Timeout

The command have to display propagation timeout and polling interval into Stdout.

The values must be formatted as JSON, and times are in seconds. Example: `{"timeout": 30, "interval": 5}`

If an error occurs or if the command is not provided: the default display propagation timeout and polling interval are used.

Mode	Command
default	`myprogram timeout`
RAW	`myprogram timeout`

NOTE: The following documentation is auto-generated from the ACME provider's API library lego (https://go-acme.github.io/lego/). Some sections may refer to lego directly - in most cases, these sections apply to the Terraform provider as well.

Exoscale DNS Challenge Provider

The `exoscale` DNS challenge provider can be used to perform DNS challenges for the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource with Exoscale (https://www.exoscale.com/).

For complete information on how to use this provider with the `acme_certifiate` resource, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

Example

```
resource "acme_certificate" "certificate" {
  ...

  dns_challenge {
    provider = "exoscale"
  }
}
```

Argument Reference

The following arguments can be either passed as environment variables, or directly through the `config` block in the `dns_challenge` (/docs/providers/acme/r/certificate.html#dns_challenge) argument in the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource. For more details, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

In addition, arguments can also be stored in a local file, with the path supplied by supplying the argument with the `_FILE` suffix. See here (/docs/providers/acme/r/certificate.html#using-variable-files-for-provider-arguments) for more information.

- `EXOSCALE_API_KEY` - API key.
- `EXOSCALE_API_SECRET` - API secret.
- `EXOSCALE_ENDPOINT` - API endpoint URL.
- `EXOSCALE_HTTP_TIMEOUT` - API request timeout.
- `EXOSCALE_POLLING_INTERVAL` - Time between DNS propagation check.
- `EXOSCALE_PROPAGATION_TIMEOUT` - Maximum waiting time for DNS propagation.
- `EXOSCALE_TTL` - The TTL of the TXT record used for the DNS challenge.

NOTE: The following documentation is auto-generated from the ACME provider's API library lego (https://go-acme.github.io/lego/). Some sections may refer to lego directly - in most cases, these sections apply to the Terraform provider as well.

FastDNS DNS Challenge Provider

The `fastdns` DNS challenge provider can be used to perform DNS challenges for the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource with FastDNS (https://www.akamai.com/us/en/products/security/fast-dns.jsp).

For complete information on how to use this provider with the `acme_certifiate` resource, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

Example

```
resource "acme_certificate" "certificate" {
  ...

  dns_challenge {
    provider = "fastdns"
  }
}
```

Argument Reference

The following arguments can be either passed as environment variables, or directly through the `config` block in the `dns_challenge` (/docs/providers/acme/r/certificate.html#dns_challenge) argument in the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource. For more details, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

In addition, arguments can also be stored in a local file, with the path supplied by supplying the argument with the `_FILE` suffix. See here (/docs/providers/acme/r/certificate.html#using-variable-files-for-provider-arguments) for more information.

- `AKAMAI_ACCESS_TOKEN` - Access token.
- `AKAMAI_CLIENT_SECRET` - Client secret.
- `AKAMAI_CLIENT_TOKEN` - Client token.
- `AKAMAI_HOST` - API host.
- `AKAMAI_POLLING_INTERVAL` - Time between DNS propagation check.
- `AKAMAI_PROPAGATION_TIMEOUT` - Maximum waiting time for DNS propagation.
- `AKAMAI_TTL` - The TTL of the TXT record used for the DNS challenge.

NOTE: The following documentation is auto-generated from the ACME provider's API library lego (https://go-acme.github.io/lego/). Some sections may refer to lego directly - in most cases, these sections apply to the Terraform provider as well.

Gandi DNS Challenge Provider

The `gandi` DNS challenge provider can be used to perform DNS challenges for the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource with Gandi (https://www.gandi.net).

For complete information on how to use this provider with the `acme_certifiate` resource, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

Example

```
resource "acme_certificate" "certificate" {
  ...

  dns_challenge {
    provider = "gandi"
  }
}
```

Argument Reference

The following arguments can be either passed as environment variables, or directly through the `config` block in the `dns_challenge` (/docs/providers/acme/r/certificate.html#dns_challenge) argument in the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource. For more details, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

In addition, arguments can also be stored in a local file, with the path supplied by supplying the argument with the `_FILE` suffix. See here (/docs/providers/acme/r/certificate.html#using-variable-files-for-provider-arguments) for more information.

- `GANDI_API_KEY` - API key.
- `GANDI_HTTP_TIMEOUT` - API request timeout.
- `GANDI_POLLING_INTERVAL` - Time between DNS propagation check.
- `GANDI_PROPAGATION_TIMEOUT` - Maximum waiting time for DNS propagation.
- `GANDI_TTL` - The TTL of the TXT record used for the DNS challenge.

NOTE: The following documentation is auto-generated from the ACME provider's API library lego (https://go-acme.github.io/lego/). Some sections may refer to lego directly - in most cases, these sections apply to the Terraform provider as well.

Gandi Live DNS (v5) DNS Challenge Provider

The `gandiv5` DNS challenge provider can be used to perform DNS challenges for the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource with Gandi Live DNS (v5) (https://www.gandi.net).

For complete information on how to use this provider with the `acme_certifiate` resource, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

Example

```
resource "acme_certificate" "certificate" {
  ...

  dns_challenge {
    provider = "gandiv5"
  }
}
```

Argument Reference

The following arguments can be either passed as environment variables, or directly through the `config` block in the `dns_challenge` (/docs/providers/acme/r/certificate.html#dns_challenge) argument in the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource. For more details, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

In addition, arguments can also be stored in a local file, with the path supplied by supplying the argument with the `_FILE` suffix. See here (/docs/providers/acme/r/certificate.html#using-variable-files-for-provider-arguments) for more information.

- `GANDIV5_API_KEY` - API key.
- `GANDIV5_HTTP_TIMEOUT` - API request timeout.
- `GANDIV5_POLLING_INTERVAL` - Time between DNS propagation check.
- `GANDIV5_PROPAGATION_TIMEOUT` - Maximum waiting time for DNS propagation.
- `GANDIV5_TTL` - The TTL of the TXT record used for the DNS challenge.

NOTE: The following documentation is auto-generated from the ACME provider's API library lego (https://go-acme.github.io/lego/). Some sections may refer to lego directly - in most cases, these sections apply to the Terraform provider as well.

Google Cloud DNS Challenge Provider

The `gcloud` DNS challenge provider can be used to perform DNS challenges for the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource with Google Cloud (https://cloud.google.com).

For complete information on how to use this provider with the `acme_certifiate` resource, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

Example

```
resource "acme_certificate" "certificate" {
  ...

  dns_challenge {
    provider = "gcloud"
  }
}
```

Argument Reference

The following arguments can be either passed as environment variables, or directly through the `config` block in the `dns_challenge` (/docs/providers/acme/r/certificate.html#dns_challenge) argument in the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource. For more details, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

In addition, arguments can also be stored in a local file, with the path supplied by supplying the argument with the `_FILE` suffix. See here (/docs/providers/acme/r/certificate.html#using-variable-files-for-provider-arguments) for more information.

- `Application Default Credentials` - Documentation (https://cloud.google.com/docs/authentication/production#providing_credentials_to_your_application).
- `GCE_PROJECT` - Project name.
- `GCE_SERVICE_ACCOUNT` - Account.
- `GCE_SERVICE_ACCOUNT_FILE` - Account file path.
- `GCE_POLLING_INTERVAL` - Time between DNS propagation check.
- `GCE_PROPAGATION_TIMEOUT` - Maximum waiting time for DNS propagation.
- `GCE_TTL` - The TTL of the TXT record used for the DNS challenge.

NOTE: The following documentation is auto-generated from the ACME provider's API library lego (https://go-acme.github.io/lego/). Some sections may refer to lego directly - in most cases, these sections apply to the Terraform provider as well.

Glesys DNS Challenge Provider

The `glesys` DNS challenge provider can be used to perform DNS challenges for the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource with Glesys (https://glesys.com/).

For complete information on how to use this provider with the `acme_certifiate` resource, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

Example

```
resource "acme_certificate" "certificate" {
  ...

  dns_challenge {
    provider = "glesys"
  }
}
```

Argument Reference

The following arguments can be either passed as environment variables, or directly through the `config` block in the `dns_challenge` (/docs/providers/acme/r/certificate.html#dns_challenge) argument in the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource. For more details, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

In addition, arguments can also be stored in a local file, with the path supplied by supplying the argument with the `_FILE` suffix. See here (/docs/providers/acme/r/certificate.html#using-variable-files-for-provider-arguments) for more information.

- `GLESYS_API_KEY` - API key.
- `GLESYS_API_USER` - API user.
- `GLESYS_HTTP_TIMEOUT` - API request timeout.
- `GLESYS_POLLING_INTERVAL` - Time between DNS propagation check.
- `GLESYS_PROPAGATION_TIMEOUT` - Maximum waiting time for DNS propagation.
- `GLESYS_TTL` - The TTL of the TXT record used for the DNS challenge.

