Building scalable AWS Based Splunk Architectures Using Cloud Formation in 30 Minutes or Less

How to Build an AWS Splunk Environment Fast

Arthur Spencer

(CISSP, CISA, GCIH, GCDA, GCFA, CEH)
Sr Professional Services Security Consultant
| Splunk

Neha Doshi

Perficient Splunk Practice Lead
Sr Professional Services Consultant
| Perficient

Forward-Looking Statements

During the course of this presentation, we may make forward-looking statements regarding future events or plans 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 may differ materially. 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, it may not contain current or accurate information. We do not assume any obligation to update any forward-looking statements made herein.

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 functionalities described or to include any such feature or functionality in a future release.

Splunk, Splunk>, Turn Data Into Doing, The Engine for Machine Data, Splunk Cloud, Splunk Light and SPL are trademarks and registered trademarks of Splunk Inc. in the United States and other countries. All other brand names, product names, or trademarks belong to their respective owners. © 2019 Splunk Inc. All rights reserved.

What Are We Building

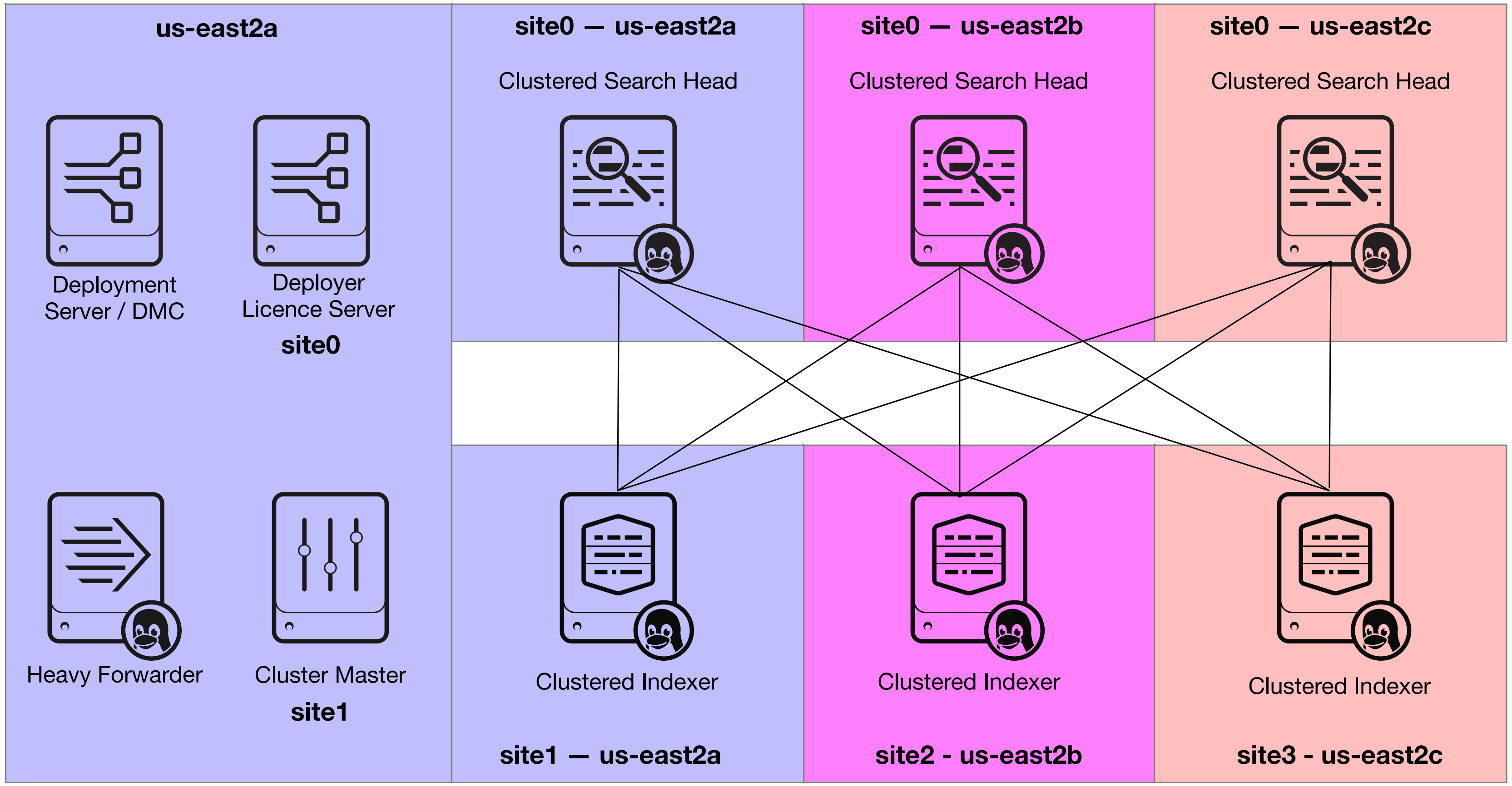
Always Start by defining requirements

Splunk Environment High Level Requirements

- Clustered SH and Clustered Indexer Environment
- Built on AWS
- Geographically Distributed
- Horizontally and Vertically Scalable
- Configuration Managed using GIT
- Development, UAT, and Production environment
- Configured according to Splunk and AWS Best Practices
- Built using a repeatable process driven by AWS CloudFormation Templates
- Deployment Size from 500 MB - 10 TB per day and beyond

What Are We Building

Multi-site Clustered Indexers and Search Heads with a Deployment Server and HF



How Splunk Environments are Established

Many Options

Methods to Build your Splunk Environment

- Manually build by hand
- Bash Scripts, Ansible / Puppet / Chef / Salt / AWS CloudFormation / Some other tool
- Hire Splunk Professional services to build it for you
- Buy Splunk Cloud and call it a day

Base Configurations

- Minimum configuration files to consistently establish a standalone, distributed, or clustered Splunk Environments.
- Ensures that systems are configured and installed in the correct order
- Customizes the environment to improve security and enhance user experience

What you need to know

Thousands of pages of documentation

Splunk

- Splunk Server Roles
- Base Configurations to establish environment
- Search Head Clustering
- Index Clustering
- How to manage Splunk apps
- How to manage user access

AWS

- Regions and Availability Zones
- VPC – Networking
- Security Groups
- EC2 – Compute
- EBS + S3 – Storage
- AMI – Machine Images
- IAM – Security Roles
- Cloud Formation

Management

- Creating a GIT repository
- Syncing GIT repo
- Moving configs to S3
- Pushing Configurations to Splunk
- High Availability (HA) / Business Continuity and Disaster Recovery (BCDR)

AWS 101

High level overview of components and features used

AWS Regions and Availability Zones

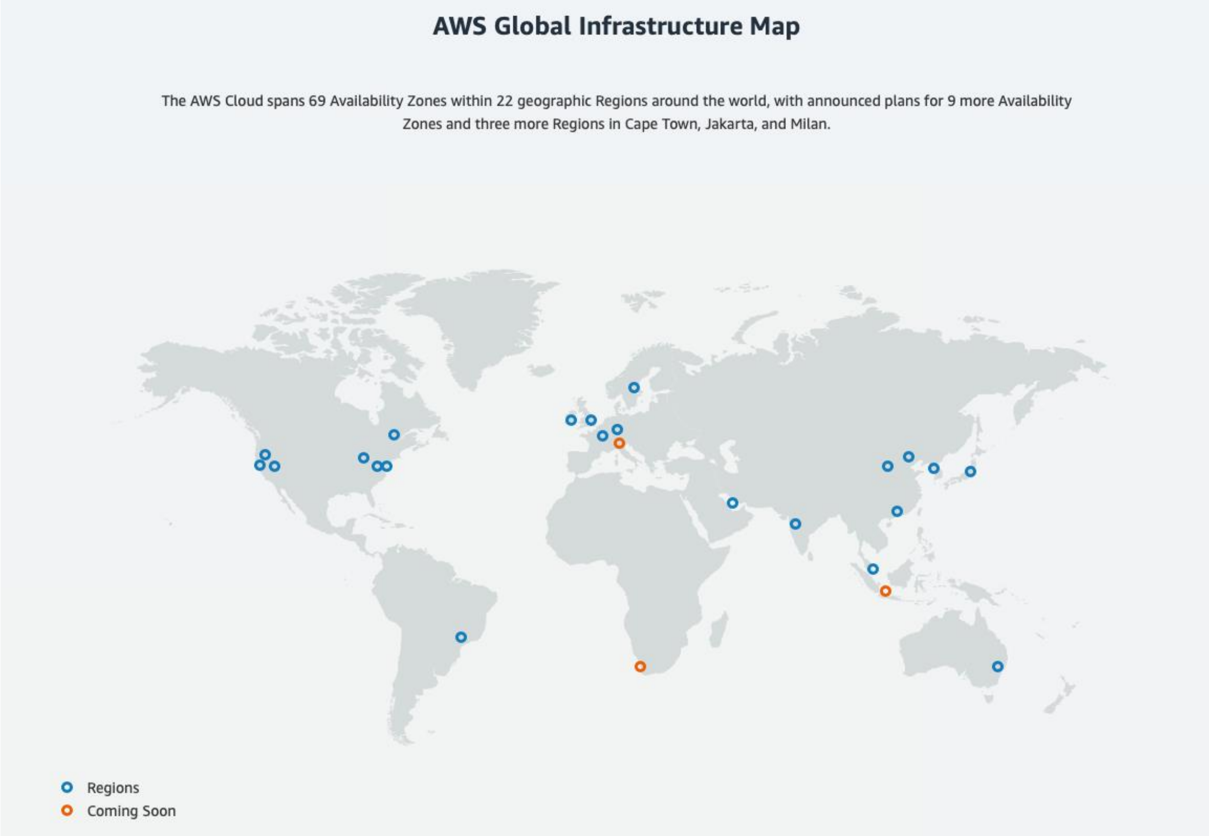
https://aws.amazon.com/about-aws/global-infrastructure/

Regions

- Geographically separated across Globe
- Consists of multiple 3+ Availability Zones

Availability Zones

- Multiple Data Centers
- Fully isolated and redundant
- 100,000+ servers



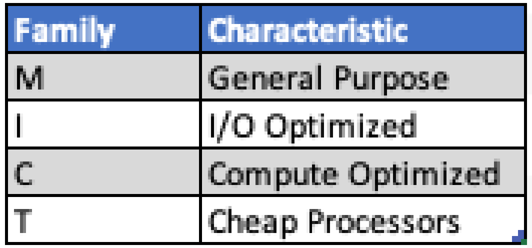
AWS Global Infrastructure Map

The AWS Cloud spans 69 Availability Zones within 22 geographic Regions around the world, with announced plans for 9 more Availability Zones and three more Regions in Cape Town, Jakarta, and Milan.

