AHEAD Innovation Days

AWS – Automation, Security, and Governance

WIFI – On Whiteboard on the right

AHEAD®

These Are Complex Balancing Acts

Innovation

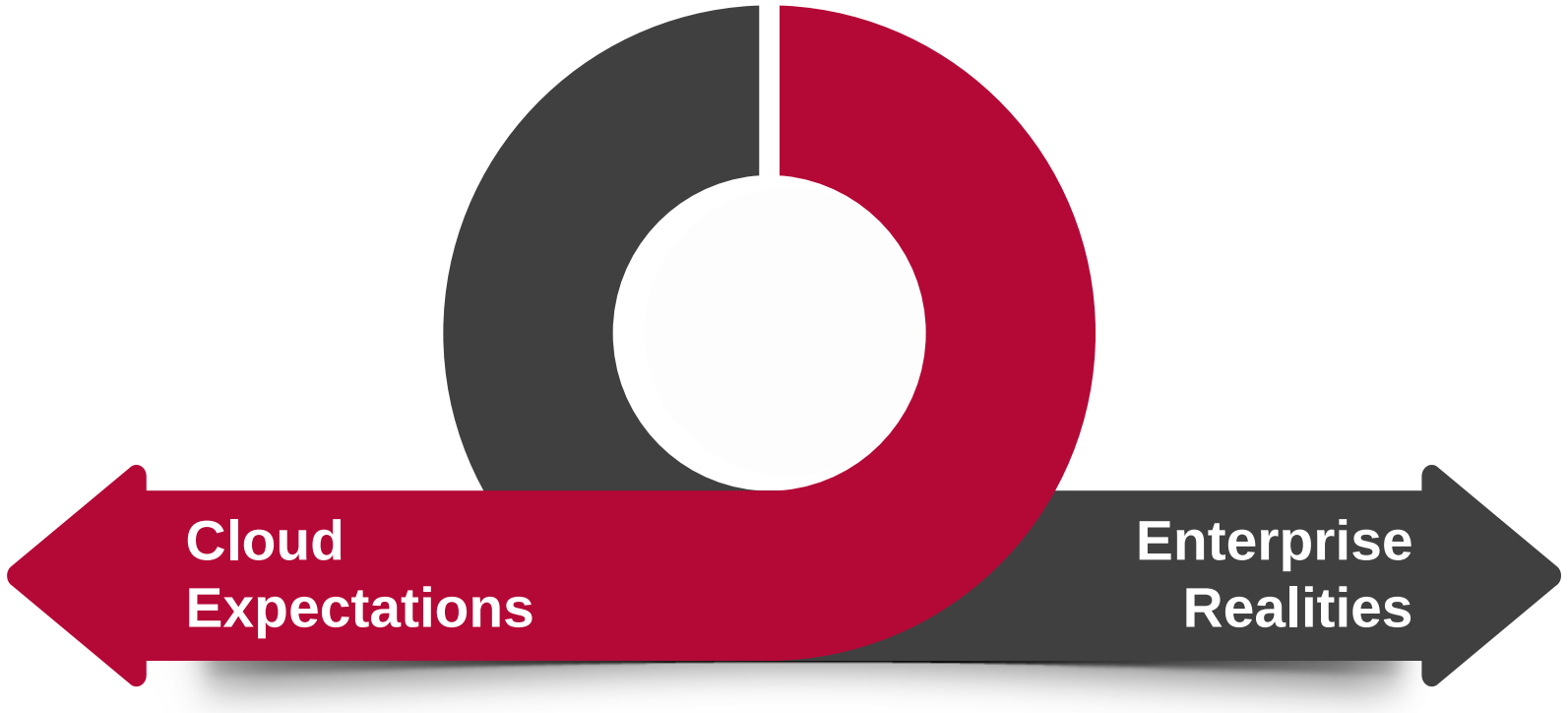
Disruption

Speed

Agility

Elasticity

Experimentation



Governance

Compliance

Visibility

Security

Efficiency

Standardization

Our Purpose

To help you deliver the speed and flexibility that your business **demands** from technology

along with the security, simplicity, and savings it **requires**

The Enterprise Cloud

Is where your **applications run** and where your **data live**

The Optimal Enterprise Cloud

Is responsive, elastic, and highly automated

Offers full transparency into cost, inventory, and risk

Incorporates rigorous security and management practices

AHEAD Enterprise Cloud Delivery Framework©

Cloud Management
Automation and Operations

- Enterprise Service Management
- Cloud Management Platform
- Configuration Management
- Application Performance Management

DevOps
Process and Integration

- Continuous Integration and Deployment
- DevOps Toolchain
- Process Optimization and Automation

Security and Risk
Compliance and Protection

- Governance, Risk, and Compliance
- Endpoint Protection
- Network Control and Threat Detection
- Log Aggregation and Analytics

Data Center

- Hyper/Converged Infrastructure
- End-User Computing
- Compute and Virtualization
- Data Protection and Availability
- Network
- Storage

Infrastructure
Platforms and Services

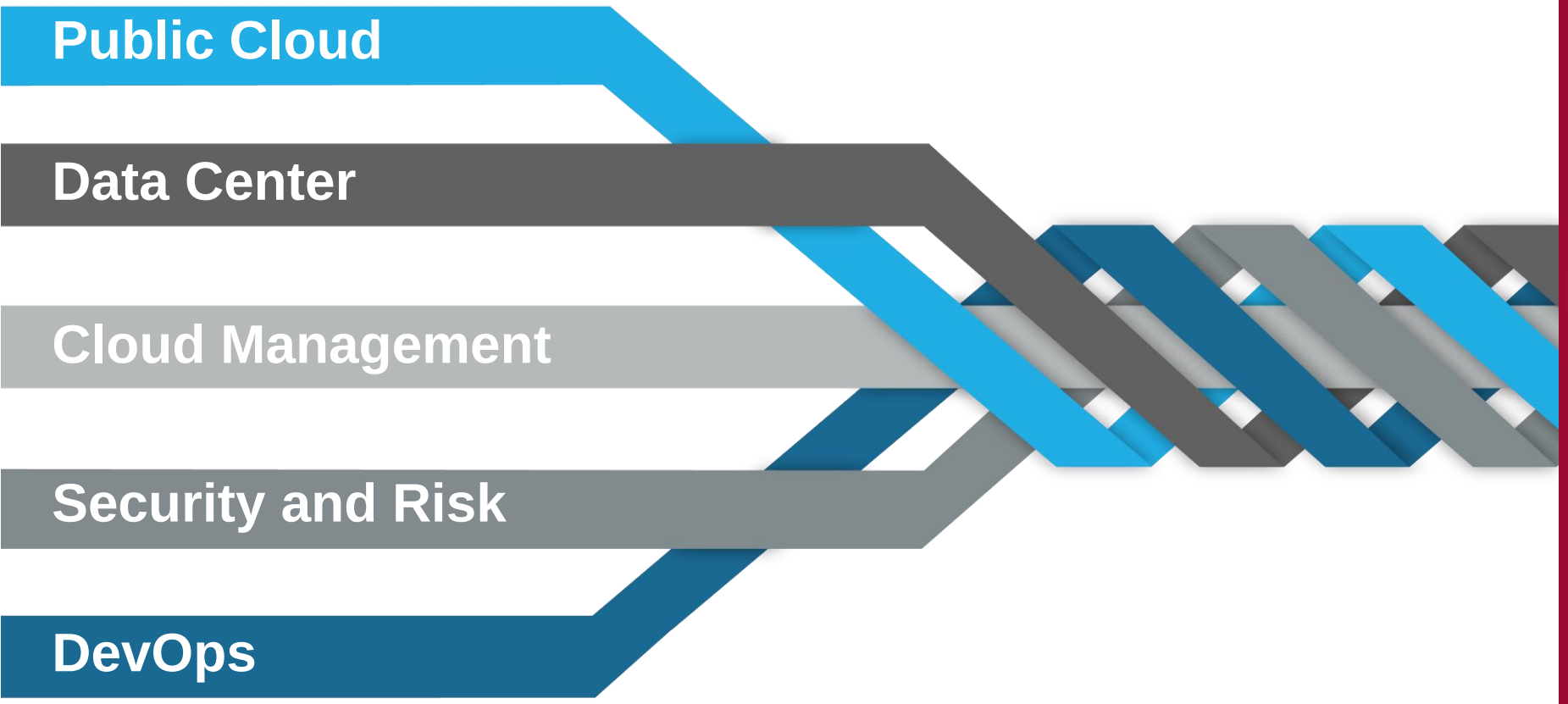
- Serverless Architectures
- Containers

Public Cloud

Enterprise Enablement

- Organizational Readiness
- Automation and Orchestration
- Extensibility and APIs
- Service Catalog
- CMDB
- Financial Management
- Marketing

Our Unique Value Is in the Stitching



Integration of ESM and cloud management

Integration of security and automation

Integration of Dev and Ops pipelines to speed delivery

AHEAD®

AHEAD Labs and Briefing Center

- Review and refine your technology roadmaps
- Translate technologies into possible outcomes
- Opportunities to dive deep into seventy enterprise cloud technologies
- Select from 100 demonstrations in our Labs
- Engage our experts who have more than 500 certifications from across AWS, AppD, Chef, Cisco, EMC, Microsoft, Nutanix, Puppet, ServiceNow, and VMware

Technical Briefings

AHEAD Innovation Days

AWS – Automation, Security, and Governance

AHEAD®

Agenda

- Lab 1 – AWS CloudFormation Overview
- Lab 2 – Infrastructure as Code – Operations
- Lab 3 – Image Operations
- Lab 4 – Governance – Automating Security
- Lab 5 – Auto Scaling Workloads
- Innovation Demo / Raffle

- Account Cleanup (Homework)
- Additional Education Labs (Homework)

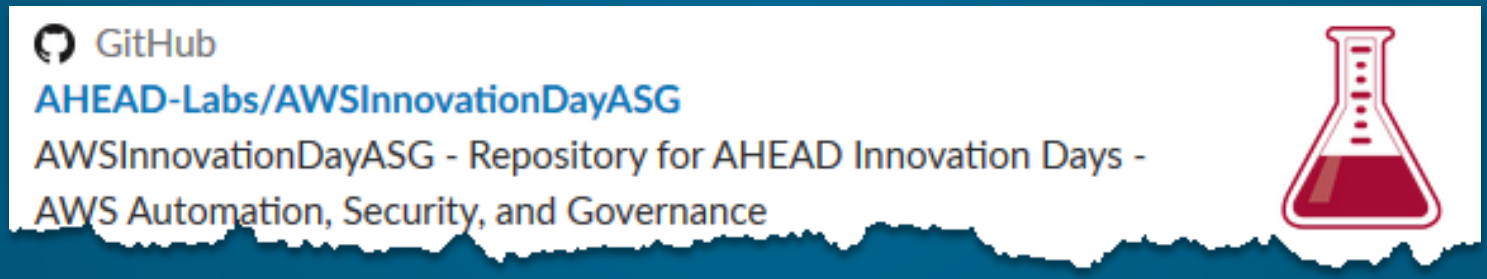
Getting Started

- You should have:
  - Lab Guide
  - Laptop
  - AWS Account and Login
  - Internet Access
  - AWS Promo Code

DISCLAIMER: AHEAD IS NOT RESPONSIBLE FOR ANY COSTS INCURRED DURING THIS TRAINING. PLEASE DELETE YOUR RESOURCES AT THE END OF THE COURSE.

Accessing the Labs and Code

https://github.com/AHEAD-Labs/AWSInnovationDayASG



Bryan Krausen

- Sr. Solutions Architect
- Blog: itdiversified.com
- Twitter: @btkrausen
- Holds (6) AWS Certifications
- Background in Data Center & Virtualization

AHEAD®

Greg Thursam

- Solutions Principal
- Twitter: @gregthursam
- Blog: Between2Clouds.blog
- Holds (4) AWS Certifications
- Background Cloud, D.C., and Linux

AHEAD®

Eric Shanks

- Senior Solutions Architect
- Twitter: @eric_shanks
- Blog: theITHollow.com
- VCDX #195 DCV/CMA
- Holds (5) AWS Certifications

AHEAD®

Adam Youngblood

- Cloud Technical Architect
- Twitter: @automation_adam
- Blog: CloudAutomation.blog
- Holds (3) AWS Certifications

AHEAD®

AHEAD Innovation Day

AWS CloudFormation

AHEAD®

AWS CloudFormation

- Often referred to as Infrastructure as Code
- Describe and provision infrastructure and resources in AWS in code
- ‘Codify’ your environment
- Create consistent, repeatable environments
- Standardize environments across accounts, regions, or business units

AWS CloudFormation - Benefits

CloudFormation

- Can be stored in source control for auditable change-log to infrastructure
- Infrastructure can be easily recreated at any state
- Combine with orchestration to enable automation patterns (CI/CD, rollback, blue-green, etc)
- Best practices are embedded into the template

