Deploying Splunk on AWS

Patrick Shumate, Solution Architect
Nate Kwong, Staff SE
Bill Bartlett, Senior SE

October 2018 | Version 1.0

Forward-Looking Statements

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 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.

splunk> .conf18

Objective

Learn how to architect a highly available and resilient Splunk Enterprise on the AWS Cloud by leveraging best practices from both Splunk and AWS technologies.

Today's Agenda

- Best Practices
- Hybrid Architectures
- Getting Data In (GDI)

splunk> .conf18

Expectations

- Familiarity with AWS services.
  - ~6 months experience
  - Actively running at least a development workload in AWS.
- Certified Splunk Administrator

Presenters

- Bill Bartlett
  - Sr. SE, GSA
  - Seattle
- Nate Kwong
  - Staff SE, Strategics
  - San Francisco
- Patrick Shumate
  - Solutions Architect, GSA
  - Virginia

Best Practices

Deployment Decisions

- Instance selection should factor in availability, durability, use-case, and cost
- Availability vs Durability: what's the difference?
- What if I need high availability?
  - Index replicatin / indexer clustering
  - Requires significantly more storage than a non-HA environment
- If I only need durabili9ty?
  - You must use EBS

Splunk and Hardware

- Splunk consumes high I/O due to indexing and searching
- Load != GB/day
- Search drives a large portion of the load
  - Rare vs. Sparse vs. Reporting
  - Real-time vs. Historic
- Rule of thumb – up to 300 GB/day

Storage Options

- Some instances have local ephemeral storage
  - Recommended instance types have very fast SSD attached
  - Data is lost when the instance dies
- EBS – Elastic Block Storage
  - Persistent block level storage volumes for use with EC2 instances
  - Cost associated – 1 TB gp2 costs $100/month for 3000 IOPS
  - Data is not lost when instance dies – can be remounted with new instance
  - For storage needs larger than 16 TB, RAID required
  - Built-in resiliency – data is backed up
- S3 – Simple Storage Service
  - Smart Store
  - Backups (EBS Snapshots)

Storage Best Practices

Single Instances / Non-Replicated Distributed Deployments

- Use EBS Volumes
- RAID can be an extra measure of reliability, but will consume CPU
- Use snapshots to backup the instance

Instance Selection

Distributed Deployments

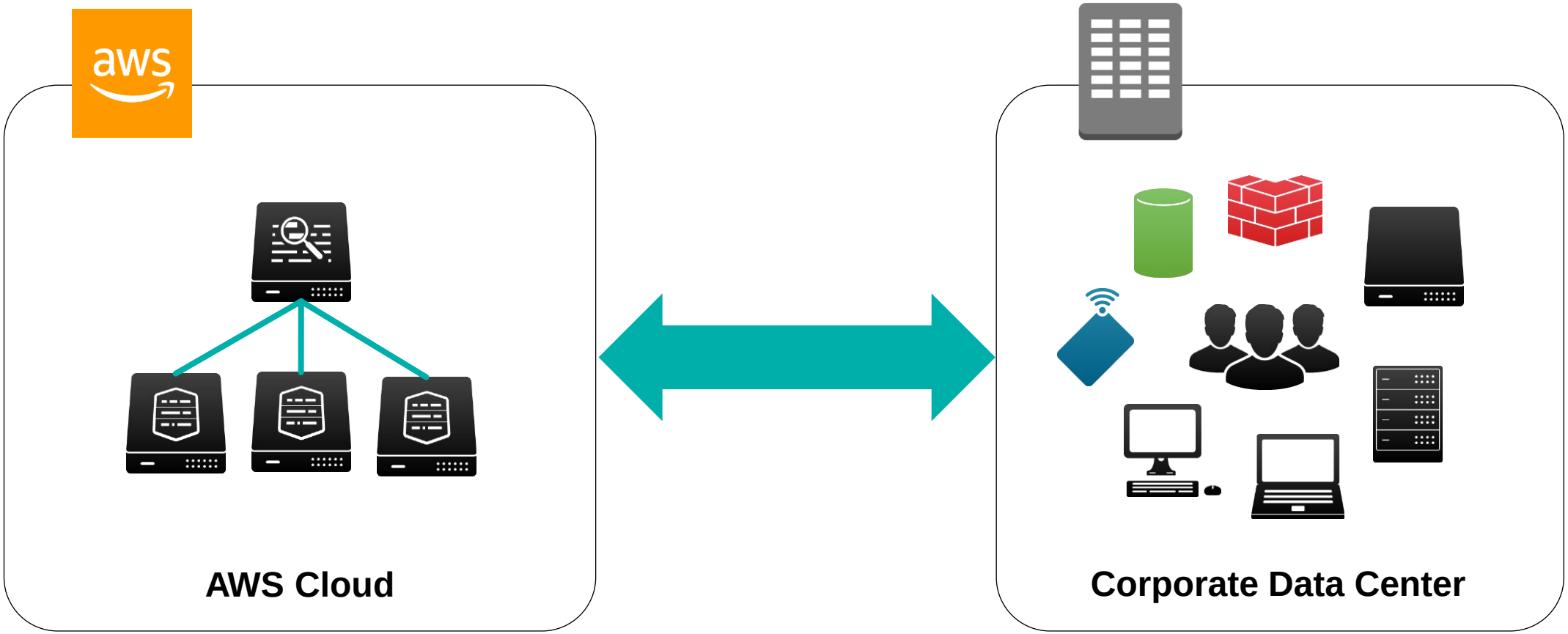
- Using Index Replication (IR)
  - Fast local SSDs may perform better than EBS
  - Search/Replication Factor determines availability of data for searching
  - IR adds load and typically requires more servers and storage
- Using EBS volumes, no IR*
  - Typically fewer instances to manage vs. IR
  - EBS durability is fantastic. (99.999%)
    - Very easy backups via EBS snapshots.
  - Availability is driven by the capability to remount a volume to a new instance (automatically or manually)
  - Cost can be largely driven by retention and daily volume
  - * You can absolutely use a IR cluster in an EBS-based deployment.

Hybrid Architectures

What is a Hybrid Architecture?

- Splunk Enterprise deployed in AWS cloud
- On-premises datacenter, labs, offices, another AWS account, etc.
- How do we connect the two together?