NOTE: The following documentation is auto-generated from the ACME provider's API library lego (https://go-acme.github.io/lego/). Some sections may refer to lego directly - in most cases, these sections apply to the Terraform provider as well.

Go Daddy DNS Challenge Provider

The `godaddy` DNS challenge provider can be used to perform DNS challenges for the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource with Go Daddy (https://godaddy.com).

For complete information on how to use this provider with the `acme_certifiate` resource, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

Example

```
resource "acme_certificate" "certificate" {
  ...

  dns_challenge {
    provider = "godaddy"
  }
}
```

Argument Reference

The following arguments can be either passed as environment variables, or directly through the `config` block in the `dns_challenge` (/docs/providers/acme/r/certificate.html#dns_challenge) argument in the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource. For more details, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

In addition, arguments can also be stored in a local file, with the path supplied by supplying the argument with the `_FILE` suffix. See here (/docs/providers/acme/r/certificate.html#using-variable-files-for-provider-arguments) for more information.

- `GODADDY_API_KEY` - API key.
- `GODADDY_API_SECRET` - API secret.
- `GODADDY_HTTP_TIMEOUT` - API request timeout.
- `GODADDY_POLLING_INTERVAL` - Time between DNS propagation check.
- `GODADDY_PROPAGATION_TIMEOUT` - Maximum waiting time for DNS propagation.
- `GODADDY_SEQUENCE_INTERVAL` - Interval between iteration.
- `GODADDY_TTL` - The TTL of the TXT record used for the DNS challenge.

NOTE: The following documentation is auto-generated from the ACME provider's API library lego (https://go-acme.github.io/lego/). Some sections may refer to lego directly - in most cases, these sections apply to the Terraform provider as well.

Hosting.de DNS Challenge Provider

The `hostingde` DNS challenge provider can be used to perform DNS challenges for the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource with Hosting.de (https://www.hosting.de/).

For complete information on how to use this provider with the `acme_certifiate` resource, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

Example

```
resource "acme_certificate" "certificate" {
  ...

  dns_challenge {
    provider = "hostingde"
  }
}
```

Argument Reference

The following arguments can be either passed as environment variables, or directly through the `config` block in the `dns_challenge` (/docs/providers/acme/r/certificate.html#dns_challenge) argument in the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource. For more details, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

In addition, arguments can also be stored in a local file, with the path supplied by supplying the argument with the `_FILE` suffix. See here (/docs/providers/acme/r/certificate.html#using-variable-files-for-provider-arguments) for more information.

- `HOSTINGDE_API_KEY` - API key.
- `HOSTINGDE_ZONE_NAME` - Zone name in ACE format.
- `HOSTINGDE_HTTP_TIMEOUT` - API request timeout.
- `HOSTINGDE_POLLING_INTERVAL` - Time between DNS propagation check.
- `HOSTINGDE_PROPAGATION_TIMEOUT` - Maximum waiting time for DNS propagation.
- `HOSTINGDE_TTL` - The TTL of the TXT record used for the DNS challenge.

NOTE: The following documentation is auto-generated from the ACME provider's API library lego (https://go-acme.github.io/lego/). Some sections may refer to lego directly - in most cases, these sections apply to the Terraform provider as well.

HTTP request DNS Challenge Provider

The `httpreq` DNS challenge provider can be used to perform DNS challenges for the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource with HTTP request.

For complete information on how to use this provider with the `acme_certifiate` resource, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

Example

```
resource "acme_certificate" "certificate" {
  ...

  dns_challenge {
    provider = "httpreq"
  }
}
```

Argument Reference

The following arguments can be either passed as environment variables, or directly through the `config` block in the `dns_challenge` (/docs/providers/acme/r/certificate.html#dns_challenge) argument in the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource. For more details, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

In addition, arguments can also be stored in a local file, with the path supplied by supplying the argument with the `_FILE` suffix. See here (/docs/providers/acme/r/certificate.html#using-variable-files-for-provider-arguments) for more information.

- `HTTPREQ_ENDPOINT` - The URL of the server.
- `HTTPREQ_MODE` - `RAW`, none.
- `HTTPREQ_HTTP_TIMEOUT` - API request timeout.
- `HTTPREQ_PASSWORD` - Basic authentication password.
- `HTTPREQ_POLLING_INTERVAL` - Time between DNS propagation check.
- `HTTPREQ_PROPAGATION_TIMEOUT` - Maximum waiting time for DNS propagation.
- `HTTPREQ_USERNAME` - Basic authentication username.

Description

The server must provide:

- `POST /present`
- `POST /cleanup`

The URL of the server must be define by `HTTPREQ_ENDPOINT`.

Mode

There are 2 modes (`HTTPREQ_MODE`):

- default mode: `json { "fqdn": "_acme-challenge.domain.", "value": "LHDhK3oGRvkiefQnx7OOczTY5Tic_xZ6HcMOc_gmtoM" }`
- `RAW json { "domain": "domain", "token": "token", "keyAuth": "key" }`

Authentication

Basic authentication (optional) can be set with some environment variables:

- `HTTPREQ_USERNAME` and `HTTPREQ_PASSWORD`
- both values must be set, otherwise basic authentication is not defined.

NOTE: The following documentation is auto-generated from the ACME provider's API library lego (https://go-acme.github.io/lego/). Some sections may refer to lego directly - in most cases, these sections apply to the Terraform provider as well.

Internet Initiative Japan DNS Challenge Provider

The `iij` DNS challenge provider can be used to perform DNS challenges for the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource with Internet Initiative Japan (https://www.iij.ad.jp/en/).

For complete information on how to use this provider with the `acme_certifiate` resource, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

Example

```
resource "acme_certificate" "certificate" {
  ...

  dns_challenge {
    provider = "iij"
  }
}
```

Argument Reference

The following arguments can be either passed as environment variables, or directly through the `config` block in the `dns_challenge` (/docs/providers/acme/r/certificate.html#dns_challenge) argument in the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource. For more details, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

In addition, arguments can also be stored in a local file, with the path supplied by supplying the argument with the `_FILE` suffix. See here (/docs/providers/acme/r/certificate.html#using-variable-files-for-provider-arguments) for more information.

- `IIJ_API_ACCESS_KEY` - API access key.
- `IIJ_API_SECRET_KEY` - API secret key.
- `IIJ_DO_SERVICE_CODE` - DO service code.
- `IIJ_POLLING_INTERVAL` - Time between DNS propagation check.
- `IIJ_PROPAGATION_TIMEOUT` - Maximum waiting time for DNS propagation.
- `IIJ_TTL` - The TTL of the TXT record used for the DNS challenge.

NOTE: The following documentation is auto-generated from the ACME provider's API library lego (https://go-acme.github.io/lego/). Some sections may refer to lego directly - in most cases, these sections apply to the Terraform provider as well.

INWX DNS Challenge Provider

The `inwx` DNS challenge provider can be used to perform DNS challenges for the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource with INWX (https://www.inwx.de/en).

For complete information on how to use this provider with the `acme_certifiate` resource, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

Example

```
resource "acme_certificate" "certificate" {
  ...

  dns_challenge {
    provider = "inwx"
  }
}
```

Argument Reference

The following arguments can be either passed as environment variables, or directly through the `config` block in the `dns_challenge` (/docs/providers/acme/r/certificate.html#dns_challenge) argument in the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource. For more details, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

In addition, arguments can also be stored in a local file, with the path supplied by supplying the argument with the `_FILE` suffix. See here (/docs/providers/acme/r/certificate.html#using-variable-files-for-provider-arguments) for more information.

- `INWX_PASSWORD` - Password.
- `INWX_USERNAME` - Username.
- `INWX_POLLING_INTERVAL` - Time between DNS propagation check.
- `INWX_PROPAGATION_TIMEOUT` - Maximum waiting time for DNS propagation.
- `INWX_SANDBOX` - Activate the sandbox (boolean).
- `INWX_TTL` - The TTL of the TXT record used for the DNS challenge.

NOTE: The following documentation is auto-generated from the ACME provider's API library lego (https://go-acme.github.io/lego/). Some sections may refer to lego directly - in most cases, these sections apply to the Terraform provider as well.

Joker DNS Challenge Provider

The `joker` DNS challenge provider can be used to perform DNS challenges for the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource with Joker (https://joker.com).

For complete information on how to use this provider with the `acme_certifiate` resource, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

Example

```
resource "acme_certificate" "certificate" {
  ...

  dns_challenge {
    provider = "joker"
  }
}
```

Argument Reference

The following arguments can be either passed as environment variables, or directly through the `config` block in the `dns_challenge` (/docs/providers/acme/r/certificate.html#dns_challenge) argument in the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource. For more details, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

In addition, arguments can also be stored in a local file, with the path supplied by supplying the argument with the `_FILE` suffix. See here (/docs/providers/acme/r/certificate.html#using-variable-files-for-provider-arguments) for more information.

- `JOKER_API_KEY` - API key.
- `JOKER_PASSWORD` - Joker.com password.
- `JOKER_USERNAME` - Joker.com username (email address).
- `JOKER_HTTP_TIMEOUT` - API request timeout.
- `JOKER_POLLING_INTERVAL` - Time between DNS propagation check.
- `JOKER_PROPAGATION_TIMEOUT` - Maximum waiting time for DNS propagation.
- `JOKER_TTL` - The TTL of the TXT record used for the DNS challenge.