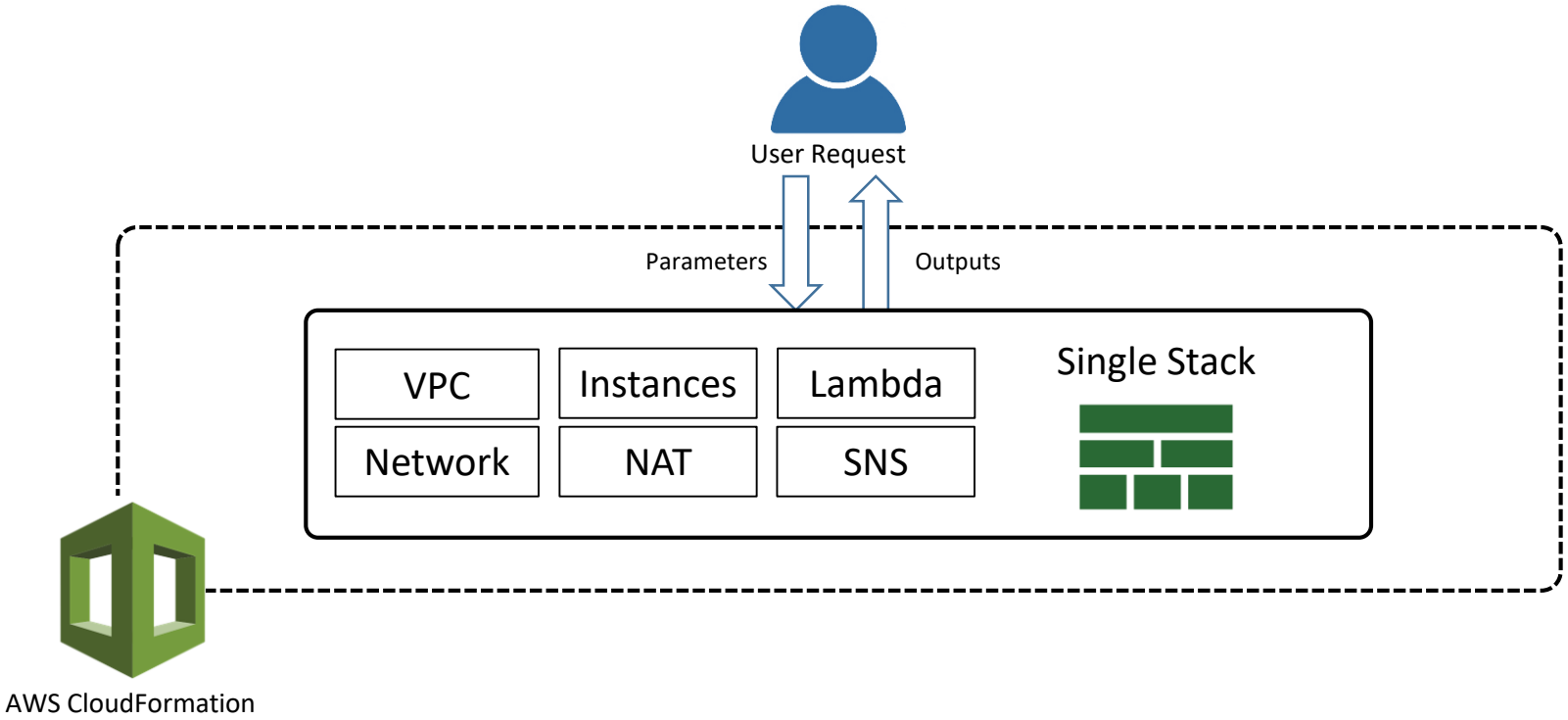
Manual Deployment

- Requires third-party systems to monitor changes
- Reverting infrastructure changes becomes detective work
- Human error and drift are probable
- Automating deployments is limited or requires complex coding

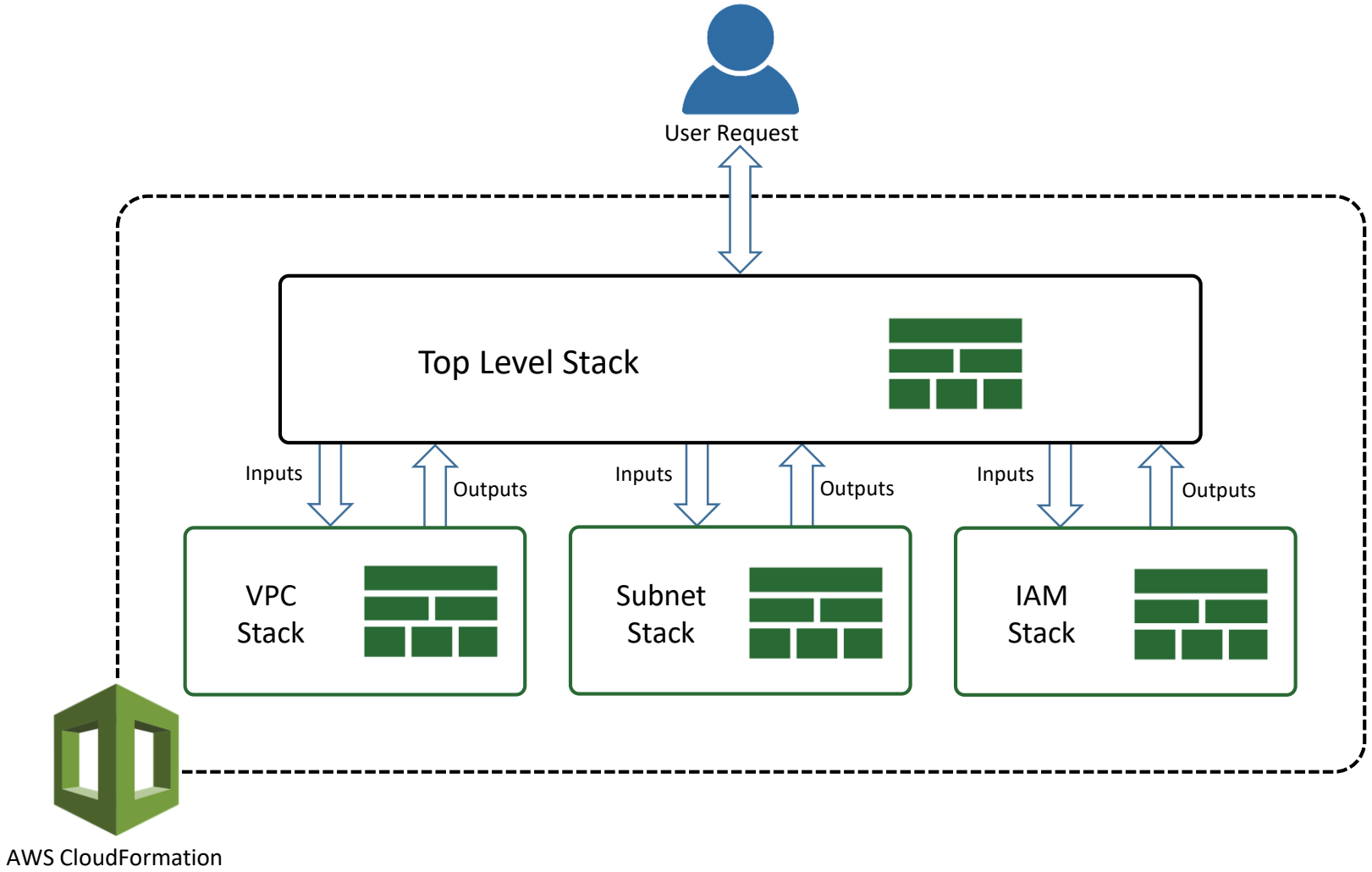
AWS CloudFormation - Template

- Text file that can be written in YAML or JSON
- Defines the resources to be provisioned in AWS
- Resources can be in a single stack or use nested stacks
- Manage relationships between created resources
  - Assign an EC2 instance to an Application Load Balancer
  - Create a Security Group and assign to an EC2 instance
- Gather attributes from resources to apply to others
- Deploy templates from the console, API, SDKs, or AWS CLI

AWS CloudFormation – Single Stack



AWS CloudFormation – Nested Stacks



AWS CloudFormation - Parameters

Ability to pass values to your template upon creation or during updates

Define inputs to gather information from user

- Parameter types include:
  - String
  - Number
  - List
  - CommaDelimitedList
  - AWS-Specific Type
  - SSM Parameter
- Set default values for parameters
- For sensitive parameters, use ‘NoEcho’ to obscure the entry

```
"Parameters" : {
  "InstanceTypeParameter" : {
    "Type" : "String",
    "Default" : "t2.micro",
    "AllowedValues" : ["t2.micro", "m1.small", "m1.large"],
    "Description" : "Enter t2.micro, m1.small, or m1.large. "
  }
}
```

AWS CloudFormation – Mappings

Match a key to a corresponding set of named values

Use Fn::FindInMap function to return a named value based on a specified key

- Most often used with ‘Region’ selection for resources
  - Enables a template to be written without regard to a certain region

```
"Mappings" : {
  "RegionMap" : {
    "us-east-1" : { "32" : "ami-6411e20d", "64" : "ami-7a11e213" },
    "us-west-1" : { "32" : "ami-c9c7978c", "64" : "ami-cfc7978a" },
}
```

AWS CloudFormation – Conditions

Ability to conditionally create resources

Define when a resource is created or when a property is defined

- Combine with Parameters to determine what resources to build or how big to build them
  - Example: If deploying in Production, instance size is 5m.large. If dev, use t2.small.

```
"Conditions" : {
  "CreateProdResources" : {"Fn::Equals" : [{"Ref" : "EnvType"}, "prod"]}
 }
```

AWS CloudFormation – Resources

Declares the AWS resources that should be created

Only *Required section of a CloudFormation template

- Resources are defined using a Logical ID and are used to reference other provisioned resources within the template
- Each resource will declare a Resource Type (e.g., AWS::EC2::Instance)
- Resources section makes use of all the other defined sections of the template (i.e., parameters, conditions, mappings, etc)

AWS CloudFormation – Outputs

Declares output values from resources created in template

Outputs are defined with a Logical ID followed by the desired value

- Can be imported into other stacks, return in response (to describe stack calls, or viewed on the CloudFormation console.
- Example: Output the S3 bucket name created in a stack to make it easier to find.

AWS CloudFormation – Pseudo Parameters

Used to return critical data within the Resources section

Parameters predefined by AWS CloudFormation

- Examples:
  - AWS::AccountId – returns the account ID where the stack is created
  - AWS::Region – returns the region where the stack is being created
  - AWS::StackId – returns the ID of the CloudFormation stack

AWS CloudFormation – Intrinsic Functions

Built-in functions to help manage your stacks and resources

Used to assign values to properties that are not available until runtime

- Examples:
  - Fn::FindInMap – returns a value corresponding to a map in Mapping section
  - Fn::GetAtt – returns the value of an attribute from a resource in the template
  - Fn:Join – appends a set of values into a single value
  - Ref – returns the value of a parameter or resource

AWS CloudFormation – Recommendations

- Develop a library of CloudFormation templates that serve as the *backbone* for new infrastructure deployments. This enables you to ensure that best configuration and security practices are part of any new resource deployment, as those resource deployments are based off CloudFormation templates.
- Build templates to be as parameterized and regionally abstract as possible, to maximize template reuse and enable multi-region deployments.
- Store your CloudFormation templates in source control, and follow best practices for continuous integration and delivery when updating your templates.
- Use a CI/CD (build/automation) server to orchestrate and automate the steps in your CloudFormation development pipeline (change → test -->delivery/deployment)
- Generate change-sets for staging and production deployments, and analyze these change sets before running updates to stacks in these environments.

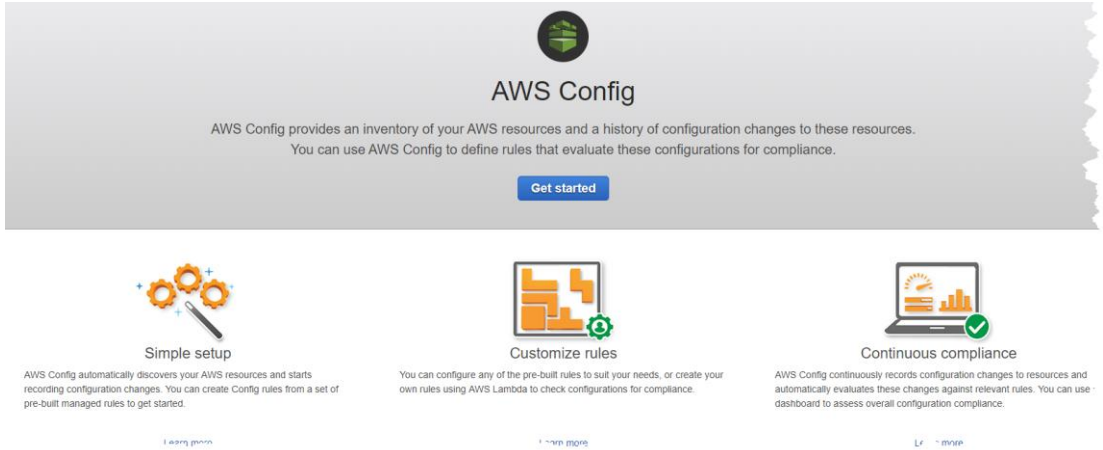
Lab 1 - CloudFormation

In this lab, participants will use CloudFormation to create the base infrastructure and additional AWS resources to use for subsequent labs. The lab will make use of nested CloudFormation templates to deploy the resources.