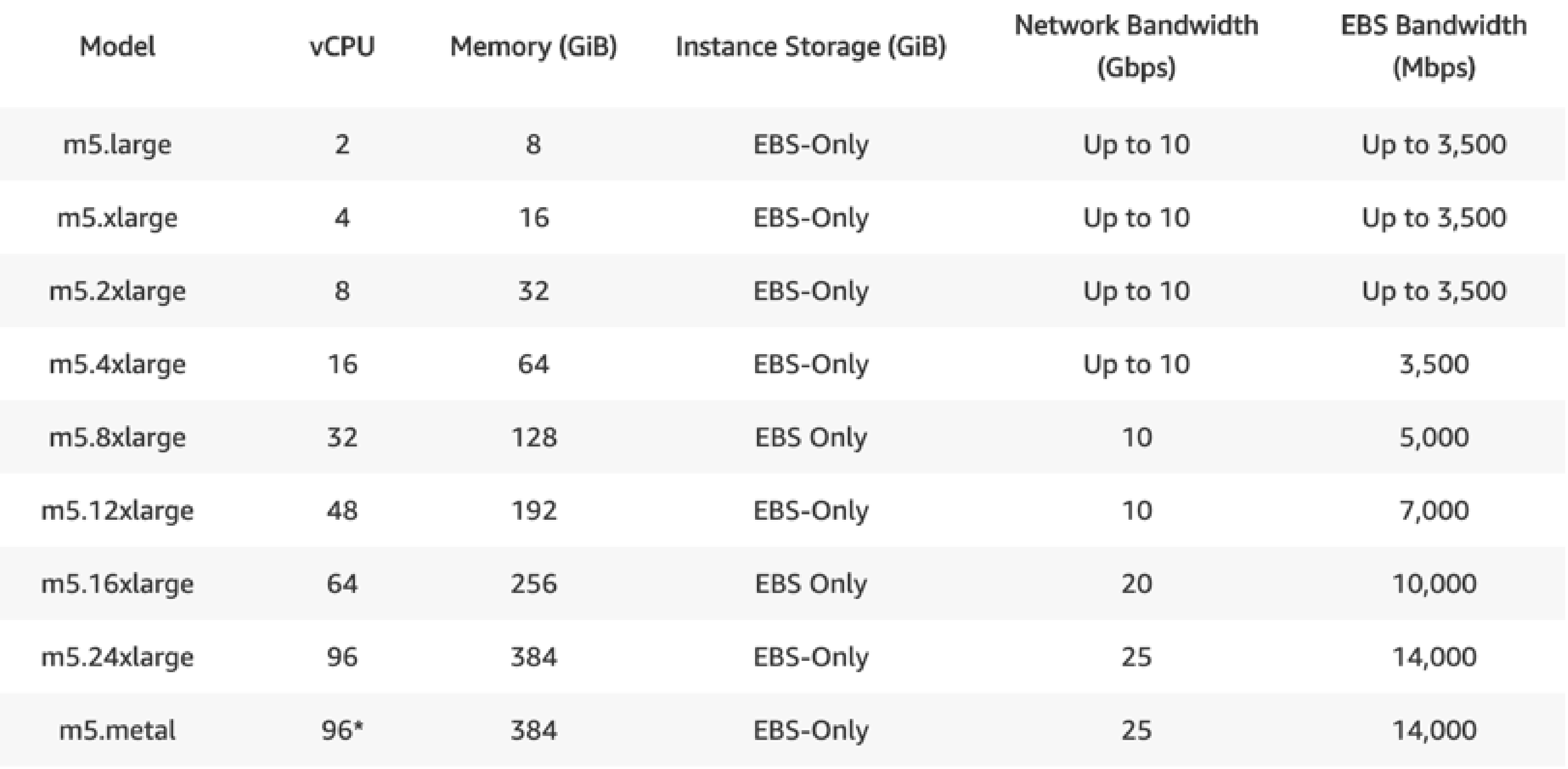
COMPUTE - EC2 Instance types

https://aws.amazon.com/ec2/instance-types/

Processor Families

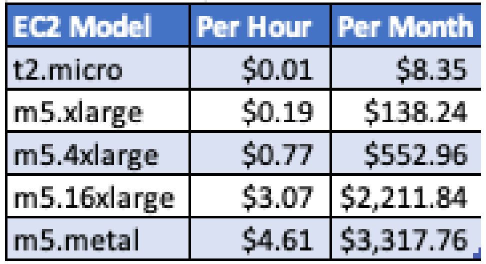


Family	Characteristic
M	General Purpose
I	I/O Optimized
C	Compute Optimized
T	Cheap Processors



Model	vCPU	Memory (GiB)	Instance Storage (GiB)	Network Bandwidth (Gbps)	EBS Bandwidth (Mbps)
m5.large	2	8	EBS-Only	Up to 10	Up to 3,500
m5.xlarge	4	16	EBS-Only	Up to 10	Up to 3,500
m5.2xlarge	8	32	EBS-Only	Up to 10	Up to 3,500
m5.4xlarge	16	64	EBS-Only	Up to 10	3,500
m5.8xlarge	32	128	EBS Only	10	5,000
m5.12xlarge	48	192	EBS-Only	10	7,000
m5.16xlarge	64	256	EBS Only	20	10,000
m5.24xlarge	96	384	EBS-Only	25	14,000
m5.metal	96*	384	EBS-Only	25	14,000

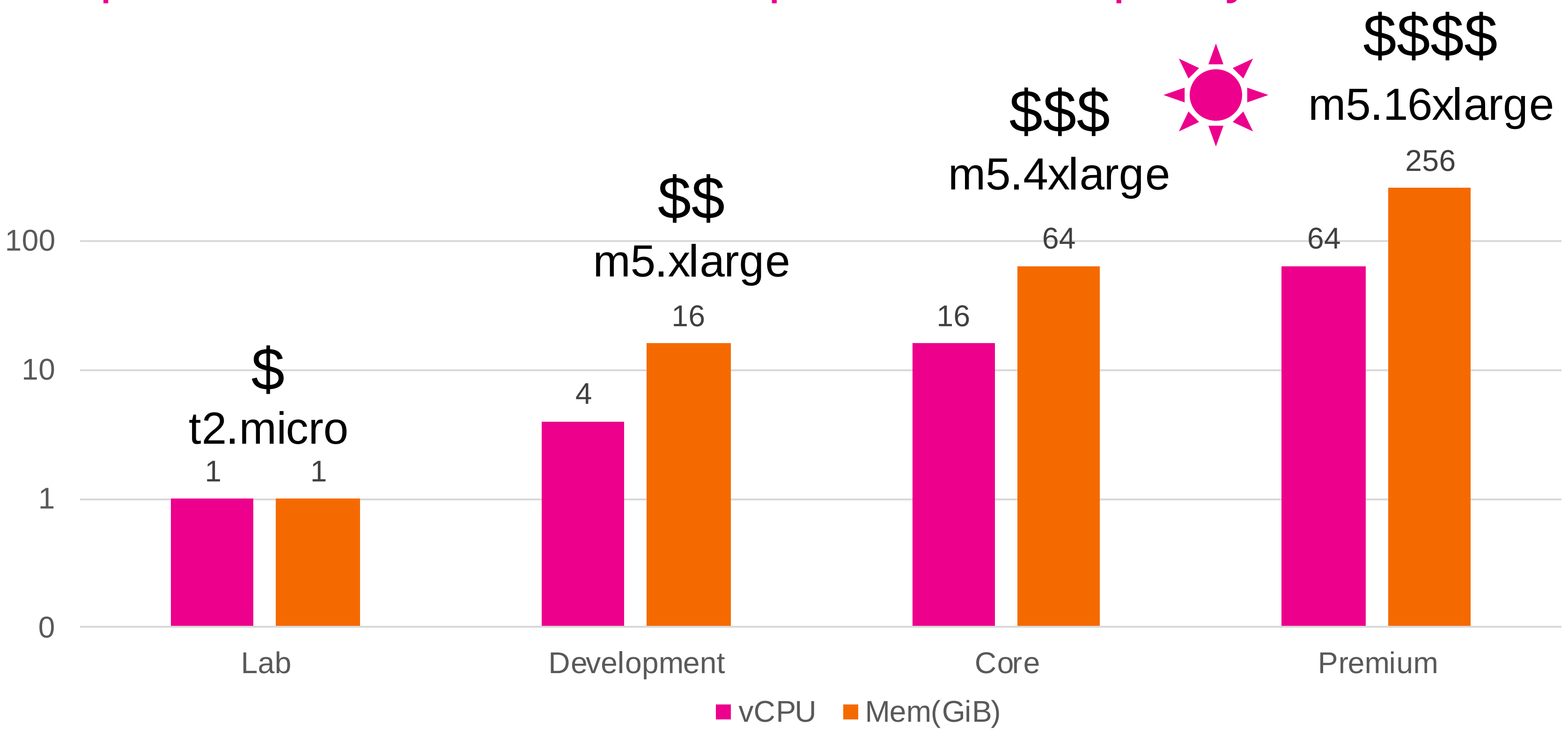
On Demand Cost – 9.7.2019



EC2 Model	Per Hour	Per Month
t2.micro	$0.01	$8.35
m5.xlarge	$0.19	$138.24
m5.4xlarge	$0.77	$552.96
m5.16xlarge	$3.07	$2,211.84
m5.metal	$4.61	$3,317.76

EC2 Instance Types for Different Environments

https://docs.splunk.com/Documentation/Splunk/7.3.1/Capacity



Pick the Instance type and size that is right for you

Storage – EBS and S3

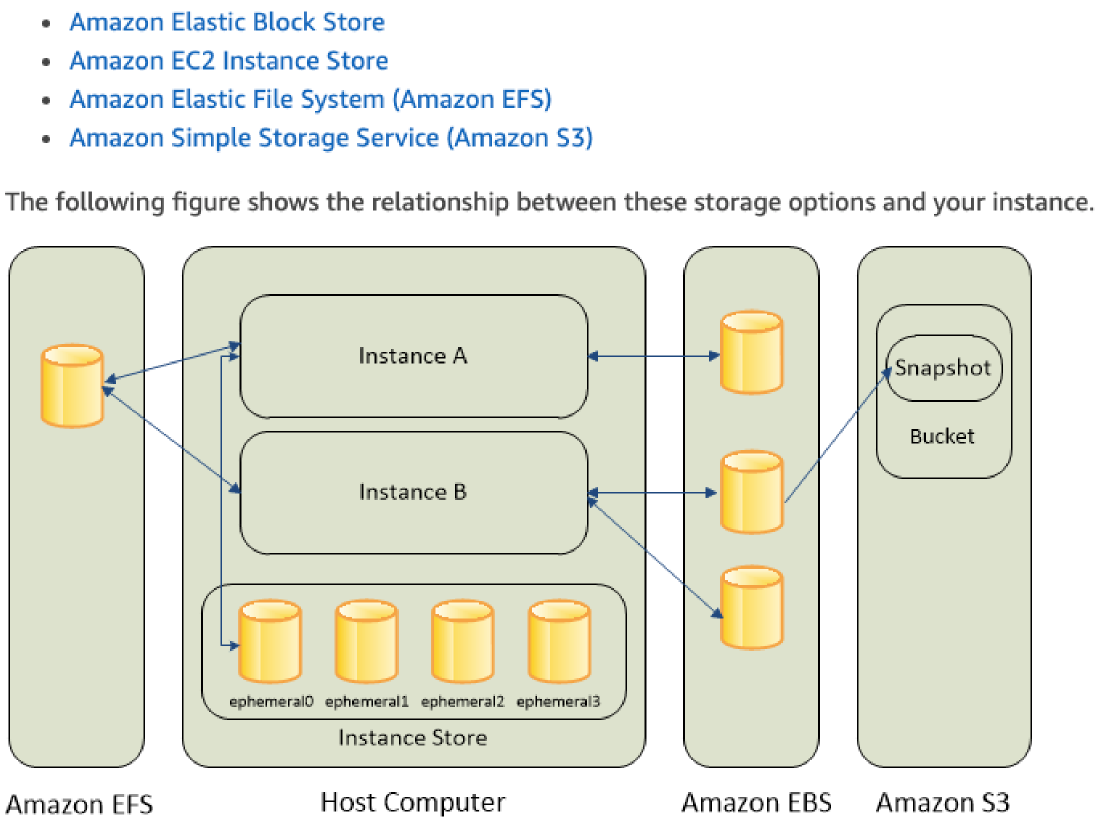
https://aws.amazon.com/ebs/features/

EBS

- Volumes created in an AZ and are attached to and EC2 instance
- Pay for performance
- Can support up to 64,000 IOPS