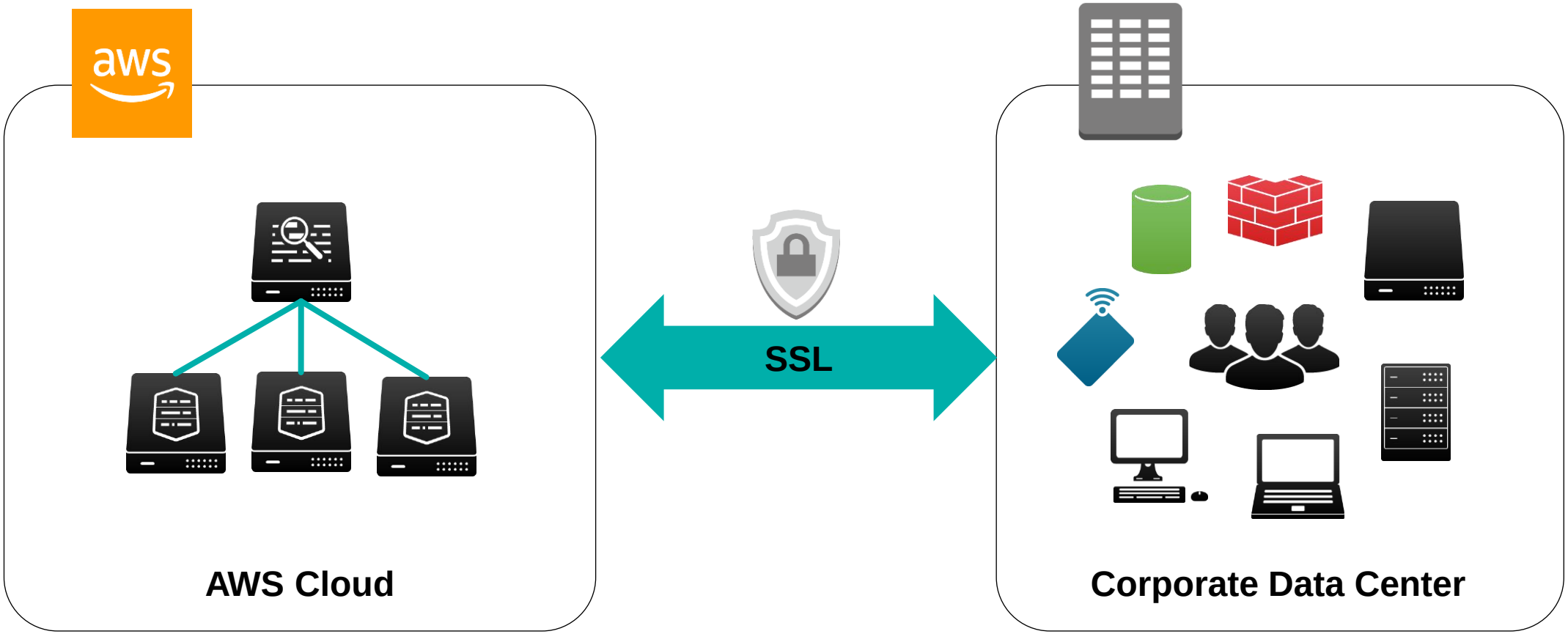
Hybrid Architecture



Hybrid Architecture Options

- Option 1: SSL with Forwarders and Search Heads
- Option 2: VPN Connection
  - Option 2a: VPN Connection – Virtual Private Gateway
  - Option 2b: VPN Connection – Software VPN Gateway
- Option 3: Direct Connect
- Option 4: VPC Peering (for customers with AWS as their datacenter)
- Bonus option: AWS PrivateLink

SSL with Forwarders and Search Heads

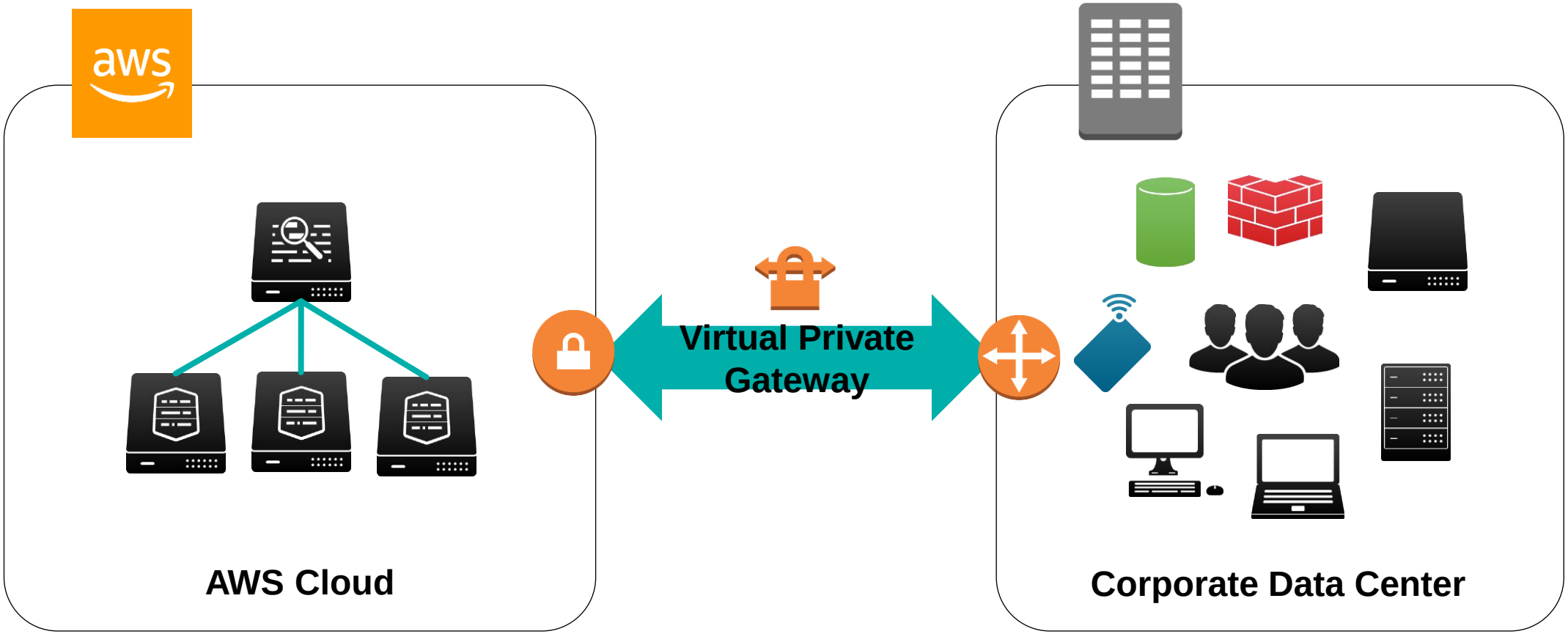


SSL with Forwarders and Search Heads

- Pros
  - Low costs
  - No extra equipment is needed
  - SSL via the forwarder provides data compression ratio up to 10:1

- Considerations
  - Recommended to use a CA-signed certificate for security purposes
  - SSL settings are not on by default for forwarders or search heads

VPN Connection – Virtual Private Gateway

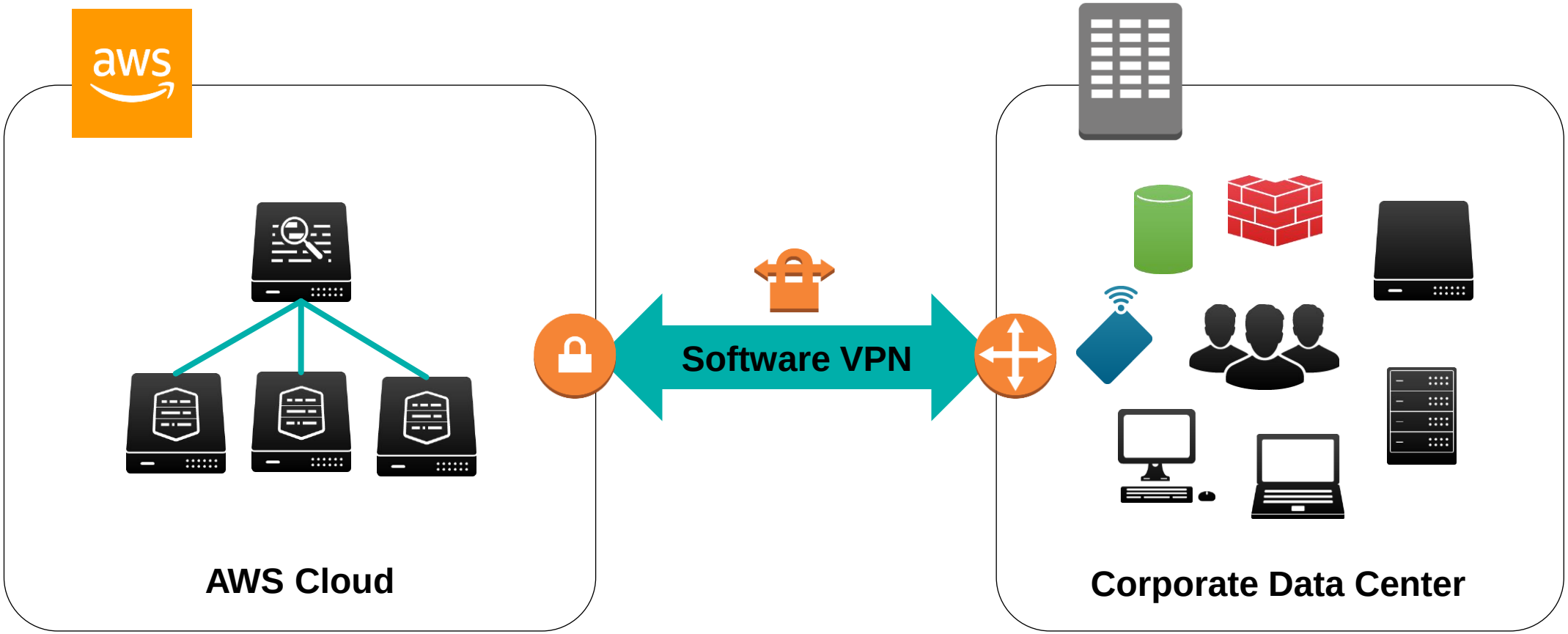


VPN Connection – Virtual Private Gateway

- Pros
  - Easy setup of the Virtual Private Gateway in AWS
  - Relatively low cost of $0.05 per VPN Connection-hour

