Splunk .conf 2014

Running Splunk on Amazon Web Services

Alan Williams
Principal Engineer
alanwill on Twitter & GitHub

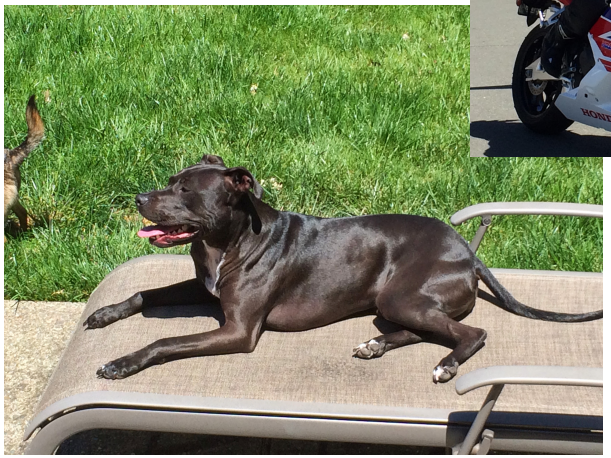
AUTODESK.

Disclaimer

During the course of this presentation, we may make forward-looking statements regarding future events or the expected performance of the company. We caution you that such statements reflect our current expectations and estimates based on factors currently known to us and that actual events or results could differ materially. For important factors that may cause actual results to differ from those contained in our forward-looking statements, please review our filings with the SEC. The forward-looking statements made in the this presentation are being made as of the time and date of its live presentation. If reviewed after its live presentation, this presentation may not contain current or accurate information. We do not assume any obligation to update any forward-looking statements we may make. In addition, any information about our roadmap outlines our general product direction and is subject to change at any time without notice. It is for informational purposes only, and shall not be incorporated into any contract or other commitment. Splunk undertakes no obligation either to develop the features or functionality described or to include any such feature or functionality in a future release.

Who Am I?

- Engineer @ Autodesk
- General technologist
- AWS for ~4 years
- Splunk for ~1 year
- Motorcyclist
- Soft spot for pit bulls



Who is Autodesk?

- Leader in 3D design, engineering and entertainment software
- Introduced AutoCAD in 1982
- Empowering the Maker movement
- Help our customers imagine, design and create a better world

Why

…did we choose to run Splunk in AWS?

AUTODESK.

Make This Better!