NOTE: The following documentation is auto-generated from the ACME provider's API library lego (https://go-acme.github.io/lego/). Some sections may refer to lego directly - in most cases, these sections apply to the Terraform provider as well.

Amazon Lightsail DNS Challenge Provider

The `lightsail` DNS challenge provider can be used to perform DNS challenges for the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource with Amazon Lightsail (https://aws.amazon.com/lightsail/).

For complete information on how to use this provider with the `acme_certifiate` resource, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

Example

```
resource "acme_certificate" "certificate" {
  ...

  dns_challenge {
    provider = "lightsail"
  }
}
```

Argument Reference

The following arguments can be either passed as environment variables, or directly through the `config` block in the `dns_challenge` (/docs/providers/acme/r/certificate.html#dns_challenge) argument in the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource. For more details, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

In addition, arguments can also be stored in a local file, with the path supplied by supplying the argument with the `_FILE` suffix. See here (/docs/providers/acme/r/certificate.html#using-variable-files-for-provider-arguments) for more information.

- `AWS_ACCESS_KEY_ID` - Access key ID.
- `AWS_SECRET_ACCESS_KEY` - Secret access key.
- `DNS_ZONE` - DNS zone.
- `LIGHTSAIL_POLLING_INTERVAL` - Time between DNS propagation check.
- `LIGHTSAIL_PROPAGATION_TIMEOUT` - Maximum waiting time for DNS propagation.

NOTE: The following documentation is auto-generated from the ACME provider's API library lego (https://go-acme.github.io/lego/). Some sections may refer to lego directly - in most cases, these sections apply to the Terraform provider as well.

Linode (deprecated) DNS Challenge Provider

The `linode` DNS challenge provider can be used to perform DNS challenges for the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource with Linode (deprecated) (https://www.linode.com/).

For complete information on how to use this provider with the `acme_certifiate` resource, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

Example

```
resource "acme_certificate" "certificate" {
  ...

  dns_challenge {
    provider = "linode"
  }
}
```

Argument Reference

The following arguments can be either passed as environment variables, or directly through the `config` block in the `dns_challenge` (/docs/providers/acme/r/certificate.html#dns_challenge) argument in the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource. For more details, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

In addition, arguments can also be stored in a local file, with the path supplied by supplying the argument with the `_FILE` suffix. See here (/docs/providers/acme/r/certificate.html#using-variable-files-for-provider-arguments) for more information.

- `LINODE_API_KEY` - API key.
- `LINODE_HTTP_TIMEOUT` - API request timeout.
- `LINODE_POLLING_INTERVAL` - Time between DNS propagation check.
- `LINODE_TTL` - The TTL of the TXT record used for the DNS challenge.

NOTE: The following documentation is auto-generated from the ACME provider's API library lego (https://go-acme.github.io/lego/). Some sections may refer to lego directly - in most cases, these sections apply to the Terraform provider as well.

Linode (v4) DNS Challenge Provider

The `linodev4` DNS challenge provider can be used to perform DNS challenges for the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource with Linode (v4) (https://www.linode.com/).

For complete information on how to use this provider with the `acme_certifiate` resource, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

Example

```
resource "acme_certificate" "certificate" {
  ...

  dns_challenge {
    provider = "linodev4"
  }
}
```

Argument Reference

The following arguments can be either passed as environment variables, or directly through the `config` block in the `dns_challenge` (/docs/providers/acme/r/certificate.html#dns_challenge) argument in the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource. For more details, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

In addition, arguments can also be stored in a local file, with the path supplied by supplying the argument with the `_FILE` suffix. See here (/docs/providers/acme/r/certificate.html#using-variable-files-for-provider-arguments) for more information.

- `LINODE_TOKEN` - API token.
- `LINODE_HTTP_TIMEOUT` - API request timeout.
- `LINODE_POLLING_INTERVAL` - Time between DNS propagation check.
- `LINODE_PROPAGATION_TIMEOUT` - Maximum waiting time for DNS propagation.
- `LINODE_TTL` - The TTL of the TXT record used for the DNS challenge.

NOTE: The following documentation is auto-generated from the ACME provider's API library lego (https://go-acme.github.io/lego/). Some sections may refer to lego directly - in most cases, these sections apply to the Terraform provider as well.

Liquid Web DNS Challenge Provider

The `liquidweb` DNS challenge provider can be used to perform DNS challenges for the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource with Liquid Web (https://cart.liquidweb.com/storm/api/docs/v1/).

For complete information on how to use this provider with the `acme_certifiate` resource, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

Example

```
resource "acme_certificate" "certificate" {
  ...

  dns_challenge {
    provider = "liquidweb"
  }
}
```

Argument Reference

The following arguments can be either passed as environment variables, or directly through the `config` block in the `dns_challenge` (/docs/providers/acme/r/certificate.html#dns_challenge) argument in the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource. For more details, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

In addition, arguments can also be stored in a local file, with the path supplied by supplying the argument with the `_FILE` suffix. See here (/docs/providers/acme/r/certificate.html#using-variable-files-for-provider-arguments) for more information.

- `LIQUID_WEB_PASSWORD` - Storm API Password.
- `LIQUID_WEB_USERNAME` - Storm API Username.
- `LIQUID_WEB_ZONE` - DNS Zone.
- `LIQUID_WEB_HTTP_TIMEOUT` - Maximum waiting time for the DNS records to be created (not verified).
- `LIQUID_WEB_POLLING_INTERVAL` - Time between DNS propagation check.
- `LIQUID_WEB_PROPAGATION_TIMEOUT` - Maximum waiting time for DNS propagation.
- `LIQUID_WEB_TTL` - The TTL of the TXT record used for the DNS challenge.
- `LIQUID_WEB_URL` - Storm API endpoint.

NOTE: The following documentation is auto-generated from the ACME provider's API library lego (https://go-acme.github.io/lego/). Some sections may refer to lego directly - in most cases, these sections apply to the Terraform provider as well.

MyDNS.jp DNS Challenge Provider

The `mydnsjp` DNS challenge provider can be used to perform DNS challenges for the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource with MyDNS.jp (https://www.mydns.jp).

For complete information on how to use this provider with the `acme_certifiate` resource, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

Example

```
resource "acme_certificate" "certificate" {
  ...

  dns_challenge {
    provider = "mydnsjp"
  }
}
```

Argument Reference

The following arguments can be either passed as environment variables, or directly through the `config` block in the `dns_challenge` (/docs/providers/acme/r/certificate.html#dns_challenge) argument in the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource. For more details, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

In addition, arguments can also be stored in a local file, with the path supplied by supplying the argument with the `_FILE` suffix. See here (/docs/providers/acme/r/certificate.html#using-variable-files-for-provider-arguments) for more information.

- `MYDNSJP_MASTER_ID` - Master ID.
- `MYDNSJP_PASSWORD` - Password.
- `MYDNSJP_HTTP_TIMEOUT` - API request timeout.
- `MYDNSJP_POLLING_INTERVAL` - Time between DNS propagation check.
- `MYDNSJP_PROPAGATION_TIMEOUT` - Maximum waiting time for DNS propagation.
- `MYDNSJP_TTL` - The TTL of the TXT record used for the DNS challenge.

NOTE: The following documentation is auto-generated from the ACME provider's API library lego (https://go-acme.github.io/lego/). Some sections may refer to lego directly - in most cases, these sections apply to the Terraform provider as well.

Namecheap DNS Challenge Provider

The `namecheap` DNS challenge provider can be used to perform DNS challenges for the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource with Namecheap (https://www.namecheap.com).

For complete information on how to use this provider with the `acme_certifiate` resource, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

Example

```
resource "acme_certificate" "certificate" {
  ...

  dns_challenge {
    provider = "namecheap"
  }
}
```

Argument Reference

The following arguments can be either passed as environment variables, or directly through the `config` block in the `dns_challenge` (/docs/providers/acme/r/certificate.html#dns_challenge) argument in the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource. For more details, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

In addition, arguments can also be stored in a local file, with the path supplied by supplying the argument with the `_FILE` suffix. See here (/docs/providers/acme/r/certificate.html#using-variable-files-for-provider-arguments) for more information.

- `NAMECHEAP_API_KEY` - API key.
- `NAMECHEAP_API_USER` - API user.
- `NAMECHEAP_HTTP_TIMEOUT` - API request timeout.
- `NAMECHEAP_POLLING_INTERVAL` - Time between DNS propagation check.
- `NAMECHEAP_PROPAGATION_TIMEOUT` - Maximum waiting time for DNS propagation.
- `NAMECHEAP_TTL` - The TTL of the TXT record used for the DNS challenge.

NOTE: The following documentation is auto-generated from the ACME provider's API library lego (https://go-acme.github.io/lego/). Some sections may refer to lego directly - in most cases, these sections apply to the Terraform provider as well.