- Considerations
  - Need a customer gateway to create an IPSEC tunnel with the Virtual Private Gateway
    - Customer Gateway Options: Cisco ISR, Juniper SRX, Palo Alto Networks PANOS, etc.
  - Limited bandwidth of up to 1.25 Gbps for a single Virtual Private Gateway
  - May need to setup redundant VPN Gateway for fault tolerance

VPN Connection – Software VPN Gateway

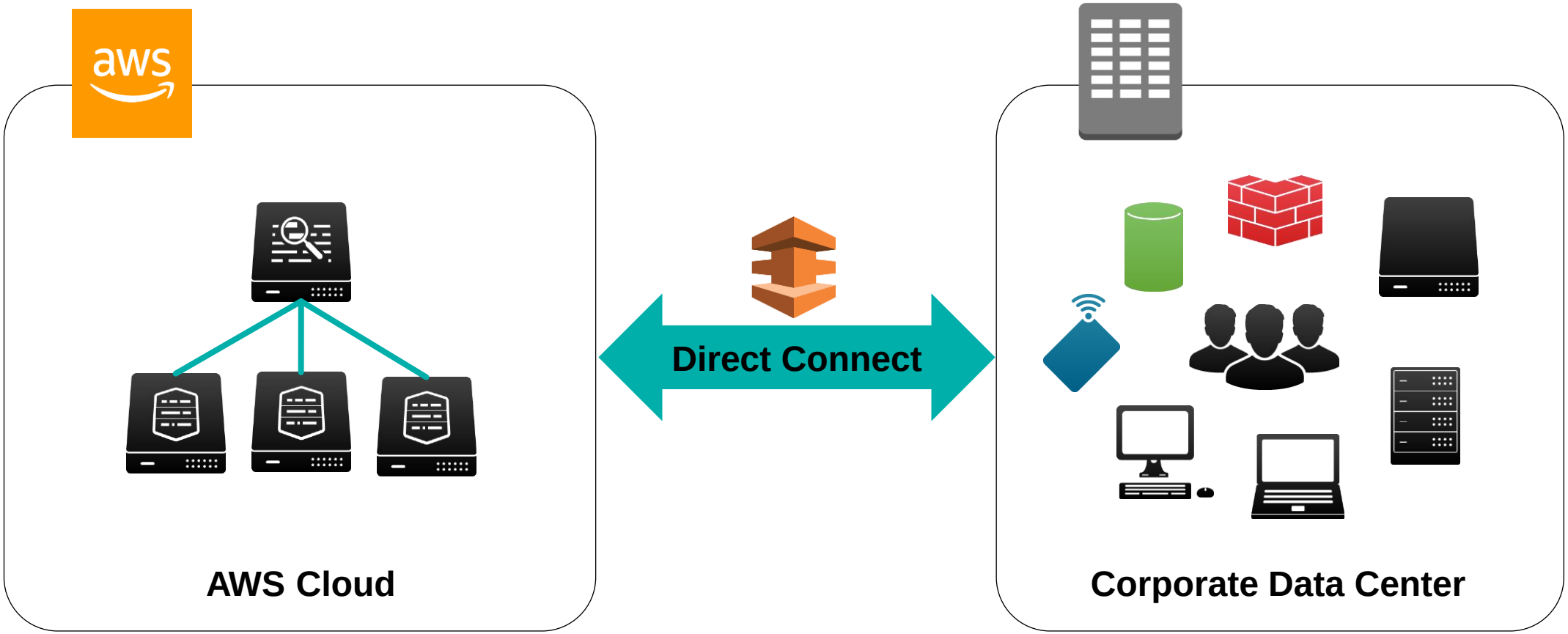


VPN Connection – Software VPN Gateway

- Pros
  - More administrative control of Software VPN Gateway
  - Can choose larger EC2 instances to increase bandwidth limit of VPN Connection

- Considerations
  - Need a customer gateway to create an IPSEC tunnel with the Software VPN Gateway
    - Customer Gateway Options: Cisco ISR, Juniper SRX, Palo Alto Networks PANOS, etc.
  - Administrative overhead of Software VPN Gateway
    - Software VPN Gateway options: Openswan, OpenVPN
  - May need to setup redundant VPN Gateway for fault tolerance

Direct Connect

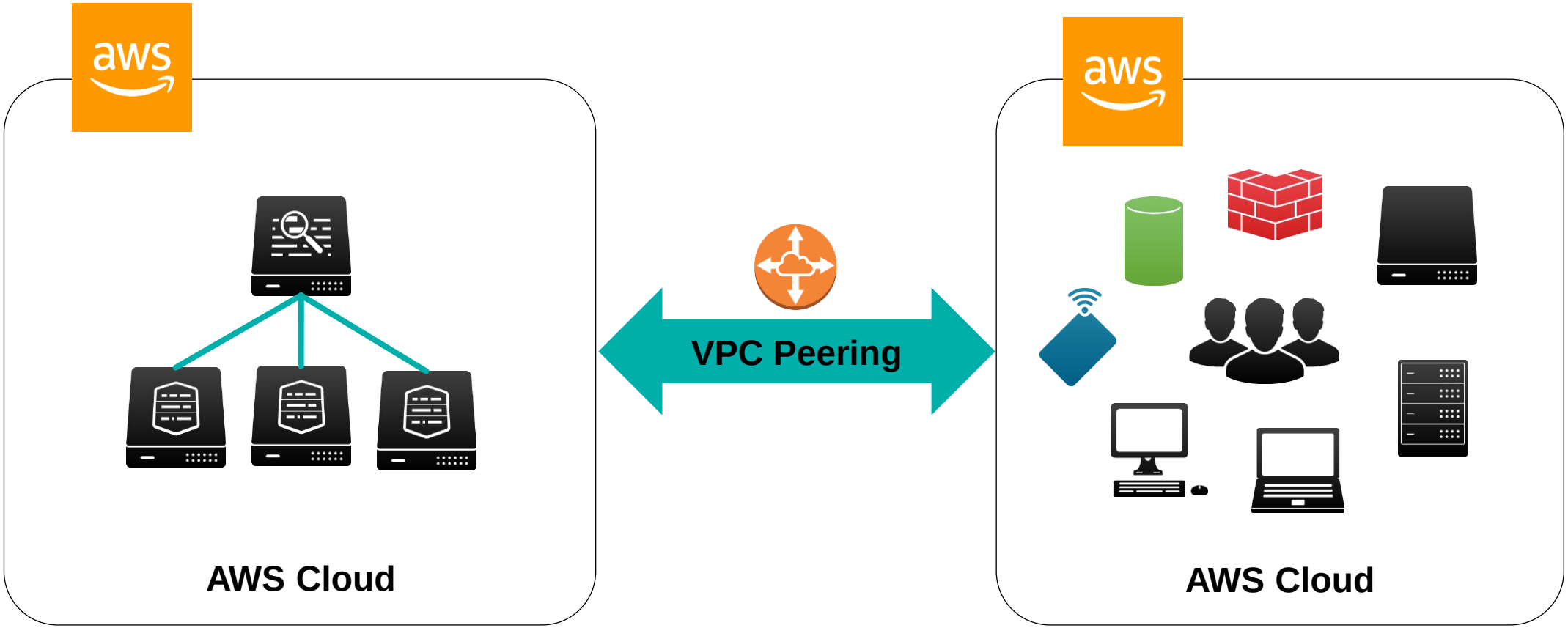


Direct Connect

- Pros
  - High bandwidth network between on-premises to AWS Cloud (from 50Mbps to 40Gbps)
  - Consistent network performance
  - Lower data transfer **out** costs