- Splunk 4.3
- 5 year old hardware
- Performance issues
- Global
- Now

Decisions… decisions

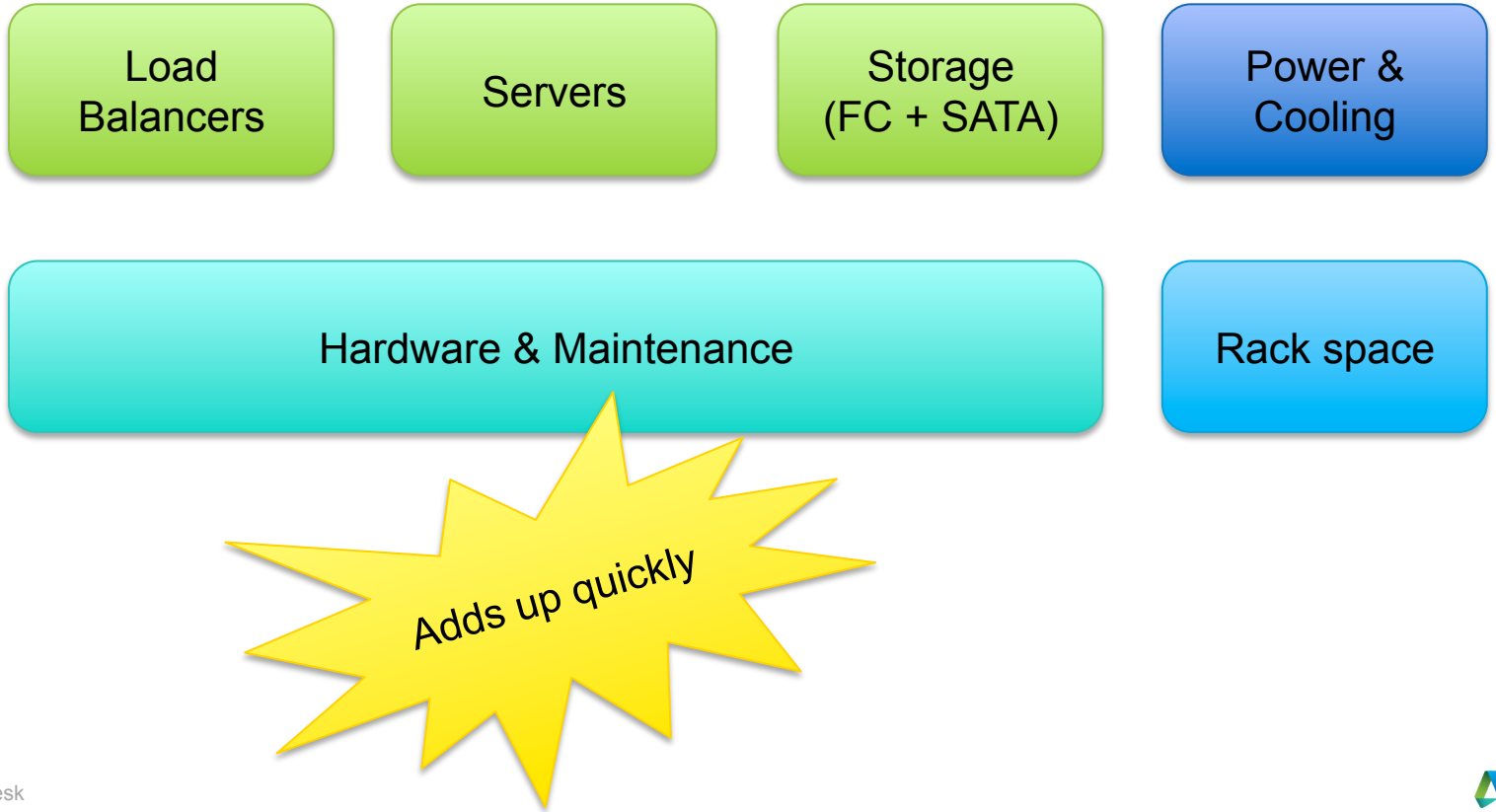
Where We Are	Where We Wanted To Be
Splunk 4.3	Latest Splunk version (6.x)
EOL hardware	Hardware refresh
Fragile environment	Resiliency

Not rocket science… can we do this NOW?

Where to Begin

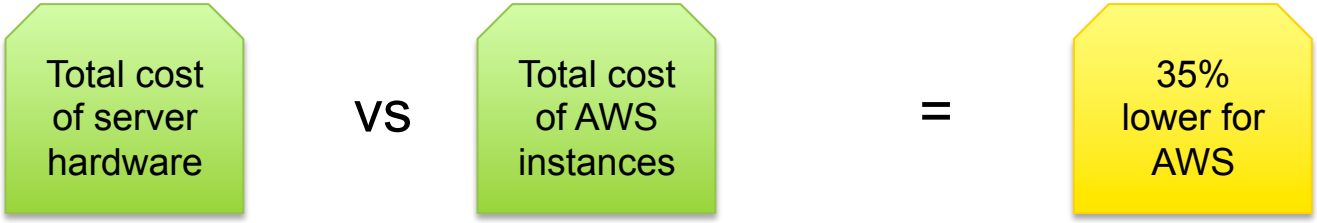
- Take inventory of existing hardware
- Use the AWS Calculator
  - http://calculator.s3.amazonaws.com/index.html
- Cost/compute analysis