Name.com DNS Challenge Provider

The `namedotcom` DNS challenge provider can be used to perform DNS challenges for the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource with Name.com (https://www.name.com).

For complete information on how to use this provider with the `acme_certifiate` resource, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

Example

```
resource "acme_certificate" "certificate" {
  ...

  dns_challenge {
    provider = "namedotcom"
  }
}
```

Argument Reference

The following arguments can be either passed as environment variables, or directly through the `config` block in the `dns_challenge` (/docs/providers/acme/r/certificate.html#dns_challenge) argument in the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource. For more details, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

In addition, arguments can also be stored in a local file, with the path supplied by supplying the argument with the `_FILE` suffix. See here (/docs/providers/acme/r/certificate.html#using-variable-files-for-provider-arguments) for more information.

- `NAMECOM_API_TOKEN` - API token.
- `NAMECOM_USERNAME` - Username.
- `NAMECOM_HTTP_TIMEOUT` - API request timeout.
- `NAMECOM_POLLING_INTERVAL` - Time between DNS propagation check.
- `NAMECOM_PROPAGATION_TIMEOUT` - Maximum waiting time for DNS propagation.
- `NAMECOM_TTL` - The TTL of the TXT record used for the DNS challenge.

NOTE: The following documentation is auto-generated from the ACME provider's API library lego (https://go-acme.github.io/lego/). Some sections may refer to lego directly - in most cases, these sections apply to the Terraform provider as well.

Namesilo DNS Challenge Provider

The `namesilo` DNS challenge provider can be used to perform DNS challenges for the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource with Namesilo (https://www.namesilo.com/).

For complete information on how to use this provider with the `acme_certifiate` resource, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

Example

```
resource "acme_certificate" "certificate" {
  ...

  dns_challenge {
    provider = "namesilo"
  }
}
```

Argument Reference

The following arguments can be either passed as environment variables, or directly through the `config` block in the `dns_challenge` (/docs/providers/acme/r/certificate.html#dns_challenge) argument in the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource. For more details, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

In addition, arguments can also be stored in a local file, with the path supplied by supplying the argument with the `_FILE` suffix. See here (/docs/providers/acme/r/certificate.html#using-variable-files-for-provider-arguments) for more information.

- `NAMESILO_API_KEY` - Client ID.
- `NAMESILO_POLLING_INTERVAL` - Time between DNS propagation check.
- `NAMESILO_PROPAGATION_TIMEOUT` - Maximum waiting time for DNS propagation, it is better to set larger than 15m.
- `NAMESILO_TTL` - The TTL of the TXT record used for the DNS challenge, should be in [3600, 2592000].

NOTE: The following documentation is auto-generated from the ACME provider's API library lego (https://go-acme.github.io/lego/). Some sections may refer to lego directly - in most cases, these sections apply to the Terraform provider as well.

Netcup DNS Challenge Provider

The `netcup` DNS challenge provider can be used to perform DNS challenges for the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource with Netcup (https://www.netcup.eu/).

For complete information on how to use this provider with the `acme_certifiate` resource, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

Example

```
resource "acme_certificate" "certificate" {
  ...

  dns_challenge {
    provider = "netcup"
  }
}
```

Argument Reference

The following arguments can be either passed as environment variables, or directly through the `config` block in the `dns_challenge` (/docs/providers/acme/r/certificate.html#dns_challenge) argument in the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource. For more details, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

In addition, arguments can also be stored in a local file, with the path supplied by supplying the argument with the `_FILE` suffix. See here (/docs/providers/acme/r/certificate.html#using-variable-files-for-provider-arguments) for more information.

- `NETCUP_API_KEY` - API key.
- `NETCUP_API_PASSWORD` - API password.
- `NETCUP_CUSTOMER_NUMBER` - Customer number.
- `NETCUP_HTTP_TIMEOUT` - API request timeout.
- `NETCUP_POLLING_INTERVAL` - Time between DNS propagation check.
- `NETCUP_PROPAGATION_TIMEOUT` - Maximum waiting time for DNS propagation.
- `NETCUP_TTL` - The TTL of the TXT record used for the DNS challenge.

NOTE: The following documentation is auto-generated from the ACME provider's API library lego (https://go-acme.github.io/lego/). Some sections may refer to lego directly - in most cases, these sections apply to the Terraform provider as well.

NIFCloud DNS Challenge Provider

The `nifcloud` DNS challenge provider can be used to perform DNS challenges for the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource with NIFCloud (https://www.nifcloud.com/).

For complete information on how to use this provider with the `acme_certifiate` resource, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

Example

```
resource "acme_certificate" "certificate" {
  ...

  dns_challenge {
    provider = "nifcloud"
  }
}
```

Argument Reference

The following arguments can be either passed as environment variables, or directly through the `config` block in the `dns_challenge` (/docs/providers/acme/r/certificate.html#dns_challenge) argument in the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource. For more details, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

In addition, arguments can also be stored in a local file, with the path supplied by supplying the argument with the `_FILE` suffix. See here (/docs/providers/acme/r/certificate.html#using-variable-files-for-provider-arguments) for more information.

- `NIFCLOUD_ACCESS_KEY_ID` - Access key.
- `NIFCLOUD_SECRET_ACCESS_KEY` - Secret access key.
- `NIFCLOUD_HTTP_TIMEOUT` - API request timeout.
- `NIFCLOUD_POLLING_INTERVAL` - Time between DNS propagation check.
- `NIFCLOUD_PROPAGATION_TIMEOUT` - Maximum waiting time for DNS propagation.
- `NIFCLOUD_TTL` - The TTL of the TXT record used for the DNS challenge.

NOTE: The following documentation is auto-generated from the ACME provider's API library lego (https://go-acme.github.io/lego/). Some sections may refer to lego directly - in most cases, these sections apply to the Terraform provider as well.

NS1 DNS Challenge Provider

The `ns1` DNS challenge provider can be used to perform DNS challenges for the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource with NS1 (https://ns1.com).

For complete information on how to use this provider with the `acme_certifiate` resource, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

Example

```
resource "acme_certificate" "certificate" {
  ...

  dns_challenge {
    provider = "ns1"
  }
}
```

Argument Reference

The following arguments can be either passed as environment variables, or directly through the `config` block in the `dns_challenge` (/docs/providers/acme/r/certificate.html#dns_challenge) argument in the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource. For more details, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

In addition, arguments can also be stored in a local file, with the path supplied by supplying the argument with the `_FILE` suffix. See here (/docs/providers/acme/r/certificate.html#using-variable-files-for-provider-arguments) for more information.

- `NS1_API_KEY` - API key.
- `NS1_HTTP_TIMEOUT` - API request timeout.
- `NS1_POLLING_INTERVAL` - Time between DNS propagation check.
- `NS1_PROPAGATION_TIMEOUT` - Maximum waiting time for DNS propagation.
- `NS1_TTL` - The TTL of the TXT record used for the DNS challenge.

NOTE: The following documentation is auto-generated from the ACME provider's API library lego (https://go-acme.github.io/lego/). Some sections may refer to lego directly - in most cases, these sections apply to the Terraform provider as well.

Oracle Cloud DNS Challenge Provider

The `oraclecloud` DNS challenge provider can be used to perform DNS challenges for the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource with Oracle Cloud (https://cloud.oracle.com/home).

For complete information on how to use this provider with the `acme_certifiate` resource, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

Example

```
resource "acme_certificate" "certificate" {
  ...

  dns_challenge {
    provider = "oraclecloud"
  }
}
```

Argument Reference

The following arguments can be either passed as environment variables, or directly through the `config` block in the `dns_challenge` (/docs/providers/acme/r/certificate.html#dns_challenge) argument in the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource. For more details, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

In addition, arguments can also be stored in a local file, with the path supplied by supplying the argument with the `_FILE` suffix. See here (/docs/providers/acme/r/certificate.html#using-variable-files-for-provider-arguments) for more information.

- `OCI_COMPARTMENT_OCID` - Compartment OCID.
- `OCI_PRIVKEY_FILE` - Private key file.
- `OCI_PRIVKEY_PASS` - Private key password.
- `OCI_PUBKEY_FINGERPRINT` - Public key fingerprint.
- `OCI_REGION` - Region.
- `OCI_TENANCY_OCID` - Tenanct OCID.
- `OCI_USER_OCID` - User OCID.
- `OCI_POLLING_INTERVAL` - Time between DNS propagation check.

- `OCI_PROPAGATION_TIMEOUT` - Maximum waiting time for DNS propagation.
- `OCI_TTL` - The TTL of the TXT record used for the DNS challenge.

NOTE: The following documentation is auto-generated from the ACME provider's API library lego (https://go-acme.github.io/lego/). Some sections may refer to lego directly - in most cases, these sections apply to the Terraform provider as well.

Open Telekom Cloud DNS Challenge Provider

The `otc` DNS challenge provider can be used to perform DNS challenges for the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource with Open Telekom Cloud (https://cloud.telekom.de/en).