- Considerations
  - You need network equipment at a Direct Connect location or you need to leverage a partner
  - Higher port costs
  - May need to setup redundant Direct Connect connection for fault tolerance

VPC Peering

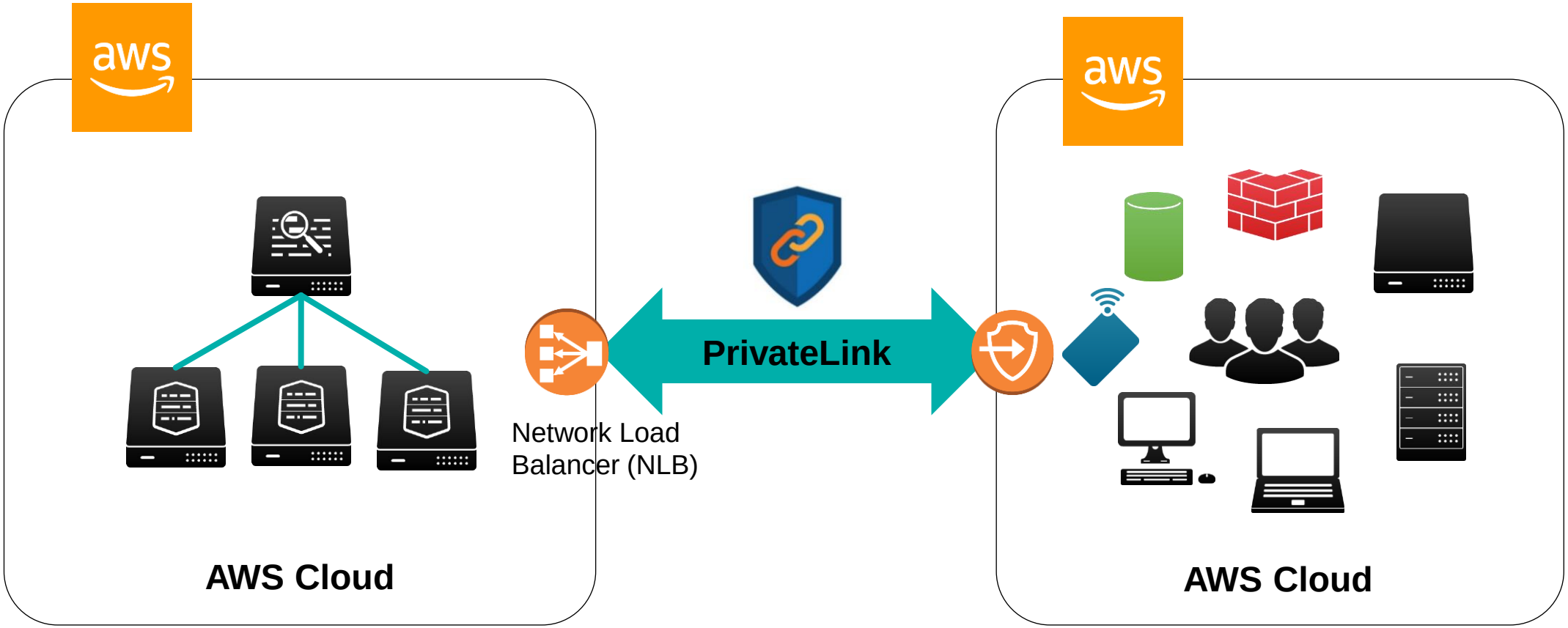


VPC Peering

- Pros
  - Easy to setup
  - No cost for VPC Peering except for data transfer costs
  - Traffic stays private and isolated in AWS network

- Considerations
  - VPC Peering is only for AWS VPCs in either the same or different AWS accounts
  - No overlapping IP ranges between peered VPCs
  - Transitive Peering and Edge-to-Edge Routing configuration is not supported

AWS PrivateLink



AWS PrivateLink

- Pros
  - Easy to setup
  - Traffic stays private and isolated in AWS network
  - Simplified private network connection (no need for firewall rules, route tables, etc.)
  - Access to AWS Services (i.e. CloudFormation, SNS, Kinesis, etc.) via private network

- Considerations
  - PrivateLink is only for AWS VPCs in either the same or different AWS accounts
  - PrivateLink Endpoints support IPv4 and TCP traffic only.
  - PrivateLink Endpoints are supported in the same AWS region only.
  - Fairly new AWS feature and architecture design, so we are still learning….

Hybrid Architecture Matrix

Connection Type	Costs	Complexity	Considerations
SSL	Low – Data transfer costs	Low to Medium	Use a CA-signed certificate, SSL is not on by default
AWS VPN	Low to Medium – cost for each VPN connection	Low to Medium	A customer gateway is needed to establish VPN, limited bandwidth
Software VPN	Low to Medium – cost for each VPN connection	Low to Medium	Administrative overhead and redundancy not included
Direct Connect	Medium to High – higher port costs, but lower bandwidth costs	Medium to High	Proximity of Direct Connect location
VPC Peering	Low - Data transfer costs	Low	No overlapping IP address space, VPC to VPC connection
PrivateLink	Low – VPC Endpoint and data transfer costs	Low	Same AWS region only, support IPv4 and TCP only, new architecture design.

Getting data in (GDI)