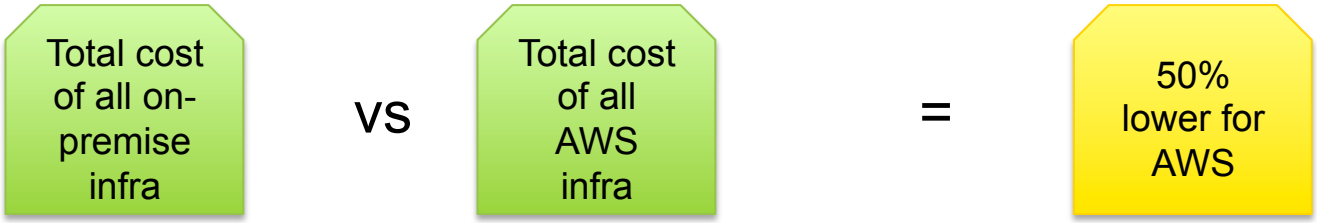
Cost Analysis – Account for Everything



AUTODESK.

What We Noticed…

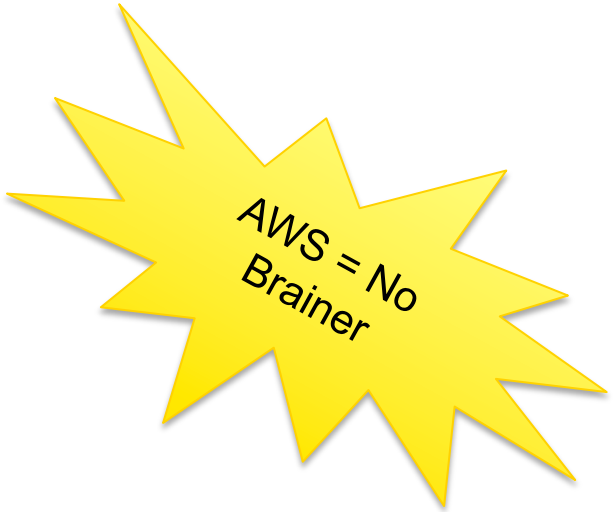




AUTODESK.

Outcome

- We can’t compete on price
  - Economies of scale
- We can’t compete on speed
  - Time to provision
  - Time to deliver new features



How

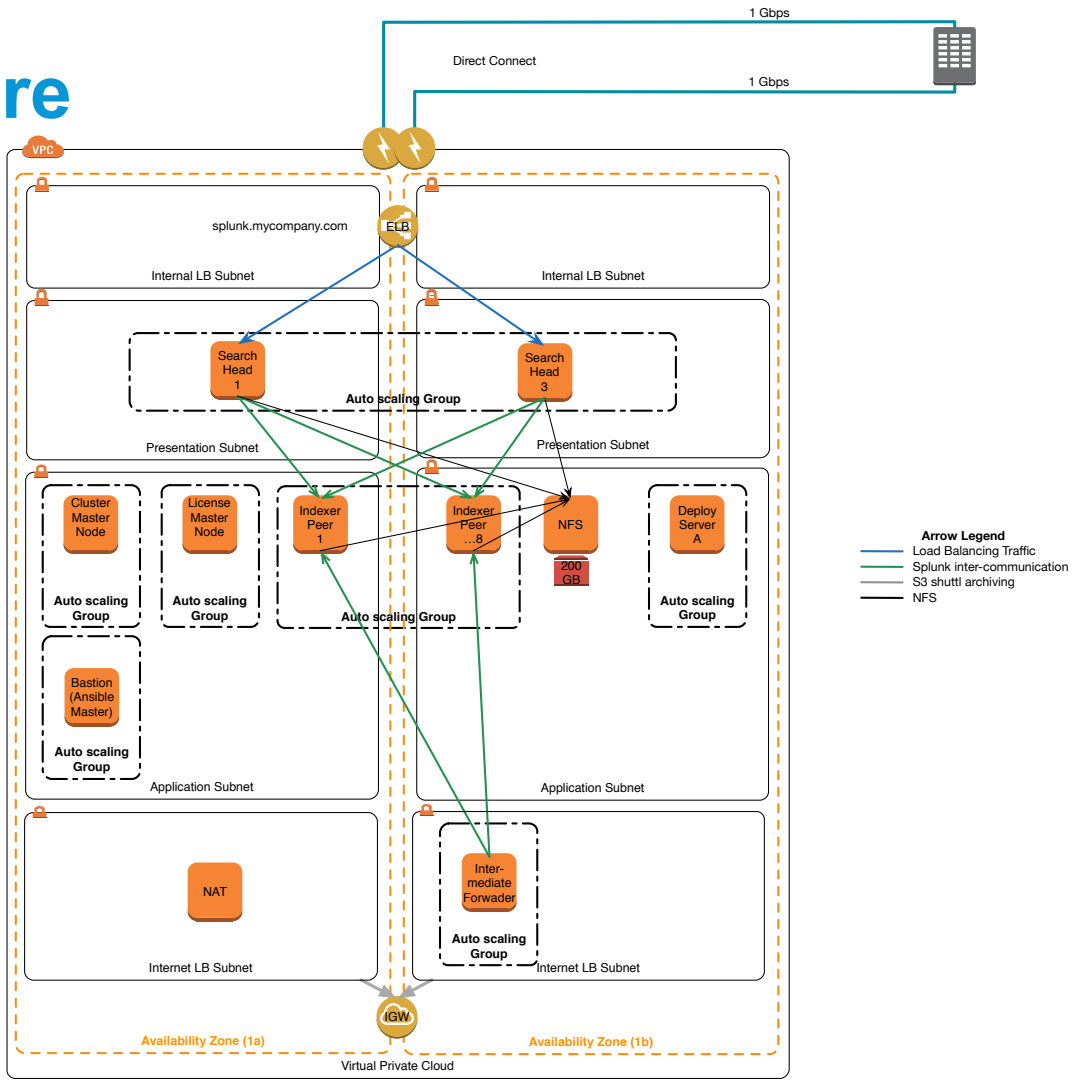
…do we run Splunk in AWS?