For complete information on how to use this provider with the `acme_certifiate` resource, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

Example

```
resource "acme_certificate" "certificate" {
  ...

  dns_challenge {
    provider = "otc"
  }
}
```

Argument Reference

The following arguments can be either passed as environment variables, or directly through the `config` block in the `dns_challenge` (/docs/providers/acme/r/certificate.html#dns_challenge) argument in the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource. For more details, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

In addition, arguments can also be stored in a local file, with the path supplied by supplying the argument with the `_FILE` suffix. See here (/docs/providers/acme/r/certificate.html#using-variable-files-for-provider-arguments) for more information.

- `OTC_DOMAIN_NAME` - Domain name.
- `OTC_IDENTITY_ENDPOINT` - Identity endpoint URL.
- `OTC_PASSWORD` - Password.
- `OTC_PROJECT_NAME` - Project name.
- `OTC_USER_NAME` - User name.
- `OTC_HTTP_TIMEOUT` - API request timeout.
- `OTC_POLLING_INTERVAL` - Time between DNS propagation check.
- `OTC_PROPAGATION_TIMEOUT` - Maximum waiting time for DNS propagation.

- `0TC_TTL` - The TTL of the TXT record used for the DNS challenge.

NOTE: The following documentation is auto-generated from the ACME provider's API library lego (https://go-acme.github.io/lego/). Some sections may refer to lego directly - in most cases, these sections apply to the Terraform provider as well.

OVH DNS Challenge Provider

The `ovh` DNS challenge provider can be used to perform DNS challenges for the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource with OVH (https://www.ovh.com/).

For complete information on how to use this provider with the `acme_certifiate` resource, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

Example

```
resource "acme_certificate" "certificate" {
  ...

  dns_challenge {
    provider = "ovh"
  }
}
```

Argument Reference

The following arguments can be either passed as environment variables, or directly through the `config` block in the `dns_challenge` (/docs/providers/acme/r/certificate.html#dns_challenge) argument in the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource. For more details, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

In addition, arguments can also be stored in a local file, with the path supplied by supplying the argument with the `_FILE` suffix. See here (/docs/providers/acme/r/certificate.html#using-variable-files-for-provider-arguments) for more information.

- `OVH_APPLICATION_KEY` - Application key.
- `OVH_APPLICATION_SECRET` - Application secret.
- `OVH_CONSUMER_KEY` - Consumer key.
- `OVH_ENDPOINT` - Endpoint URL (ovh-eu or ovh-ca).
- `OVH_HTTP_TIMEOUT` - API request timeout.
- `OVH_POLLING_INTERVAL` - Time between DNS propagation check.
- `OVH_PROPAGATION_TIMEOUT` - Maximum waiting time for DNS propagation.
- `OVH_TTL` - The TTL of the TXT record used for the DNS challenge.

NOTE: The following documentation is auto-generated from the ACME provider's API library lego (https://go-acme.github.io/lego/). Some sections may refer to lego directly - in most cases, these sections apply to the Terraform provider as well.

PowerDNS DNS Challenge Provider

The `pdns` DNS challenge provider can be used to perform DNS challenges for the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource with PowerDNS (https://www.powerdns.com/).

For complete information on how to use this provider with the `acme_certifiate` resource, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

Example

```
resource "acme_certificate" "certificate" {
  ...

  dns_challenge {
    provider = "pdns"
  }
}
```

Argument Reference

The following arguments can be either passed as environment variables, or directly through the `config` block in the `dns_challenge` (/docs/providers/acme/r/certificate.html#dns_challenge) argument in the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource. For more details, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

In addition, arguments can also be stored in a local file, with the path supplied by supplying the argument with the `_FILE` suffix. See here (/docs/providers/acme/r/certificate.html#using-variable-files-for-provider-arguments) for more information.

- `PDNS_API_KEY` - API key.
- `PDNS_API_URL` - API url.
- `PDNS_HTTP_TIMEOUT` - API request timeout.
- `PDNS_POLLING_INTERVAL` - Time between DNS propagation check.
- `PDNS_PROPAGATION_TIMEOUT` - Maximum waiting time for DNS propagation.
- `PDNS_TTL` - The TTL of the TXT record used for the DNS challenge.

Information

Tested and confirmed to work with PowerDNS authoritative server 3.4.8 and 4.0.1. Refer to PowerDNS documentation (https://doc.powerdns.com/md/httpapi/README/) instructions on how to enable the built-in API interface.

PowerDNS Notes: - PowerDNS API does not currently support SSL, therefore you should take care to ensure that traffic between lego and the PowerDNS API is over a trusted network, VPN etc. - In order to have the SOA serial automatically increment each time the `_acme-challenge` record is added/modified via the API, set `SOA-EDIT-API` to `INCEPTION-INCREMENT` for the zone in the `domainmetadata` table

NOTE: The following documentation is auto-generated from the ACME provider's API library lego (https://go-acme.github.io/lego/). Some sections may refer to lego directly - in most cases, these sections apply to the Terraform provider as well.

Rackspace DNS Challenge Provider

The `rackspace` DNS challenge provider can be used to perform DNS challenges for the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource with Rackspace (https://www.rackspace.com/).

For complete information on how to use this provider with the `acme_certifiate` resource, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

Example

```
resource "acme_certificate" "certificate" {
  ...

  dns_challenge {
    provider = "rackspace"
  }
}
```

Argument Reference

The following arguments can be either passed as environment variables, or directly through the `config` block in the `dns_challenge` (/docs/providers/acme/r/certificate.html#dns_challenge) argument in the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource. For more details, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

In addition, arguments can also be stored in a local file, with the path supplied by supplying the argument with the `_FILE` suffix. See here (/docs/providers/acme/r/certificate.html#using-variable-files-for-provider-arguments) for more information.

- `RACKSPACE_API_KEY` - API key.
- `RACKSPACE_USER` - API user.
- `RACKSPACE_HTTP_TIMEOUT` - API request timeout.
- `RACKSPACE_POLLING_INTERVAL` - Time between DNS propagation check.
- `RACKSPACE_PROPAGATION_TIMEOUT` - Maximum waiting time for DNS propagation.
- `RACKSPACE_TTL` - The TTL of the TXT record used for the DNS challenge.

NOTE: The following documentation is auto-generated from the ACME provider's API library lego (https://go-acme.github.io/lego/). Some sections may refer to lego directly - in most cases, these sections apply to the Terraform provider as well.

RFC2136 DNS Challenge Provider

The `rfc2136` DNS challenge provider can be used to perform DNS challenges for the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource with RFC2136 (https://tools.ietf.org/html/rfc2136).

For complete information on how to use this provider with the `acme_certifiate` resource, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

Example

```
resource "acme_certificate" "certificate" {
  ...

  dns_challenge {
    provider = "rfc2136"
  }
}
```

Argument Reference

The following arguments can be either passed as environment variables, or directly through the `config` block in the `dns_challenge` (/docs/providers/acme/r/certificate.html#dns_challenge) argument in the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource. For more details, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

In addition, arguments can also be stored in a local file, with the path supplied by supplying the argument with the `_FILE` suffix. See here (/docs/providers/acme/r/certificate.html#using-variable-files-for-provider-arguments) for more information.

- `RFC2136_NAMESERVER` - Network address in the form "host" or "host:port".
- `RFC2136_TSIG_ALGORITHM` - TSIG algorythm. See miekg/dns#tsig.go (https://github.com/miekg/dns/blob/master/tsig.go) for supported values. To disable TSIG authentication, leave the `RFC2136_TSIG*` variables unset..
- `RFC2136_TSIG_KEY` - Name of the secret key as defined in DNS server configuration. To disable TSIG authentication, leave the `RFC2136_TSIG*` variables unset..
- `RFC2136_TSIG_SECRET` - Secret key payload. To disable TSIG authentication, leave the `RFC2136_TSIG*` variables unset..
- `RFC2136_DNS_TIMEOUT` - API request timeout.

- `RFC2136_POLLING_INTERVAL` - Time between DNS propagation check.
- `RFC2136_PROPAGATION_TIMEOUT` - Maximum waiting time for DNS propagation.
- `RFC2136_SEQUENCE_INTERVAL` - Interval between iteration.
- `RFC2136_TTL` - The TTL of the TXT record used for the DNS challenge.

NOTE: The following documentation is auto-generated from the ACME provider's API library lego (https://go-acme.github.io/lego/). Some sections may refer to lego directly - in most cases, these sections apply to the Terraform provider as well.

Amazon Route 53 DNS Challenge Provider

The `route53` DNS challenge provider can be used to perform DNS challenges for the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource with Amazon Route 53 (https://aws.amazon.com/route53/).

For complete information on how to use this provider with the `acme_certifiate` resource, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

Example

```
resource "acme_certificate" "certificate" {
  ...

  dns_challenge {
    provider = "route53"
  }
}
```

Argument Reference

The following arguments can be either passed as environment variables, or directly through the `config` block in the `dns_challenge` (/docs/providers/acme/r/certificate.html#dns_challenge) argument in the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource. For more details, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

In addition, arguments can also be stored in a local file, with the path supplied by supplying the argument with the `_FILE` suffix. See here (/docs/providers/acme/r/certificate.html#using-variable-files-for-provider-arguments) for more information.