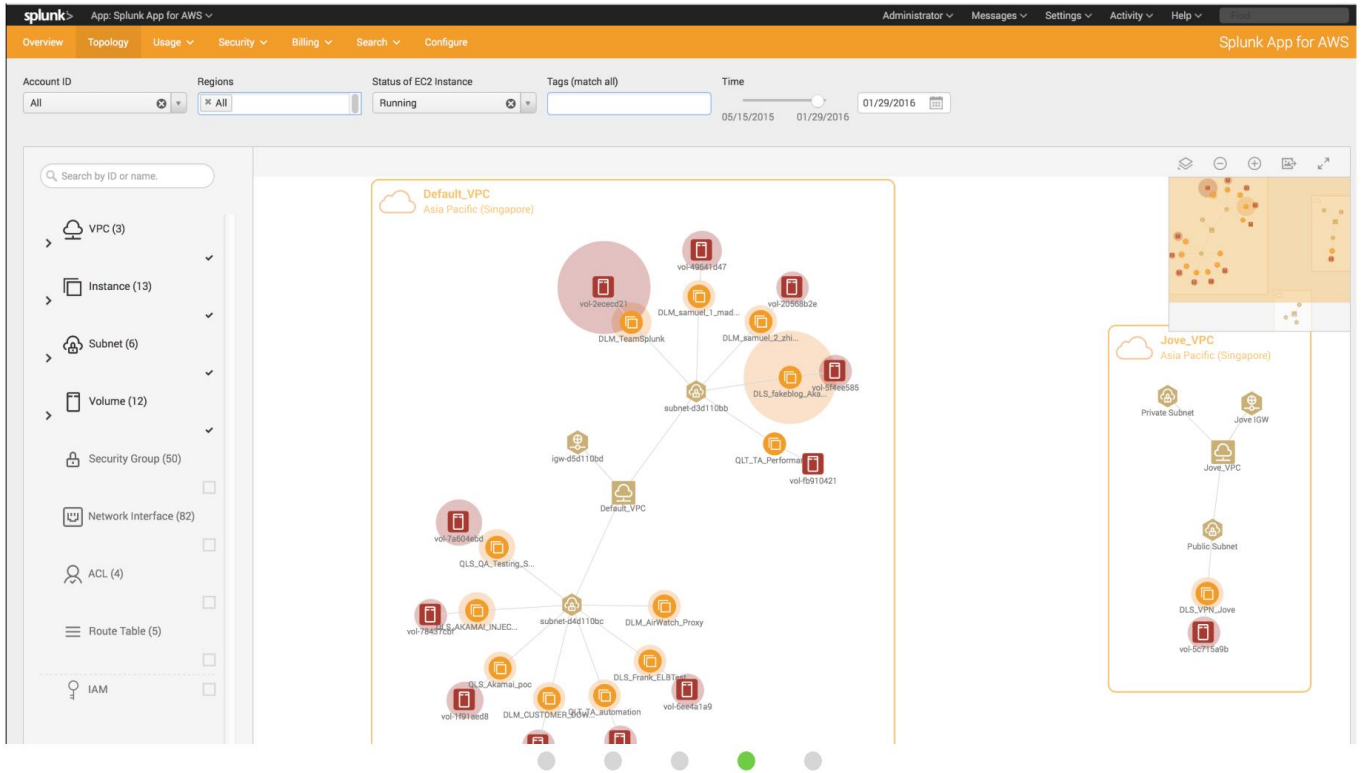
Sources, transports and the cloud

splunk> .conf18

Splunk's App for AWS

This goes without saying but just in case

- Rich set of pre-built dashboards & reports
- Analyze and visualize data from numerous AWS services
  - AWS CloudTrail
  - AWS Config
  - AWS Config Rules
  - Amazon Inspector
  - Amazon RDS
  - Amazon CloudWatch
  - Amazon VPC Flow Logs
  - Amazon S3
  - Amazon EC2
  - Amazon CloudFront
  - Amazon EBS
  - Amazon ELB
- AWS Billing Free App on Splunkbase



splunk> .conf18

Getting Data In

GDI, automation, n accounts

- Data Sources
- AWS
  - CloudWatch Metrics
  - CloudWatch Logs
  - CloudWatch Events
  - AWS Config Notifications
  - AWS Config Snapshots

  Custom Sources

- Data Transports
  - S3
  - AWS Kinesis
  - AWS Firehose
  - SNS
  - SQS
  - HEC

- *n* Accounts
  - Automation
  - Planning
  - Executions

splunk> .conf18

Getting Data In

GDI, automation, n accounts

- Data Sources
- AWS
  - CloudWatch Metrics
  - CloudWatch Logs
  - CloudWatch Events
  - AWS Config Notifications
  - AWS Config Snapshots

  Custom Sources

- Data Transports
  - S3
  - AWS Kinesis
  - AWS Firehose
  - SNS
  - SQS
  - HEC

- *n* Accounts
  - Automation
  - Planning
  - Executions

splunk> .conf18

AWS Simple Storage Service (S3)

Fast(ish) Limitless Object Storage

Use for

- Event time to Index time –slow is OK
- _raw events forever

GDI

- AWS App TA
- Bespoke SNS > λ
- Bespoke SNS > λ > Firehose

splunk> .conf18

Amazon Kinesis Data Streams (KDS)

massively scalable and durable real-time data streaming service

Use for

- Realtime
- Very high rate of write from many many emitters
- More than one target system

GDI

- AWS App TA
- Bespoke λ >HEC
- KDS to Firehose > HEC

Amazon Kinesis Data Firehose (KDF)

reliably load streaming data into data stores and analytics tools

Use For:

- Near real-time data
- Events to files via S3

GDI

- Top level target from AWS

splunk> .conf18

Amazon Simple Notification Service (SNS)

fully managed pub/sub messaging and mobile notifications service

Use For

- Not a source for Splunk
- Event Stimulus Services
- Notifications

GDI

- Message bus for other GDI workers

splunk> .conf18

Amazon Simple Queue Service (SQS)

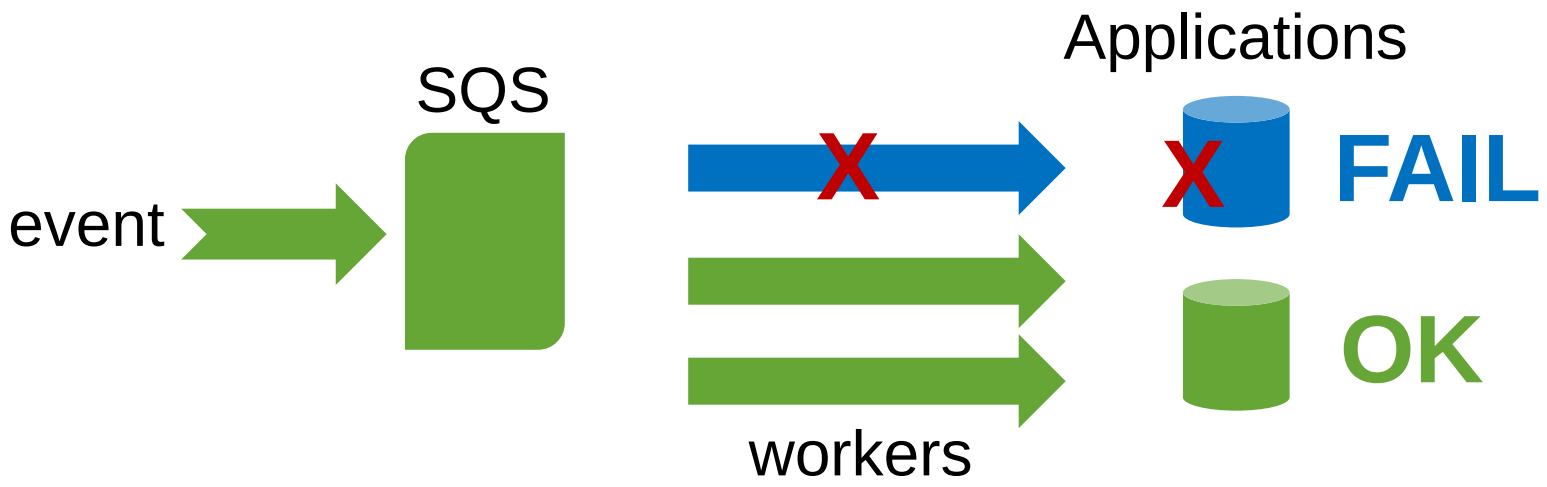
fully managed message queuing service

Use For:

- Message bus for Splunk
- Serverless Apps
- SOA

GDI

- AWS App TA
- Message queue for distributed workers



Splunk Universal Forwarder

reliable, secure data collection from remote sources

Splunk Universal Forwarder

Still the most distributed way to GDI

Is your source very Cloudy?