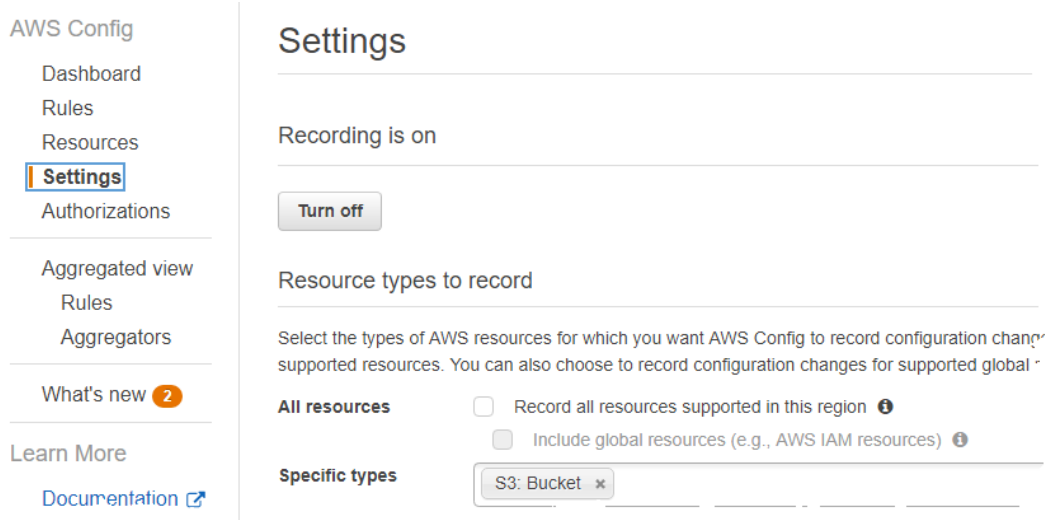
AHEAD®

Lab 1 appendix

Config NOT in Use



Config in Use



AHEAD Innovation Day

AWS - Day 2 Operations

AHEAD®

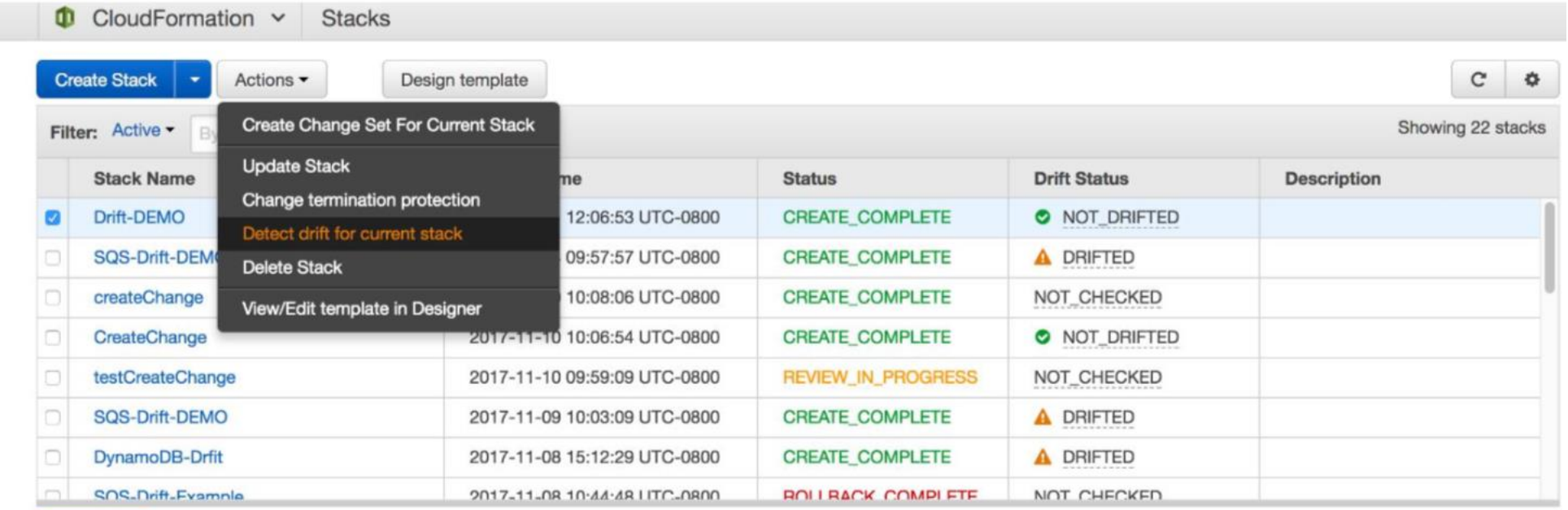
Day 2 - Operations

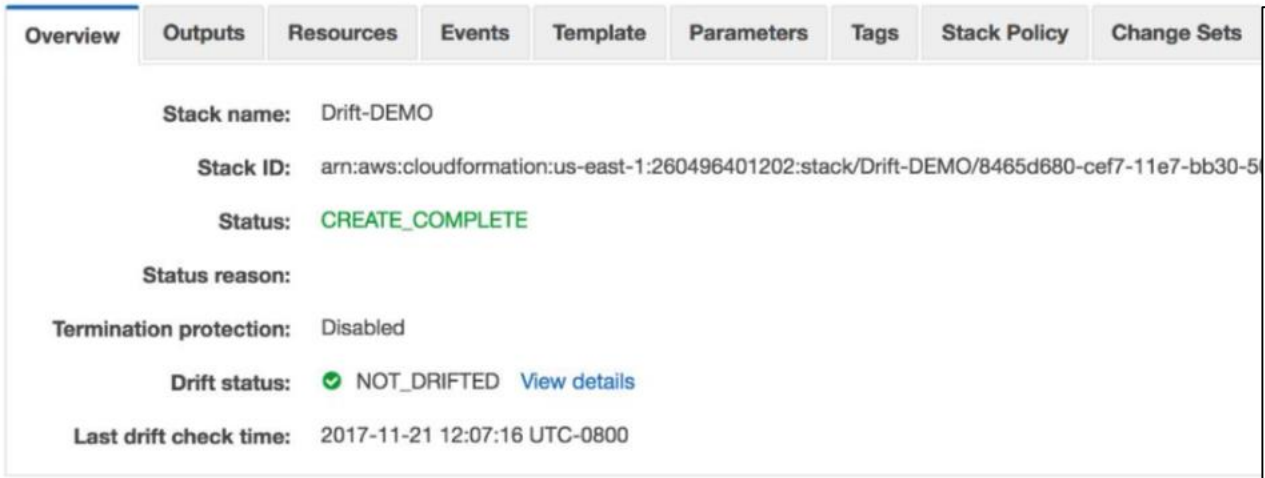
- CloudFormation Update Stacks
- CloudFormation: Coming Soon…..
- Tagging Strategies
- CloudWatch Overview
- Configuration Management
- Lambda as Operations
- Real-World Examples
- LAB TIME !

AWS CloudFormation – Updating a Stack

- Updating a stack requires the creation and execution of a change set
- Resources may require replacement
- Update at the parent stack, not child stack

CloudFormation Drift Detection





DRIFT DETECTION – COMING SOON

Coming soon in 2018, Configuration drift detection capability in AWS CloudFormation will be generally available in all AWS commercial regions.

AHEAD

AWS Tags

Tags

- Key Value Pairs
  - Key: Name
  - Value: John Smith
- Metadata
- Can be inherited
  - ASG, CFn, Beanstalk

Use Benefits

- Simplified organize method
- Instant and automated real-time discover of resources
- Can be used in IAM to automatically restrict permissions
  - For security you must then restrict who can set tags

Tagging Categories

Technical Tags

Name – Used to identify individual resources

Application ID – Used to identify disparate resources that are related to a specific application

Application Role – Used to describe the function of a particular resource (e.g. web server, message broker, database)

Cluster – Used to identify resource farms that share a common configuration and that perform a specific function for an application

Environment – Used to distinguish between development, test, and production infrastructure

Version – Used to help distinguish between different versions of resources or applications

Tags for Automation

Date/Time – Used to identify the date a resource should be started, stopped, deleted, or rotated

Opt in/Opt out – Used to indicate whether a resource should be automatically included in an automated activity such as starting, stopping, or resizing instances

Security – Used to determine security requirements, such as encryption, the enabling of Amazon VPC Flow Logs, and also to identify route tables or security groups that deserve extra scrutiny

Business Tags

Owner – Used to identify who is responsible for the resource

Cost Center/Business Unit – Used to identify the cost center or business unit associated with a resource; typically for cost allocation and tracking

Customer – Used to identify a specific client that a particular group of resources serves

Project – Used to identify the project(s) the resource supports

Security Tags

Confidentiality – An identifier for the specific data-confidentiality level a resource supports

Compliance – An identifier for workloads designed to adhere to specific compliance requirements

Tagging Design Best Practices/Guiding Principles

Tagging should represent organizationally relevant dimensions and adheres to tagging best practices:

- Always use a standardized, case-sensitive format for tags, and implement it *consistently* across all resource types.
- Consider tag dimensions that support the ability to manage **resource access control**, **cost tracking**, **automation**, **security**, and **organization**.
- It is easy to modify tags to accommodate changing business requirements
  - Consider ramifications of future changes, especially in relation to tag-based access control, automation, or upstream billing reports.
- **Standardize, Communicate and Enforce**
- Err on the side of using too many tags rather than too few tags.

Configuration Management

OLD WAY

Focus on pushing configuration to legacy systems in legacy ways
(i.e. PowerShell, Shell scripting, etc.)

NEW WAY

Focus on massive scale, new development, bridging legacy gap. Shifting mindset to engineering a *desired state configuration.*

Configuration Management - Overview

Benefit

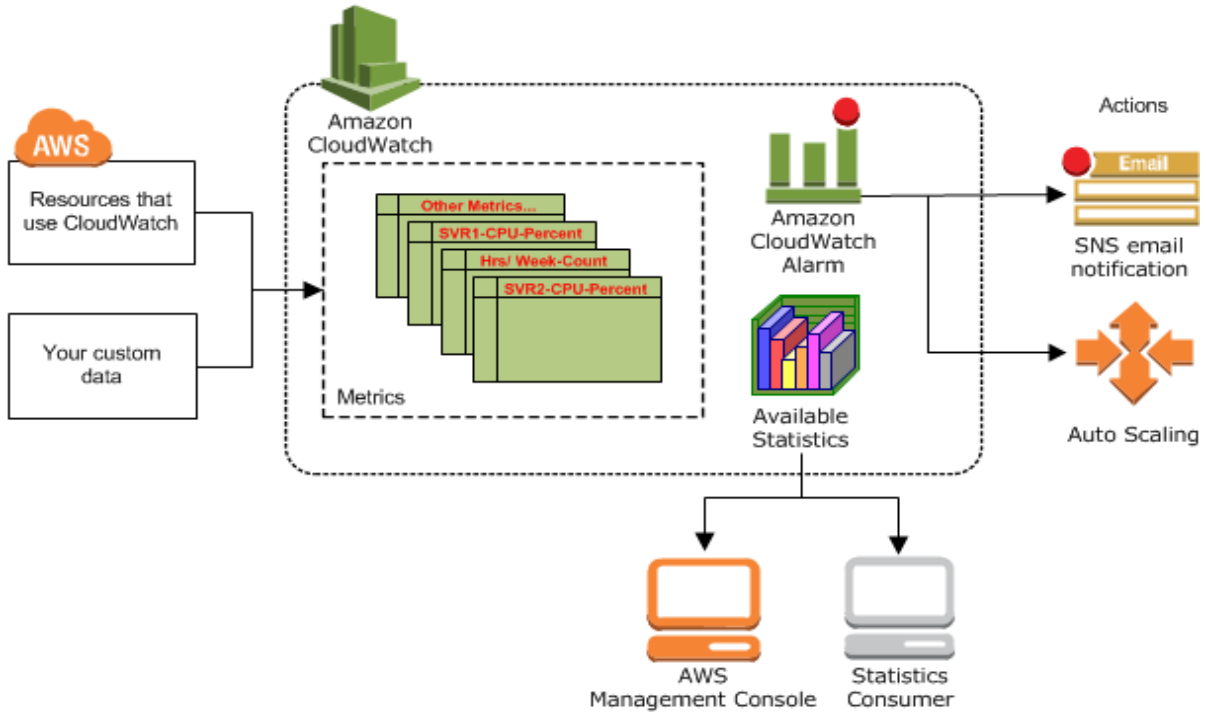
- "Templatized" server configuration
- Embed Security and Organizational Best Practice
- Large OEM and Community Libraries
- Standardize
- Auditability
- Avoid AMI sprawl

Use Cases

- Auto Scaling servers
- Self-Healing
- Security / Compliance
- Blue-Green, Zero-Downtime, and advanced automation patterns
- Every EC2 use case!