AUTODESK.

Splunk Infrastructure Goals

- Automated and Dynamic
- Scalable
- Responsive

Architecture



AUTODESK.

Automated and Dynamic

AUTODESK.

Automated and Dynamic

- AWS CloudFormation Template
  - Infrastructure provisioning
- Ansible Playbook
  - Software install and configuration

CloudFormation Template (splunk-app.json)

- Search Heads
- Peer Nodes
- Cluster Master
- License Master
- Deployment Server
- NFS Instance

- Elastic Load Balancer
- Security Groups
- IAM Roles
- EBS Volumes
- Auto Scaling Groups

https://github.com/alanwill/splunk-on-aws

Ansible Playbook (ansible-splunk)

- Update latest OS packages
- Update hostname
- Download & install Splunk
- Configure inputs.conf
- Deploy custom certs
- Change default password
- Install sysstat

- Install & configure:
  - License Master
  - Cluster Master
  - Peer Nodes
  - Search Heads
  - Deployment Server

https://github.com/alanwill/ansible-splunk

Scalable

AUTODESK.

Scalable

- Easy to add/remove nodes
  - Cloudformation + Ansible
- Dynamic
  - Auto Scaling Groups for everything
  - …even single instanced nodes (1/1/1)
- Splunk Search Head Pooling (NFS)

Auto Scaling Groups

- Can be applied to all Splunk components
- Bootstrap Ansible playbook
- Could pre-bake but haven't tried
  - Consider dynamic portions

Auto Scaling Groups

- Search Heads
  - CPU based policy
- Peer Nodes
  - Manual scaling, no policies
- Cluster/License Master, Deployment instance
  - 1/1/1 ASG (Single instance)
  - Use EBS for persistent data

Search Head Provisioning Code Example

AUTODESK.