- Is my source:
- Transient
- Ephemeral
- Extremely light weight

- Do I want to be responsible for:
- Delivery
- Retry
- Load balancing
- Queue and DLQ

Splunk HTTP Event Collector

Use For:

- Short lived, transient, ephemeral workers
- Transactional Applications
- Anything that you can code to emit
- Cloudy

GDI

- Now this is This GDI

Getting Data In

GDI, automation, n accounts

- Data Sources
- AWS
  - CloudWatch Metrics
  - CloudWatch Logs
  - CloudWatch Events
  - AWS Config Notifications
  - AWS Config Snapshots

  Custom Sources

- Data Transports
  - S3
  - AWS Kinesis
  - AWS Firehose
  - SNS
  - SQS
  - HEC

- *n* Accounts
  - Automation
  - Planning
  - Executions

splunk> .conf18

Choose wisely

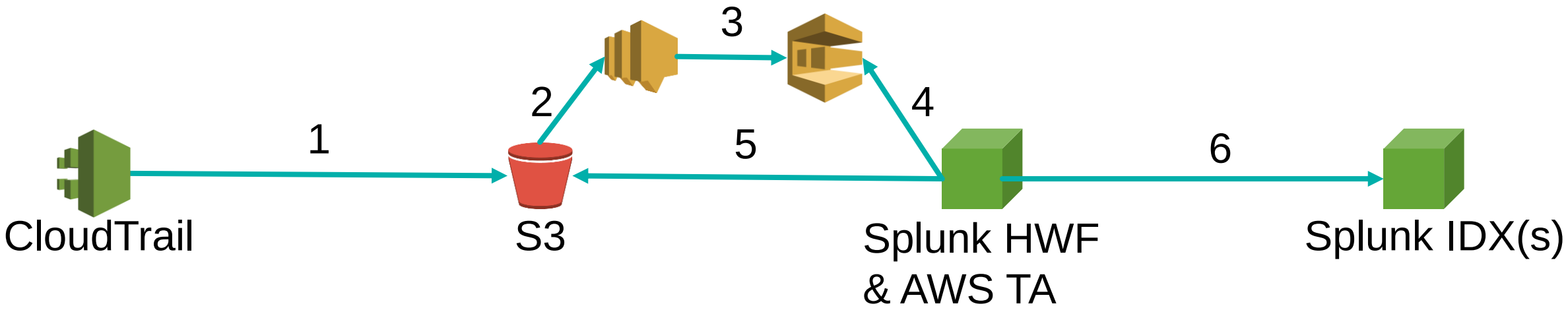
Poor choices make for good stories

- Many many ways to solve
- Solve for operational stability
  - Once it is set up it: It works reliably, fails predictably, recovers automatically, has metrics
  - Alarms, reports success and failure
- Reduced operational complexity
  - If the setup is 1000 clicks – automate
  - Declarative Operations

Data Sources >>> Transports

All the ways the data flows

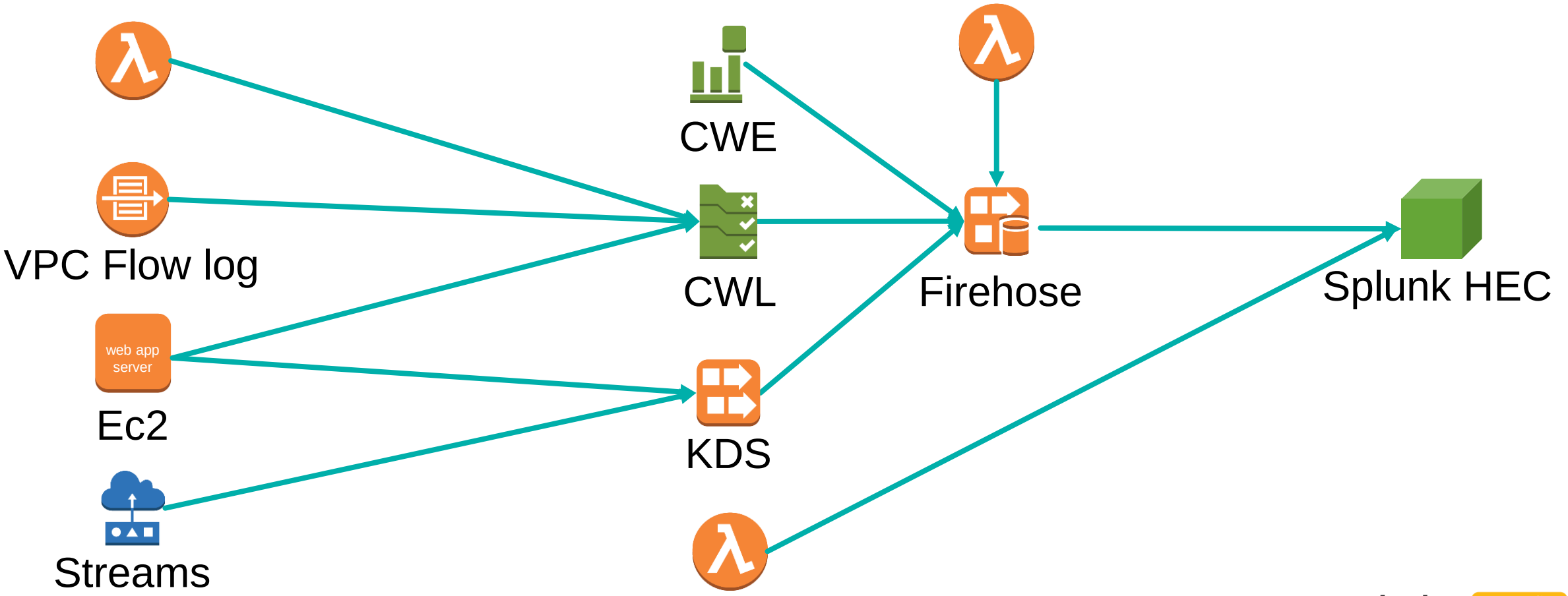
CloudTrail (poll)



Data Sources >>> Transports

All the ways the data flows

Common Data and Transport Patterns (push)

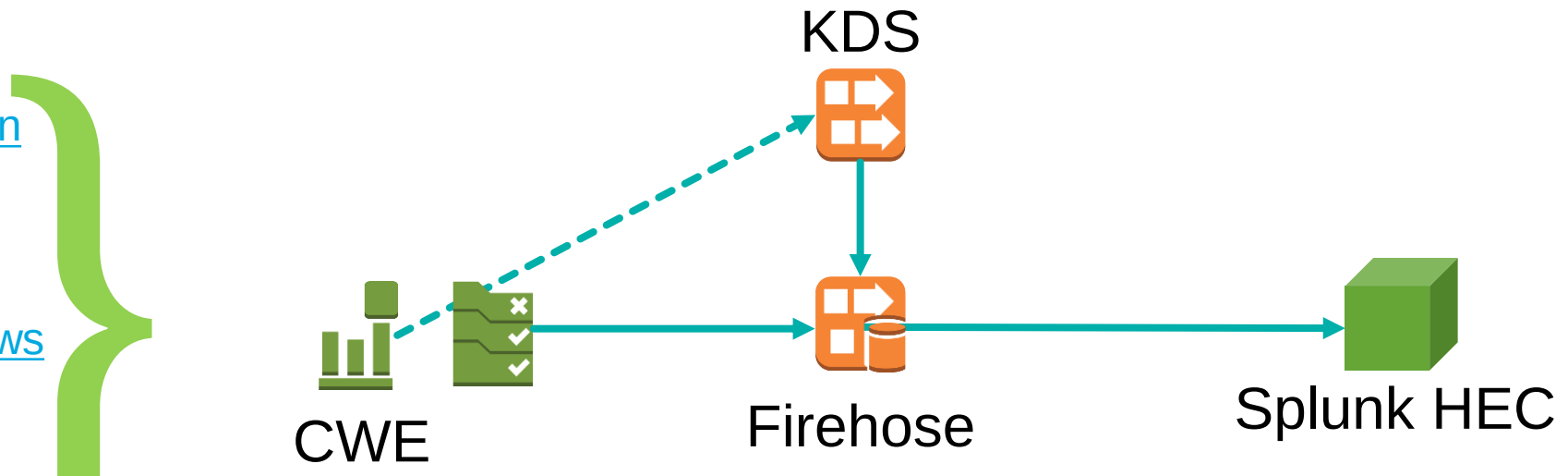


Data Sources >>> Transports

All the ways the data flows

Common Data and Transport Patterns

- AWS API Call Events
- AWS Management Console Sign-in Events
- Amazon EC2 Events
- AWS Systems Manager Events
- Amazon EC2 Maintenance Windows Events
- Amazon ECS Events
- Amazon GuardDuty Events
- AWS Health Events
- AWS KMS Events
- Amazon Macie Events
- Scheduled Events
- AWS Trusted Advisor Events



