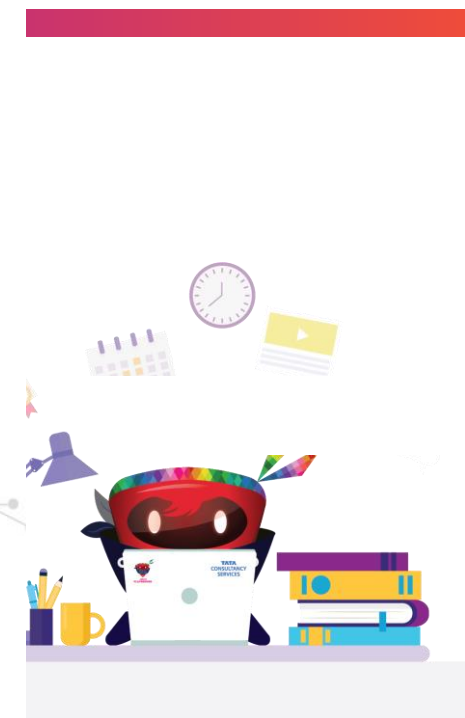


## AWS Solution Architect Associate – Certification Training

North America Talent Development Team



A background network diagram consisting of numerous grey nodes connected by thin grey lines. Several clusters of nodes are highlighted in different colors: a purple cluster on the left, an orange cluster in the upper middle, a teal cluster on the right, and a pink cluster at the bottom center. A grey rectangular box with rounded corners is positioned on the left side of the slide.

## Session 2

# *AWS Security Features*

- AWS Security: Overview
- Shared Responsibility Model
- AWS Directory Service
- Encryption
- AWS Certificate Manager
- AWS Key Management Service
- AWS CloudHSM

## Overview

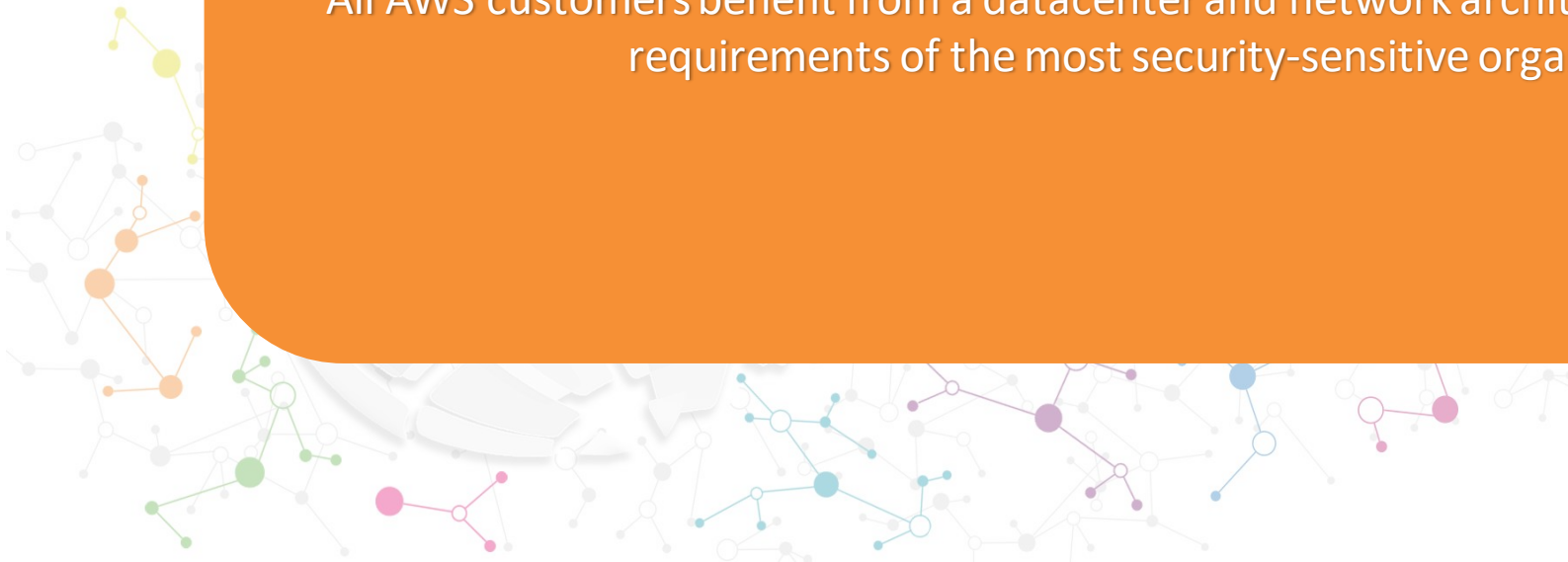


# What are your perceptions on cloud security?



## *At AWS, cloud security is job zero.*

All AWS customers benefit from a datacenter and network architecture built to satisfy the requirements of the most security-sensitive organizations.

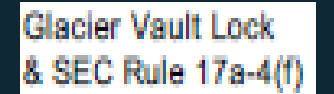


Where would some of the world's top security people like to work? At scale on huge challenges with huge rewards

So AWS has world-class security and compliance teams watching your back!