CloudWatch Overview

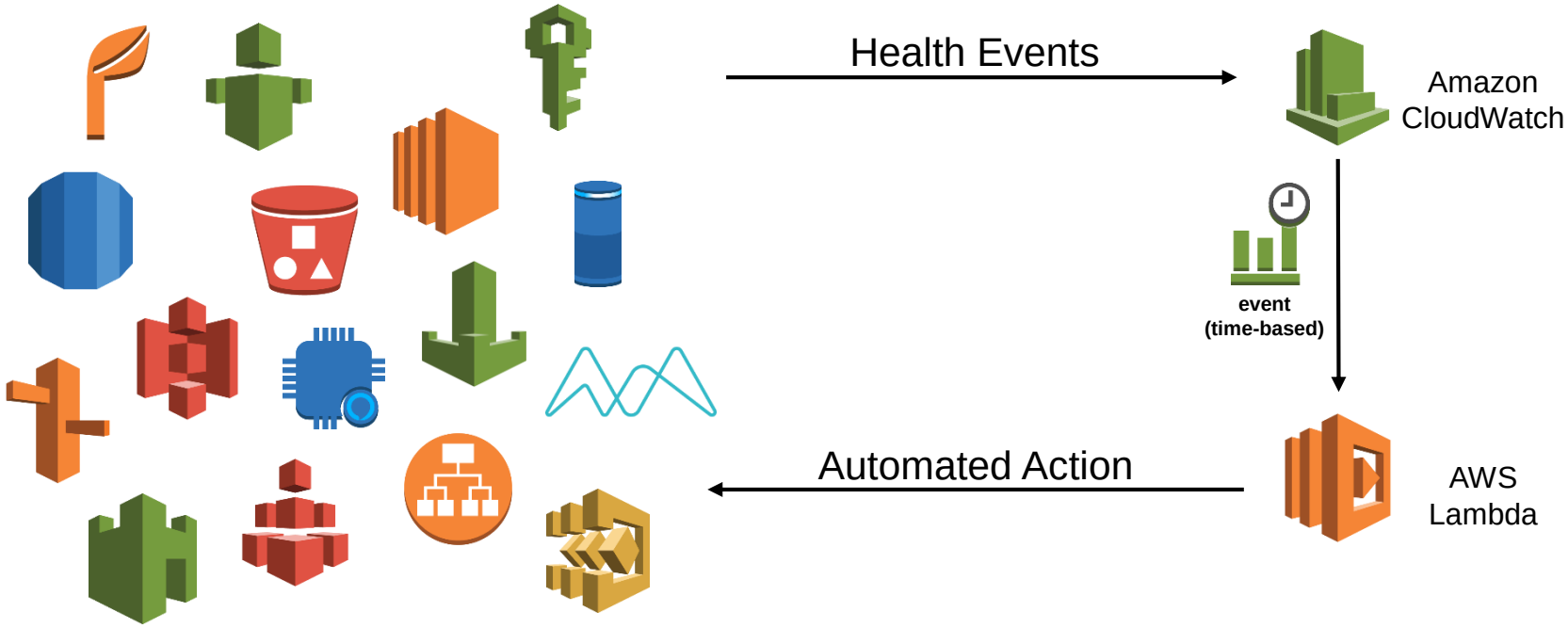
Amazon CloudWatch is a monitoring service for AWS cloud resources and the applications you run on AWS. You can use Amazon CloudWatch to collect and track metrics, collect and monitor log files, set alarms, and automatically react to changes in your AWS resources.



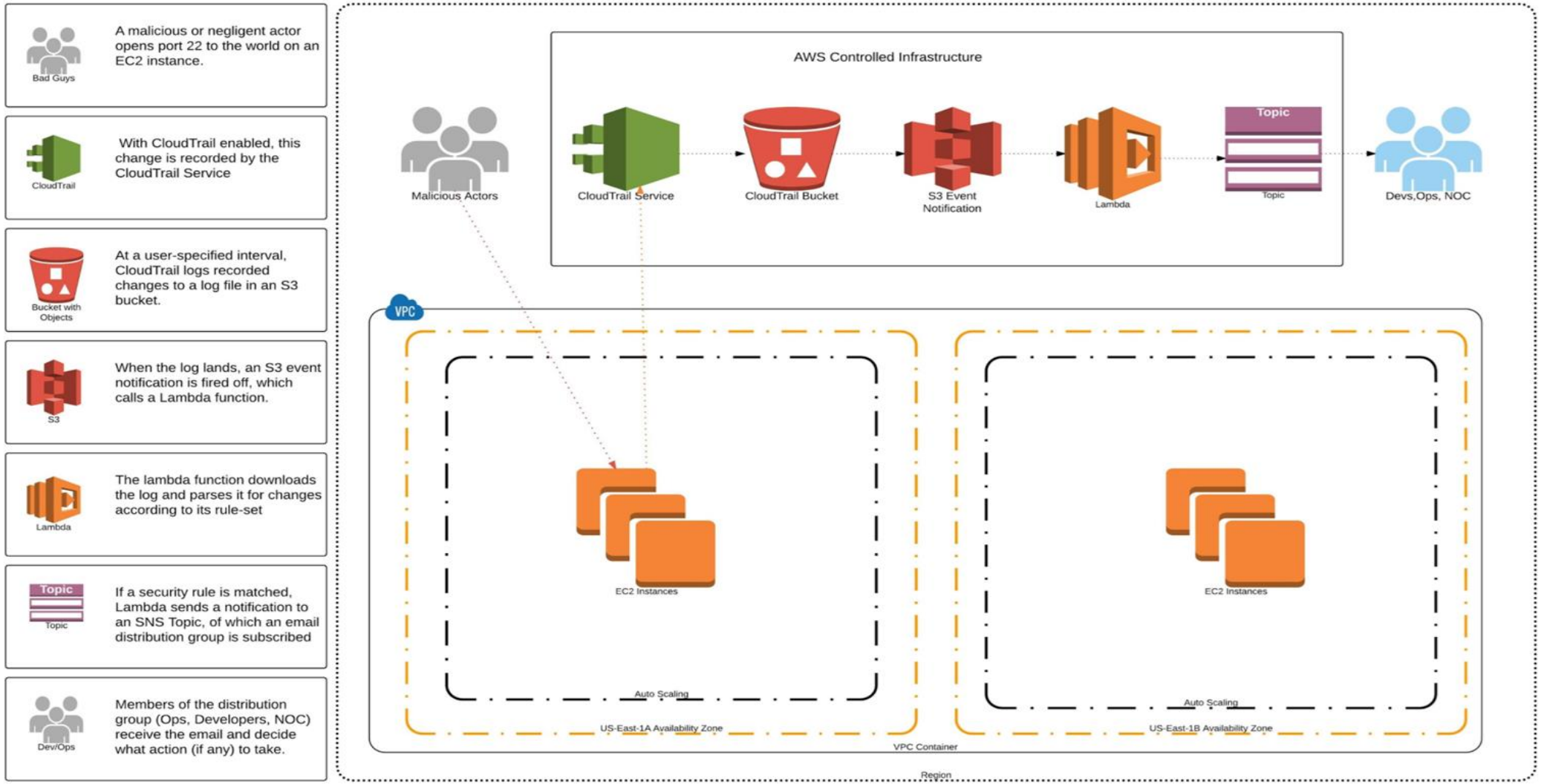
Lambda as Operations

Function as a Service (FaaS)

- Run code without provisioning or managing servers
- Trigged from other AWS services (i.e. CloudWatch alarm) or schedule based
- AWS Lambda is often used as the "engine" to run code or scripts in response to specific events



Example: Automating Events in AWS



Lab Account - Governance Update

AWS Governance + Standards

EC2 SHUTDOWN

1

- Shutdown instances daily
- TAG: AutoOff = **True**
- Launches instances daily
- TAG: AutoOn = **True**
- Notify Owner

EBS LIFECYCLE & TERMINATION

2

- Snapshots EBS Volumes after 14 days of being **available**
- Terminate EBS Volumes after 14 days of being **available**
- Setup Lifecycle policy to glacier after 14 days
- Terminate from Glacier after 14 days
- Notify owner

TAG ENFORCER

3

- ***Terminate*** instances ***immediately*** if missing required tags
- TAGS: **Name, Owner, ProjectID**
- Notify Owner

EC2 LIFECYCLE & TERMINATION

4

- Terminate instances after 30 days of being **stopped**
- Invoke **'EBS LIFECYCLE and TERMINATION'** function
- TAG: DoNotDelete = **False**
- Notify Owner

AHEAD®

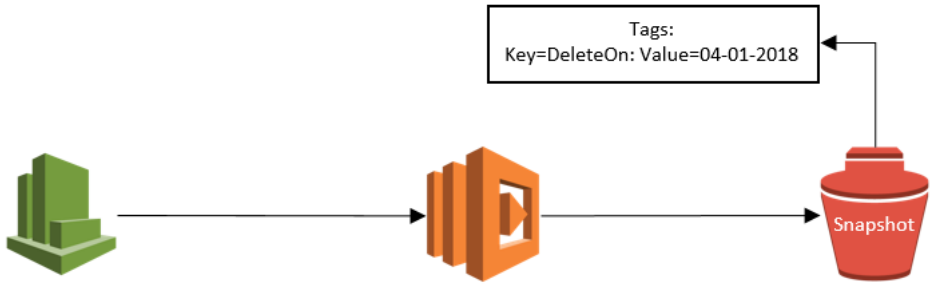
Lab 2 - Infrastructure as Code (Operations)

In this lab, participants will learn different techniques and use cases on how to manage changes to your automated AWS environment. Users will learn how to update existing stacks and add operational controls to the environment, such as data protection.

AHEAD®

Lab 2 diagram

EBS-Snapshot-Create-Function



CloudWatch Event

A scheduled CloudWatch Event runs at 11p (configurable) and calls the EBS-Snapshot function.

EBS-Snapshot Lambda

The EBS-Snapshot Lambda function runs and creates a snapshot for any EC2 instances tagged with 'backup'

Snapshot

Based on the retention period defined, an EBS snapshot is stored in S3 with a 'DeleteOn' tag that defines the date the snapshot should be deleted.

EBS-Snapshot-Delete-Function

CloudWatch Event

A scheduled CloudWatch Event runs at 10p (configurable) and calls the Delete-Snapshot function.

EBS-Delete-Snapshot Lambda

The delete Snapshot Lambda looks for any EBS snapshots that have a 'DeleteOn' tag equal to the current day and deletes the snapshot.

AHEAD Innovation Day

Managing Compliance with AWS Systems Manager

AHEAD®

AWS Systems Manager

- Provides a list of capabilities including:
  - Run Command
  - State Manager
  - Patch Compliance
  - Maintenance Windows
  - Parameter Store
  - and more….

Relatable Services

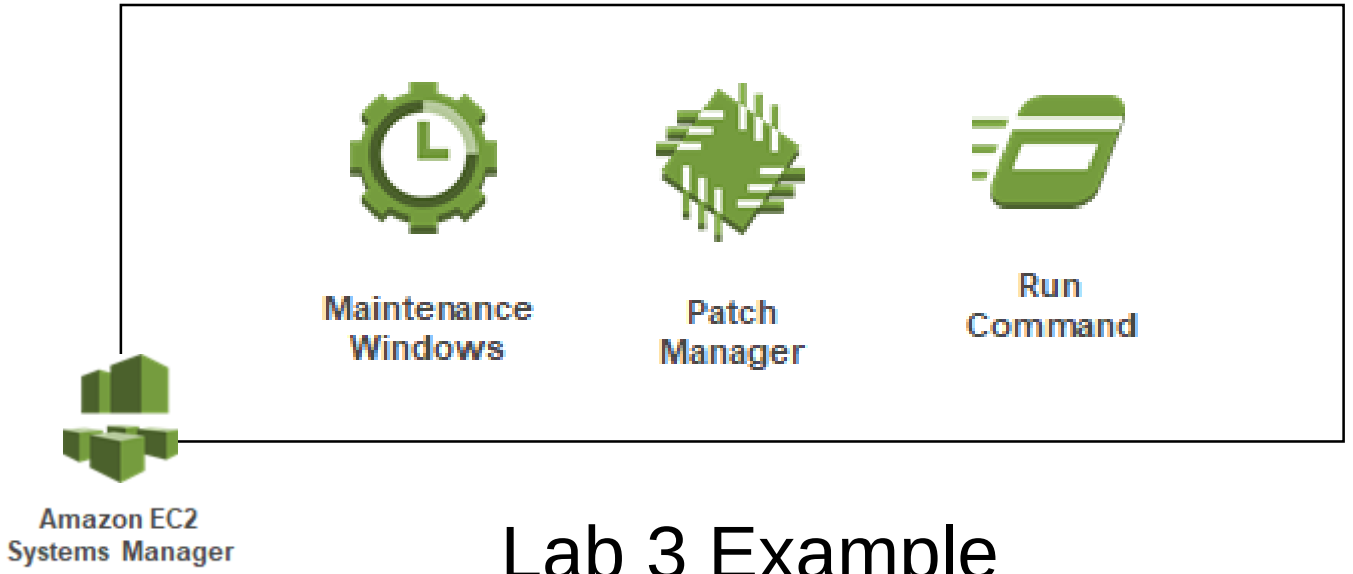
AWS System Manager	Comparable Products
AWS Run Command	BIGFIX
AWS Patch Manager	Windows Server Update Services 3.0
AWS Parameter Store	HashiCorp Vault

AWS Systems Manager

- Some System Manager Services are used in the background
- Patch Manager
  - Uses SSM Run Command to install and check for patches
- Microsoft AD, AD Connector, Simple AD
  - Use Run Command to automatically join Windows Servers to the domain