Search Head Provisioning Code Example

- Create EC2 instance with CloudFormation
- Run Ansible Playbook
  - Install and configure Splunk
  - Mount Search Head Pooling NFS volume

Add New Search Head – Create EC2 Instance

```
"SearchHeadInstance5" : {
      "Type" : "AWS::EC2::Instance",
      "Properties" : {
        "InstanceType" : { "Ref" : "SearchHeadInstanceType" },
        "KeyName" : { "Ref" : "AppKeyName" },
        "SubnetId" : { "Ref" : "PresentationSubnetAZ1" },
        "ImageId"  : ...
        "SecurityGroupIds" : ...
        "IamInstanceProfile": { "Ref": "SplunkInternalComponentsInstanceProfile" },
        "BlockDeviceMappings" : [
          { "DeviceName" : "/dev/xvda", "Ebs" : { "VolumeSize" : "10", "VolumeType":"gp2" } },
          { "DeviceName" : "/dev/sdb", "VirtualName" : "ephemeral0" },
          { "DeviceName" : "/dev/sdc", "VirtualName" : "ephemeral1" }
        ] ,
        "Tags" : [
          { "Key" : "purpose", "Value" : "Search Head" },
          { "Key" : "stack", "Value" : { "Ref" : "EnvironmentName" } },
          { "Key" : "app", "Value" : { "Ref" : "AppName" } },
          { "Key" : "Name", "Value" : "Splunk Search Head" }
        ] } }
```

Add New Search Head – Ansible Splunk Build

```
- name: Dynamically change hostname
  shell: "hostname `curl http://169.254.169.254/latest/meta-data/instance-id`.
{{ splunk_host_domain }}"

- name: Download Splunk server binary
  get_url: dest=/home/ec2-user  url={{ splunk_binary_url }} sha256sum={{ splunk_binary_sha256sum }}
  when: splunk_installed_result|failed

- name: Install Splunk server binary
  yum: pkg=/home/ec2-user/{{ splunk_binary_file }} state=installed
  when: splunk_installer_present.stat.exists == true

- name: Execute config_splunk_inputs.sh script
  shell: /home/ec2-user/config_splunk_inputs.sh
  when: splunk_running|failed

- name: Start Splunk for the first time
  command: /bin/su --shell=/bin/bash --session-command="/opt/splunk/bin/splunk start --accept-
license" splunk
  when: splunk_running|failed
```

Peer Node Provisioning Code Example

AUTODESK.

Peer Node Provisioning Code Example

- CloudFormation
  - Crete EC2 instance
  - Create EBS volumes and attach to instance
  - Mount EBS volumes
- Run Ansible Playbook
  - Install and configure Splunk
  - Mount Search Head Pooling NFS volume

Add New Peer Node – Create EC2 Instance

```
"PeerNodeInstance8" : {
      "Type" : "AWS::EC2::Instance",
      "Properties" : {
        "InstanceType" : { "Ref" : "PeerNodeInstanceType" },
        "KeyName" : { "Ref" : "AppKeyName" },
        "SubnetId" : { "Ref" : "ApplicationSubnetAZ1" },
        "ImageId" : ... ,
        "SecurityGroupIds" : ...,
        "IamInstanceProfile": { "Ref": "SplunkInternalComponentsInstanceProfile" },
        "EbsOptimized" : true,
        "BlockDeviceMappings" : [
          { "DeviceName" : "/dev/xvda", "Ebs" : { "VolumeSize" : "10", "VolumeType":"gp2" } },
          { "DeviceName" : "/dev/sdb", "VirtualName" : "ephemeral0" },
          { "DeviceName" : "/dev/sdc", "VirtualName" : "ephemeral1" }
        ],
        "Tags" : [
          { "Key" : "purpose", "Value" : "Peer Node" },
          { "Key" : "stack", "Value" : { "Ref" : "EnvironmentName" } },
          { "Key" : "app", "Value" : { "Ref" : "AppName" } },
          { "Key" : "Name", "Value" : "Splunk Peer Node" }
        ] } },
```