Every customer benefits from the tough scrutiny of other AWS customers





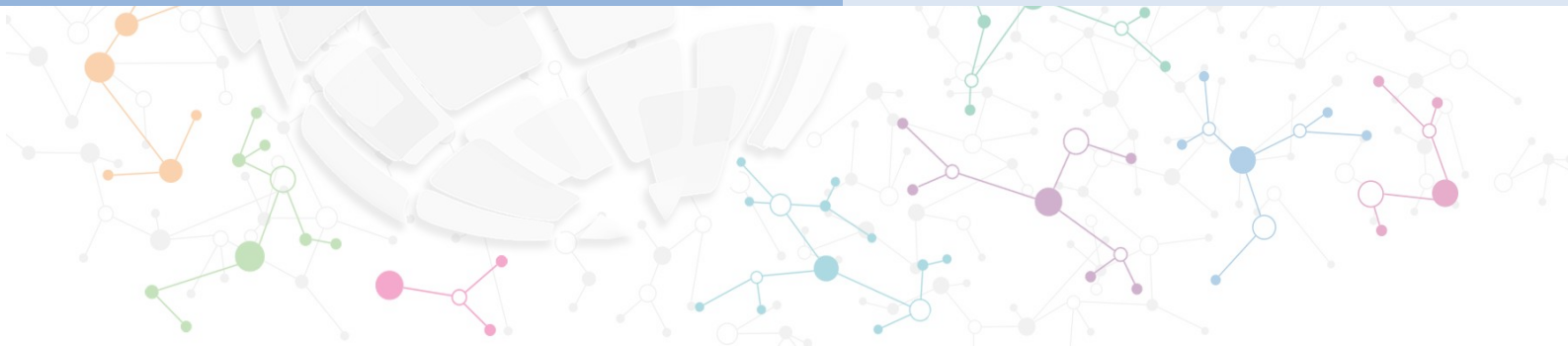


# AWS SAA Boot Camp Session2



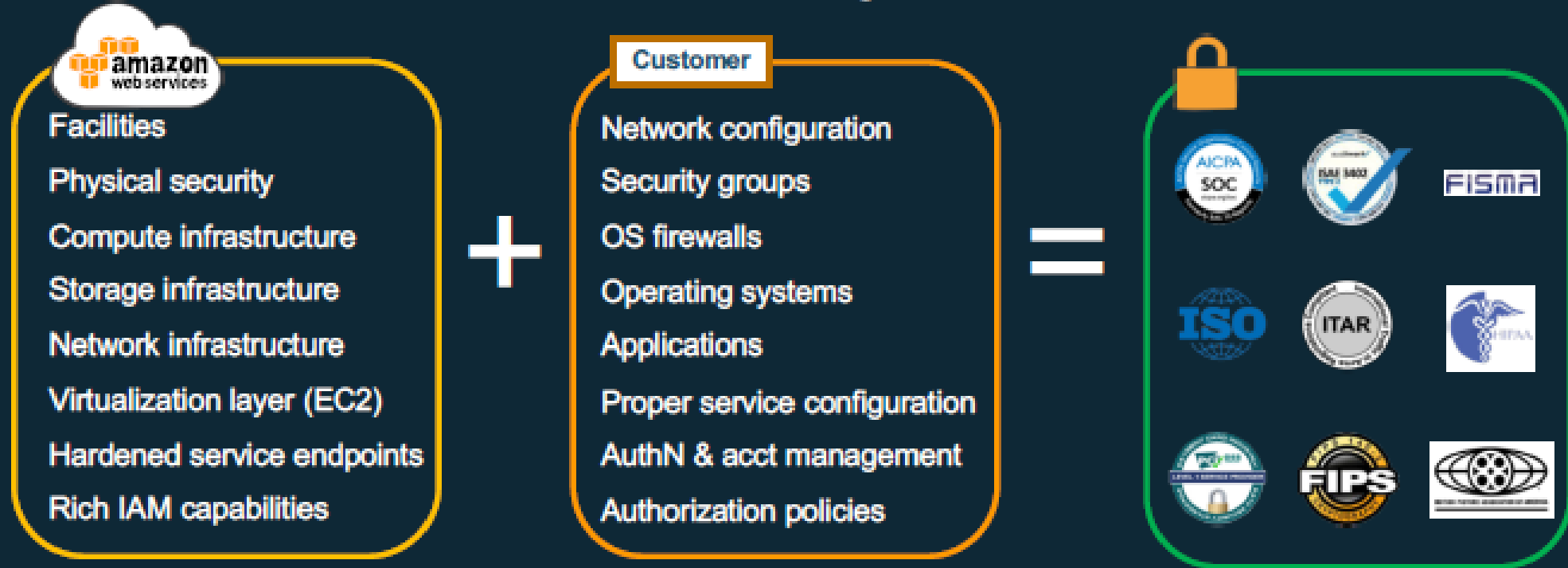
## AWS Security

## Shared Responsibility Model

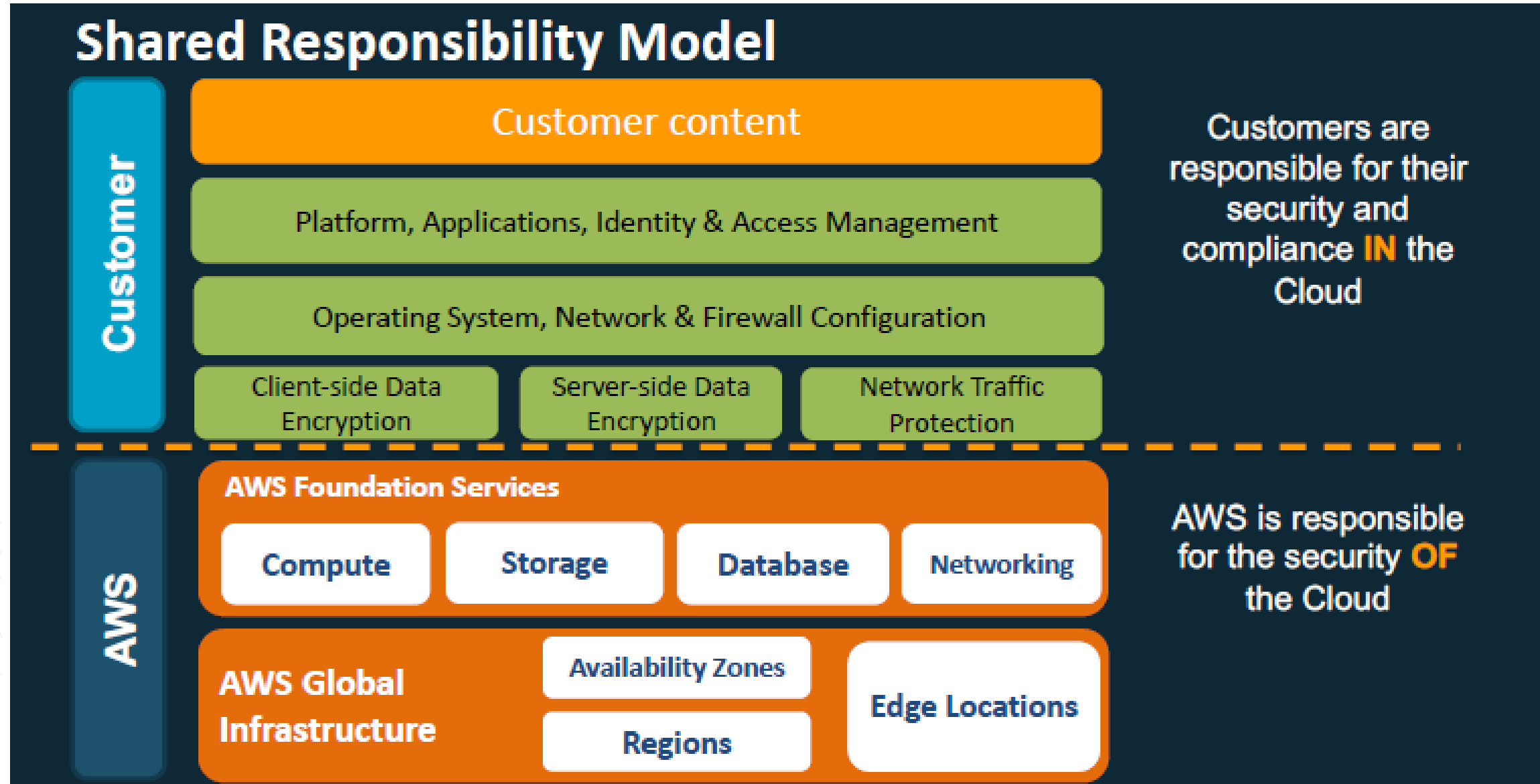




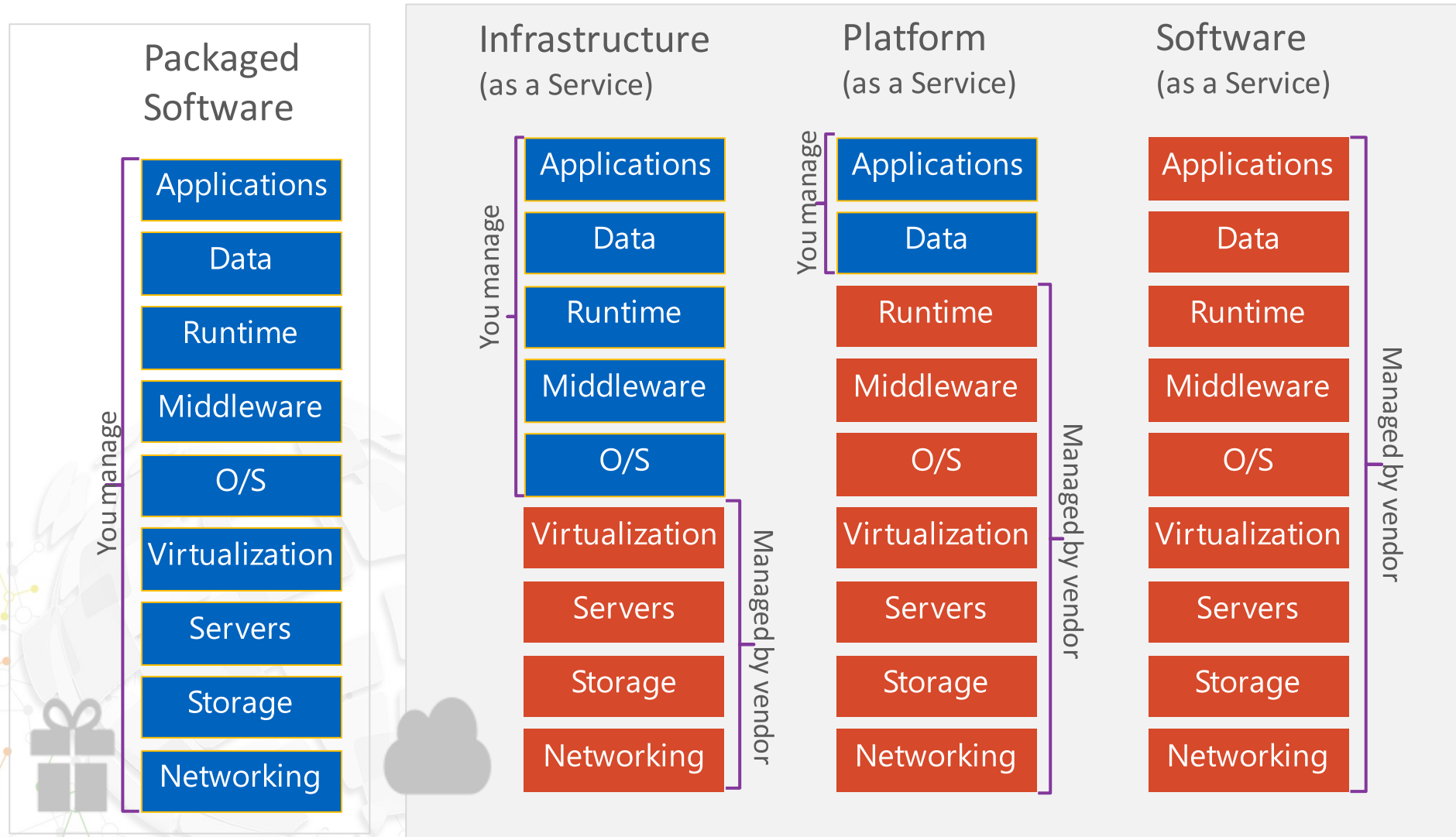
## AWS Shared Responsibility Model



- Scope of responsibility depends on the type of service offered by AWS:  
**Infrastructure, Container, Abstracted Services**
- Understanding who is responsible for what is critical to ensuring your AWS data and systems are secure!



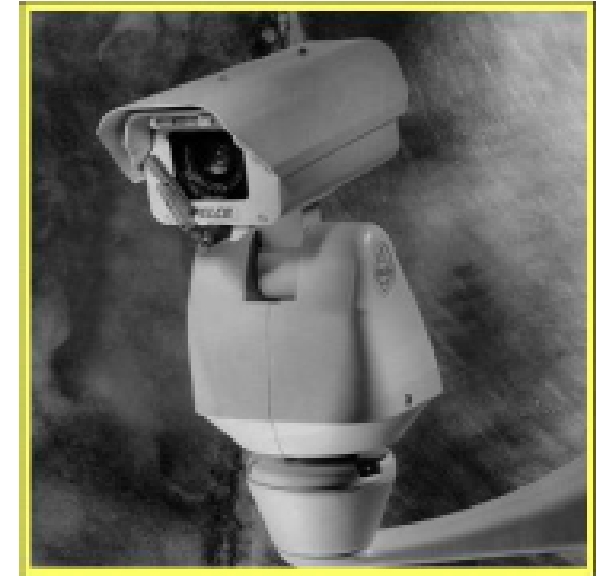
# Service Models





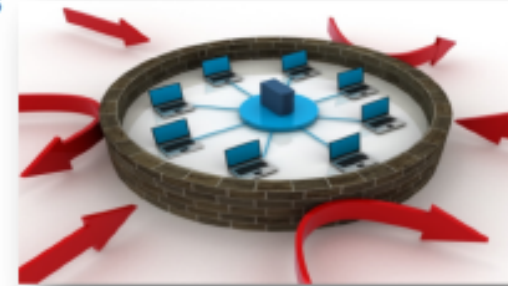
## Physical Security of Data Center

- **Amazon has been building large-scale data centers for many years.**
- **Important attributes:**
  - Non-descript facilities
  - Robust perimeter controls
  - Strictly controlled physical access
  - Two or more levels of two-factor authentication
- **Controlled, need-based access.**
- **All access is logged and reviewed.**
- **Separation of Duties**
  - Employees with physical access don't have logical privileges.