Combine System Manager Services

- Multiple Services are often Chained Together



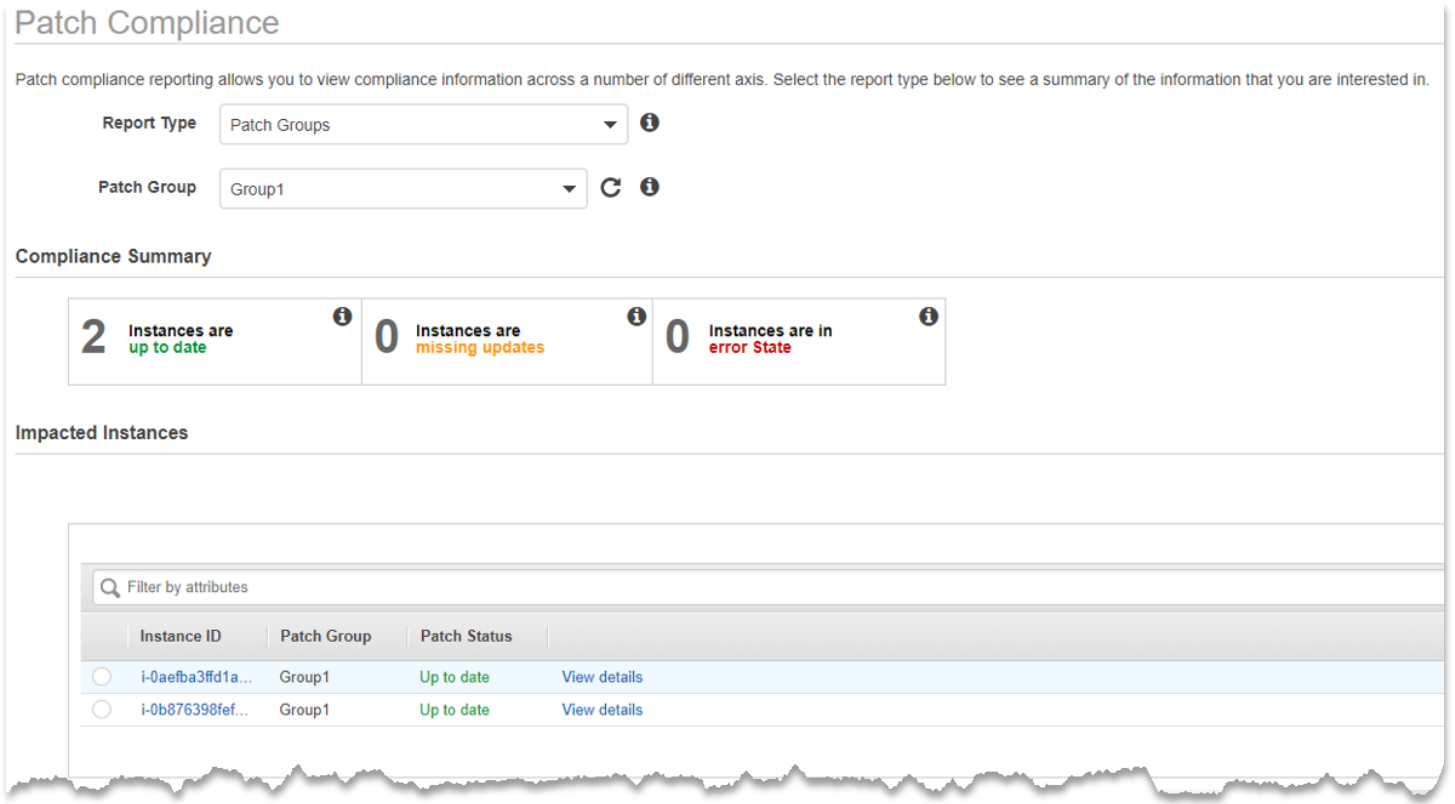
Lab 3 Example

AWS Systems Manager - Prerequisites

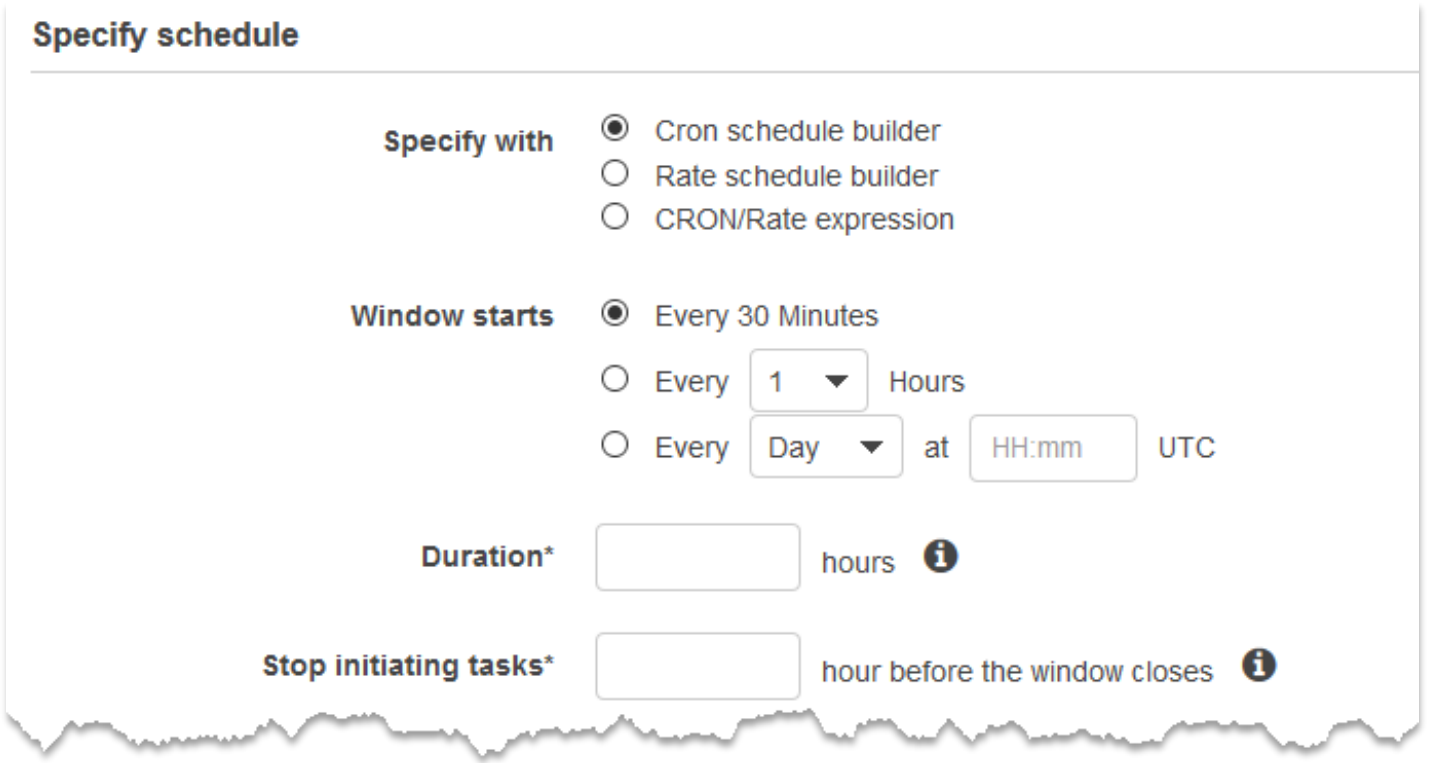
- Requires an agent on any EC2 instance or managed instance running in a non-AWS environment
- EC2 role with permissions to interact with the EC2 Systems Manager services
- A PassRole policy to register tasks on your behalf.

AWS Systems Manager – Patch Compliance

- Built-in patching mechanism provided by AWS
- Not as robust as SCCM or WSUS but effective...and free.

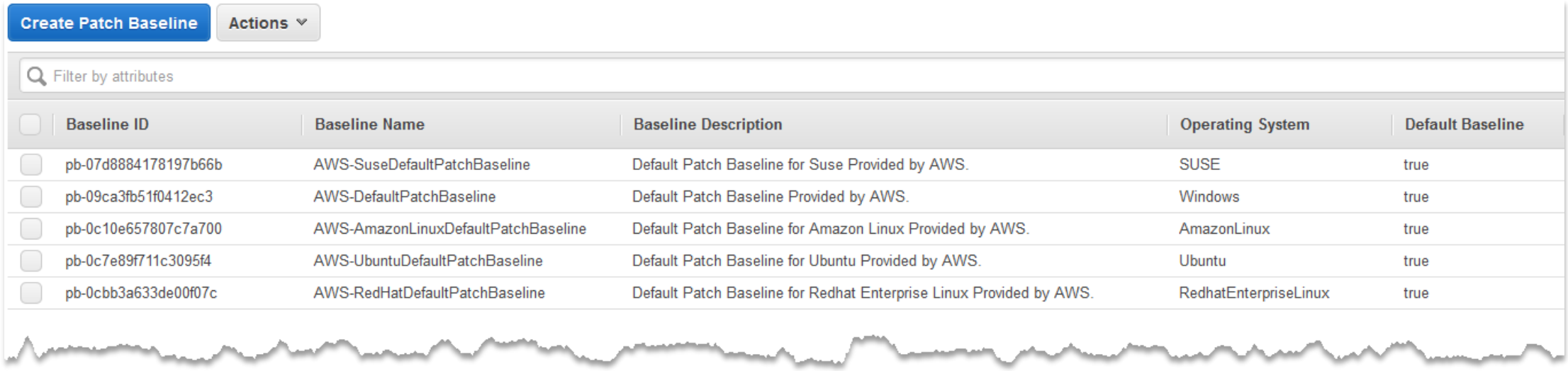


AWS Systems Manager – Maintenance Window



AWS Systems Manager – Patch Baselines

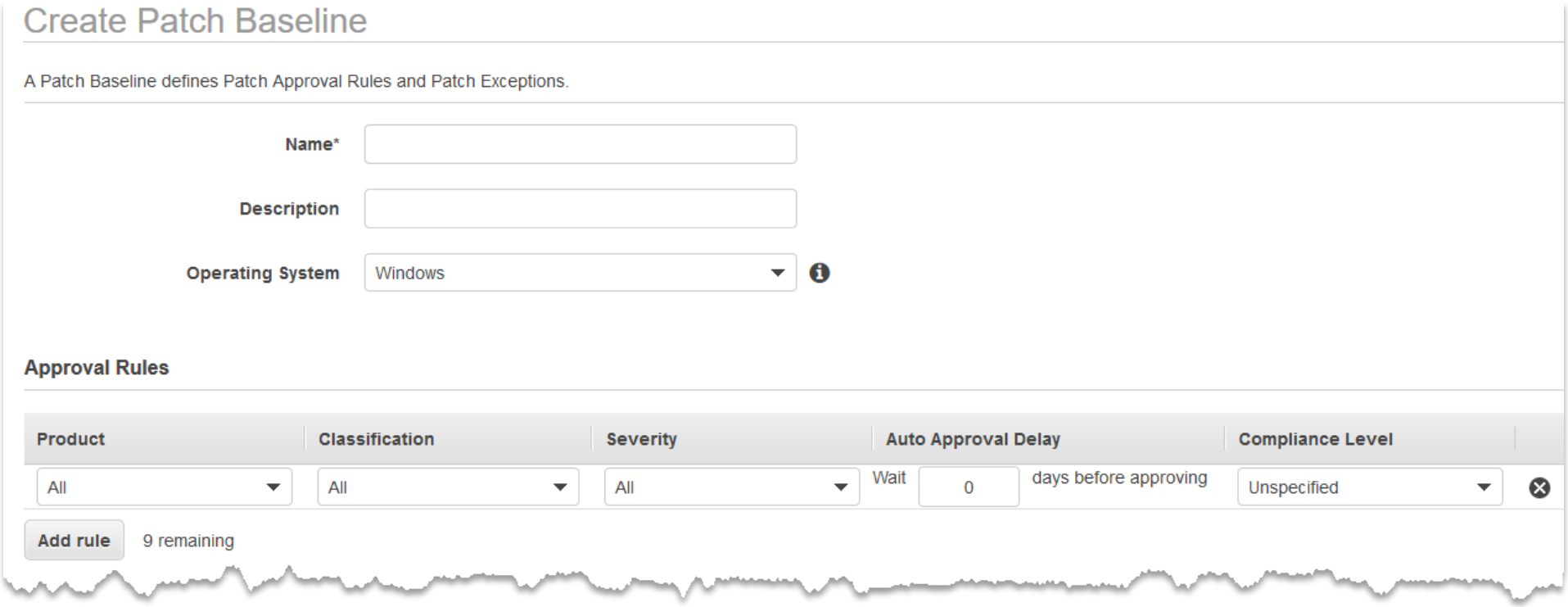
- Pre-configured patch baselines for general defaults.
- Custom patch baselines based on your own organizational policies



Baseline ID	Baseline Name	Baseline Description	Operating System	Default Baseline
pb-07d8884178197b66b	AWS-SuseDefaultPatchBaseline	Default Patch Baseline for Suse Provided by AWS.	SUSE	true
pb-09ca3fb51f0412ec3	AWS-DefaultPatchBaseline	Default Patch Baseline Provided by AWS.	Windows	true
pb-0c10e657807c7a700	AWS-AmazonLinuxDefaultPatchBaseline	Default Patch Baseline for Amazon Linux Provided by AWS.	AmazonLinux	true
pb-0c7e89f711c3095f4	AWS-UbuntuDefaultPatchBaseline	Default Patch Baseline for Ubuntu Provided by AWS.	Ubuntu	true
pb-0cbb3a633de00f07c	AWS-RedHatDefaultPatchBaseline	Default Patch Baseline for Redhat Enterprise Linux Provided by AWS.	RedhatEnterpriseLinux	true