Add New Peer Node – Create EBS Volumes

```
"PeerNodeInstance8Volume1" : {
     "Type" : "AWS::EC2::Volume",
     "Properties" : {
       "Size" : { "Ref" : "PeerNodeVolumeSize" }, "VolumeType" : "gp2",
       "AvailabilityZone" : { "Fn::GetAtt" : [ "PeerNodeInstance1", "AvailabilityZone" ] },
       "Tags" : [
         { "Key" : "purpose", "Value" : "Peer Node Instance 1 storage" },
         { "Key" : "stack", "Value" : { "Ref" : "EnvironmentName" } },
         { "Key" : "app", "Value" : { "Ref" : "AppName" } },
         { "Key" : "Name", "Value" : "Splunk Data" } ] },
     "DeletionPolicy" : "Snapshot" },

   "PeerNodeInstance8Volume2" : {
     "Type" : "AWS::EC2::Volume",
     "Properties" : {
       "Size" : { "Ref" : "PeerNodeVolumeSize" }, "VolumeType" : "gp2",
       "AvailabilityZone" : { "Fn::GetAtt" : [ "PeerNodeInstance1", "AvailabilityZone" ] },
       "Tags" : [
         { "Key" : "purpose", "Value" : "Peer Node Instance 1 storage" },
         { "Key" : "stack", "Value" : { "Ref" : "EnvironmentName" } },
         { "Key" : "app", "Value" : { "Ref" : "AppName" } },
         { "Key" : "Name", "Value" : "Splunk Data" } ] },
     "DeletionPolicy" : "Snapshot" },
```

Add New Peer Node – Mount EBS Volumes

```
"PeerNodeInstance8Mount1" : {
     "Type" : "AWS::EC2::VolumeAttachment",
     "Properties" : {
       "InstanceId" : { "Ref" : "PeerNodeInstance1" },
       "VolumeId"  : { "Ref" : "PeerNodeInstance1Volume1" },
       "Device" : "/dev/sdf"
     }
   },

   "PeerNodeInstance8Mount2" : {
     "Type" : "AWS::EC2::VolumeAttachment",
     "Properties" : {
       "InstanceId" : { "Ref" : "PeerNodeInstance1" },
       "VolumeId"  : { "Ref" : "PeerNodeInstance1Volume2" },
       "Device" : "/dev/sdg"
     }
   },
```

Add New Peer Node – Ansible Add to Cluster

```
- name: Enable Peer Nodes
  command: runuser -l splunk -c "splunk edit cluster-config -mode slave -master_uri https://
{{ splunk_cluster_master_ip }}:8089 -replication_port 9887 -secret {{ replication_key }}"
  when: peer_nodes_clustering_enabled|failed
  register: peer_nodes_cluster_configure

- name: Prewarm EBS volume1
  command: dd if=/dev/zero of=/dev/sdf bs=1M
  when: splunk_volume_exists|failed
  ignore_errors: True

- name: Create RAID 0 device
  command: mdadm --create --verbose /dev/md0 --level=stripe --raid-devices=2 /dev/sdf /dev/sdg
  when: splunk_volume_exists|failed

- name: Create filesystem
  filesystem: fstype=ext4 dev=/dev/md0
  when: splunk_volume_exists|failed

- name: Mount volume
  mount: name=/opt/splunk/data src=/dev/md0 fstype=ext4 state=mounted
  when: splunk_volume_exists|failed
```

Responsive

AUTODESK.

Responsive

- Search Heads – CPU bound
  - C3 instances
- Peer Nodes/Indexers – IO bound
  - C3 instances + EBS
  - I2 instances
  - HS1 instances

Responsive

- Maximize IOPs with RAID 0

Looking Back…

- Project took ~4 weeks
- Took longer to co-ordinate cutover
- Time to delivery = biggest win
- Repeatable builds enables new use cases
- Very happy with results

Still to Do…

- Increase the “idempotency” of Ansible scripts
- Test on Google Compute Engine
- Make CFN more dynamic for varied sized clusters
- Auto Scaling Groups Lifecycle actions
  - Termination hooks for clean removal from cluster

Contribute, PRs Encouraged…

- CloudFormation Splunk Cluster Template
  - https://github.com/alanwill/splunk-on-aws
- Ansible Splunk Playbook
  - https://github.com/alanwill/ansible-splunk
- Follow Me: @alanwill