S3

- Reliable, Fast, inexpensive storage
- Stores Files as Objects

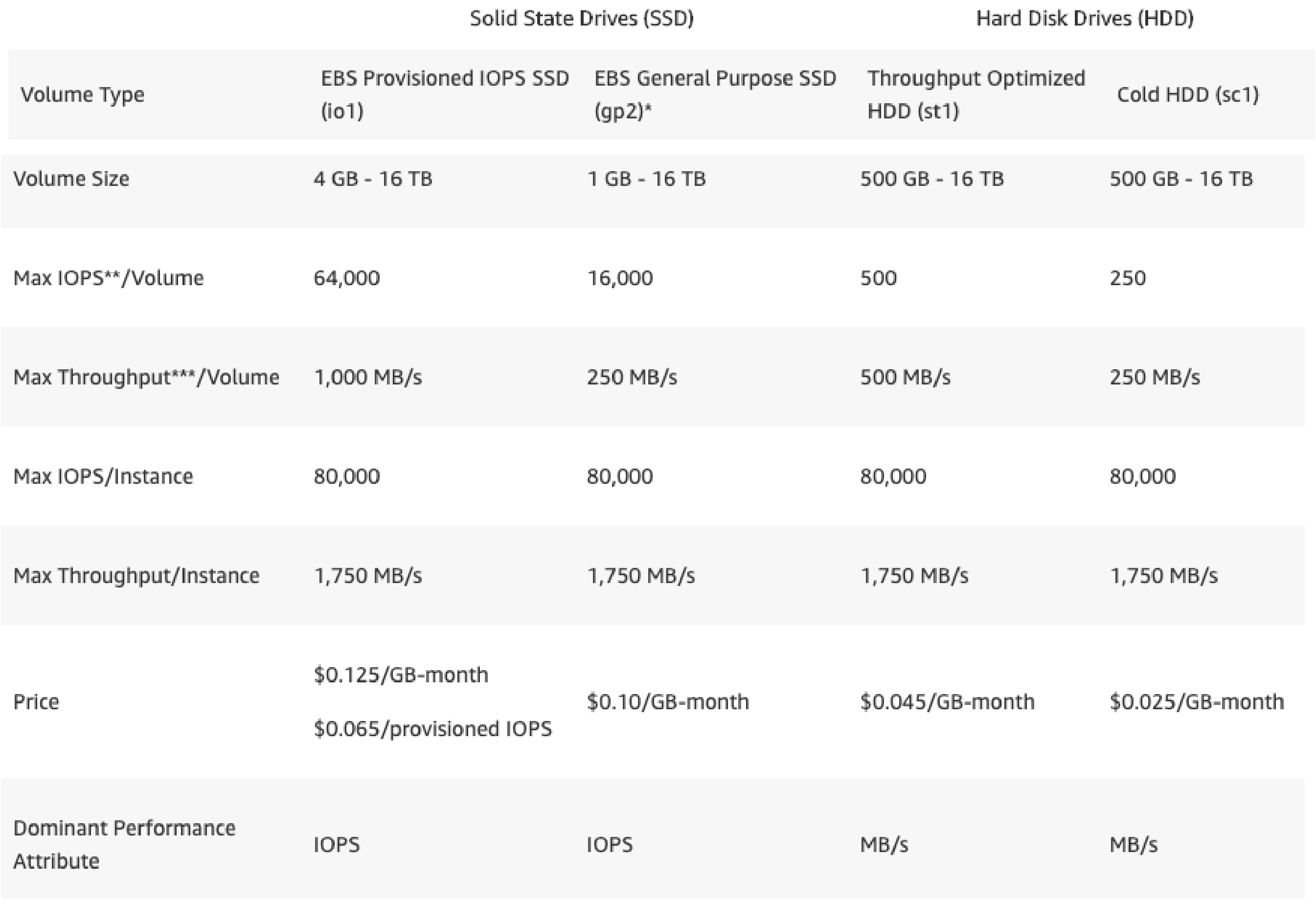


- Amazon Elastic Block Store
- Amazon EC2 Instance Store
- Amazon Elastic File System (Amazon EFS)
- Amazon Simple Storage Service (Amazon S3)

The following figure shows the relationship between these storage options and your instance.

EBS Storage

https://aws.amazon.com/ebs/features/



	Solid State Drives (SSD)		Hard Disk Drives (HDD)	
Volume Type	EBS Provisioned IOPS SSD (io1)	EBS General Purpose SSD (gp2)*	Throughput Optimized HDD (st1)	Cold HDD (sc1)
Volume Size	4 GB - 16 TB	1 GB - 16 TB	500 GB - 16 TB	500 GB - 16 TB
Max IOPS**/Volume	64,000	16,000	500	250
Max Throughput***/Volume	1,000 MB/s	250 MB/s	500 MB/s	250 MB/s
Max IOPS/Instance	80,000	80,000	80,000	80,000
Max Throughput/Instance	1,750 MB/s	1,750 MB/s	1,750 MB/s	1,750 MB/s
Price	$0.125/GB-month $0.065/provisioned IOPS	$0.10/GB-month	$0.045/GB-month	$0.025/GB-month
Dominant Performance Attribute	IOPS	IOPS	MB/s	MB/s

EBS

Networking – VPC, Security Groups, and NACLs

https://docs.aws.amazon.com/vpc/latest/userguide/VPC_Security.html

VPC

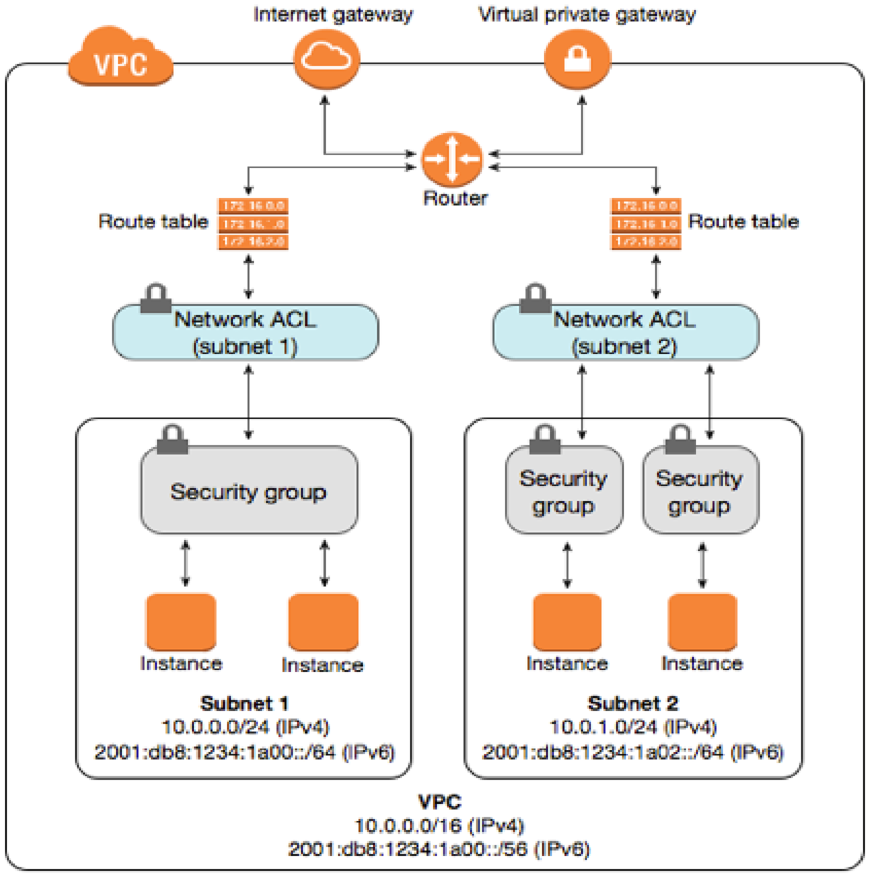
- Virtual Private Cloud
- Network Subnet

Security Groups

- Firewall for inbound and outbound access for EC2 instances

NACL

- Firewall for inbound and outbound traffic between VPC subnets



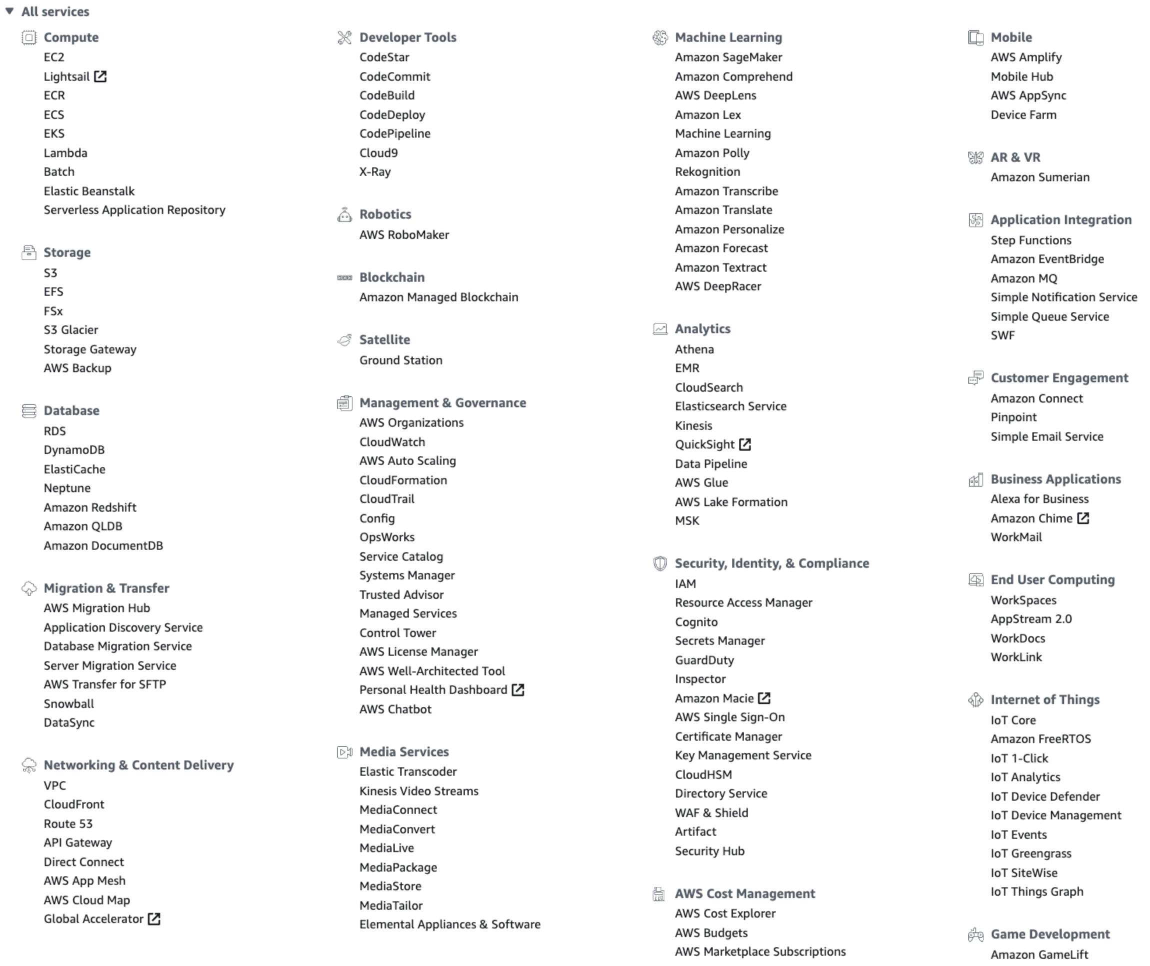
AWS Management Console

Many Options – what to choose

EC2

VPC

S3



▼ All services

Compute
EC2
Lightsail
ECR
ECS
EKS
Lambda
Batch
Elastic Beanstalk
Serverless Application Repository

Storage
S3
EFS
FSx
S3 Glacier
Storage Gateway
AWS Backup

Database
RDS
DynamoDB
ElastiCache
Neptune
Amazon Redshift
Amazon QLDB
Amazon DocumentDB

Migration & Transfer
AWS Migration Hub
Application Discovery Service
Database Migration Service
Server Migration Service
AWS Transfer for SFTP
Snowball
DataSync

Networking & Content Delivery
VPC
CloudFront
Route 53
API Gateway
Direct Connect
AWS App Mesh
AWS Cloud Map
Global Accelerator

Developer Tools
CodeStar
CodeCommit
CodeBuild
CodeDeploy
CodePipeline
Cloud9
X-Ray

Robotics
AWS RoboMaker

Blockchain
Amazon Managed Blockchain

Satellite
Ground Station

Management & Governance
AWS Organizations
CloudWatch
AWS Auto Scaling
CloudFormation
CloudTrail
Config
OpsWorks
Service Catalog
Systems Manager
Trusted Advisor
Managed Services
Control Tower
AWS License Manager
AWS Well-Architected Tool
Personal Health Dashboard
AWS Chatbot