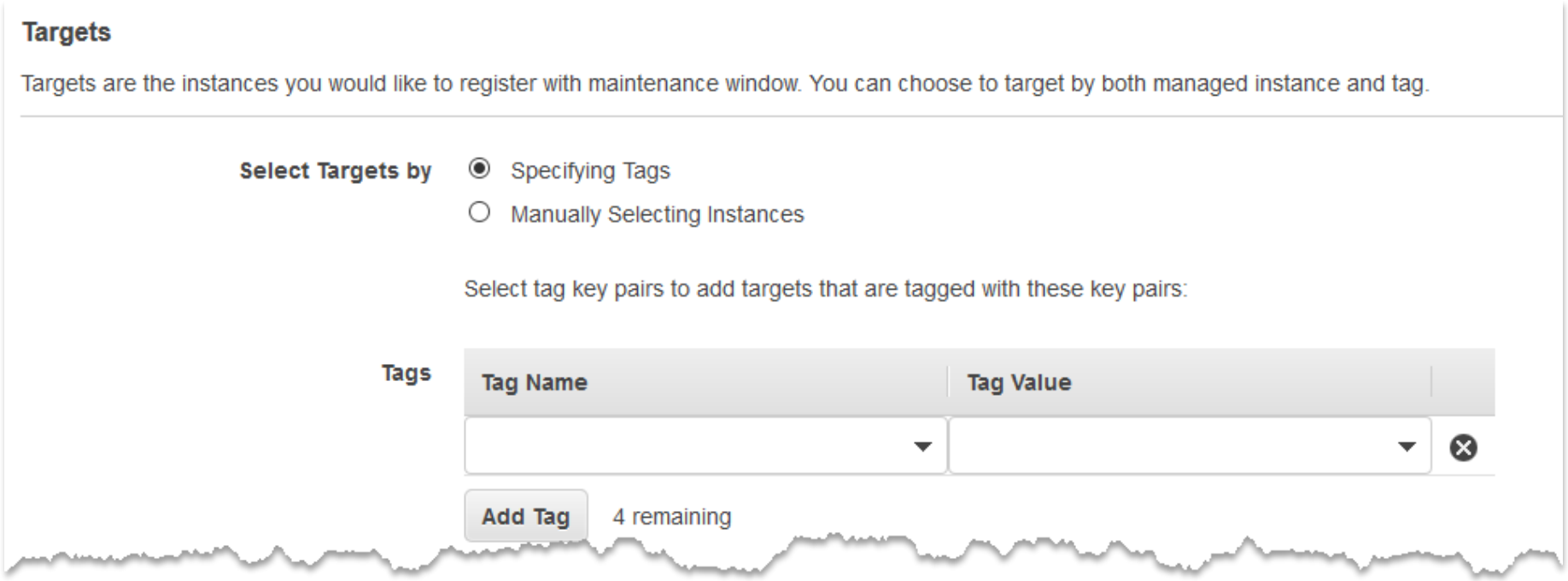
AWS Systems Manager – Custom Patches

- Customize patches for the baseline, including selecting the OS, classification, severity, and product with auto approval delay options.



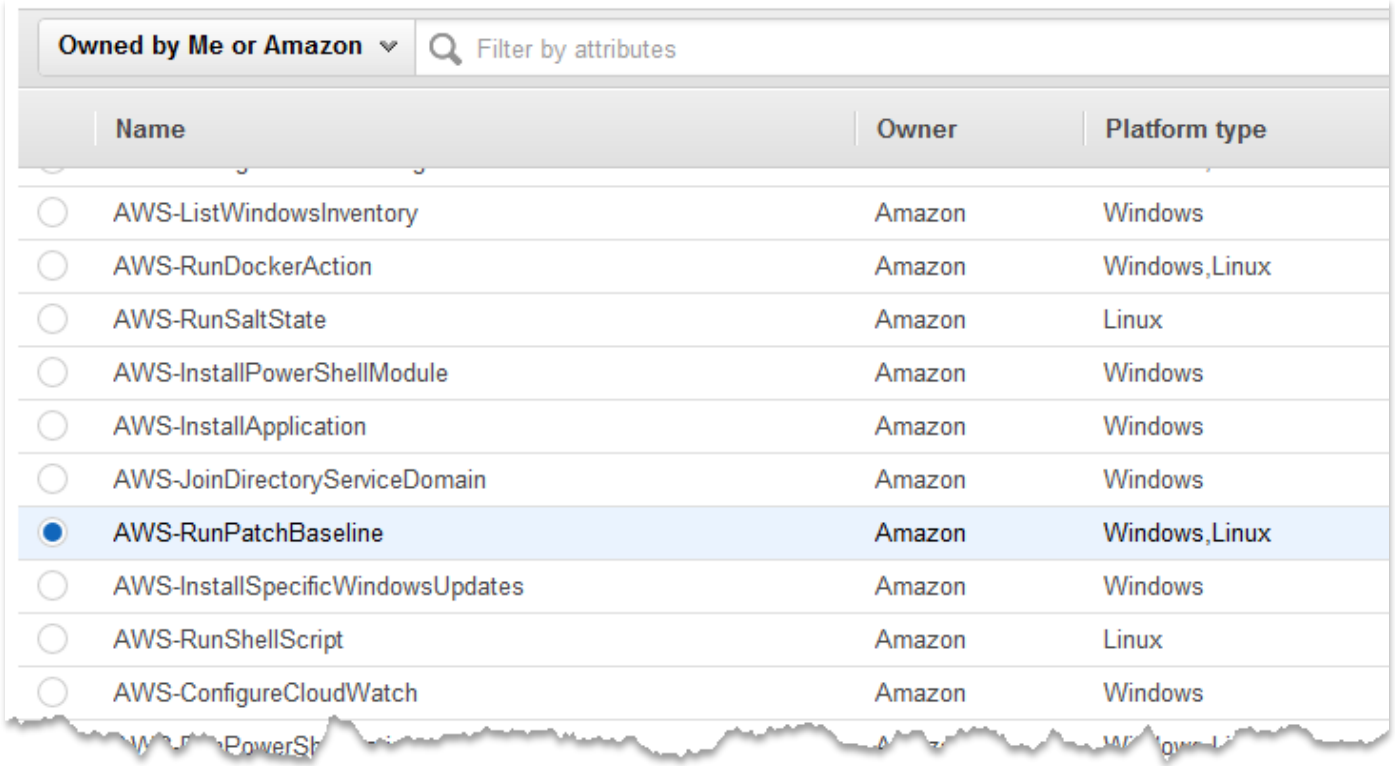
AWS Systems Manager – Target Groups

- Group instances by Tags to configure different patching schedules for different environments, such as production, dev, and test.



AWS Systems Manager – Run Command

- Run Patch Baseline executes the patching mechanism on EC2 instances.
- Baseline can either scan or scan & install.



Lab 3 – Image Operations

In this session, participants will learn how to use AWS native services for image patch management of EC2 instances. The lab will cover the integration of multiple services and how they work together to build a comprehensive operational strategy.

AHEAD®

AHEAD Innovation Day

Automating Security with AWS Config

AHEAD

Breach!!!

SC Media US > News > Cloud Security > Open AWS S3 bucket exposes private info on thousands of Fedex customers

by Teri Robinson, Executive E

Follow @TeriRsNY

February 15, 2018

Open AWS S3 bucket ex
thous

by Lindsey O'Donnell

A misconfigured Amazon (S3) Simple Storag
jewelry partner, left personal details and co
exposed to the public internet.

PRIVACY SECURITY

Data breach of 48 million records was scraped from public social network pages

Brian Jackson @brianjjackson
Published: April 18th, 2018

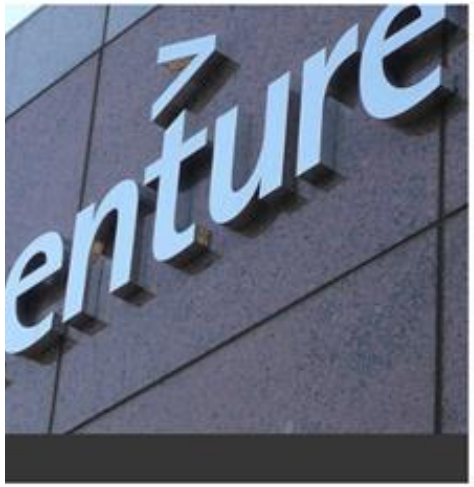
A data breach of 48 million personal records first revealed to a House of Commons committee was caused by an error made by business data search service LocalBlox, according to cyber security vendor UpGuard.

A misconfigured Amazon Web Services (AWS) S3 bucket was discovered by Chris Vickery, a member of the UpGuard Cyber Risk Team on Feb. 18. As has been the case in many misconfigured S3 buckets over the past 18 months, the data was externally exposed to the public Internet.

Accenture latest to breach
ue to misconfigured

f sensitive client and company data
tech and cloud giant accidentally left
ts open to the public.