AmazonCloudWatch/latest/events/EventTypes.html

Getting Data In

GDI, automation, n accounts

- Data Sources
- AWS
  - CloudWatch Metrics
  - CloudWatch Logs
  - CloudWatch Events
  - AWS Config Notifications
  - AWS Config Snapshots

  Custom Sources

- Data Transports
  - S3
  - AWS Kinesis
  - AWS Firehose
  - SNS
  - SQS
  - HEC

- *n* Accounts
  - Automation
  - Planning
  - Executions

One organization, many accounts

- Every organization is going to have many accounts
  - Separation of environments
  - Separation of duties
- Most accounts will have more than one region
  - All regions should be monitored
  - N x R x S = UGH

Accommodations and solutions

Things that will work for everyone

- Get out in front
  - Deliver a standard service with end points
  - Build on boarding automation
  - Use the tools to adapt
    - AWS moves very fast – new features for old services , new services with new sources
    - A new mouse may not require a new trap

No Unique one off solutions

Bespoke is for suits, and macaroni art

Best Practices

(your mileage may vary)

Bring the data to one place

- Single Points of Aggregation
- One transport - many accounts
  - Streamlined Splunk set-up

Many Eggs – One basket

Ship the data locally

- Minimize Blast Radius
- Transports in each region, each account

Many eggs individually wrapped

splunk> .conf18

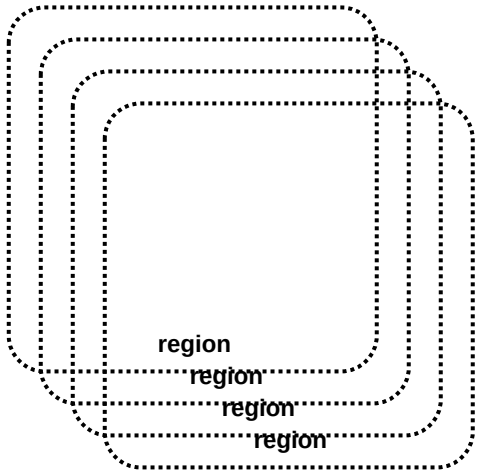
Best Practices

(your mileage may vary)

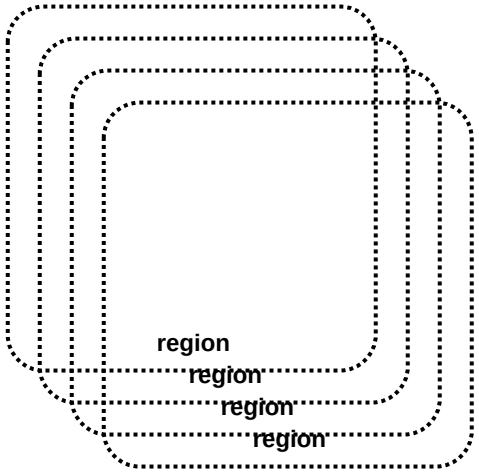
All in one

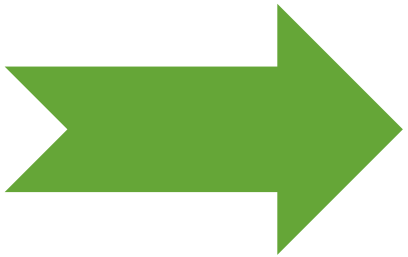
(your mileage may vary)

ACCT A



ACCT B

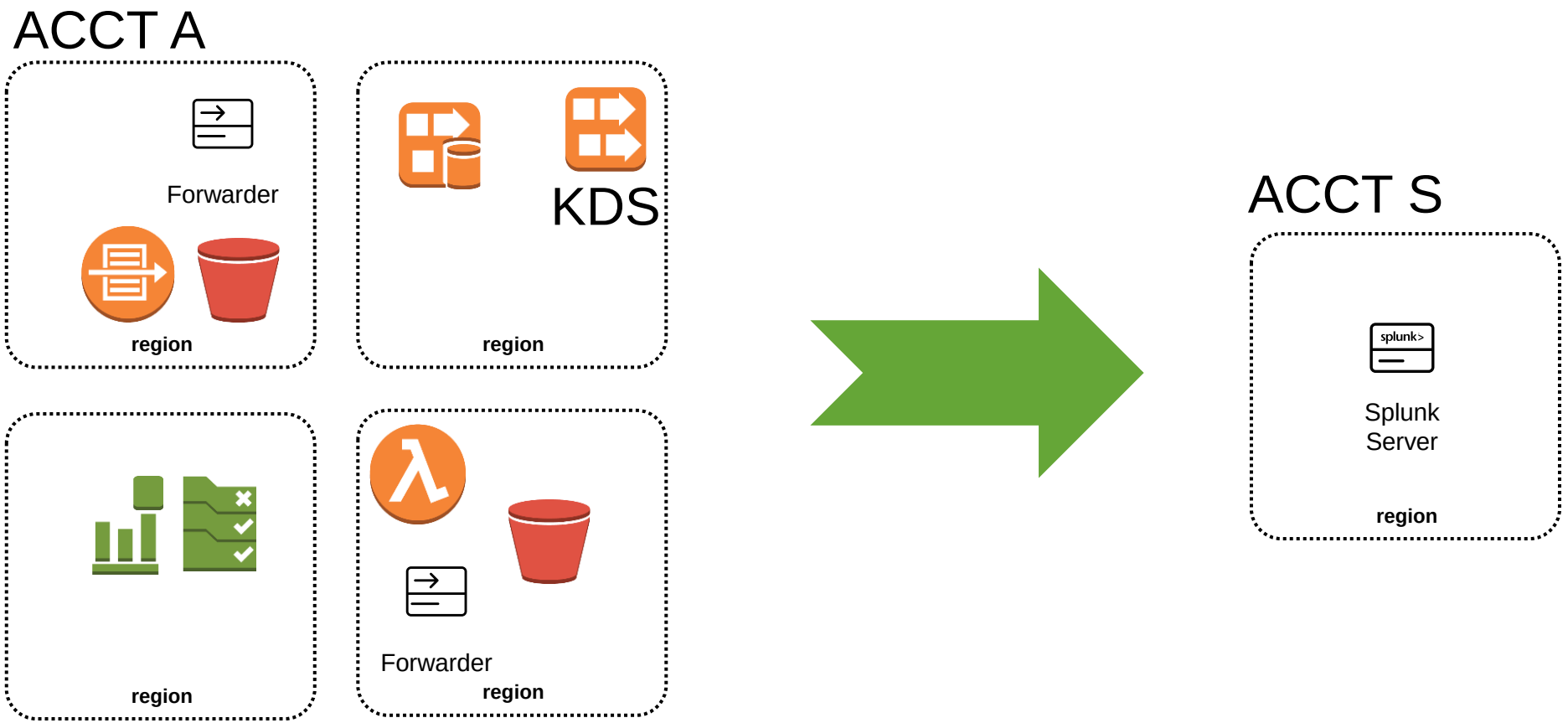




ACCT S

Location, location, location

(your mileage may vary)



Automation with AWS

AWS CodeDeploy

Service that automates software deployments to compute services including Amazon EC2, Lambda, and on-premises.