Media Services
Elastic Transcoder
Kinesis Video Streams
MediaConnect
MediaConvert
MediaLive
MediaPackage
MediaStore
MediaTailor
Elemental Appliances & Software

Machine Learning
Amazon SageMaker
Amazon Comprehend
AWS DeepLens
Amazon Lex
Machine Learning
Amazon Polly
Rekognition
Amazon Transcribe
Amazon Translate
Amazon Personalize
Amazon Forecast
Amazon Textract
AWS DeepRacer

Analytics
Athena
EMR
CloudSearch
Elasticsearch Service
Kinesis
QuickSight
Data Pipeline
AWS Glue
AWS Lake Formation
MSK

Security, Identity, & Compliance
IAM
Resource Access Manager
Cognito
Secrets Manager
GuardDuty
Inspector
Amazon Macie
AWS Single Sign-On
Certificate Manager
Key Management Service
CloudHSM
Directory Service
WAF & Shield
Artifact
Security Hub

AWS Cost Management
AWS Cost Explorer
AWS Budgets
AWS Marketplace Subscriptions

Mobile
AWS Amplify
Mobile Hub
AWS AppSync
Device Farm

AR & VR
Amazon Sumerian

Application Integration
Step Functions
Amazon EventBridge
Amazon MQ
Simple Notification Service
Simple Queue Service
SWF

Customer Engagement
Amazon Connect
Pinpoint
Simple Email Service

Business Applications
Alexa for Business
Amazon Chime
WorkMail

End User Computing
WorkSpaces
AppStream 2.0
WorkDocs
WorkLink

Internet of Things
IoT Core
Amazon FreeRTOS
IoT 1-Click
IoT Analytics
IoT Device Defender
IoT Device Management
IoT Events
IoT Greengrass
IoT SiteWise
IoT Things Graph

Game Development
Amazon GameLift

IAM

CloudFormation

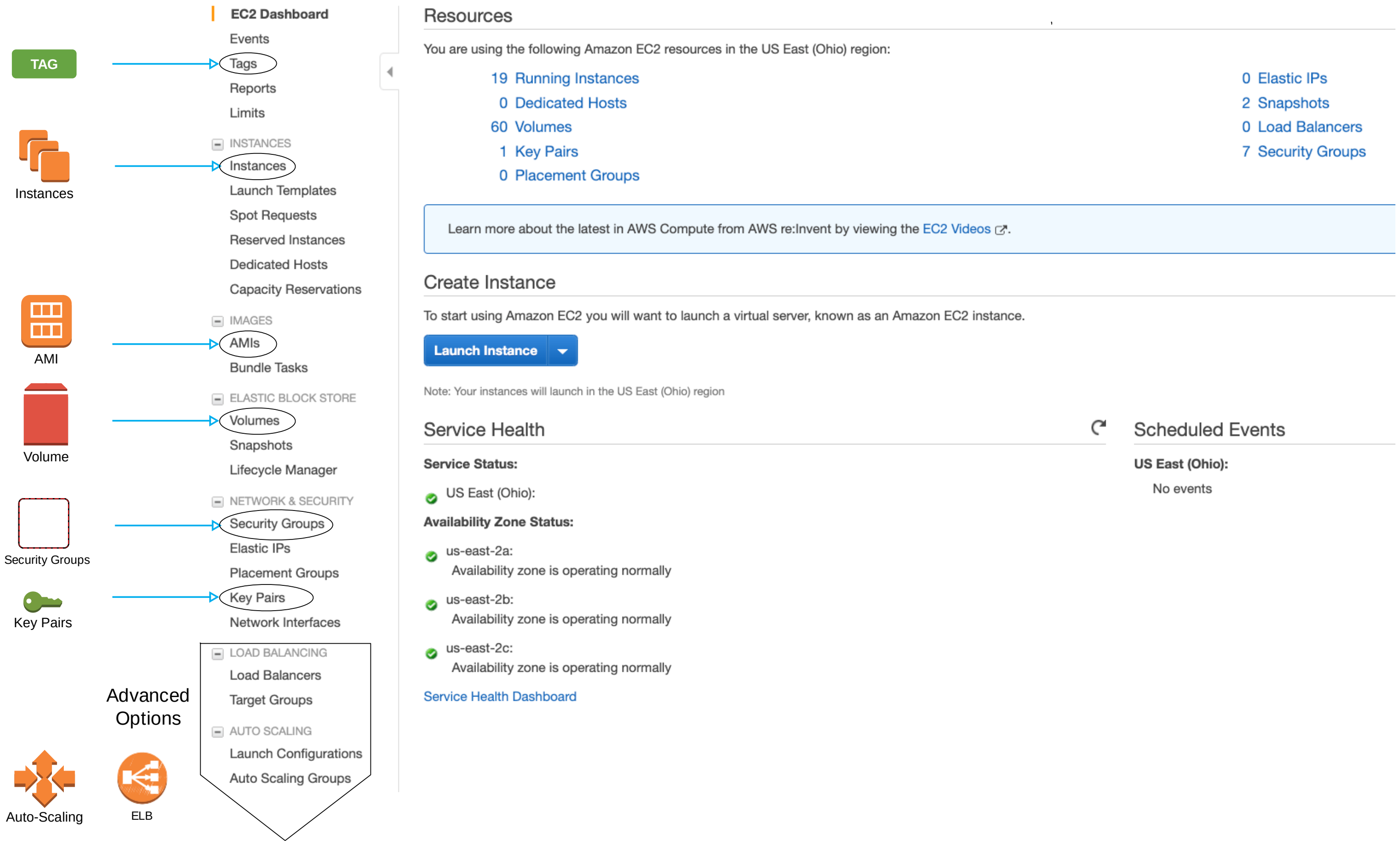
Billing

Not even here
Located in
the user
dropdown
at top

splunk> .conf19

EC2 Dashboard

Primary Dashboard for controlling AWS EC2 instances



Resources

You are using the following Amazon EC2 resources in the US East (Ohio) region:

19 Running Instances	0 Elastic IPs
0 Dedicated Hosts	2 Snapshots
60 Volumes	0 Load Balancers
1 Key Pairs	7 Security Groups
0 Placement Groups	

Learn more about the latest in AWS Compute from AWS re:Invent by viewing the EC2 Videos.

Create Instance

To start using Amazon EC2 you will want to launch a virtual server, known as an Amazon EC2 instance.

Note: Your instances will launch in the US East (Ohio) region

Service Health

Service Status:

US East (Ohio):

Availability Zone Status:

us-east-2a:
Availability zone is operating normally

us-east-2b:
Availability zone is operating normally

us-east-2c:
Availability zone is operating normally

Service Health Dashboard

Scheduled Events

US East (Ohio):

No events

Other AWS Menus accessed

Many Many options and features are available

VPC

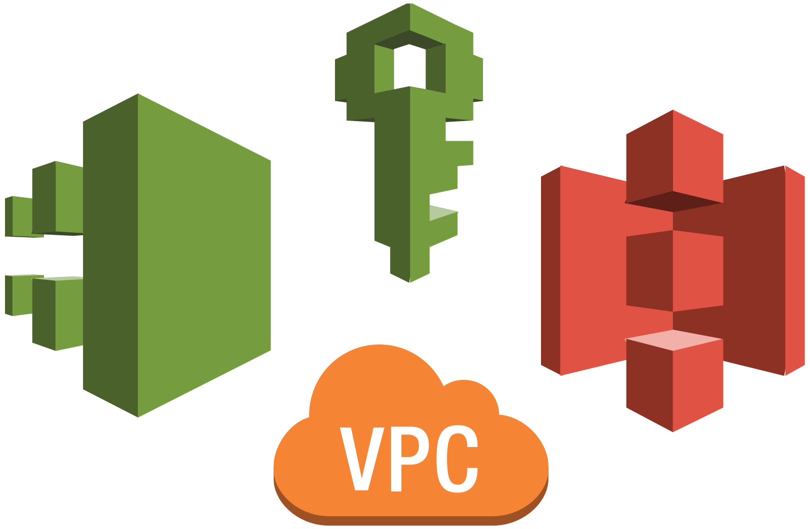
- Virtual Private Cloud
  - Your VPCs
  - Subnets
  - Route Tables
  - Internet Gateways
  - Endpoints
- Security
  - Network ACLs
  - Security Groups

Other Menus Accessed

- IAM – Identity and Access Management
- Route 53 - DNS
- Secrets Manager
- S3 + S3 Glacier
- System Manager
- Lambda
- Billing

Data Sources

- Cloudwatch, Cloudtrail, VPC flow logs, Kinesis, Firehose



CloudFormation

Common Language to describe and provision AWS Cloud Infrastructure

What is CloudFormation

- Script used for CI/CD. Infrastructure as code. Data Center in a Single file.

AWS Marketplace and AMI’s

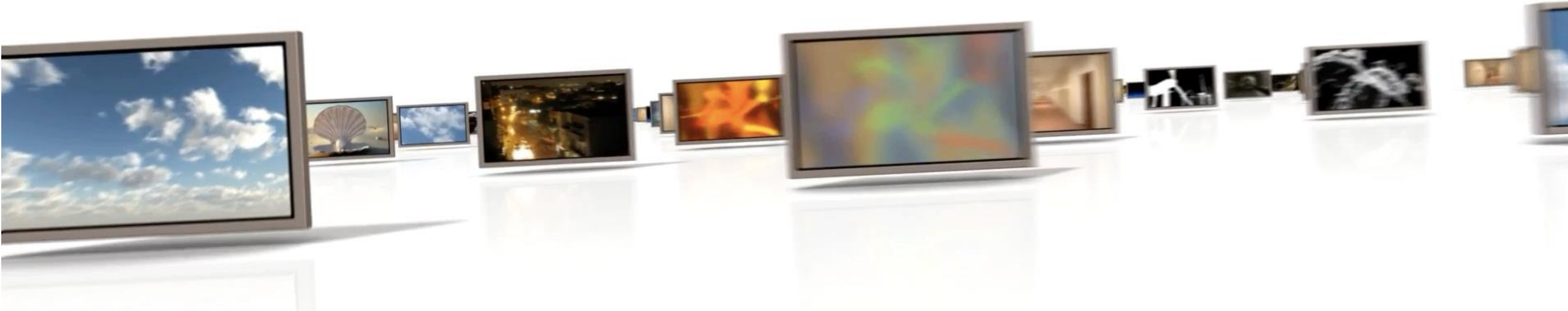
- https://aws.amazon.com/marketplace/pp/B00PUXWXNE

AWS Quick Start

- https://aws.amazon.com/quickstart/architecture/splunk-enterprise/
- Awesome starting point created by Splunkers Bill Bartlet and Roy Arsen
- Over 2500 lines of highly structured JSON code containing many embedded Bash commands and needs to be optimized for you environment
- Asks many questions and builds an entire Splunk environment with many options
- Script is difficult to understand, configure, and optimize by someone new to AWS
- Requires additional steps to be production ready