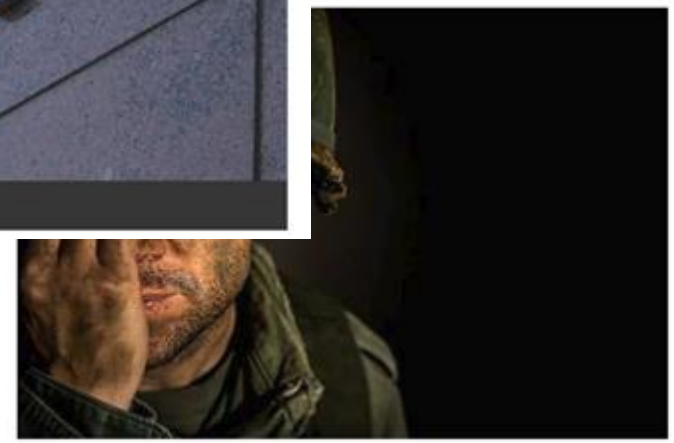
7 | 03:35 PM



ry social media
t wide open in AWS

osed, your tax dollars at

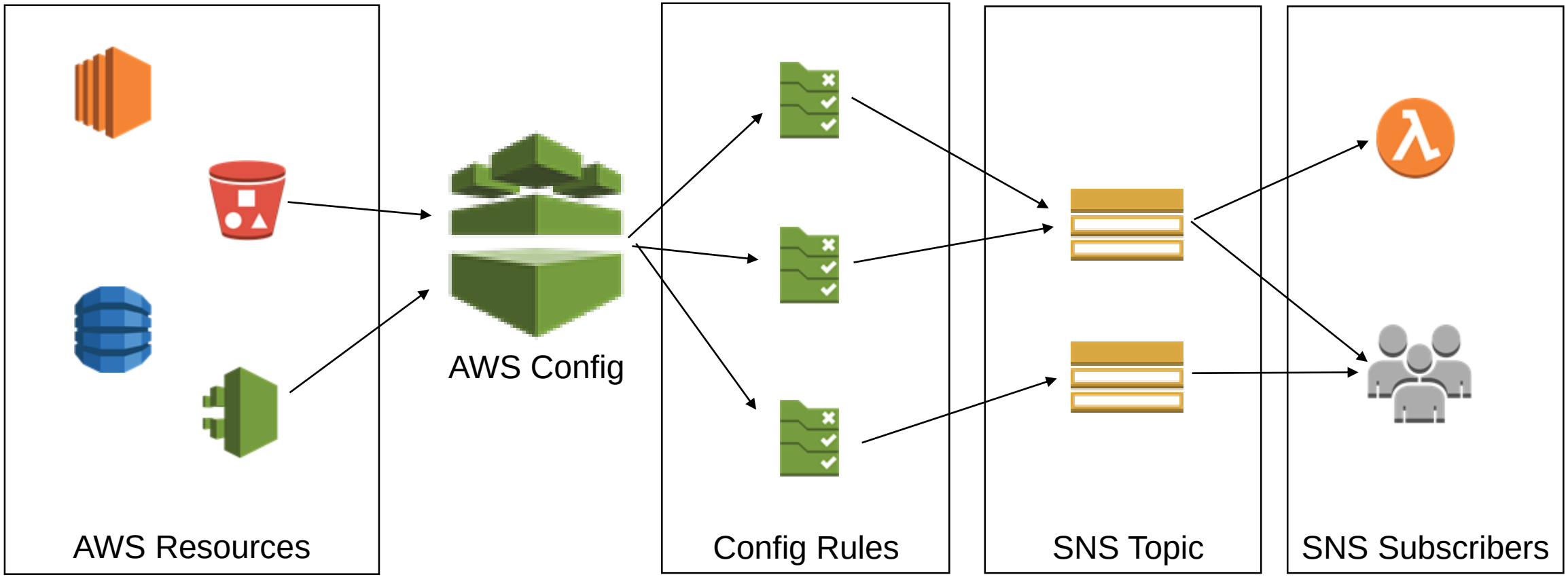
Nov 2017 at 20:08 66 SHARE



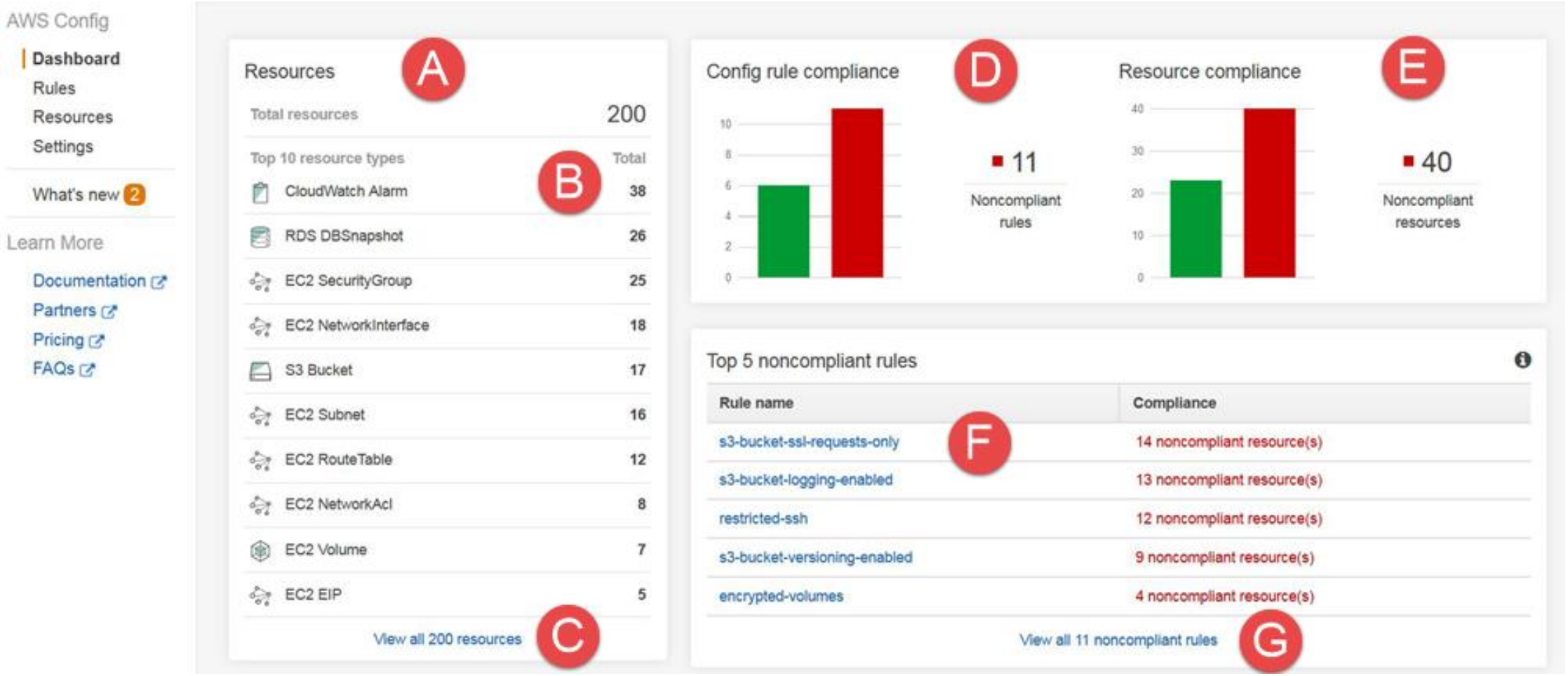
AWS Config

- Provides an inventory of AWS resources and a history of configuration changes to these resources
- All configuration changes are recorded and stored on S3
- Define Config rules to evaluate configurations to ensure compliance
- Select from the pre-defined rules or customize your own to suit your needs.

Config Overview

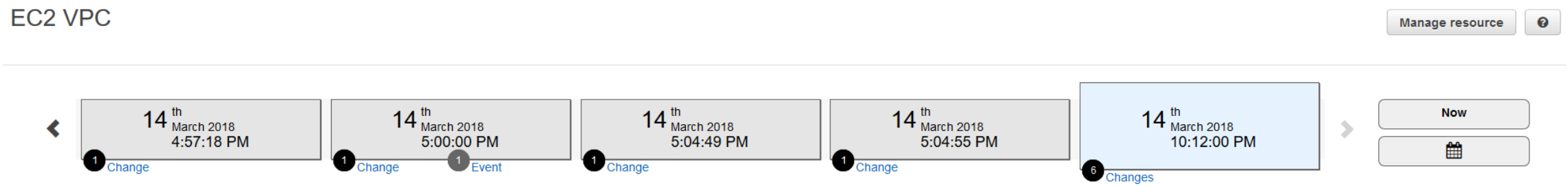


AWS Config Dashboard



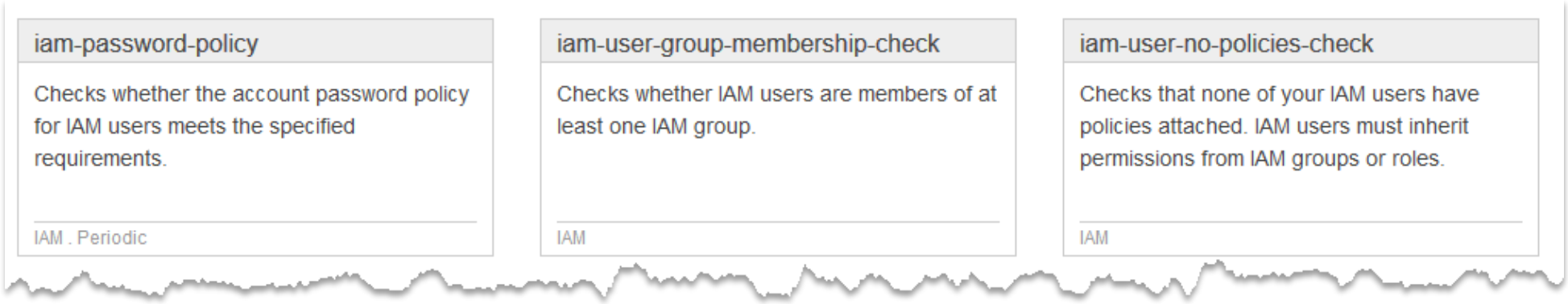
AWS Config – Change Tracking

- Config continuously tracks changes throughout the AWS environment
- Config can monitor a subset of specific AWS resources, such as S3 buckets or it can monitor the entire environment
- Config can record changes from the environment in real-time, or periodically, such as once a day



AWS Config – Config Rules

- Managed Config Rules:
  - Managed by AWS and ready to immediately use

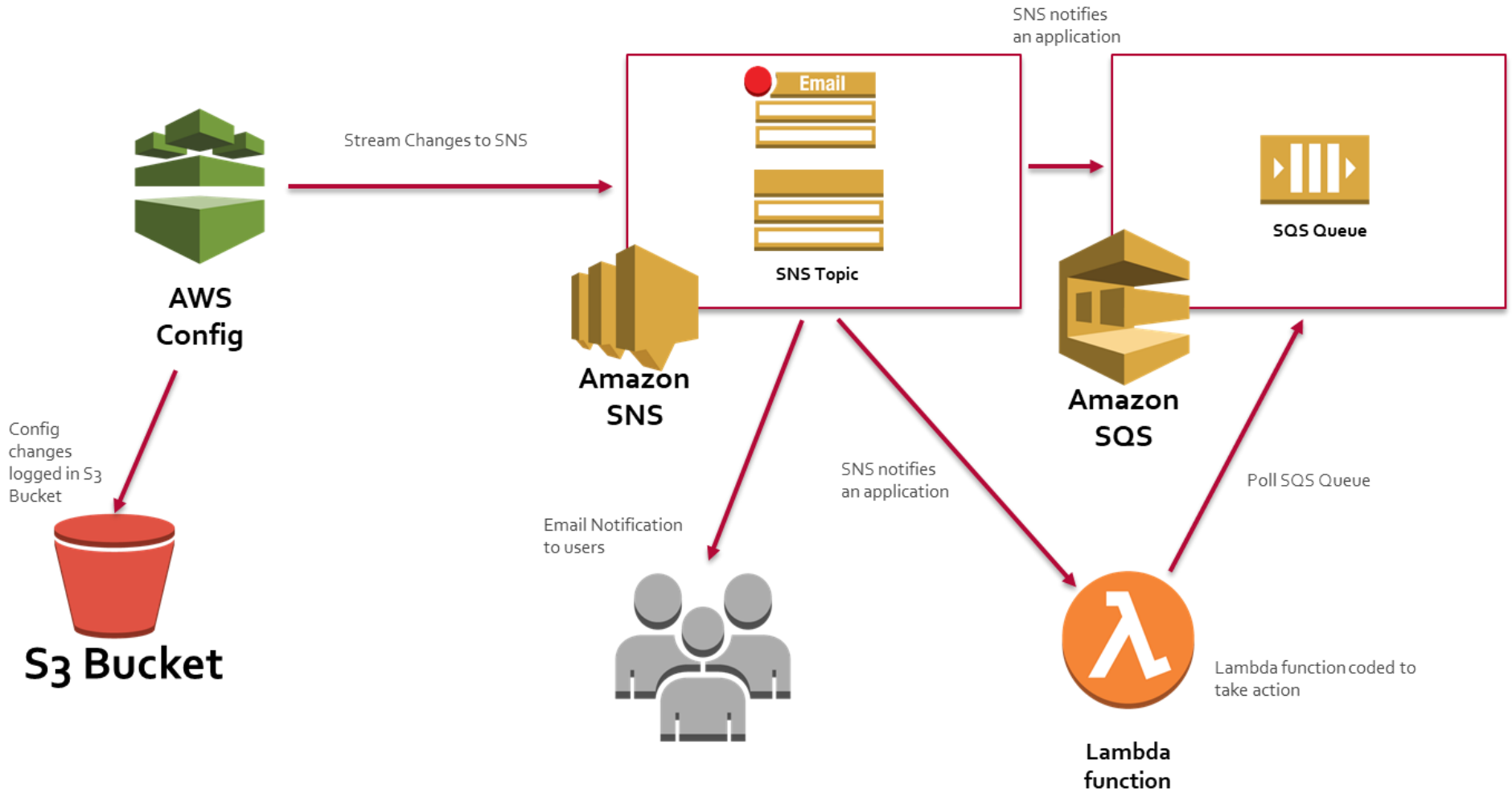


- Custom Config Rules:
  - Lambda functions written by customers used to check and remediate any number of configurations deemed unwanted.

AWS Config – Config Remediation

- AWS Config can not only record changes to the environment, but can take actions based upon undesired resources configurations.
- Config stores log files in a specified S3 bucket but can also send changes to an Amazon Simple Notification Service (SNS) topic.
- Subscribers to these SNS topics can be notified by email, or applications can be notified to take action.

AWS Config - Actions



Lab 4 – Automating Security

In this lab, participants will learn how to automate the security and self-remediation of their AWS environments using AWS native tools. Using the combination of Config, SNS, and Lambda, users will ensure that their sensitive data stored in S3 remains private and out of the hands of unauthorized users.

AHEAD®

AHEAD Innovation Day

Autoscaling Workloads on AWS

AHEAD®

AWS Auto Scaling

- Monitors applications and adjusts capacity to meet demands
- Scale Up to meet demands, Scale Down to save costs
- Will provide fault tolerance by replacing impaired instances
- Dynamically scale or set a schedule for scaling activities
- Define your own metrics to invoke a scaling event
- Previously was limited to EC2 but has been expanded to ECS, DynamoDB, and Amazon Aurora

AWS Auto Scaling – Launch Configuration

- Launch Configuration – template that an Auto Scaling group uses to launch EC2 instances.
  - AMI
  - Instance Type
  - Key Pair
  - Security Groups
  - Block Device Mappings
- Cannot modify a launch configuration
  - Must create a new launch configuration and modify ASG to point to new one

AWS Auto Scaling – Auto Scaling Groups (ASG)

- Collection of EC2 instances sharing similar functions
- Treated as a logical group for scaling and management
- A single application may have multiple ASGs (i.e., web tier, app tier)
- Configuration items include:
  - Minimum Capacity
  - Maximum Capacity
  - Desired Capacity
  - Network Configuration
  - Scaling Rules
  - Notifications
  - Load Balancing

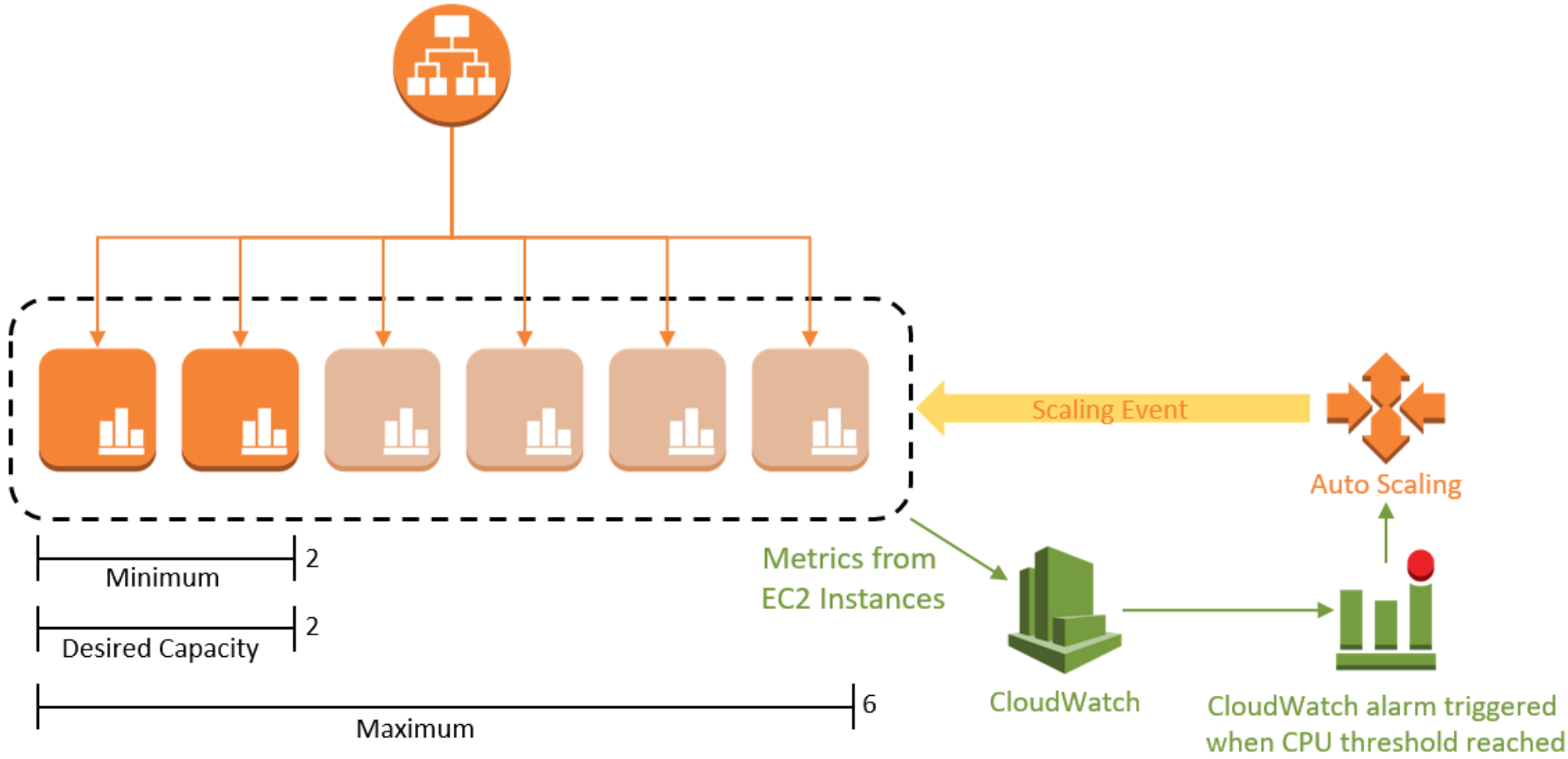
AWS Auto Scaling – Scaling Workloads

- Scaling means an increase *or* decrease in resources
- Three ways to configure scaling for your applications
  - Manual - modifications in the console)
  - Scheduled – if application has predictable traffic patterns
  - Dynamic – scale based on EC2 instance or application metrics
- Should have a minimum of two scaling policies for each ASG
  - Scale Out – Adding Resources to handle workloads
  - Scale In – Remove Resources to save costs

AWS Auto Scaling – Scaling Policies

- Can use CloudWatch metrics to trigger scaling event
  - CPU Utilization
  - Number of Connections to Load Balancer
  - Custom application metrics (customer created)
- Can use SQS to trigger scaling as well
  - Example - if messages in queue > 100, add capacity

AWS Auto-Scaling



AHEAD Cloud Delivery Framework

Below are the areas that will be architected and deployed as part of AHEAD foundation or enterprise public cloud environment.