Amazon EC2 Systems Manager

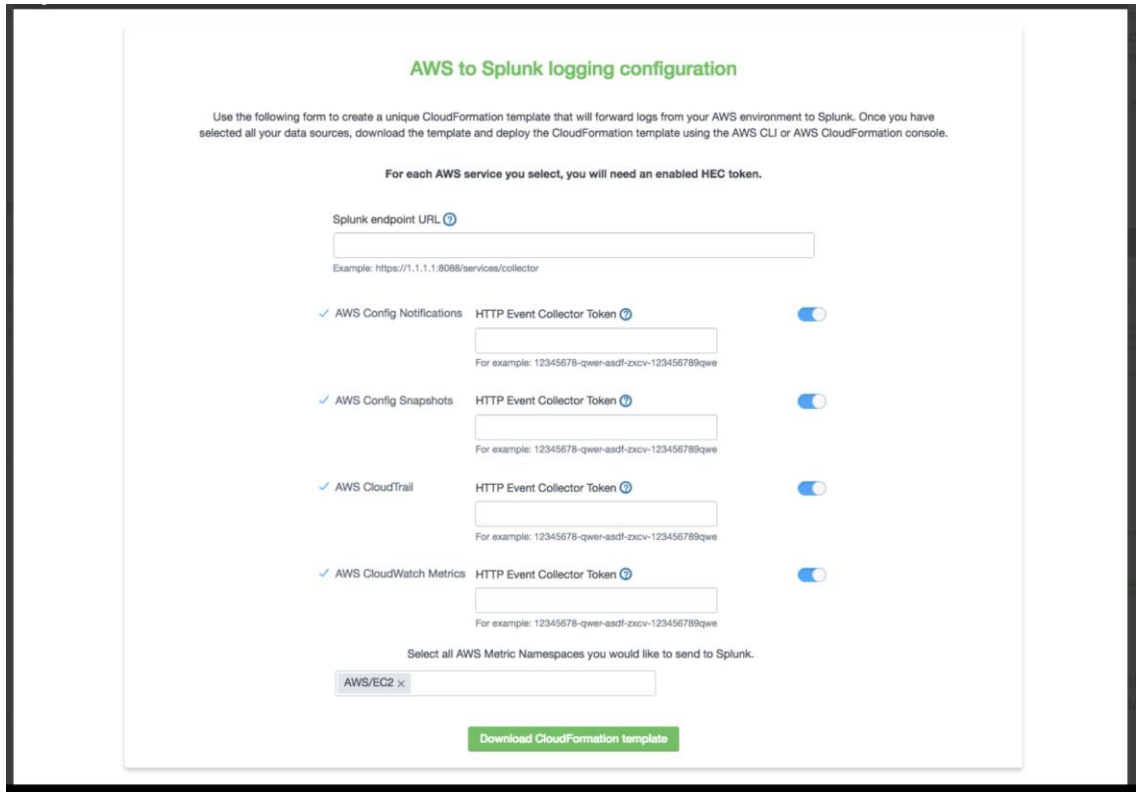
Visibility and control of your infrastructure on AWS, provides a unified user interface so you can view operational data from multiple AWS services and allows you to automate operational tasks across your AWS resources.

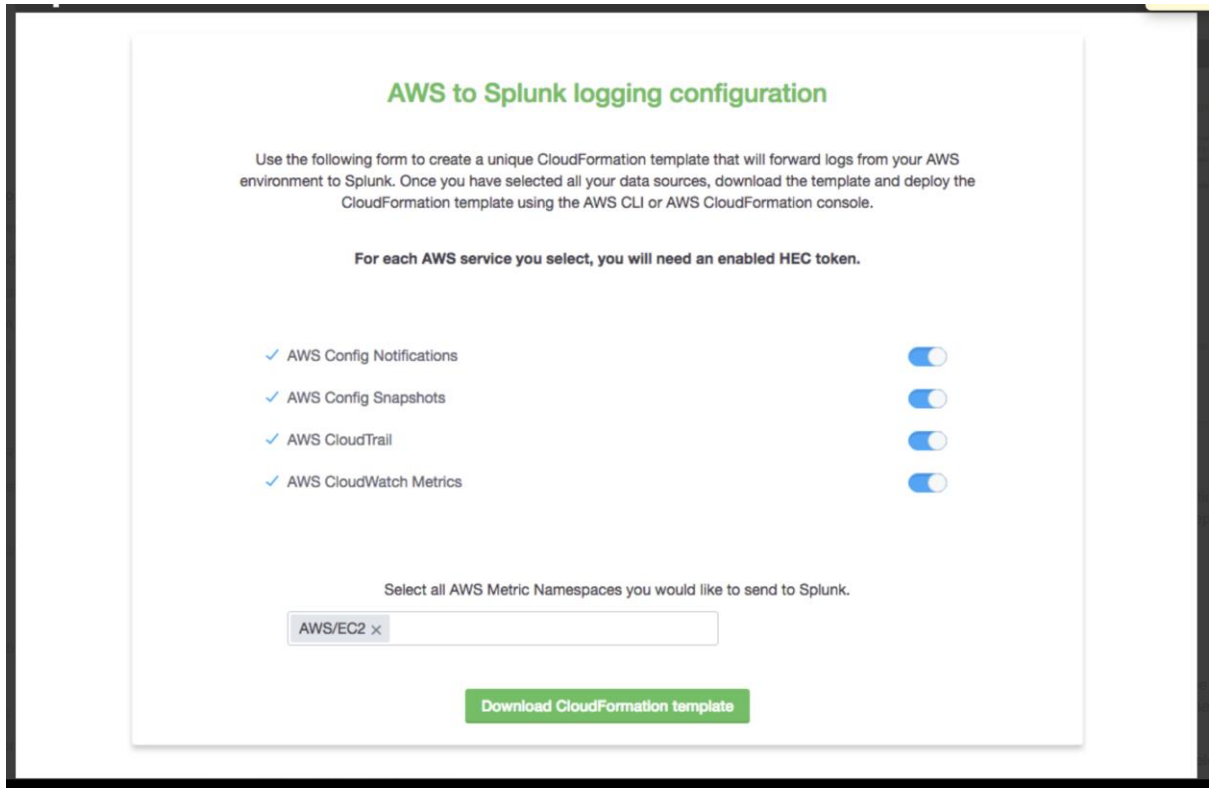
AWS CloudFormation

Provides a common language for you to describe and provision all the infrastructure resources in your cloud environment.

Splunk GDI for AWS

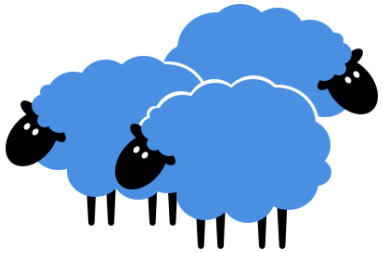
open source project to provide advanced AWS and Splunk automation for configuring common AWS data sources as serverless solution





Blog Post link

Git Hub Link post launch 9/26



CT Grazer

https://github.com/FINRAOS/CTGrazer

CTGrazer is code you can use to create an *AWS Lambda* Function that will collect all of your *AWS CloudTrail* logs and efficiently send them to your *Splunk HEC (HTTP Event Collector)* server.

Why?

Using **CTGrazer** to port your AWS CloudTrail logs into Splunk has many advantages

- **Speed** *CloudTrail* logs are processed as soon as they become available
- **Security** All data is **encrypted** in transit and it does not rely on AWS IAM Access Keys
- **Scalable** CTGrazer will automatically **scale up** and down according to your needs
- **Reliable** If CTGrazer can't get your logs to their destination, it will **automatically retry** until it can
- **Cost Effectiveness** Pulling in **400K objects** a month will cost you about the same as a **cheeseburger!**

Key Takeaways

Thanks for stopping by, here's a roadie.

1. Splunk runs the same anywhere you run it, CPU is CPU and memory is memory. The cloud makes it easy to try something, change it and keep iterating.

2. You can run Splunk in, on, and between clouds and datacenters. The Cloud brings new tools for connectivity.

3. The cloud is cloudy but it provides new tools to GDI.

Q&A

splunk> .conf18

Thank You

Don't forget to rate this session in the .conf18 mobile app