Demo

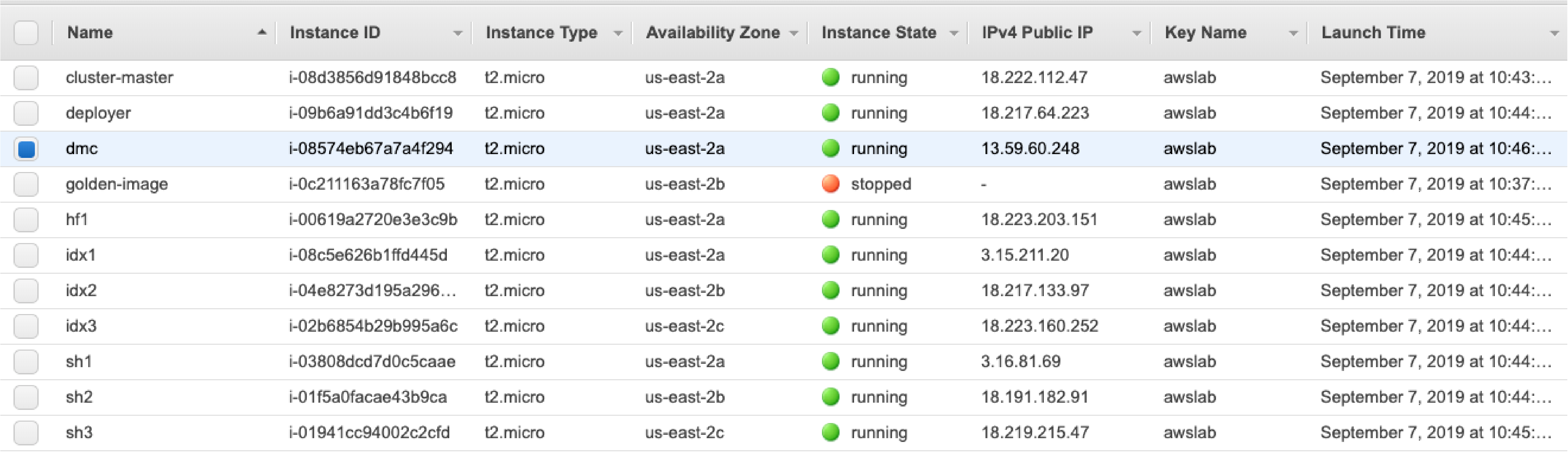
FN2195 - Building Scalable Splunk Architectures with CloudFormation



in 30 minutes or less

Running EC2 Instances – across 3 AZs

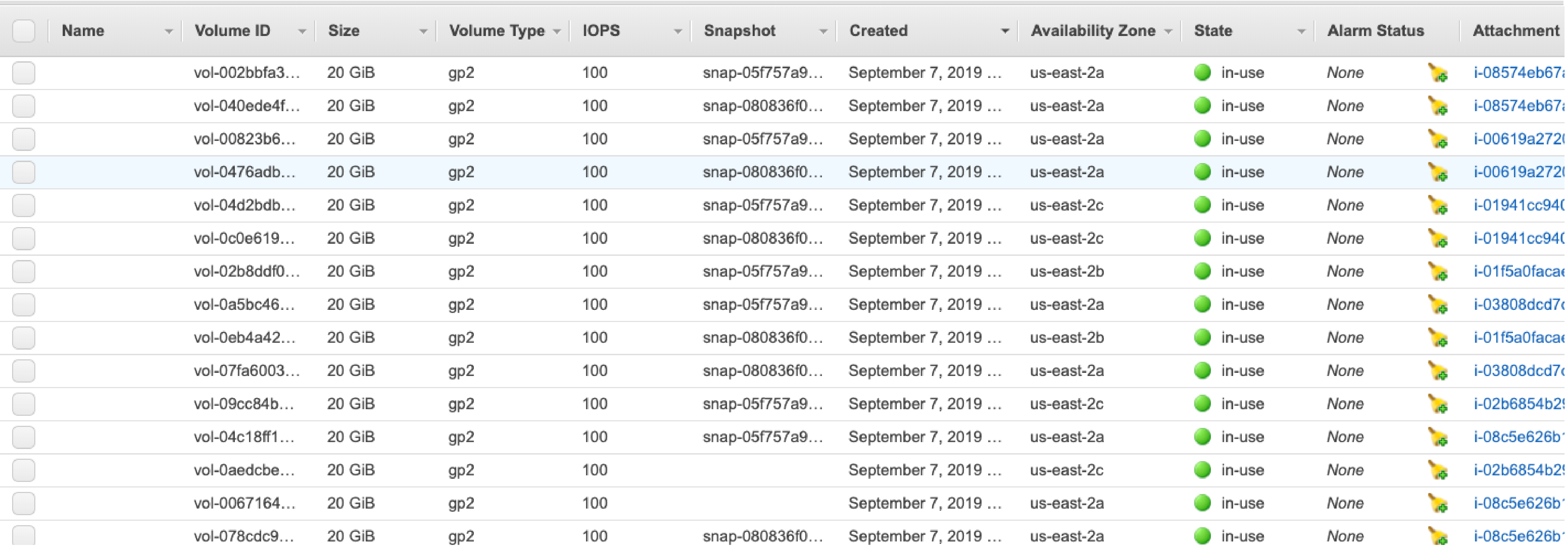
Servers at end of CloudFormation Template execution



☐	Name	Instance ID	Instance Type	Availability Zone	Instance State	IPv4 Public IP	Key Name	Launch Time
☐	cluster-master	i-08d3856d91848bcc8	t2.micro	us-east-2a	● running	18.222.112.47	awslab	September 7, 2019 at 10:43:...
☐	deployer	i-09b6a91dd3c4b6f19	t2.micro	us-east-2a	● running	18.217.64.223	awslab	September 7, 2019 at 10:44:...
☑	dmc	i-08574eb67a7a4f294	t2.micro	us-east-2a	● running	13.59.60.248	awslab	September 7, 2019 at 10:46:...
☐	golden-image	i-0c211163a78fc7f05	t2.micro	us-east-2b	● stopped	-	awslab	September 7, 2019 at 10:37:...
☐	hf1	i-00619a2720e3e3c9b	t2.micro	us-east-2a	● running	18.223.203.151	awslab	September 7, 2019 at 10:45:...
☐	idx1	i-08c5e626b1ffd445d	t2.micro	us-east-2a	● running	3.15.211.20	awslab	September 7, 2019 at 10:44:...
☐	idx2	i-04e8273d195a296...	t2.micro	us-east-2b	● running	18.217.133.97	awslab	September 7, 2019 at 10:44:...
☐	idx3	i-02b6854b29b995a6c	t2.micro	us-east-2c	● running	18.223.160.252	awslab	September 7, 2019 at 10:44:...
☐	sh1	i-03808dcd7d0c5caae	t2.micro	us-east-2a	● running	3.16.81.69	awslab	September 7, 2019 at 10:44:...
☐	sh2	i-01f5a0facae43b9ca	t2.micro	us-east-2b	● running	18.191.182.91	awslab	September 7, 2019 at 10:44:...
☐	sh3	i-01941cc94002c2cfd	t2.micro	us-east-2c	● running	18.219.215.47	awslab	September 7, 2019 at 10:45:...

Running EBS Volumes – across 3 AZs

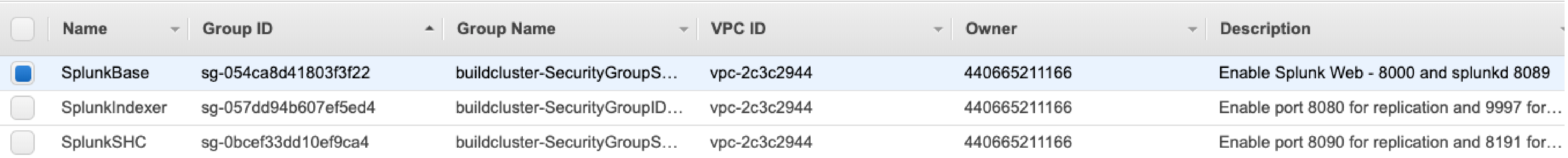
Disk Volumes after Execution of Script



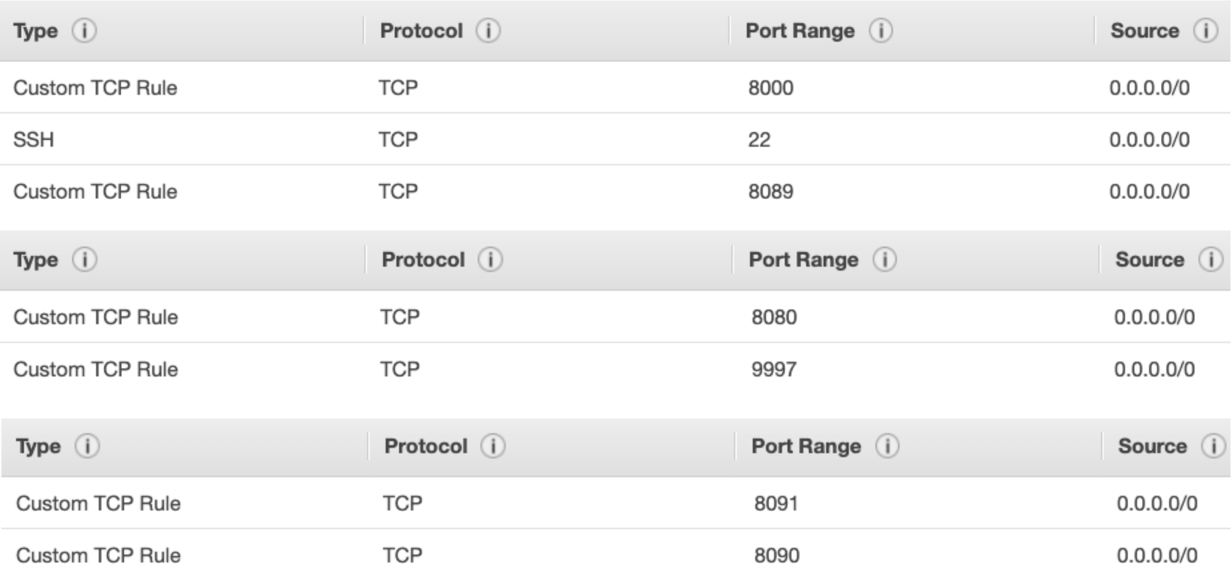
□	Name	Volume ID	Size	Volume Type	IOPS	Snapshot	Created	Availability Zone	State	Alarm Status	Attachment
□		vol-002bbfa3…	20 GiB	gp2	100	snap-05f757a9…	September 7, 2019 …	us-east-2a	● in-use	None	i-08574eb67
□		vol-040ede4f…	20 GiB	gp2	100	snap-080836f0…	September 7, 2019 …	us-east-2a	● in-use	None	i-08574eb67
□		vol-00823b6…	20 GiB	gp2	100	snap-05f757a9…	September 7, 2019 …	us-east-2a	● in-use	None	i-00619a272
□		vol-0476adb…	20 GiB	gp2	100	snap-080836f0…	September 7, 2019 …	us-east-2a	● in-use	None	i-00619a272
□		vol-04d2bdb…	20 GiB	gp2	100	snap-05f757a9…	September 7, 2019 …	us-east-2c	● in-use	None	i-01941cc94
□		vol-0c0e619…	20 GiB	gp2	100	snap-080836f0…	September 7, 2019 …	us-east-2c	● in-use	None	i-01941cc94
□		vol-02b8ddf0…	20 GiB	gp2	100	snap-05f757a9…	September 7, 2019 …	us-east-2b	● in-use	None	i-01f5a0faca
□		vol-0a5bc46…	20 GiB	gp2	100	snap-05f757a9…	September 7, 2019 …	us-east-2a	● in-use	None	i-03808dcd7
□		vol-0eb4a42…	20 GiB	gp2	100	snap-080836f0…	September 7, 2019 …	us-east-2b	● in-use	None	i-01f5a0faca
□		vol-07fa6003…	20 GiB	gp2	100	snap-080836f0…	September 7, 2019 …	us-east-2a	● in-use	None	i-03808dcd7
□		vol-09cc84b…	20 GiB	gp2	100	snap-05f757a9…	September 7, 2019 …	us-east-2c	● in-use	None	i-02b6854b2
□		vol-04c18ff1…	20 GiB	gp2	100	snap-05f757a9…	September 7, 2019 …	us-east-2a	● in-use	None	i-08c5e626b
□		vol-0aedcbe…	20 GiB	gp2	100		September 7, 2019 …	us-east-2c	● in-use	None	i-02b6854b2
□		vol-0067164…	20 GiB	gp2	100		September 7, 2019 …	us-east-2a	● in-use	None	i-08c5e626b
□		vol-078cdc9…	20 GiB	gp2	100	snap-080836f0…	September 7, 2019 …	us-east-2a	● in-use	None	i-08c5e626b

Security Groups

Security Groups after execution of CloudFormation Template



□	Name	Group ID	Group Name	VPC ID	Owner	Description
☑	SplunkBase	sg-054ca8d41803f3f22	buildcluster-SecurityGroupS…	vpc-2c3c2944	440665211166	Enable Splunk Web - 8000 and splunkd 8089
□	SplunkIndexer	sg-057dd94b607ef5ed4	buildcluster-SecurityGroupID…	vpc-2c3c2944	440665211166	Enable port 8080 for replication and 9997 for…
□	SplunkSHC	sg-0bcef33dd10ef9ca4	buildcluster-SecurityGroupS…	vpc-2c3c2944	440665211166	Enable port 8090 for replication and 8191 for…