## EC2 Security

- **Host operating system**
  - Individual SSH keyed logins via bastion host for AWS admins
  - All accesses logged and audited
- **Guest (a.k.a. Instance) operating system**
  - Customer controlled (customer owns root/admin)
  - AWS admins cannot log in
  - Customer-generated keypairs
- **Stateful firewall**
  - Mandatory inbound firewall, default deny mode
  - Customer controls configuration via Security Groups



## Network Security

- IP Spoofing prohibited at host OS level.
- Packet sniffing is ineffective (protected at hypervisor level).
- Unauthorized Port Scanning a violation of TOS and is detected/blocked.
- Inbound ports blocked by default.

## Configuration Management

- Most updates are done in such a manner that they will not impact the customer.
- Changes are authorized, logged, tested, approved, and documented.
- AWS will communicate with customers, either via email, or through the AWS Service Health Dashboard (<http://status.aws.amazon.com/>) when there is a potential for service being affected.

## Built for “Continuous Availability”

- **Scalable, fault tolerant services.**
- **All availability zones (AZs) are always on.**
  - There is no “Disaster Recovery Datacenter”
  - All managed to the same standards
- **Robust Internet connectivity**
  - Each AZ has redundant, Tier 1 ISP Service Providers
  - Resilient network infrastructure



## Disk Management

- Proprietary disk management prevents customers from accessing each other's data.
- Disks wiped prior to use.
- Disks can also be encrypted by the customer for additional security.

## Storage Device Decommissioning

- All storage devices go through process using techniques from:
  - DoD 5220.22-M ("National Industrial Security Program Operating Manual").
  - NIST 800-88 ("Guidelines for Media Sanitization").
- Ultimately devices are:
  - Degaussed.
  - Physically destroyed.

## Under the AWS Shared Responsibility Model

### AWS Responsibility? or Customer Responsibility?

Configuring the Security Group rules that determine which ports are open on the EC2 Linux instance

Preventing packet sniffing at the hypervisor level

Patching the operating system with the latest security patches

Shredding disk drives before they leave a datacenter

Securing the internal network inside the AWS datacenters

Installing camera systems to monitor the physical datacenters

Toggling on the Server-side encryption feature for S3 buckets

## Under the AWS Shared Responsibility Model

### AWS Responsibility? or Customer Responsibility?

Configuring the Security Group rules that determine which ports are open on the EC2 Linux instance

Preventing packet sniffing at the hypervisor level

Patching the operating system with the latest security patches

Shredding disk drives before they leave a datacenter

Securing the internal network inside the AWS datacenters

Installing camera systems to monitor the physical datacenters

Toggling on the Server-side encryption feature for S3 buckets

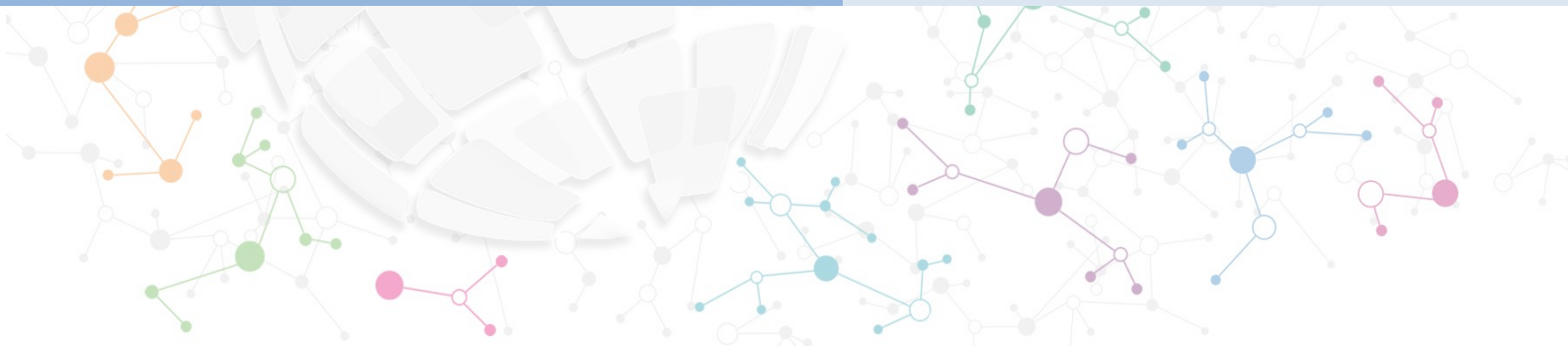


## AWS SAA Boot Camp Session2



## AWS Security

## AWS Directory Service





## AWS Directory Service

*Managed service for Active Directory*

Use your existing Corporate Credentials for

- AWS-based applications
- AWS Management Console



### Microsoft AD

Based on Microsoft Active Directory in Windows Server 2012 R2. Supports adding trust relationships with on-premises domains. Extend your schema using MS AD



### Simple AD

A Microsoft Active-Directory compatible directory powered by Samba 4.



### AD Connector

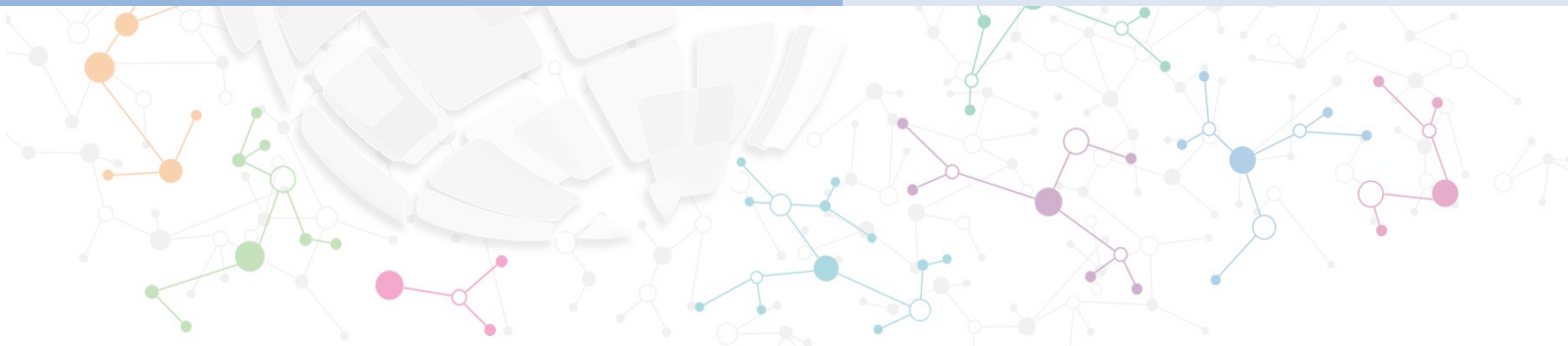
Connect to your on-premises Active Directory. Integrates with existing RADIUS MFA solutions.

# AWS SAA Boot Camp Session2



## AWS Security

## Encryption



## How are you currently encrypting data?





## Encryption

*Protecting data in-transit and at-rest.*



### Encryption In-Transit

HTTPS

SSL/TLS

VPN / IPSEC

SSH

### Encryption At-Rest

Object

Database

Filesystem

Disk

*Details about encryption can be found in the AWS Whitepaper,  
["Securing Data at Rest with Encryption"](#).*

## Encryption at Rest

### Volume Encryption

EBS Encryption

Filesystem Tools

AWS  
Marketplace/Partner

### Object Encryption

S3 Server Side  
Encryption (SSE)

S3 SSE w/ Customer  
Provided Keys

Client-Side Encryption

### Database Encryption

RDS  
MSSQL  
TDE

RDS  
ORACLE  
TDE/HSM

RDS  
MYSQL  
KMS

RDS  
PostgreSQL  
KMS

Redshift  
Encryption

## AWS Certificate Manager

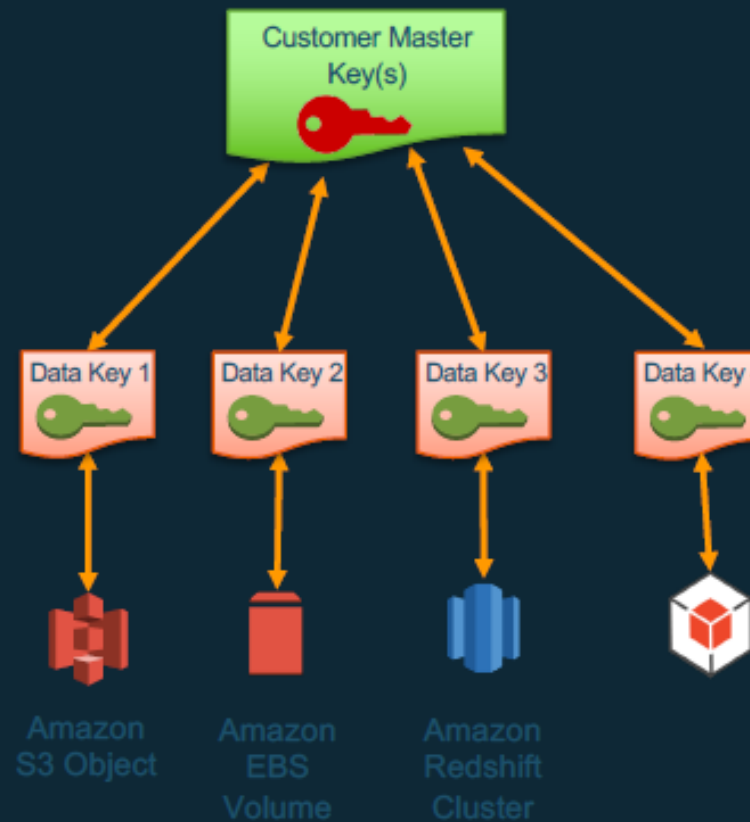


AWS Certificate Manager (ACM) makes it easy to provision, manage, deploy, and renew SSL/TLS certificates on the AWS platform.