- `AWS_ACCESS_KEY_ID` - Managed by the AWS client.
- `AWS_HOSTED_ZONE_ID` - Override the hosted zone ID.
- `AWS_REGION` - Managed by the AWS client.
- `AWS_SECRET_ACCESS_KEY` - Managed by the AWS client.
- `AWS_MAX_RETRIES` - The number of maximum returns the service will use to make an individual API request.
- `AWS_POLLING_INTERVAL` - Time between DNS propagation check.
- `AWS_PROPAGATION_TIMEOUT` - Maximum waiting time for DNS propagation.
- `AWS_TTL` - The TTL of the TXT record used for the DNS challenge.

Description

AWS Credentials are automatically detected in the following locations and prioritized in the following order:

1. Environment variables: `AWS_ACCESS_KEY_ID`, `AWS_SECRET_ACCESS_KEY`, `AWS_REGION`, [`AWS_SESSION_TOKEN`]
2. Shared credentials file (defaults to `~/.aws/credentials`)
3. Amazon EC2 IAM role

If `AWS_HOSTED_ZONE_ID` is not set, Lego tries to determine the correct public hosted zone via the FQDN.

See also: sessions (https://docs.aws.amazon.com/sdk-for-go/v1/developer-guide/sessions.html)

Policy

The following AWS IAM policy document describes the permissions required for lego to complete the DNS challenge.

```
{
   "Version": "2012-10-17",
   "Statement": [
       {
           "Sid": "",
           "Effect": "Allow",
           "Action": [
               "route53:GetChange",
               "route53:ChangeResourceRecordSets",
               "route53:ListResourceRecordSets"
           ],
           "Resource": [
               "arn:aws:route53:::hostedzone/*",
               "arn:aws:route53:::change/*"
           ]
       },
       {
           "Sid": "",
           "Effect": "Allow",
           "Action": "route53:ListHostedZonesByName",
           "Resource": "*"
       }
   ]
}
```

NOTE: The following documentation is auto-generated from the ACME provider's API library lego (https://go-acme.github.io/lego/). Some sections may refer to lego directly - in most cases, these sections apply to the Terraform provider as well.

Sakura Cloud DNS Challenge Provider

The `sakuracloud` DNS challenge provider can be used to perform DNS challenges for the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource with Sakura Cloud (https://cloud.sakura.ad.jp/).

For complete information on how to use this provider with the `acme_certifiate` resource, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

Example

```
resource "acme_certificate" "certificate" {
  ...

  dns_challenge {
    provider = "sakuracloud"
  }
}
```

Argument Reference

The following arguments can be either passed as environment variables, or directly through the `config` block in the `dns_challenge` (/docs/providers/acme/r/certificate.html#dns_challenge) argument in the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource. For more details, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

In addition, arguments can also be stored in a local file, with the path supplied by supplying the argument with the `_FILE` suffix. See here (/docs/providers/acme/r/certificate.html#using-variable-files-for-provider-arguments) for more information.

- `SAKURACLOUD_ACCESS_TOKEN` - Access token.
- `SAKURACLOUD_ACCESS_TOKEN_SECRET` - Access token secret.
- `SAKURACLOUD_HTTP_TIMEOUT` - API request timeout.
- `SAKURACLOUD_POLLING_INTERVAL` - Time between DNS propagation check.
- `SAKURACLOUD_PROPAGATION_TIMEOUT` - Maximum waiting time for DNS propagation.
- `SAKURACLOUD_TTL` - The TTL of the TXT record used for the DNS challenge.

NOTE: The following documentation is auto-generated from the ACME provider's API library lego (https://go-acme.github.io/lego/). Some sections may refer to lego directly - in most cases, these sections apply to the Terraform provider as well.

Selectel DNS Challenge Provider

The `selectel` DNS challenge provider can be used to perform DNS challenges for the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource with Selectel (https://kb.selectel.com/).

For complete information on how to use this provider with the `acme_certifiate` resource, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

Example

```
resource "acme_certificate" "certificate" {
  ...

  dns_challenge {
    provider = "selectel"
  }
}
```

Argument Reference

The following arguments can be either passed as environment variables, or directly through the `config` block in the `dns_challenge` (/docs/providers/acme/r/certificate.html#dns_challenge) argument in the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource. For more details, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

In addition, arguments can also be stored in a local file, with the path supplied by supplying the argument with the `_FILE` suffix. See here (/docs/providers/acme/r/certificate.html#using-variable-files-for-provider-arguments) for more information.

- `SELECTEL_API_TOKEN` - API token.
- `SELECTEL_BASE_URL` - API endpoint URL.
- `SELECTEL_HTTP_TIMEOUT` - API request timeout.
- `SELECTEL_POLLING_INTERVAL` - Time between DNS propagation check.
- `SELECTEL_PROPAGATION_TIMEOUT` - Maximum waiting time for DNS propagation.
- `SELECTEL_TTL` - The TTL of the TXT record used for the DNS challenge.

NOTE: The following documentation is auto-generated from the ACME provider's API library lego (https://go-acme.github.io/lego/). Some sections may refer to lego directly - in most cases, these sections apply to the Terraform provider as well.

Stackpath DNS Challenge Provider

The `stackpath` DNS challenge provider can be used to perform DNS challenges for the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource with Stackpath (https://www.stackpath.com/).

For complete information on how to use this provider with the `acme_certifiate` resource, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

Example

```
resource "acme_certificate" "certificate" {
  ...

  dns_challenge {
    provider = "stackpath"
  }
}
```

Argument Reference

The following arguments can be either passed as environment variables, or directly through the `config` block in the `dns_challenge` (/docs/providers/acme/r/certificate.html#dns_challenge) argument in the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource. For more details, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

In addition, arguments can also be stored in a local file, with the path supplied by supplying the argument with the `_FILE` suffix. See here (/docs/providers/acme/r/certificate.html#using-variable-files-for-provider-arguments) for more information.

- `STACKPATH_CLIENT_ID` - Client ID.
- `STACKPATH_CLIENT_SECRET` - Client secret.
- `STACKPATH_STACK_ID` - Stack ID.
- `STACKPATH_POLLING_INTERVAL` - Time between DNS propagation check.
- `STACKPATH_PROPAGATION_TIMEOUT` - Maximum waiting time for DNS propagation.
- `STACKPATH_TTL` - The TTL of the TXT record used for the DNS challenge.

NOTE: The following documentation is auto-generated from the ACME provider's API library lego (https://go-acme.github.io/lego/). Some sections may refer to lego directly - in most cases, these sections apply to the Terraform provider as well.

TransIP DNS Challenge Provider

The `transip` DNS challenge provider can be used to perform DNS challenges for the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource with TransIP (https://www.transip.nl/).

For complete information on how to use this provider with the `acme_certifiate` resource, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

Example

```
resource "acme_certificate" "certificate" {
  ...

  dns_challenge {
    provider = "transip"
  }
}
```

Argument Reference

The following arguments can be either passed as environment variables, or directly through the `config` block in the `dns_challenge` (/docs/providers/acme/r/certificate.html#dns_challenge) argument in the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource. For more details, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

In addition, arguments can also be stored in a local file, with the path supplied by supplying the argument with the `_FILE` suffix. See here (/docs/providers/acme/r/certificate.html#using-variable-files-for-provider-arguments) for more information.

- `TRANSIP_ACCOUNT_NAME` - Account name.
- `TRANSIP_PRIVATE_KEY_PATH` - Private key path.
- `TRANSIP_POLLING_INTERVAL` - Time between DNS propagation check.
- `TRANSIP_PROPAGATION_TIMEOUT` - Maximum waiting time for DNS propagation.
- `TRANSIP_TTL` - The TTL of the TXT record used for the DNS challenge.

NOTE: The following documentation is auto-generated from the ACME provider's API library lego (https://go-acme.github.io/lego/). Some sections may refer to lego directly - in most cases, these sections apply to the Terraform provider as well.

VegaDNS DNS Challenge Provider

The `vegadns` DNS challenge provider can be used to perform DNS challenges for the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource with VegaDNS (https://github.com/shupp/VegaDNS-API).

For complete information on how to use this provider with the `acme_certifiate` resource, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

Example

```
resource "acme_certificate" "certificate" {
  ...

  dns_challenge {
    provider = "vegadns"
  }
}
```

Argument Reference

The following arguments can be either passed as environment variables, or directly through the `config` block in the `dns_challenge` (/docs/providers/acme/r/certificate.html#dns_challenge) argument in the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource. For more details, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

In addition, arguments can also be stored in a local file, with the path supplied by supplying the argument with the `_FILE` suffix. See here (/docs/providers/acme/r/certificate.html#using-variable-files-for-provider-arguments) for more information.

- `SECRET_VEGADNS_KEY` - API key.
- `SECRET_VEGADNS_SECRET` - API secret.
- `VEGADNS_URL` - API endpoint URL.
- `VEGADNS_POLLING_INTERVAL` - Time between DNS propagation check.
- `VEGADNS_PROPAGATION_TIMEOUT` - Maximum waiting time for DNS propagation.
- `VEGADNS_TTL` - The TTL of the TXT record used for the DNS challenge.