SplunkBase – Inbound Rules Applied to All Splunk Servers

Type	Protocol	Port Range	Source
Custom TCP Rule	TCP	8000	0.0.0.0/0
SSH	TCP	22	0.0.0.0/0
Custom TCP Rule	TCP	8089	0.0.0.0/0

SplunkIndexer – Inbound Rules Applied to Indexers

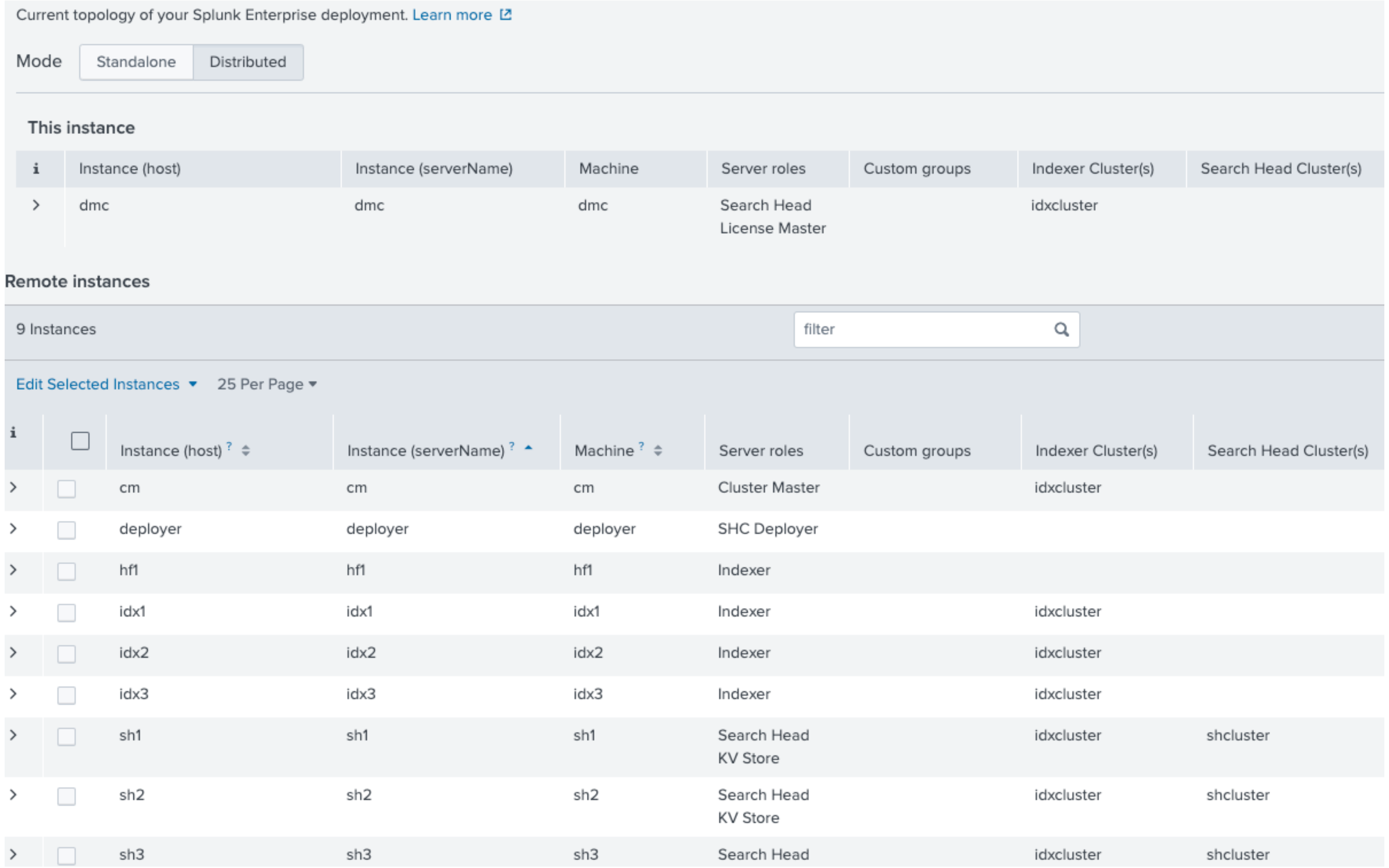
Type	Protocol	Port Range	Source
Custom TCP Rule	TCP	8080	0.0.0.0/0
Custom TCP Rule	TCP	9997	0.0.0.0/0

SplunkSHC – Inbound Rules Applied to all Non Indexers

Type	Protocol	Port Range	Source
Custom TCP Rule	TCP	8091	0.0.0.0/0
Custom TCP Rule	TCP	8090	0.0.0.0/0

Manually Configure Management Console

Management Console after Manual Configuration



Current topology of your Splunk Enterprise deployment. Learn more

Mode: Standalone | Distributed

This instance

i	Instance (host)	Instance (serverName)	Machine	Server roles	Custom groups	Indexer Cluster(s)	Search Head Cluster(s)
›	dmc	dmc	dmc	Search Head License Master		idxcluster	

Remote instances

9 Instances

filter

Edit Selected Instances ▾ 25 Per Page ▾

i	□	Instance (host) ?	Instance (serverName) ? ▲	Machine ?	Server roles	Custom groups	Indexer Cluster(s)	Search Head Cluster(s)
›	□	cm	cm	cm	Cluster Master		idxcluster	
›	□	deployer	deployer	deployer	SHC Deployer			
›	□	hf1	hf1	hf1	Indexer			
›	□	idx1	idx1	idx1	Indexer		idxcluster	
›	□	idx2	idx2	idx2	Indexer		idxcluster	
›	□	idx3	idx3	idx3	Indexer		idxcluster	
›	□	sh1	sh1	sh1	Search Head KV Store		idxcluster	shcluster
›	□	sh2	sh2	sh2	Search Head KV Store		idxcluster	shcluster
›	□	sh3	sh3	sh3	Search Head		idxcluster	shcluster

AWS CloudFormation Demo Deconstructed

Using CloudFormation to create a Splunk Golden Image and Lab Environment

splunk> .conf19

CloudFormation Automation

How the Demo Environment was Constructed

Launch a CloudFormation Template that performs the following

- Define VPC and Security Groups
- Create a Golden Image
  - Start with AMI Linux
  - Install Splunk, Configure Linux System, setup OS environment
  - Create Golden Image AMI
- Install Machines using the Golden Image AMI and configure systems in this order
  - Cluster Master
  - Indexer 1, Indexer 2, and Indexer 3
  - Deployer
  - Search Head 1, Search Head 2, Search Head 3
  - Deployment Server / Distributed Monitoring Console (DMC)

Decisions required to Bring up the Splunk Stack

Many parameters need to be set when the Cloud Formation Script is run

- What is the base AMI that is to be used to create the Splunk Golden Image
- Instance type(s) to launch EC2 instances
- What VPC will the instances be created in
- Size and Type of Disks to use
- What IAM Policy will be applied to these machines
- What Security Group(s) will be used to control access to the instances
- What AWS Key is to be used for SSH access to this Instance
- Name of the S3 bucket used to store Splunk install scripts
- Name of the tarball used to install Splunk
- Name of the directory that Splunk is to be installed into
- Name of the *NIX user that Splunk is to be installed as
- Name of the Splunk Admin user and password

Cloud Formation Template Sections

docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/template-anatomy.html

• Format Version (optional)	- Version that the template conforms to.
• Description (optional)	- Describe what the template is for.
• Metadata (optional)	- Information that can be referenced by template.
• Parameters (optional)	- Values to pass to template at runtime.
• Mappings (optional)	- Similar to a lookup table
• Conditions (optional)	- Circumstances under which entities are created
• Transforms (optional)	- Specifies macros that process the Template.
• Outputs (optional)	- Output produced by the template
• Resources (required)	- Describes stack resources and properties

** Templates can be in either JSON or YAML Format.

** The AWS CloudFormation Designer makes it easy to switch between formats.

Cloud Formation Sections

Parameters to set variables and Outputs displays them

```
Parameters:
  IndexerInstanceType:
    Description: Instance type to launch EC2 Indexer
    Type: String
    Default: t2.micro
    AllowedValues:
      - ‘t2.micro’
      - ‘m5.large’
      - ‘m5.xlarge’
      - ‘m5.4xlarge’
      - ‘m5.16xlarge’
```

Assign a Value to a user variable

Read it with the !Ref function.

```
Outputs:
  StackVPC:
    Description: The ID of the VPC
    Value: !Ref MyVPC
    Export:
      Name: !Sub "${AWS::StackName}-VPCID”
```

Create Output and export for other stacks to use

Other Cloud Formation Details

Mappings are like lookup tables and Conditions make decisions

Mappings:

```
RegionMap:
  us-east-1:
    "HVM64":  "ami-0ff8a91507f77f867"
  us-west-1:
    "HVM64":  "ami-0bdb828fd58c52235”
```

Define a different AMI for every Availability Zone

Conditions:

```
CreateSHC: !Equals
  - !Ref SHCEnabled
  - 'yes’
```

Resources:

```
  SplunkSHCMember1:
    Type: 'AWS::EC2::Instance'
    Condition:  CreateSHC
```

Set a variable based on a parameter and use it in a Resource Section

Intrinsic Functions:

* Used in Resources sections to assign values that are not available until runtime.

Returns a Base64 encoded string

Fn::Base64: *valueToEncode*

!Base64 *valueToEncode*

Substitutes variables in an input string

Fn::Sub: - *String*

!Sub - *String*

Define VPC and Security Groups

Create Security Groups

Security Group Applied to All Splunk Instances

```
SecurityGroupSplunkBase:
  Type: 'AWS::EC2::SecurityGroup’
  Properties:
    VpcId: !Ref VPCID
    GroupDescription: Enable Splunk Web - 8000 and Splunkd 8089
    SecurityGroupIngress:
      - IpProtocol: tcp
        FromPort: 8000
        ToPort: 8000
        CidrIp: '0.0.0.0/0’
      - IpProtocol: tcp
        FromPort: 8089
        ToPort: 8089
        CidrIp: '0.0.0.0/0’
      - IpProtocol: tcp
        FromPort: 22
        ToPort: 22
        CidrIp: '0.0.0.0/0’
```

Create Golden Image

Resources Section to create our Golden Image

With Default Values

Define EC2

Define Storage

Define Security

```
Resources:
        Instance:
                Type: "AWS::EC2::Instance"
                Properties:
                        ImageId: ami-00c79db59589996b9
                        InstanceType: t2.micro
                        KeyName: awslab
                        BlockDeviceMappings:
                                - Ebs:
                                        DeleteOnTermination: True
                                        Encrypted: True
                                        VolumeSize: 20
                                        VolumeType: gp2
                                DeviceName: /dev/sdb
                        SecurityGroups:
                                - SecurityGroupSplunkBase
                        IamInstanceProfile: splunk
```

Parameterized

Define EC2

Define Storage

Define Security

```
Resources:
        Instance:
                Type: "AWS::EC2::Instance"
                Properties:
                        ImageId: !Ref AmiBaseId
                        InstanceType: !Ref InstanceType
                        BlockDeviceMappings:
                                - Ebs:
                                        DeleteOnTermination: True
                                        Encrypted: True
                                        VolumeSize: 20
                                        VolumeType: gp2
                                DeviceName: /dev/sdb
                        KeyName: !Ref KeyName
                        SecurityGroups:
                                - !Ref SecurityGroup
                        IamInstanceProfile: !Ref IamInstanceProfile
```

Create Golden Image

UserData Section – Just a Bash Script – Set Env variables, copy script from S3, run script, call AMICreate