## AWS Key Management Service

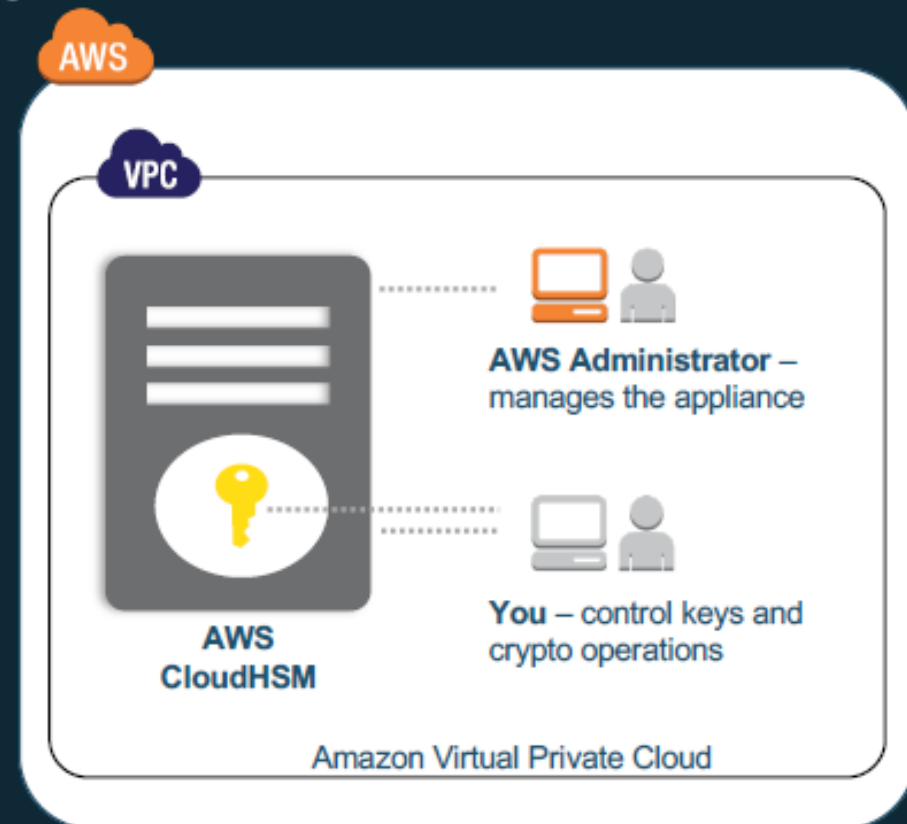
*Managed service to securely create, control, rotate, and use encryption keys.*



## AWS CloudHSM

*Help meet compliance requirements for data security by using a dedicated Hardware Security Module appliance with AWS.*

- Dedicated, single-tenant hardware device
- Can be deployed as HA and load balanced
- Customer use cases:
  - Oracle TDE
  - MS SQL Server TDE
  - Setup SSL connections
  - Digital Rights Management (DRM)
  - Document Signing