NOTE: The following documentation is auto-generated from the ACME provider's API library lego (https://go-acme.github.io/lego/). Some sections may refer to lego directly - in most cases, these sections apply to the Terraform provider as well.

Versio.[nl|eu|uk] DNS Challenge Provider

The `versio` DNS challenge provider can be used to perform DNS challenges for the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource with Versio.[nl|eu|uk] (https://www.versio.nl/domeinnamen).

For complete information on how to use this provider with the `acme_certifiate` resource, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

Example

```
resource "acme_certificate" "certificate" {
  ...

  dns_challenge {
    provider = "versio"
  }
}
```

Argument Reference

The following arguments can be either passed as environment variables, or directly through the `config` block in the `dns_challenge` (/docs/providers/acme/r/certificate.html#dns_challenge) argument in the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource. For more details, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

In addition, arguments can also be stored in a local file, with the path supplied by supplying the argument with the `_FILE` suffix. See here (/docs/providers/acme/r/certificate.html#using-variable-files-for-provider-arguments) for more information.

- `VERSIO_PASSWORD` - Basic authentication password.
- `VERSIO_USERNAME` - Basic authentication username.
- `VERSIO_ENDPOINT` - The endpoint URL of the API Server.
- `VERSIO_HTTP_TIMEOUT` - API request timeout.
- `VERSIO_POLLING_INTERVAL` - Time between DNS propagation check.
- `VERSIO_PROPAGATION_TIMEOUT` - Maximum waiting time for DNS propagation.
- `VERSIO_SEQUENCE_INTERVAL` - Interval between iteration, default 60s.
- `VERSIO_TTL` - The TTL of the TXT record used for the DNS challenge.

To test with the sandbox environment set `VERSIO_ENDPOINT=https://www.versio.nl/testapi/v1/`

NOTE: The following documentation is auto-generated from the ACME provider's API library lego (https://go-acme.github.io/lego/). Some sections may refer to lego directly - in most cases, these sections apply to the Terraform provider as well.

Vscale DNS Challenge Provider

The `vscale` DNS challenge provider can be used to perform DNS challenges for the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource with Vscale (https://vscale.io/).

For complete information on how to use this provider with the `acme_certifiate` resource, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

Example

```
resource "acme_certificate" "certificate" {
  ...

  dns_challenge {
    provider = "vscale"
  }
}
```

Argument Reference

The following arguments can be either passed as environment variables, or directly through the `config` block in the `dns_challenge` (/docs/providers/acme/r/certificate.html#dns_challenge) argument in the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource. For more details, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

In addition, arguments can also be stored in a local file, with the path supplied by supplying the argument with the `_FILE` suffix. See here (/docs/providers/acme/r/certificate.html#using-variable-files-for-provider-arguments) for more information.

- `VSCALE_API_TOKEN` - API token.
- `VSCALE_BASE_URL` - API enddpoint URL.
- `VSCALE_HTTP_TIMEOUT` - API request timeout.
- `VSCALE_POLLING_INTERVAL` - Time between DNS propagation check.
- `VSCALE_PROPAGATION_TIMEOUT` - Maximum waiting time for DNS propagation.
- `VSCALE_TTL` - The TTL of the TXT record used for the DNS challenge.

NOTE: The following documentation is auto-generated from the ACME provider's API library lego (https://go-acme.github.io/lego/). Some sections may refer to lego directly - in most cases, these sections apply to the Terraform provider as well.

Vultr DNS Challenge Provider

The `vultr` DNS challenge provider can be used to perform DNS challenges for the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource with Vultr (https://www.vultr.com/).

For complete information on how to use this provider with the `acme_certifiate` resource, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

Example

```
resource "acme_certificate" "certificate" {
  ...

  dns_challenge {
    provider = "vultr"
  }
}
```

Argument Reference

The following arguments can be either passed as environment variables, or directly through the `config` block in the `dns_challenge` (/docs/providers/acme/r/certificate.html#dns_challenge) argument in the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource. For more details, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

In addition, arguments can also be stored in a local file, with the path supplied by supplying the argument with the `_FILE` suffix. See here (/docs/providers/acme/r/certificate.html#using-variable-files-for-provider-arguments) for more information.

- `VULTR_API_KEY` - API key.
- `VULTR_HTTP_TIMEOUT` - API request timeout.
- `VULTR_POLLING_INTERVAL` - Time between DNS propagation check.
- `VULTR_PROPAGATION_TIMEOUT` - Maximum waiting time for DNS propagation.
- `VULTR_TTL` - The TTL of the TXT record used for the DNS challenge.

NOTE: The following documentation is auto-generated from the ACME provider's API library lego (https://go-acme.github.io/lego/). Some sections may refer to lego directly - in most cases, these sections apply to the Terraform provider as well.

Zone.ee DNS Challenge Provider

The `zoneee` DNS challenge provider can be used to perform DNS challenges for the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource with Zone.ee (https://www.zone.ee/).

For complete information on how to use this provider with the `acme_certifiate` resource, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

Example

```
resource "acme_certificate" "certificate" {
  ...

  dns_challenge {
    provider = "zoneee"
  }
}
```

Argument Reference

The following arguments can be either passed as environment variables, or directly through the `config` block in the `dns_challenge` (/docs/providers/acme/r/certificate.html#dns_challenge) argument in the `acme_certificate` (/docs/providers/acme/r/certificate.html) resource. For more details, see here (/docs/providers/acme/r/certificate.html#using-dns-challenges).

In addition, arguments can also be stored in a local file, with the path supplied by supplying the argument with the `_FILE` suffix. See here (/docs/providers/acme/r/certificate.html#using-variable-files-for-provider-arguments) for more information.

- `ZONEEE_API_KEY` - API key.
- `ZONEEE_API_USER` - API user.
- `ZONEEE_ENDPOINT` - API endpoint URL.
- `ZONEEE_HTTP_TIMEOUT` - API request timeout.
- `ZONEEE_POLLING_INTERVAL` - Time between DNS propagation check.
- `ZONEEE_PROPAGATION_TIMEOUT` - Maximum waiting time for DNS propagation.
- `ZONEEE_TTL` - The TTL of the TXT record used for the DNS challenge.

acme_certificate

The `acme_certificate` resource can be used to create and manage an ACME TLS certificate.

NOTE: As the usage model of Terraform generally sees it as being run on a different server than a certificate would normally be placed on, the `acme_certificate` resource only supports DNS challenges.

Example

The below example is the same example that can be found on the index page (/docs/providers/acme/index.html), and creates both an account and certificate within the same configuration. The account is created using the `acme_registration` (/docs/providers/acme/r/registration.html) resource.

NOTE: When creating accounts and certificates within the same configuration, ensure that you reference the `account_key_pem` (/docs/providers/acme/r/registration.html#account_key_pem) argument in the `acme_registration` (/docs/providers/acme/r/registration.html) resource as the corresponding `account_key_pem` argument in the `acme_certificate` resource. This will ensure that the account gets created before the certificate and avoid errors.

```
provider "acme" {
  server_url = "https://acme-staging-v02.api.letsencrypt.org/directory"
}

resource "tls_private_key" "private_key" {
  algorithm = "RSA"
}

resource "acme_registration" "reg" {
  account_key_pem = "${tls_private_key.private_key.private_key_pem}"
  email_address   = "nobody@example.com"
}

resource "acme_certificate" "certificate" {
  account_key_pem           = "${acme_registration.reg.account_key_pem}"
  common_name               = "www.example.com"
  subject_alternative_names = ["www2.example.com"]

  dns_challenge {
    provider = "route53"
  }
}
```

Using an external CSR

The `acme_certificate` resource can also take an external CSR. In this example, we create one using `tls_cert_request` (/docs/providers/tls/r/cert_request.html) first, before supplying it to the `certificate_request_pem` argument.

NOTE: Some current ACME CA implementations (including Let's Encrypt) strip most of the organization information out of a certificate request subject. You may wish to confirm with the CA what behavior to expect when using the `certificate_request_pem` argument with this resource.

NOTE: It is not a good practice to use the same private key for both your account and your certificate. Make sure you use different keys.

```
provider "acme" {
  server_url = "https://acme-staging-v02.api.letsencrypt.org/directory"
}

resource "tls_private_key" "reg_private_key" {
  algorithm = "RSA"
}

resource "acme_registration" "reg" {
  account_key_pem = "${tls_private_key.reg_private_key.private_key_pem}"
  email_address   = "nobody@example.com"
}

resource "tls_private_key" "cert_private_key" {
  algorithm = "RSA"
}

resource "tls_cert_request" "req" {
  key_algorithm   = "RSA"
  private_key_pem = "${tls_private_key.cert_private_key.private_key_pem}"
  dns_names       = ["www.example.com", "www2.example.com"]

  subject {
    common_name = "www.example.com"
  }
}

resource "acme_certificate" "certificate" {
  account_key_pem         = "${acme_registration.reg.account_key_pem}"
  certificate_request_pem = "${tls_cert_request.req.cert_request_pem}"

  dns_challenge {
    provider = "route53"
  }
}
```

Argument Reference

The resource takes the following arguments:

NOTE: All arguments in `acme_certificate`, other than `min_days_remaining`, force a new resource when changed.