UserData

```
UserData:
        "Fn::Base64":
                !Sub |
                        #!/bin/bash -v
                        export SPLUNK_HOME=${SplunkHome}
                        export AWS_S3_BUCKET=${AwsS3Bucket}
                        export SPLUNK_TARBALL=${SplunkTarball}
                        export SPLUNK_SYSTEM_USER=${SplunkSystemUser}
                        export SPLUNK_ADMIN_USER=${SplunkAdminUser}
                        export SPLUNK_ADMIN_PASSWORD=${SplunkAdminPassword}
                        aws s3 cp s3://${AwsS3Bucket}/bash/golden_image.sh /tmp/golden_image.sh
                        chmod 755 /tmp/golden_image.sh
                        sudo -E /tmp/golden_image.sh
                        rm -f /tmp/golden_image.sh
                        /opt/aws/bin/cfn-signal \
                        -e $? \
                        --stack ${AWS::StackName} \
                        --region ${AWS::Region} \
                        --resource AMICreate
                        shutdown -h now
```

Set Env vars

Run Setup Script

Build AMI

Default Values

/opt/splunk
ajs-aws-splunk
splunk-7….-Linux-x86_64.tgz
splunk
Admin
Changeme1
ajs-aws-splunk

Defined at Runtime
Defined at Runtime

Golden Image Bash Script 1of 4

Setup Environment and get ready to Install Splunk

Update System

```
#!/bin/bash -v

chmod 600 /var/log/cloud-init-output.log
yum update -y aws-cfn-bootstrap
yum install -y jq
```

Add Splunk User

```
adduser $SPLUNK_SYSTEM_USER --comment "Splunk User" --system --create-home --shell /sbin/nologin
usermod --expiredate 1 $SPLUNK_USER
```

Mount Splunk Drive

```
mkdir $SPLUNK_HOME
mkfs -t xfs /dev/sdb
echo "/dev/sdb $SPLUNK_HOME xfs defaults,nofail 0 2" >> /etc/fstab
mount -a
```

Copy and untar Splunk

```
aws s3 cp s3://${AWS_S3_BUCKET}/code/${SPLUNK_TARBALL} /tmp
tar -xzf /tmp/${SPLUNK_TARBALL} -C $SPLUNK_HOME --strip-components=1
rm -f /tmp/${SPLUNK_TARBALL}

echo "source $SPLUNK_HOME/bin/setSplunkEnv" >> /home/splunk/.bashrc
```

Golden Image Bash Script 2 of 4

Install Splunk

Set Default User

```
echo "[user_info]" > $SPLUNK_HOME/etc/system/local/user-seed.conf
echo "USERNAME = $SPLUNK_ADMIN_USER" >> $SPLUNK_HOME/etc/system/local/user-seed.conf
echo "PASSWORD = $SPLUNK_ADMIN_PASSWORD" >> $SPLUNK_HOME/etc/system/local/user-seed.conf

touch $SPLUNK_HOME/etc/.ui_login
```

Start Splunk

```
chown -R $SPLUNK_SYSTEM_USER:$SPLUNK_SYSTEM_USER $SPLUNK_HOME
sudo -u $SPLUNK_SYSTEM_USER $SPLUNK_HOME/bin/splunk start --accept-license --answer-yes --no-prompt
```

Start Splunk at boot

```
$SPLUNK_HOME/bin/splunk enable boot-start -user $SPLUNK_SYSTEM_USER
```

Golden Image Bash Script 3 of 4

Disable THP and Setup Limits

Configure init.d Script

```
cat << EOF > /tmp/initd-update
disable_thp() {
echo "Disabling transparent huge pages"
If test -f /sys/kernel/mm/transparent_hugepage/enabled; then
   echo never > /sys/kernel/mm/transparent_hugepage/enabled
fi
if test -f /sys/kernel/mm/transparent_hugepage/defrag; then
   echo never > /sys/kernel/mm/transparent_hugepage/defrag
fi
}
change_ulimit() {
   ulimit -Sn 65535
   ulimit -Hn 65535
   ulimit -Su 20480
   ulimit -Hu 20480
   ulimit -Sf unlimited
   ulimit -Hf unlimited
}
EOF
sed -i "/init\.d\/functions/r /tmp/initd-update" /etc/init.d/splunk
sed -i "/start)$/a \ disable_thp\n change_ulimit" /etc/init.d/splunk
rm /tmp/initd-update
```

Golden Image Bash Script 4 of 4

Disable THP and Setup Limits

Configure Limits

```
# Create 25-splunk.conf in limits.d to set ulimits when not using systemctl
echo "$SPLUNK_SYSTEM_USER hard core 0" >> /etc/security/limits.d/25-splunk.conf
echo "$SPLUNK_SYSTEM_USER hard maxlogins 10" >> /etc/security/limits.d/25-splunk.conf
echo "$SPLUNK_SYSTEM_USER soft nofile 65535" >> /etc/security/limits.d/25-splunk.conf
echo "$SPLUNK_SYSTEM_USER hard nofile 65535" >> /etc/security/limits.d/25-splunk.conf
echo "$SPLUNK_SYSTEM_USER soft nproc 20480" >> /etc/security/limits.d/25-splunk.conf
echo "$SPLUNK_SYSTEM_USER hard nproc 20480" >> /etc/security/limits.d/25-splunk.conf
echo "$SPLUNK_SYSTEM_USER soft fsize unlimited" >> /etc/security/limits.d/25-splunk.conf
echo "$SPLUNK_SYSTEM_USER hard fsize unlimited" >> /etc/security/limits.d/25-splunk.conf
```

Prepare system for cloning

```
$SPLUNK_HOME/bin/splunk stop
$SPLUNK_HOME/bin/splunk clone-prep-clear-config
rm -f $SPLUNK_HOME/var/log

systemctl daemon-reload
```

Create Golden Image AMI

Create AMI using CreateAmI script – uses Lambda Functions

https://stackoverflow.com/questions/21431450/create-ami-image-as-part-of-a-cloudformation-stack

Bring up an Indexer

Resources Section to bring up an Indexer

Wait for Cluster Master

```
DependsOn: CM
  Type: 'AWS::EC2::Instance'
  Properties:
```

Define EC2

```
    DisableApiTermination: !Ref DisableApiTermination
    IamInstanceProfile:  !Ref IamInstanceProfile
    ImageId:  !ImportValue  splunk-golden-ami
    AvailabilityZone:  us-east-2a
     InstanceType:  !Ref InstanceType
     KeyName: !Ref KeyName
     Tags:
       - Key: Application
        Value: !Ref 'AWS::StackId'
       - Key: Role
         Value: indexer
       - Key: Name
         Value: idx1
```

Define Network

```
     NetworkInterfaces:
         - GroupSet:
           - !Ref SecurityGroupSplunkBase
            - !Ref SecurityGroupIDXCluster
          AssociatePublicIpAddress: !Ref AssociatePublicIpAddress
          DeviceIndex:  '0'
          DeleteOnTermination:  true
```

Define Storage

```
BlockDeviceMappings:
   - DeviceName: /dev/xvda
    Ebs:
      VolumeType:  gp2
      VolumeSize:  20
      DeleteOnTermination:  !Ref DeleteOnTermination
   - DeviceName: /dev/sdb
    Ebs:
      VolumeType:  !Ref IDXHotVolumeType
      VolumeSize:  !Ref IDXHotVolumeSize
      DeleteOnTermination:  !Ref DeleteOnTermination
   - DeviceName: /dev/sdc
    Ebs:
      VolumeType:  !Ref IDXColdVolumeType
      VolumeSize:  Ref IDXColdVolumeSize
      DeleteOnTermination:  !Ref DeleteOnTermination
```

Bring up an Indexer – UserData section

UserData Script – Similar BASH for All Roles - Set Env variables, copy Indexer script from S3, run script

Define script using 2 function

```
UserData:
  "Fn::Base64":
    !Sub |
```

Set Env Vars

```
      export hostname=idx1
      export site=site1
      export ClusterMasterPrivateIp=${CM.PrivateIp}
      export SPLUNK_HOME=${SplunkHome}
      export AWS_S3_BUCKET=${AwsS3Bucket}
      export SPLUNK_SYSTEM_USER=${SplunkSystemUser}
      export SPLUNK_ADMIN_USER=${SplunkAdminUser}
      export SPLUNK_ADMIN_PASSWORD=${SplunkAdminPassword}
      export SPLUNK_GENERAL_SECRET=${SplunkGeneralSecret}
      export SPLUNK_CLUSTER_SECRET=${SplunkClusterSecret}
      export
SPLUNK_INDEX_DISCOVERY_SECRET=${SplunkIndexDiscoverySecret}
      export NumberOfAZs=${NumberOfAZs}
      export INDEX_CLUSTER_LABEL=${IndexClusterLabel}
      export SEARCH_CLUSTER_LABEL=${SearchClusterLabel}
```

Run Setup Script

```
      aws s3 cp s3://${AwsS3Bucket}/bash/idx.sh /tmp/idx.sh
      chmod 755 /tmp/idx.sh
      sudo -E /tmp/idx.sh
      rm -f /tmp/idx.sh
```

Bringing Splunk to Life

Minimum Base Configs to establish a Multi-site Distributed Splunk Cluster

splunk> .conf19

Bringing Up the Full Splunk Stack

Launch resources in the correct order using the Golden Image AMI and configure it using CLI commands

1 Build Cluster Master

2 Once Cluster Master is up

- Build Indexer 1
- Build Indexer 2
- Build Indexer 3
- Build DEPLOYER

3 Once Deployer is up

- Build Search Head 1
- Build Search Head 2

4 Once Search Head 2 is up

- Build Search Head 3 and bootstrap SHC

5 Once Search Head 3 is up

- Build Deployment Server

6 Once the Distributed Monitoring Console is up

- Configure DMC

Cluster Master – Minimum Config

Created in $SPLUNK_HOME/system/local

inputs.conf

```
[default]
host = cm
```

outputs.conf

```
[indexAndForward]
index = false

[tcpout]
defaultGroup = indexers

[tcpout:indexers]
indexerDiscovery = cluster_master
useACK = true

[indexer_discovery:cluster_master]
pass4SymmKey = $7$82oYYXA…<Redacted>…=
master_uri = https://127.0.0.1:8089
```

server.conf

```
[general]
pass4SymmKey = $7$Nv6+gqS…<Redacted>…=
serverName = cm
site = site1

[clustering]
available_sites = site1,site2,site3
cluster_label = idxcluster
mode = master
multisite = true
pass4SymmKey = $7$RBXxzg3...<Redacted>…=
site_replication_factor = origin:1,total:3
site_search_factor = origin:1,total:2

[indexer_discovery]
pass4SymmKey = $7$vhLnvpg…<Redacted>…=
indexerWeightByDiskCapacity = true
```

Clustered Indexer – Minimum Config

Created in $SPLUNK_HOME/system/local

inputs.conf

```
[default]
host = idx1

[splunktcp://9997]
```

server.conf

```
[general]
pass4SymmKey = $7$A6JoK5/...Redacted...=
serverName = idx1
site = site1

[replication_port://8080]

[clustering]
master_uri = https://172.31.13.128:8089
mode = slave
multisite = true
pass4SymmKey = $7$kErCFNG/...Redacted...=
```

Deployer – Minimum Config

Created in $SPLUNK_HOME/system/local

inputs.conf

```
[default]
host = deployer
```

outputs.conf

```
[indexAndForward]
index = false

[tcpout]
defaultGroup = indexers

[tcpout:indexers]
indexerDiscovery = cluster_master
useACK = true

[indexer_discovery:cluster_master]
pass4SymmKey = $7$zbDBWIZ…<Redacted>…=
master_uri = https://172.31.13.128:8089
```

server.conf

```
[general]
pass4SymmKey = $7$H1T9EhJ…<Redacted>…=
serverName = deployer
site = site0

[shclustering]
pass4SymmKey = $7$z0WXO24…<Redacted>…=
shcluster_label = shcluster

[clustering]
master_uri = https://172.31.13.128:8089
mode = searchhead
multisite = true
pass4SymmKey = $7$9mjcylv…<Redacted>…=
```

Clustered Search Head – Minimum Config

Created in $SPLUNK_HOME/system/local

inputs.conf

```
[default]
host = sh3
```

outputs.conf

```
[indexAndForward]
index = false

[tcpout]
defaultGroup = indexers

[tcpout:indexers]
indexerDiscovery = cluster_master
useACK = true

[indexer_discovery:cluster_master]
pass4SymmKey = $7$DzS29EO…<Redacted>…=
master_uri = https://172.31.13.128:8089
```

server.conf

```
[general]
pass4SymmKey = $7$cEehTAX…<Redacted>…=
serverName = sh3
site = site0

[clustering]
master_uri = https://172.31.13.128:8089
mode = searchhead
multisite = true
pass4SymmKey = $7$59deVBV…<Redacted>…=

[replication_port://8090]

[shclustering]
conf_deploy_fetch_url = https://172.31.14.197:8089
mgmt_uri = https://172.31.35.193:8089
pass4SymmKey = $7$220x3L5…<Redacted>…=
shcluster_label = shcluster
```

Deployment Server / DMC – Minimum Config

Created in $SPLUNK_HOME/system/local. Note: This is same configuration as a Heavy Forwarder

inputs.conf

```
[default]
host = ds_dmc
```

outputs.conf

```
[indexAndForward]
index = false

[tcpout]
defaultGroup = indexers

[tcpout:indexers]
indexerDiscovery = cluster_master
useACK = true

[indexer_discovery:cluster_master]
pass4SymmKey = $7$kRKu2oA…<Redacted>…=

master_uri = https://172.31.13.128:8089
```

server.conf

```
[general]
pass4SymmKey = $7$MEUEAft...<Redacted>…=
serverName = dmc
site = site0

[clustering]
master_uri = https://172.31.13.128:8089
mode = searchhead
multisite = true
pass4SymmKey = $7$K4g0rXi...<Redacted>…=
```

Whats Next

What to build on your Splunk Environment and how to Operationalize it into Production.