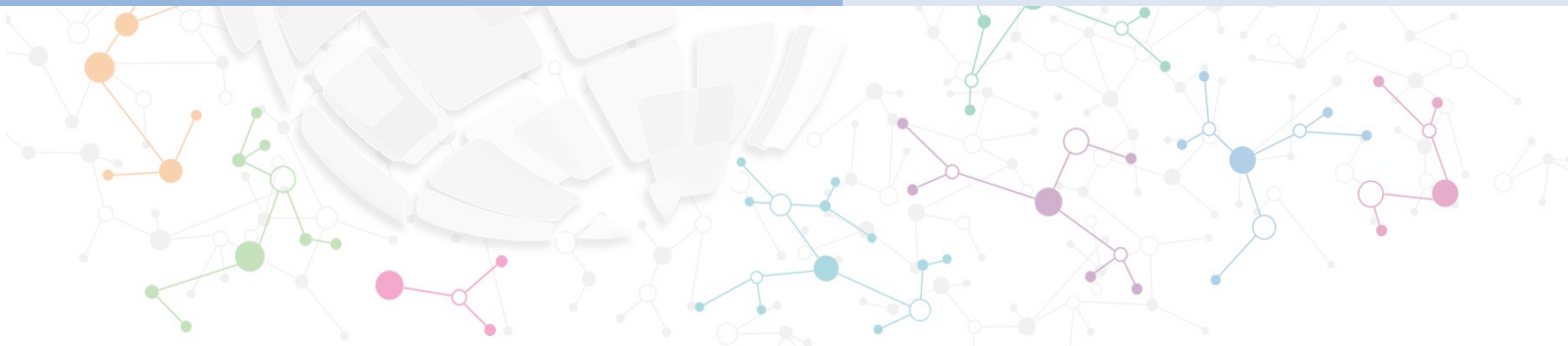


# AWS SAA Boot Camp Session2



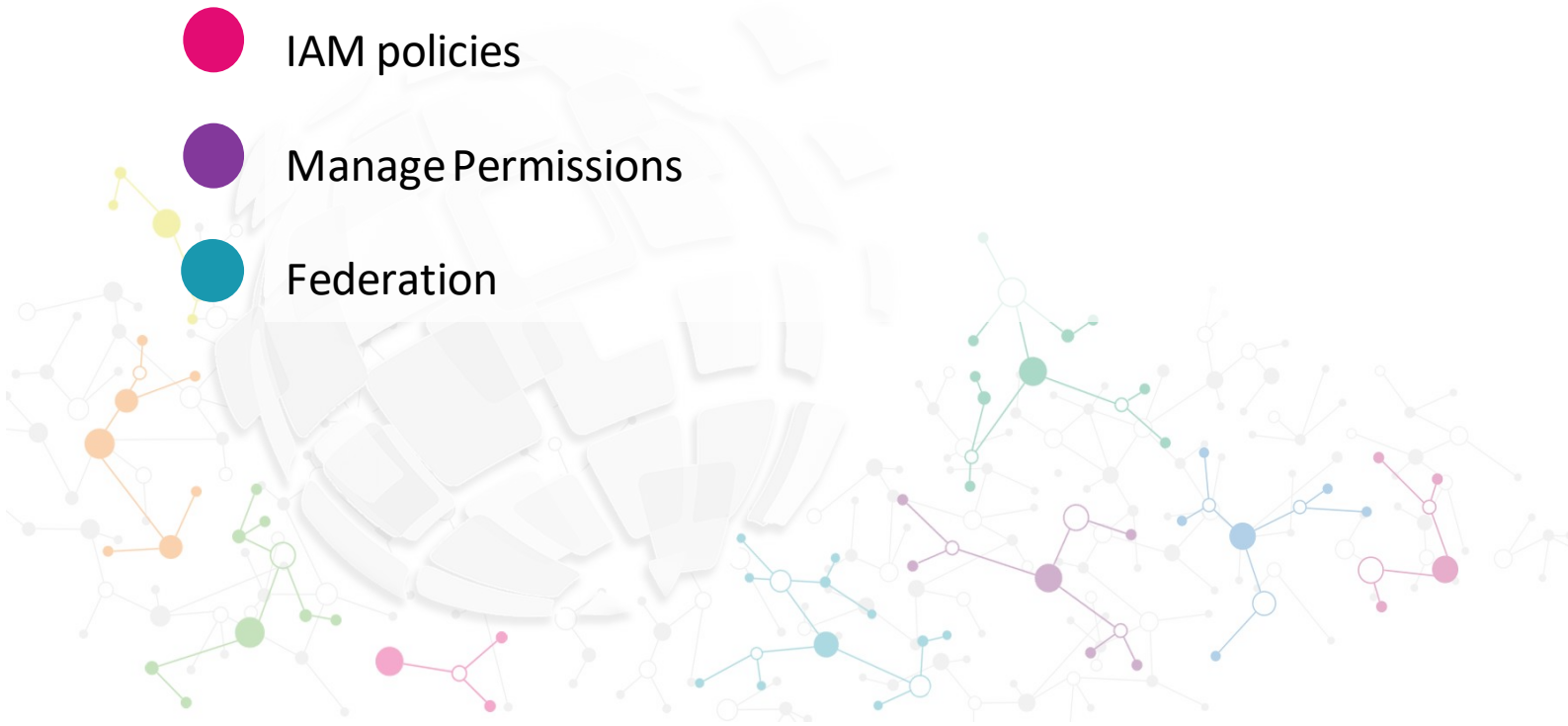
## AWS Security

## AWS IAM



# Overview

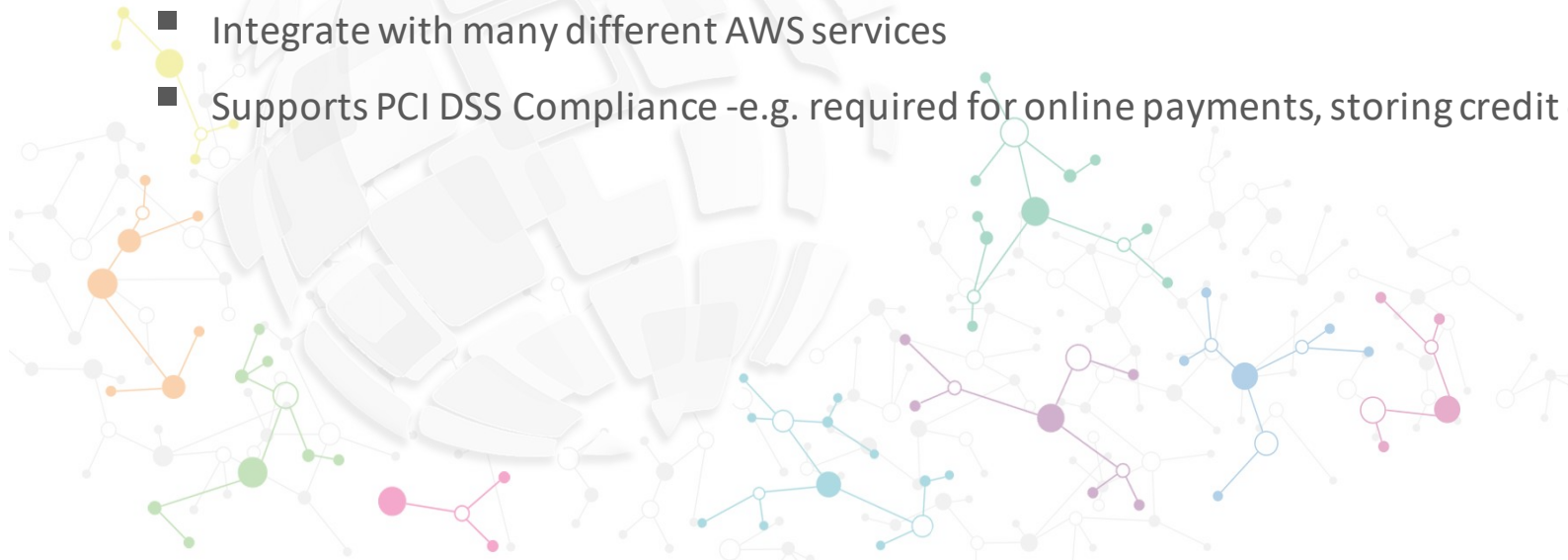
- Introduction to Amazon Identity Access Management
- IAM Users
- IAM Groups
- IAM Roles
- IAM policies
- Manage Permissions
- Federation

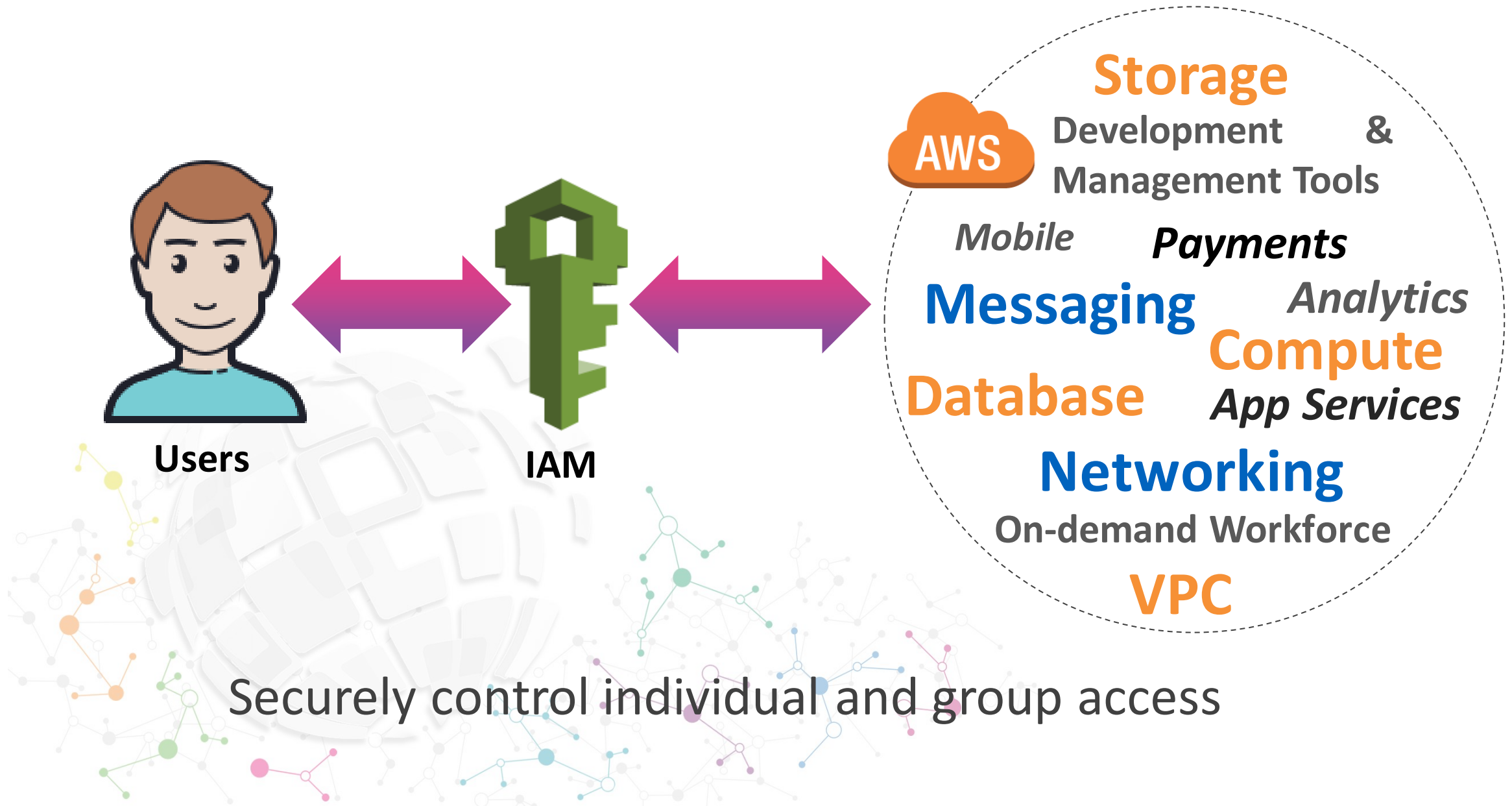




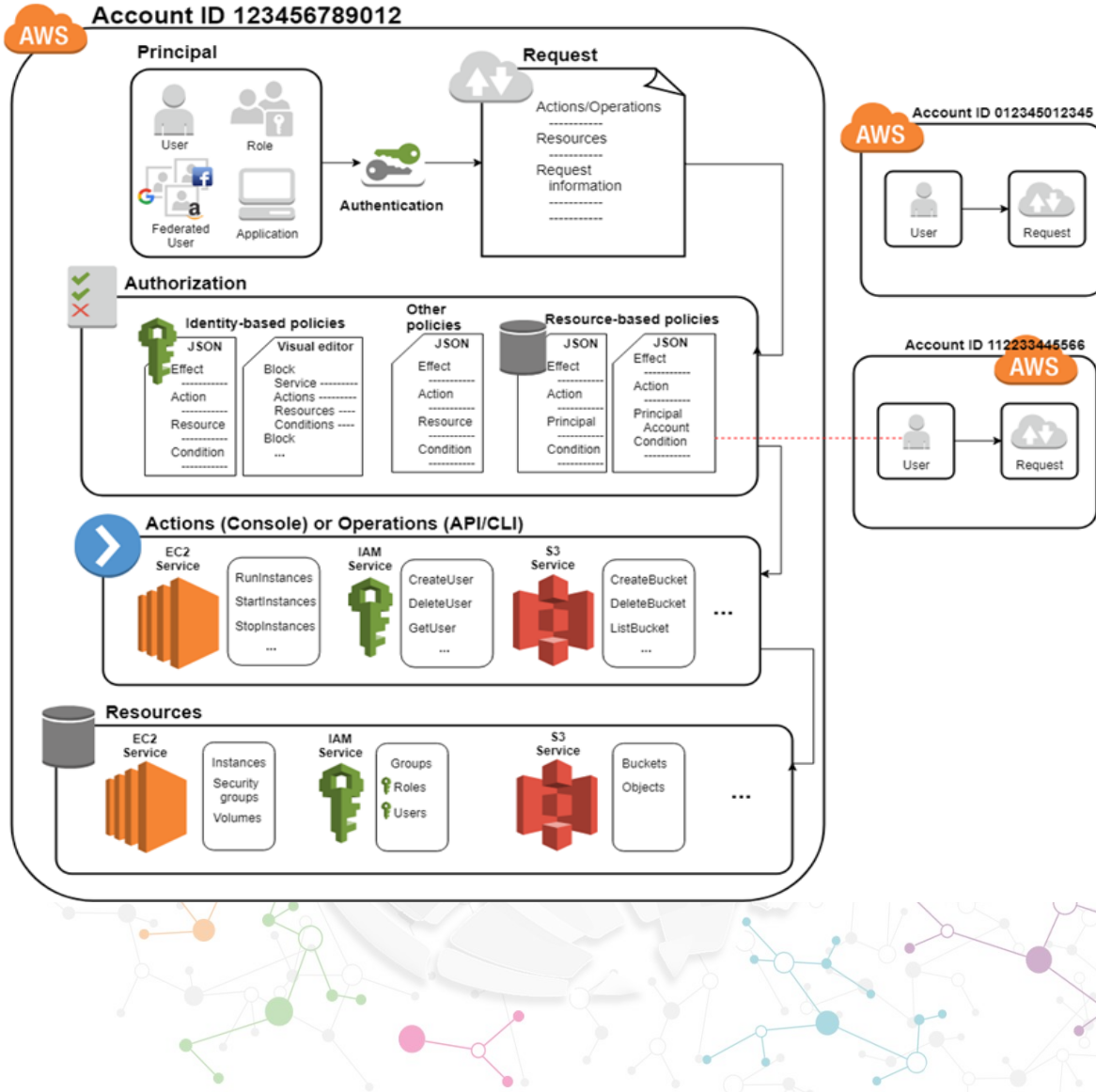
# Amazon Identity and Access Management (IAM)

- AWS Identity and Access Management (IAM) is a web service that helps you securely control access to AWS resources.
- IAM can control who can use your AWS resources (authentication) and what resources they can use and in what ways (authorization).
- Granular permission
- Identity Federation
- Multi-factor Authentication
- Can provide temporary security credentials to provide access to Users /Applications / AWS services
- Setup your own Password Policy
- Integrate with many different AWS services
- Supports PCI DSS Compliance -e.g. required for online payments, storing credit card details





# Understanding How IAM Works



The IAM infrastructure includes the following elements:

## Terms:

- Resources, Identities, Entities and Principals

## Principal

- A **principal** is a person or application that can make a request for an action or operation on an AWS resource.