- `account_key_pem` (Required) - The private key of the account that is requesting the certificate.

- `common_name` - The certificate's common name, the primary domain that the certificate will be recognized for. Required when not specifying a CSR.
- `subject_alternative_names` - The certificate's subject alternative names, domains that this certificate will also be recognized for. Only valid when not specifying a CSR.
- `key_type` - The key type for the certificate's private key. Can be one of: `P256` and `P384` (for ECDSA keys of respective length) or `2048`, `4096`, and `8192` (for RSA keys of respective length). Required when not specifying a CSR. The default is `2048` (RSA key of 2048 bits).
- `certificate_request_pem` - A pre-created certificate request, such as one from `tls_cert_request` (/docs/providers/tls/r/cert_request.html), or one from an external source, in PEM format. Either this, or the in-resource request options (`common_name`, `key_type`, and optionally `subject_alternative_names`) need to be specified.
- `dns_challenge` (Required) - The DNS challenges to use in fulfilling the request.
- `recursive_nameservers` (Optional) - The recursive nameservers that will be used to check for propagation of the challenge record. Defaults to your system-configured DNS resolvers.
- `must_staple` (Optional) Enables the OCSP Stapling Required (https://letsencrypt.org/docs/integration-guide/#implement-ocsp-stapling) TLS Security Policy extension. Certificates with this extension must include a valid OCSP Staple in the TLS handshake for the connection to succeed. Defaults to `false`. Note that this option has no effect when using an external CSR - it must be enabled in the CSR itself.

NOTE: OCSP stapling requires specific webserver configuration to support the downloading of the staple from the CA's OCSP endpoints, and should be configured to tolerate prolonged outages of the OCSP service. Consider this when using `must_staple`, and only enable it if you are sure your webserver or service provider can be configured correctly.

- `min_days_remaining` (Optional) - The minimum amount of days remaining on the expiration of a certificate before a renewal is attempted. The default is `30`. A value of less than `0` means that the certificate will never be renewed.
- `certificate_p12_password` - (Optional) Password to be used when generating the PFX file stored in `certificate_p12`. Defaults to an empty string.

Using DNS challenges

As the usage model of Terraform generally sees it as being run on a different server than a certificate would normally be placed on, the `acme_certificate` resource only supports DNS challenges. This method authenticates certificate domains by requiring the requester to place a TXT record on the FQDNs in the certificate.

The ACME provider responds to DNS challenges automatically by utilizing one of the supported DNS challenge providers. Most providers take credentials as environment variables, but if you would rather use configuration for this purpose, you can by specifying `config` blocks within a `dns_challenge` block, along with the `provider` parameter.

For a full list of providers, click here (/docs/providers/acme/dns_providers/index.html).

Example with the Route 53 provider (/docs/providers/acme/dns_providers/route53.html):

```
resource "acme_certificate" "certificate" {
  #...

  dns_challenge {
    provider = "route53"

    config = {
      AWS_ACCESS_KEY_ID     = "${var.aws_access_key}"
      AWS_SECRET_ACCESS_KEY = "${var.aws_secret_key}"
      AWS_DEFAULT_REGION    = "us-east-1"
    }
  }

  #...
}
```

Using Variable Files for Provider Arguments

Most provider arguments can be suffixed with `_FILE` to specify that you wish to store that value in a local file. This can be useful if local storage for these values is desired over configuration as variables or within the environment.

Building on the above Route 53 provider (/docs/providers/acme/dns_providers/route53.html) example, the following example uses local files to get the access key ID and secret access key.

```
resource "acme_certificate" "certificate" {
  #...

  dns_challenge {
    provider = "route53"

    config = {
      AWS_ACCESS_KEY_ID_FILE     = "/data/secrets/aws_access_key_id"
      AWS_SECRET_ACCESS_KEY_FILE = "/data/secrets/aws_secret_access_key"
      AWS_DEFAULT_REGION         = "us-east-1"
    }
  }

  #...
}
```

Manually specifying recursive nameservers for propagation checks

The ACME provider will normally use your system-configured DNS resolvers to check for propagation of the TXT records before proceeding with the certificate request. In split horizon scenarios, this check may never succeed, as the machine running Terraform may not have visibility into these public DNS records.

To override this default behavior, supply the `recursive_nameservers` to use as a list in `host:port` form within the `dns_challenge` block:

```
resource "acme_certificate" "certificate" {
  #...

  recursive_nameservers = ["8.8.8.8:53"]

  dns_challenge {
    provider = "route53"
  }

  #...
}
```

Using multiple primary DNS providers

The ACME provider will allow you to configure multiple DNS challenges in the event that you have more than one primary DNS provider.

```
resource "acme_certificate" "certificate" {
  #...

  dns_challenge {
    provider = "azure"
  }

  dns_challenge {
    provider = "gcloud"
  }

  dns_challenge {
    provider = "route53"
  }

  #...
}
```

Some considerations need to be kept in mind when using multiple providers:

- You cannot use more than one provider of the same type at once.
- Your NS records must be correctly configured so that each DNS challenge provider can correctly discover the appropriate zone to update.
- DNS propagation checks are conducted once per configured common name and subject alternative name, using the highest configured or default propagation timeout (`*_PROPAGATION_TIMEOUT`) and polling interval (`*_POLLING_INTERVAL`) settings.

Relation to Terraform provider configuration

The DNS provider configuration specified in the `acme_certificate` resource is separate from any that you supply in a corresponding provider whose functionality overlaps with the certificate's DNS providers. This ensures that there are no hard dependencies between any of these providers and the ACME provider, but it is important to note so that configuration is supplied correctly.

As an example, if you specify manual configuration for the AWS provider (/docs/providers/aws/index.html) via the `provider` (/docs/configuration/providers.html) block instead of the environment, you will still need to supply the configuration explicitly as per above.

Some of these providers have environment variable settings that overlap with the ones found here, generally depending on whether or not these variables are supported by the corresponding provider's SDK.

Check the DNS provider page (/docs/providers/acme/dns_providers/index.html) of a specific provider for more details on exactly what variables are supported.

Certificate renewal

The `acme_certificate` resource handles automatic certificate renewal so long as a plan or apply is done within the number of days specified in the `min_days_remaining` resource parameter. During refresh, if Terraform detects that the certificate is within the expiry range specified in `min_days_remaining`, or is already expired, Terraform will mark the certificate to be renewed on the next apply.

Note that a value less than `0` supplied to `min_days_remaining` will cause renewal checks to be bypassed, and the certificate will never renew.

Attribute Reference

The following attributes are exported:

- `id` - The full URL of the certificate within the ACME CA.
- `certificate_url` - The full URL of the certificate within the ACME CA. Same as `id`.
- `certificate_domain` - The common name of the certificate.
- `private_key_pem` - The certificate's private key, in PEM format, if the certificate was generated from scratch and not with `certificate_request_pem`. If `certificate_request_pem` was used, this will be blank.
- `certificate_pem` - The certificate in PEM format. This does not include the `issuer_pem`. This certificate can be concatenated with `issuer_pem` to form a full chain.
- `issuer_pem` - The intermediate certificate of the issuer.
- `certificate_p12` - The certificate, intermediate, and the private key archived as a PFX file (PKCS12 format, generally used by Microsoft products). The data is base64 encoded (including padding), and its password is configurable via the `certificate_p12_password` argument. This field is empty if creating a certificate from a CSR.

acme_registration

The `acme_registration` resource can be used to create and manage accounts on an ACME server. Once registered, the same private key that has been used for registration can be used to request authorizations for certificates.

This resource is named `acme_registration` for historical reasons - in the ACME v1 spec, a *registration* referred to the account entity. This resource name is stable and more than likely will not change until a later major version of the provider, if at all.

Keep in mind that when using this resource along with `acme_certificate` (/docs/providers/acme/r/certificate.html) within the same configuration, a change in the provider-level `server_url` (example: from the Let's Encrypt staging to production environment) within the same Terraform state will result in a resource failure, as Terraform will attempt to look for the account in the wrong CA. Consider different workspaces per environment, and/or using multiple provider instances (/docs/configuration/providers.html#multiple-provider-instances).

Example

The following creates an account off of a private key generated with the `tls_private_key` (/docs/providers/tls/r/private_key.html) resource.

```
provider "acme" {
  server_url = "https://acme-staging-v02.api.letsencrypt.org/directory"
}

resource "tls_private_key" "private_key" {
  algorithm = "RSA"
}

resource "acme_registration" "reg" {
  account_key_pem = "${tls_private_key.private_key.private_key_pem}"
  email_address   = "nobody@example.com"
}
```

Argument Reference

NOTE: All arguments in `acme_registration` force a new resource if changed.

The resource takes the following arguments:

- `account_key_pem` (Required) - The private key used to identity the account.
- `email_address` (Required) - The contact email address for the account.

Attribute Reference

The following attributes are exported:

- `id` : The original full URL of the account.
- `registration_url` : The current full URL of the account.

> `id` and `registration_url` will usually be the same and will usually only diverge when migrating protocols, ie: ACME v1 to v2.