Configuration Options are Endless

Now that you have built Splunk in less than 30-minutes what are you going to do with it

Development Server

- Stand Alone, Distributed, full blown cluster, or anything imaginable
- System to test any new Splunk features, applications, or custom designs.
- System With Pre Populated Data – Eventgen Load Generator

Production Environment

- Ad Hoc Splunk Search Environment with pre-installed apps
- Enterprise Security (ES) System
- IT Service Intelligence (ITSI) System
- Pre-Configured Heavy forwarders, enrichment and jobs servers, multi tenant portals

Special use systems

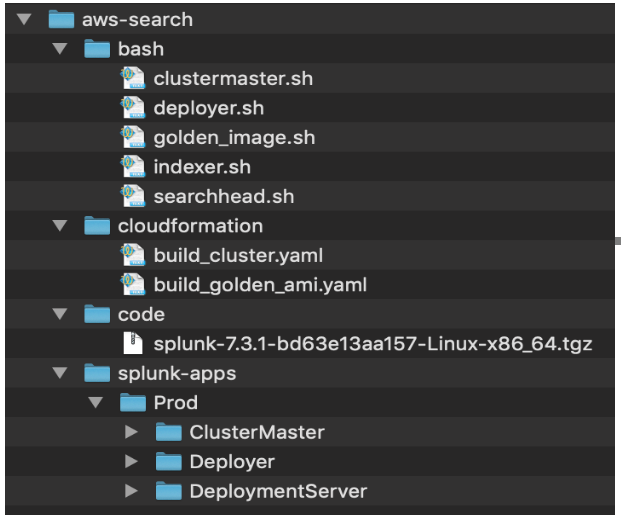
- System optimized to perform Forensic Analysis or Pen Testing
- Machine Learning Toolkit (MLTK) system
- Stream collection node

GIT

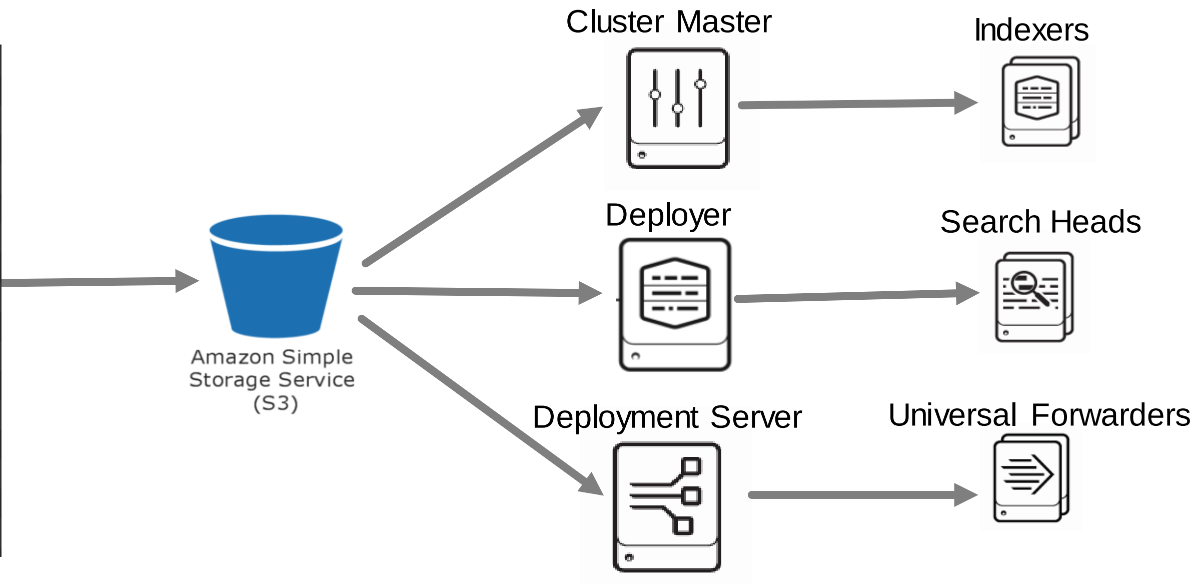
GIT is Truth

- Distributed Version control – Keep track of constant revisions to your code.
- Quickly change state between application environments.
- Continuous Integration and Continuous Deployment (CICD)

GIT Repo



```
▼ aws-search
    ▼ bash
        clustermaster.sh
        deployer.sh
        golden_image.sh
        indexer.sh
        searchhead.sh
    ▼ cloudformation
        build_cluster.yaml
        build_golden_ami.yaml
    ▼ code
        splunk-7.3.1-bd63e13aa157-Linux-x86_64.tgz
    ▼ splunk-apps
        ▼ Prod
            ▶ ClusterMaster
            ▶ Deployer
            ▶ DeploymentServer
```



Operationalizing Splunk for Production

Some things to think of as the Splunk Environment is prepared for production use

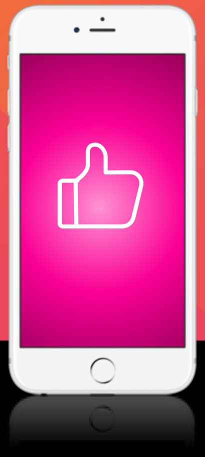
- Build the Golden image and Splunk instances according to your companies security policies preferably starting the process from a baseline AMI.
- Tier CloudFormation stacks by separating VPC, security group, and IAM from Resource creation.
- Optimize Instance EC2 and Storage size for your needs
- Secure Environment with Certificates - Web, Forwarder, management, and connectivity
- Implement a Load Balancer for SHC and HEC access
- Deploy Universal and Heavy Forwarders and configure HEC to onboard data
- Configure S3 Glacier buckets to store frozen data.
- Implement System Manager to manage Splunk
- Deploy a Bastion Host and Jump servers to control access all Instance shells
- Implement Ansible/Chef/Puppet/Jenkins/etc to support CI/CD
- Implement a monitoring & maintenance plan to quantify operation of the system
- BCDR plans and testing

Q&A

Arthur Spencer | Sr Splunk PS Security Consultant | Splunk

Neha Doshi | Splunk Practice Lead / Sr PS Consultant | Perficient

Thank You!



Go to the .conf19 mobile app to

RATE THIS SESSION

Appendix

Some information to help you on your learning path - then read everything you can find ☺

Git Repo for this project: https://github.com/arthurjspencer/aws_splunk_fn2195_cloudformation.git

Splunk Resources

- https://conf.splunk.com/files/2016/slides/best-practices-for-deploying-splunk-on-amazon-web-services.pdf
- https://www.splunk.com/pdfs/technical-briefs/deploying-splunk-enterprise-on-aws.pdf

AWS Resources

- https://aws-quickstart.s3.amazonaws.com/quickstart-splunk-enterprise/doc/splunk-enterprise-on-the-aws-cloud.pdf
- https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/gettingstarted.templatebasics.html

GIT Resources

- GIT Tutorial: https://product.hubspot.com/blog/git-and-github-tutorial-for-beginners
- Hello World: https://guides.github.com/activities/hello-world/

Project Trumpet – Getting data in

- Use AWS CloudFormation to set up all the AWS infrastructure needed to push AWS CloudTrail, AWS Config, and AWS GuardDuty data to Splunk using HTTP Event Collector (HEC). https://github.com/splunk/splunk-aws-project-trumpet

Splunk CLI Commands

Installing Splunk, Configuring it to Start at Boot, Preparing Splunk for Imaging, Peering to Standalone Indexer

```
# Golden Image - Start Splunk for the first time
splunk start --accept-license --answer-yes --no-prompt
```

```
# Golden Image - Configure Splunk to automatically start at system boot time.
splunk enable boot-start -user $SPLUNK_SYSTEM_USER
```

```
# Golden Image - Prepare the configuration for Imaging.
splunk clone-prep-clear-config
```

```
# All Roles - Login to the Splunk Instance
splunk login -auth $SPLUNK_ADMIN_USER:$SPLUNK_ADMIN_PASSWORD
```

```
#DMC - Peer the server to another Splunk Instance usually a Standalone indexer
splunk add search-server \
        -host https://$ClusterMasterPrivateIp:8089 \
        -remoteUsername $SPLUNK_ADMIN_USER \
        -remotePassword $SPLUNK_ADMIN_PASSWORD
```

Splunk CLI Commands

Establish Cluster Master and a Clustered Indexer

```
# Cluster Master - Define the Index Cluster
splunk edit cluster-config \
                -mode master \
                -multisite true \
                -available_sites $sites \
                -site site1 \
                -site_replication_factor origin:1,total:3 \
                -site_search_factor origin:1,total:2 \
                -secret $SPLUNK_CLUSTER_SECRET \
                -cluster_label $INDEX_CLUSTER_LABEL

# Indexer - Configure an Indexer to be a member of a Cluster
splunk edit cluster-config \
                -mode slave \
                -site $site \
                -master_uri https://$ClusterMasterPrivateIp:8089 \
        -replication_port 8080 \
                -secret $SPLUNK_CLUSTER_SECRET
```

Splunk CLI Commands

Establish the Deployer and Peer instances into an Index Cluster

```
# Deployer - Stage the SH cluster bundle for deployment to SH
splunk apply shcluster-bundle \
             -action stage \
             --answer-yes

# Deployer - establish the Search Head Cluster Deployer Role
splunk edit cluster-config \
             -mode searchhead \
             -site site0 \
             -master_uri https://$ClusterMasterPrivateIp:8089 \
             -secret $SPLUNK_CLUSTER_SECRET

# SHC, Deployer, DMC - Peer a Search Head to an Index Cluster
splunk edit cluster-config \
   -mode searchhead \
   -site site0 \
   -master_uri https://$ClusterMasterPrivateIp:8089 \
   -secret $SPLUNK_CLUSTER_SECRET
```

Splunk CLI Commands

Establish a Search Head Cluster

```
# SHC - Add a Search Head to a Cluster
splunk init shcluster-config \
   -mgmt_uri  https://$LOCALIP:8089 \
   -replication_port 8090 \
   -replication_factor 3 \
   -conf_deploy_fetch_url https://$DeployerPrivateIp:8089 \
   -shcluster_label $SEARCH_CLUSTER_LABEL \
   -secret $SPLUNK_CLUSTER_SECRET

# SHC Master - Bootstrap the Cluster Master
splunk bootstrap shcluster-captain \
   -servers_list " \
      https://$SHCMember1PrivateIp:8089, \
      https://$SHCMember2PrivateIp:8089, \
      https://$LOCALIP:8089"
```