## Request

- When a principal tries to use the AWS Management Console, the AWS API, or the AWS CLI, that principal sends a **request** to AWS.

## Authentication

- A principal must be **authenticated** (signed in to AWS) using their credentials to send a request to AWS.

## Authorization

Principal must also be **authorized** (allowed) to complete your request. During authorization, AWS uses values from the request context to check for policies that apply to the request.

## Actions or Operations

- After your request has been authenticated and authorized, AWS approves the **actions or operations** in your request

## Resources

- After AWS approves the operations in your request, they can be performed on the related resources within your account. A **resource** is an object that exists within a service.

# Root Account and IAM Users

- Root account = Email address with which you created AWS account. This account has Root level (Complete) access
- IAM Users = Users created by you in your AWS account.

## Root Account

- Root account is not bound by IAM Policies and Permissions
- Best Practice is to not use Root account for performing day to day tasks or provide temporary access credentials to applications etc. For that create IAM users with appropriate permissions assigned
- It is very important to secure Root account
- Best Practice is to enable Multi-Factor Authentication for your Root account
- Multi-Factor Authentication [in addition to your userid password]
  - Hardware MFA device
  - Virtual MFA device (e.g. Google Authenticator)

# IAM Users Sign-in Link

- IAM users sign-in link: (e.g. <https://401912349999.signin.aws.amazon.com/console>)
- This is the URL that will be used by IAM users to login to your account
- You can customize this link to some meaningful name –Account Alias.
- This will create a new DNS namespace. It has to be unique across AWS  
(e.g. <https://mycompany.signin.aws.amazon.com/console>)

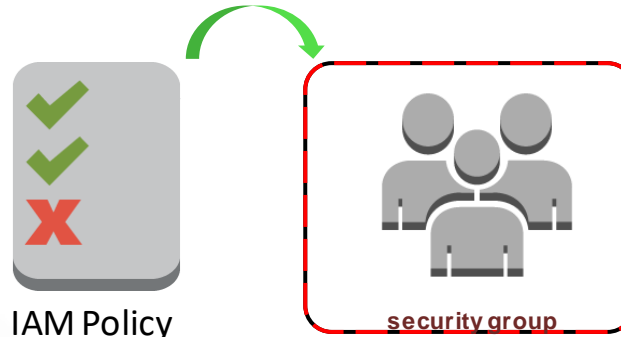


## STEP 1



Create a security group

## STEP 2



IAM Policy

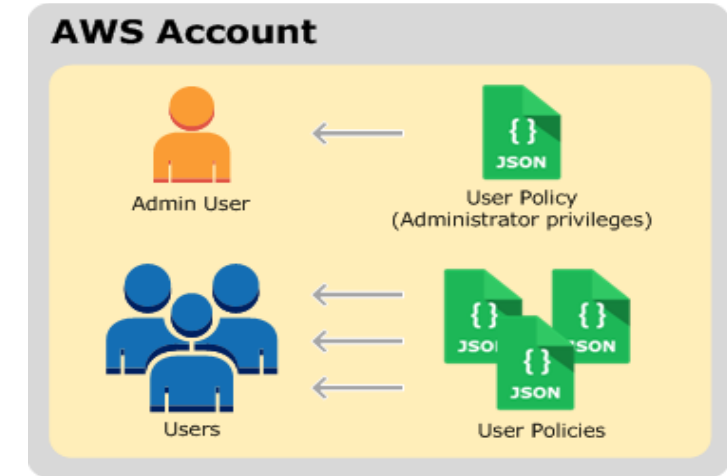
Attach IAM Policy to the group

## STEP 3



Add individual user to the group

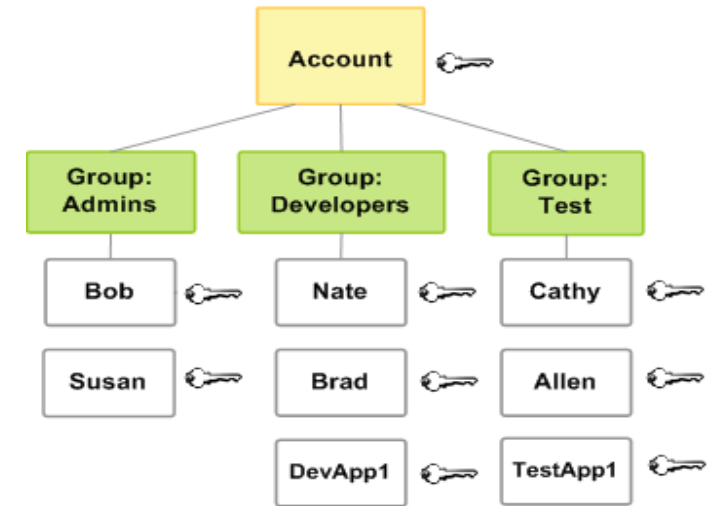
- IAM User is “user account” in your AWS account.
- Can have username/password to login to the AWS console
- IAM users can have access key and secret key to make API calls to interact with AWS services.
- Newly created IAM users have no permission to do anything, implicit deny all. Permission must be explicitly granted
- Unique identity recognized by AWS services and applications
- Security credentials needed
  - Password or Access key
- Can grant access to AWS resources for users managed outside of AWS
  - Federated users
- User will be able to make service requests based on privileges granted





# IAM Groups

- A group is a collection of IAM users
- Users can be added to or removed from a group.
- You can assign permissions to the IAM Group, all IAM users in the group inherit those permissions
- A user can belong to multiple groups.
- Groups cannot belong to other groups.
- Groups can be granted permissions using access control policies. This makes it easier to manage permissions for a collection of users, rather than having to manage permissions for each individual user.
- Groups do not have security credentials, and cannot access web services directly; they exist solely to make it easier to manage user permissions.
- AWS does not provide any default group to hold all users in it and if one is required it should be created with all users assigned to it



# IAM Roles

- A role is an AWS Identity and Access Management (IAM) entity that defines a set of permissions for making AWS service requests
- IAM roles are not associated with a specific user or group. Trusted entities (such as IAM users, applications, or AWS services such as EC2) assume roles
- IAM roles allow you to delegate access with defined permissions to trusted entities without having to share long-term access keys. You can use IAM roles to delegate access to
  - IAM users managed within your account
  - IAM users under a different AWS account
  - AWS service such as EC2
  - External authenticated user [when using Federation]
- You can only act as only one IAM role when making requests to AWS services.
- When launching an EC2 instance, you can assign (pre-configured) Role. You can also assign role to a running instance [This is a new feature available –previously it was possible only at instance launch time]
- Storing access key/secret key on instances or on S3 or inside AMIs is not a good practice

# IAM Roles Types

## ***AWS service Roles***

- AWS services need to interact with other AWS services for e.g. EC2 interacting with S3, SQS etc. AWS automatically provides temporary security credentials for these services . Amazon EC2 instance to use on behalf of its applications

## ***Cross account access roles***

- You can grant your IAM users permission to switch to roles within your AWS account or to roles defined in other AWS accounts that you own
  - Provide access between AWS accounts you own
  - Provide access between your AWS account and a 3rd party AWS account

## ***Identity Provider and Federation access Roles***

- Identity Provider can be used to grant external user identities permissions to your AWS resources without having to be created within your AWS account (Ex. Using "Gmail and Facebook id's")

## ***Trust Policy***

- Trust policy involves setting up a trust between the account that owns the resource (trusting account) and the account who owns the user that needs access to the resources (trusted account).

## ***Permissions policy***

- Permissions policy determines authorization, which grants the user of the role the needed permissions to carry out the desired tasks on the resource



# IAM Permission -Policies

- Policies are used to assign permissions
  - Can assign permissions using the AWS Management Console, the IAM API, or the AWS CLI.
- Policy is a document (in JSON format with key value pair, and can have nesting) that defines one or more permissions
- Policy is assigned to user/group/role
- Policy simulator provided by AWS to create Policy document via simple GUI interface

## ***Policy document***

- Version
- Statement
  - Sid [For Customer created policies and not copied from AWS Managed Policies]
  - Effect [Allow / Deny]
  - Action [which action(s) can be performed]
  - Resource [which target AWS resources does this policy apply to]



# IAM Policies | Identity-based policies

**Identity-based policies** control what actions the identity can perform, on which resources, and under what conditions. Identity-based policies can be further categorized:

**Inline Policies** -Policies that you create and manage, and that are embedded directly into a single user, group, or role. As per best practice, using inline policies is not recommended.

**Managed Policies** –centrally defined policy documents which can be attached to users, groups and roles. You can update the policy in one place and the permissions automatically extend to all attached entities.