Education

Accounts and Structure

Network Design

EC2 Instances

Storage

Application Architecture

Identity Management

Platform Security

Governance

Operations and Monitoring

Systems Management and Automation

Inventory, Cost, and Security Optimization

Operations Framework

AHEAD Well-Architected Framework

The Well-Architected Framework is designed to provide you with high-level guidance and best practices, not an audit, to help you build and maintain **secure, reliable, performant, cost optimized, and operationally excellent** applications in the Cloud.

Operational Excellence

Security

Reliability

Performance Efficiency

Cost Optimization

Free $5,000 in AWS Credits to Remediate Gaps

AHEAD Cloud Delivery Framework

Below are the areas that will be architected and deployed as part of AHEAD foundation or enterprise public cloud environment.

Education

Accounts and Structure

Network Design

EC2 Instances

Storage

Application Architecture

Identity Management

Platform Security

Governance

Operations and Monitoring

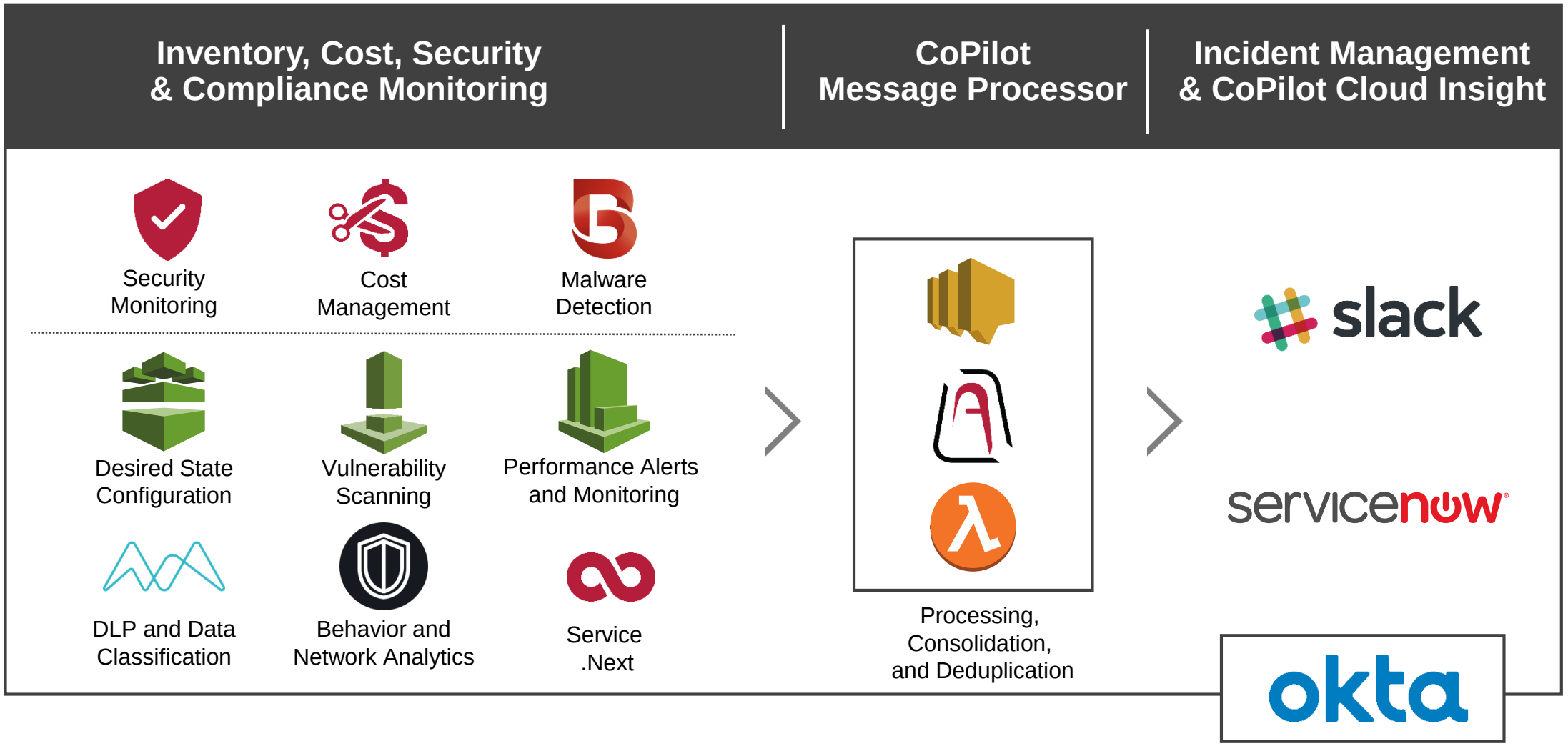
Systems Management and Automation

Inventory, Cost, and Security Optimization

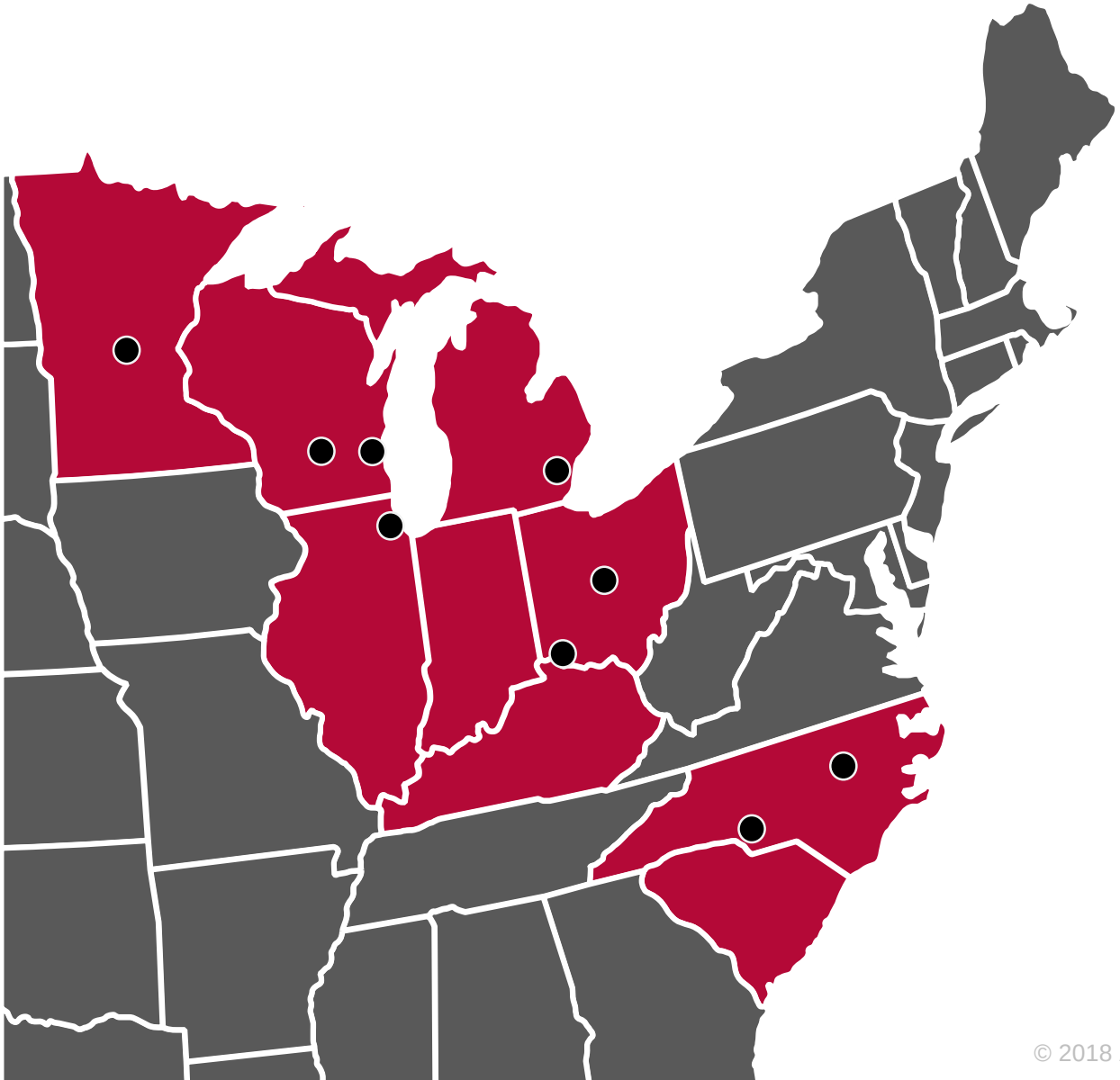
Operations Framework

AHEAD CoPilot™

for AWS

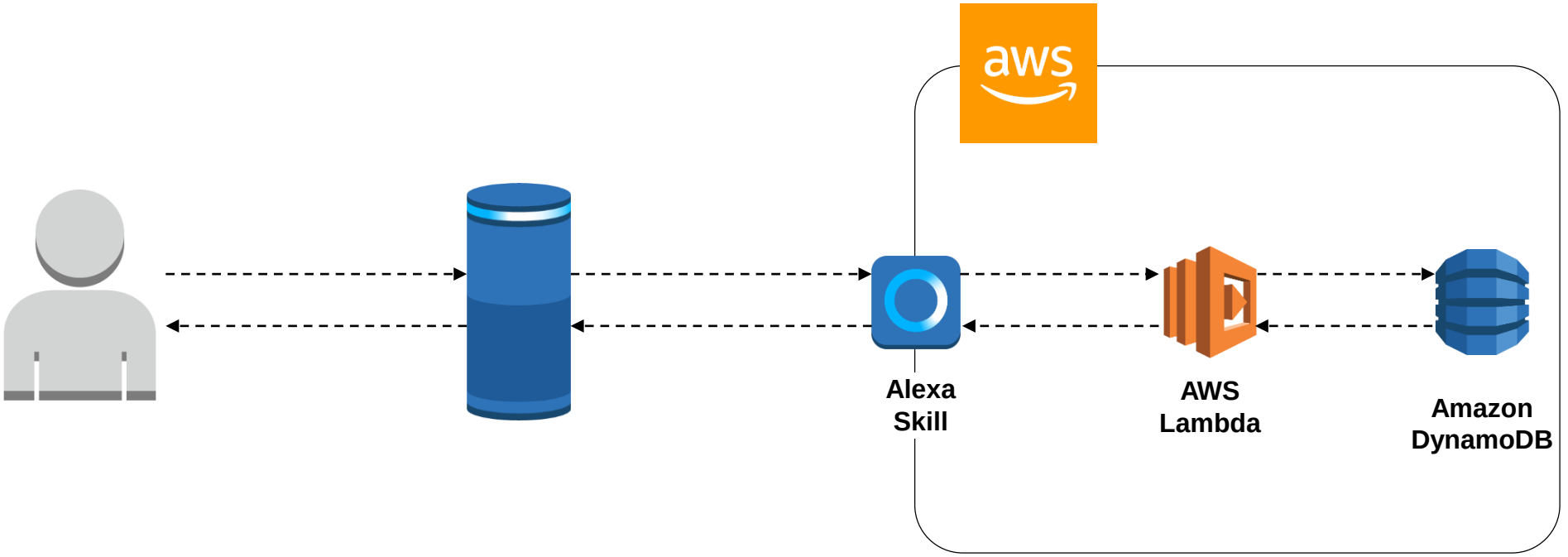


DevOps Innovation Days



- CI / CD Pipeline
- Enterprise DevOps ToolChain
- SDLC and GitFlow Practices
- Serverless and Kubernettes
- Deployment Methods
- What do YOU want to see?

Raffle Architecture



Thank You

Experts in Enterprise Cloud

thinkahead.com