## ■ **AWS Managed Policy**

- Commonly used permissions defined by AWS
- Comes with pre-configured JSON scripts , can not edit

## ■ **Customer Managed Policy**

- Can create our own policy based on JSON scripts

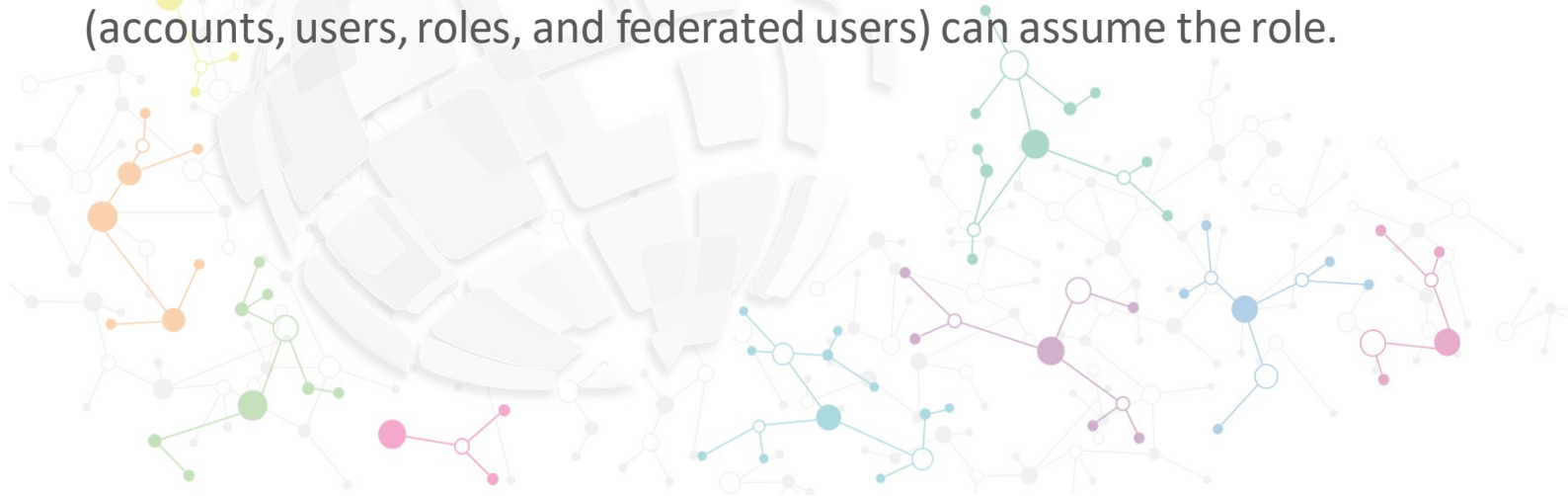




# IAM Policies | Resource-based policies

**Resource-based policies** control what actions a specified principal can perform on that resource and under what conditions. Resource-based policies are inline policies, and there are no managed resource-based policies. To enable cross-account access, you can specify an entire account or IAM entities in another account as the principal in a resource-based policy.

The IAM service supports only one type of resource-based policy called a role trust policy, which is attached to an IAM role. Because an IAM role is both an identity and a resource that supports resource-based policies, you must attach both a trust policy and an identity-based policy to an IAM role. Trust policies define which principal entities (accounts, users, roles, and federated users) can assume the role.



# IAM Policies

## AWS Managed Policy document

-e.g.

AdministratorAccessPolicy

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "*",
      "Resource": "*"
    }
  ]
}
```

## Customer Managed Policy document

-e.g.

MyCustomPolicy

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1488501897000",
      "Effect": "Allow",
      "Action": [ "ec2:*" ],
      "Resource": [ "*" ]
    }
  ]
}
```



# IAM Policy| Lets analyze the policy

```
"Sid": "Stmt1488501897000",
```

Who/what is authorized

```
"Effect": "Allow",
```

```
"Action": [
```

Which task(s) are allowed

```
  "S3: DeleteObject",
```

```
  "S3: GetObject"
```

```
],
```

```
"Condition": {
```

Which condition(s) need to be met for authorization

```
  "IpAddress": {
```

```
    "aws:sourceIp": "10.14.8.0/24"
```

```
  }
```

```
},
```

```
"Resource": [
```

Resources to which authorized tasks are performed

```
  "arn:aws:s3:::billing-marketing",
```

```
  "arn:aws:s3:::billing-sales"
```

```
]
```

When a request is made, the AWS service decides whether a given request should be allowed or denied. The evaluation logic follows these rules:

- By default, all requests are denied. (In general, requests made using the account credentials for resources in the account are always allowed.)
- An explicit allow overrides this default.
- An explicit deny overrides any allows..

The order in which the policies are evaluated has no effect on the outcome of the evaluation. All policies are evaluated, and the result is always that the request is either allowed or denied.



## AWS Managed Policies:

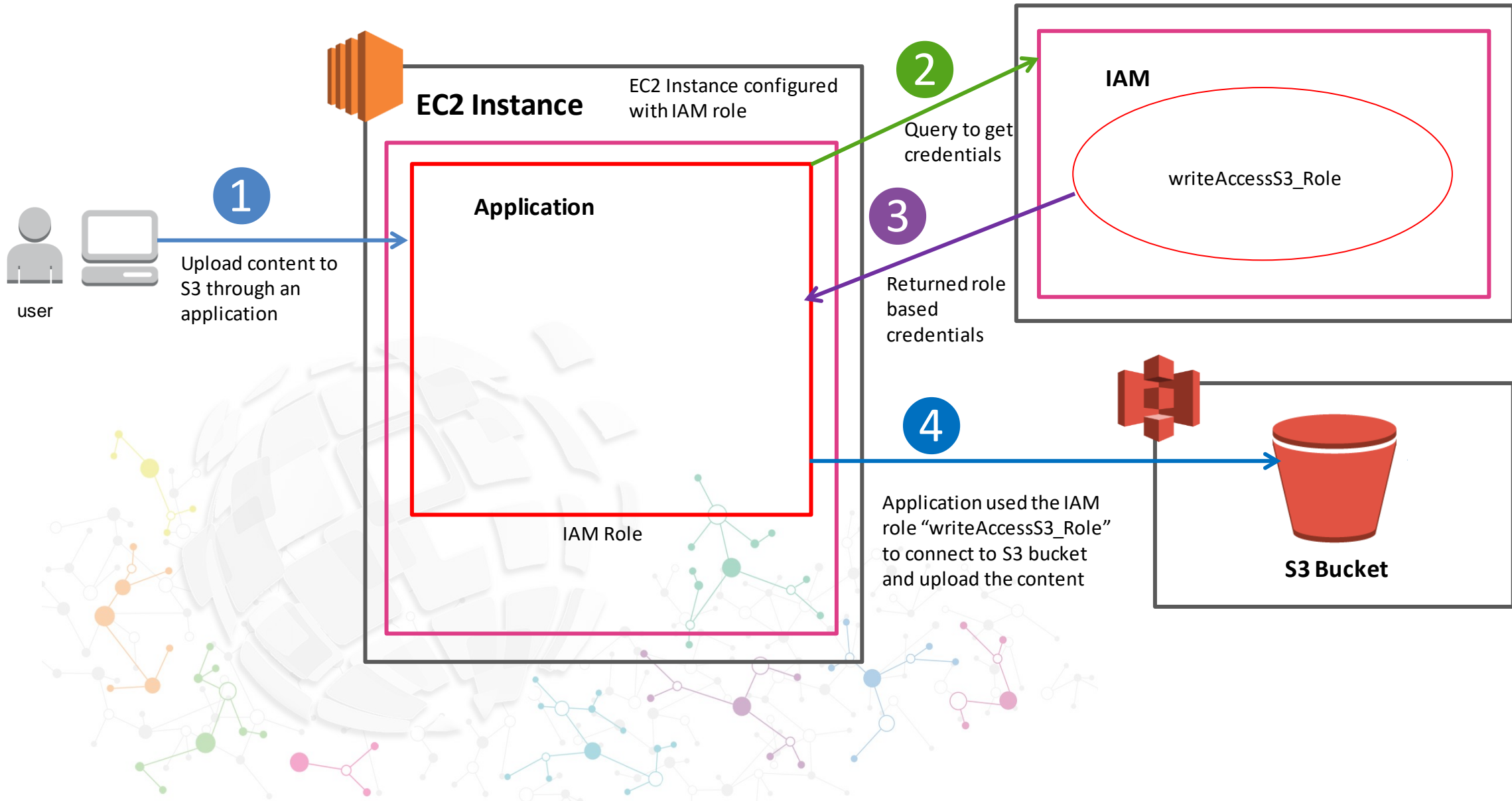
- AdministratorAccess(Full access to AWS services and resources)
- PowerUserAccess(Full access except for IAM)
- SystemAdministrator(Full access permissions necessary for resources required for application and development operations )
- SupportUser(permissions to troubleshoot and resolve issues in an AWS account, enables the user to contact AWS support )
- ReadOnlyAccess
- AmazonEC2FullAccess (Full access to EC2)
- AmazonS3ReadOnlyAccess
- AmazonS3FullAccess

## Policy versions:

- Allows to maintain up to 5 versions of the Policy documents
- You can set any version as the default version
- You can revert back to any version of the Policy Document



# Use Case | Accessing S3 Bucket using IAM Role

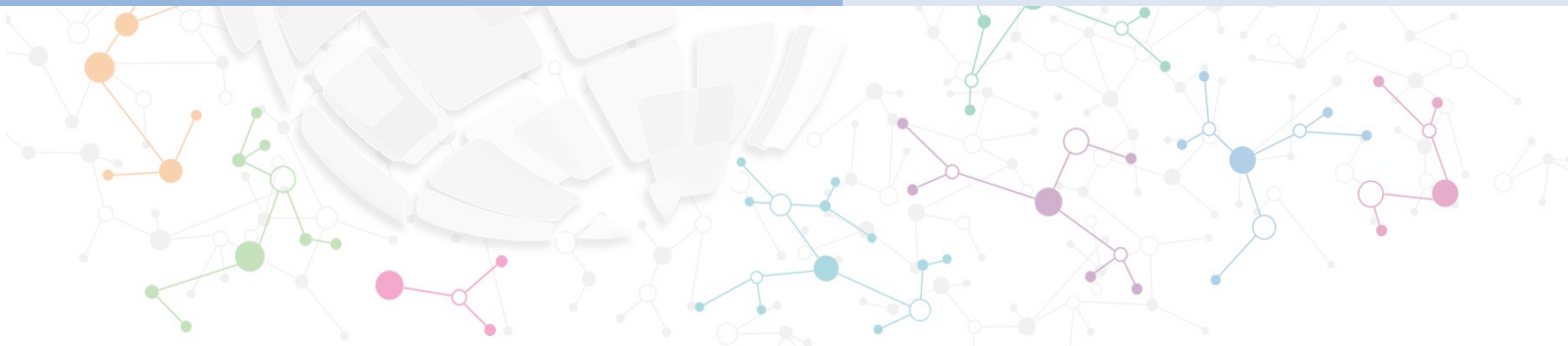


# AWS SAA Boot Camp Session2



## AWS Security

## Configuration management Tools



- Configuration Management tools are essential components in DevOps



Configuration as a Code

Consistency in deployment

Similar to other AWS Services such as CloudFormation, Elastic Beanstalk, SSM

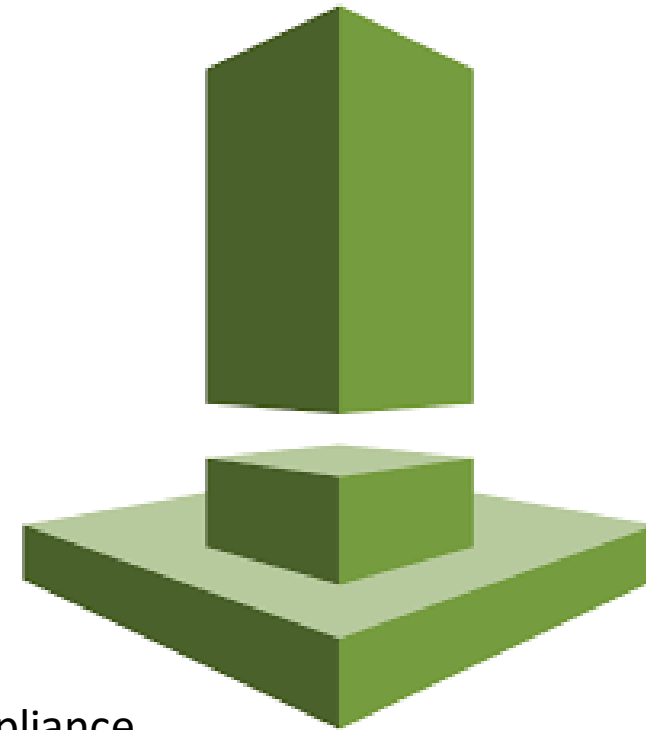
Effective option while deploying HA solution in multiple regions

AWS OpsWork is the AWS managed solution of Chef and Puppet



# Amazon Inspector

- Vulnerability Assessment Service
  - Built from the ground up to support DevSecOps
  - Automatable via APIs
  - Integrates with CI/CD tools
  - On-Demand Pricing model
  - Static & Dynamic Rules Packages
  - Generates Findings



Automated security assessment service that finds security or compliance issues when deploying application on AWS



- Detect and remediate security issues early and often with AWS inspector

What's being assessed?

- Network, VMs, OS and application configuration

Built-in content library

- Checks common security standards and vulnerabilities

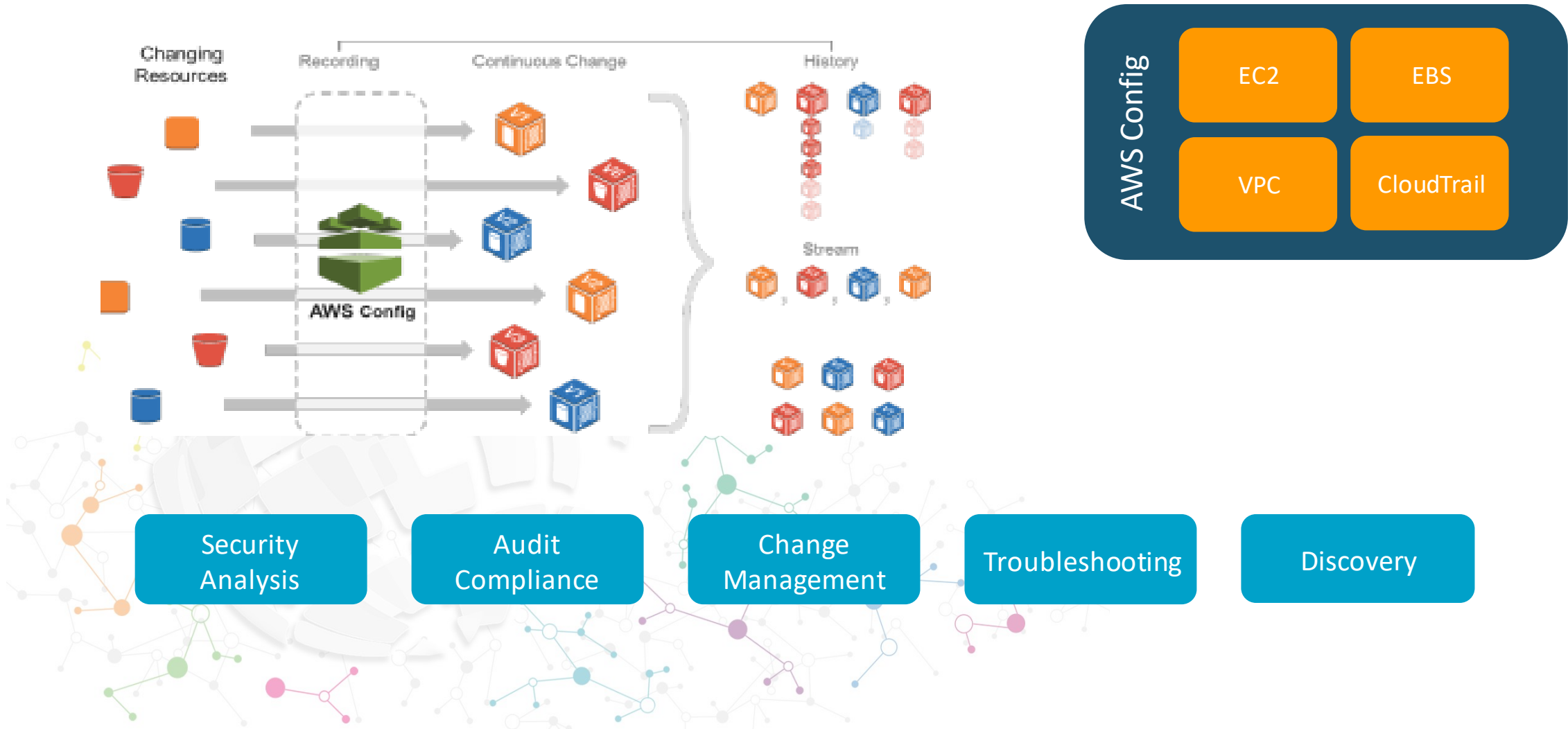
Detailed Reports

- Detailed dashboard

Full Audit Trails

- Track what tests were performed, when and their results

- Managed service for tracking AWS inventory and configuration, and configuration change notification.

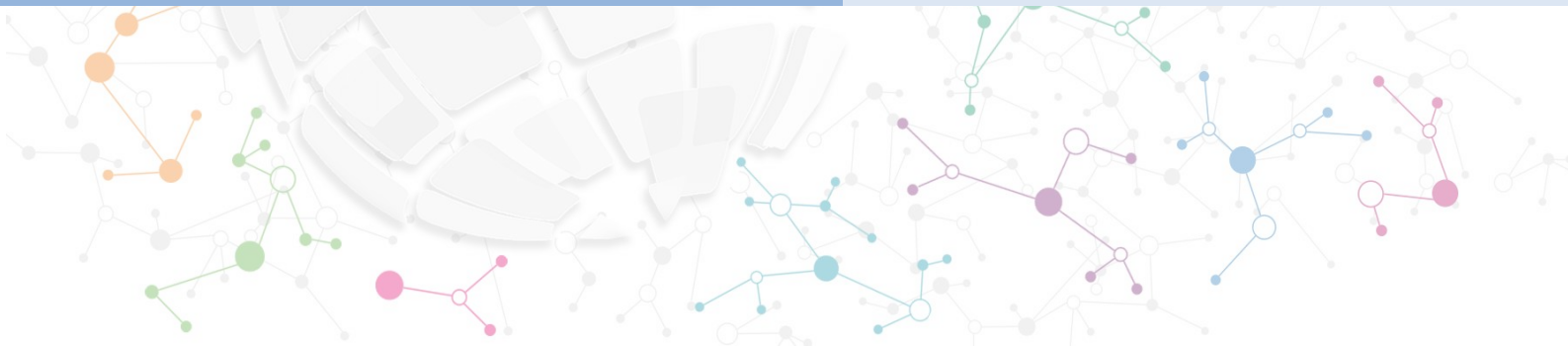


# AWS SAA Boot Camp Session2



## AWS Security

## Best Practices




# AWS Trusted Advisor

Leverage Trusted Advisor to analyze your AWS resources for best practices for availability, cost, performance and security.

## Trusted Advisor Dashboard

Download   

 Welcome to the AWS Trusted Advisor console!  
For more information, see [Meet AWS Trusted Advisor](#).

### Cost Optimization



2  5  0 

0 excluded items

**\$331.20**

Potential monthly savings

### Performance



6  2  0 

0 excluded items

### Security



4  1  4 

1 excluded items

### Fault Tolerance



8  3  2 

0 excluded items

## Security

Download   












4  1  4 

1 excluded items

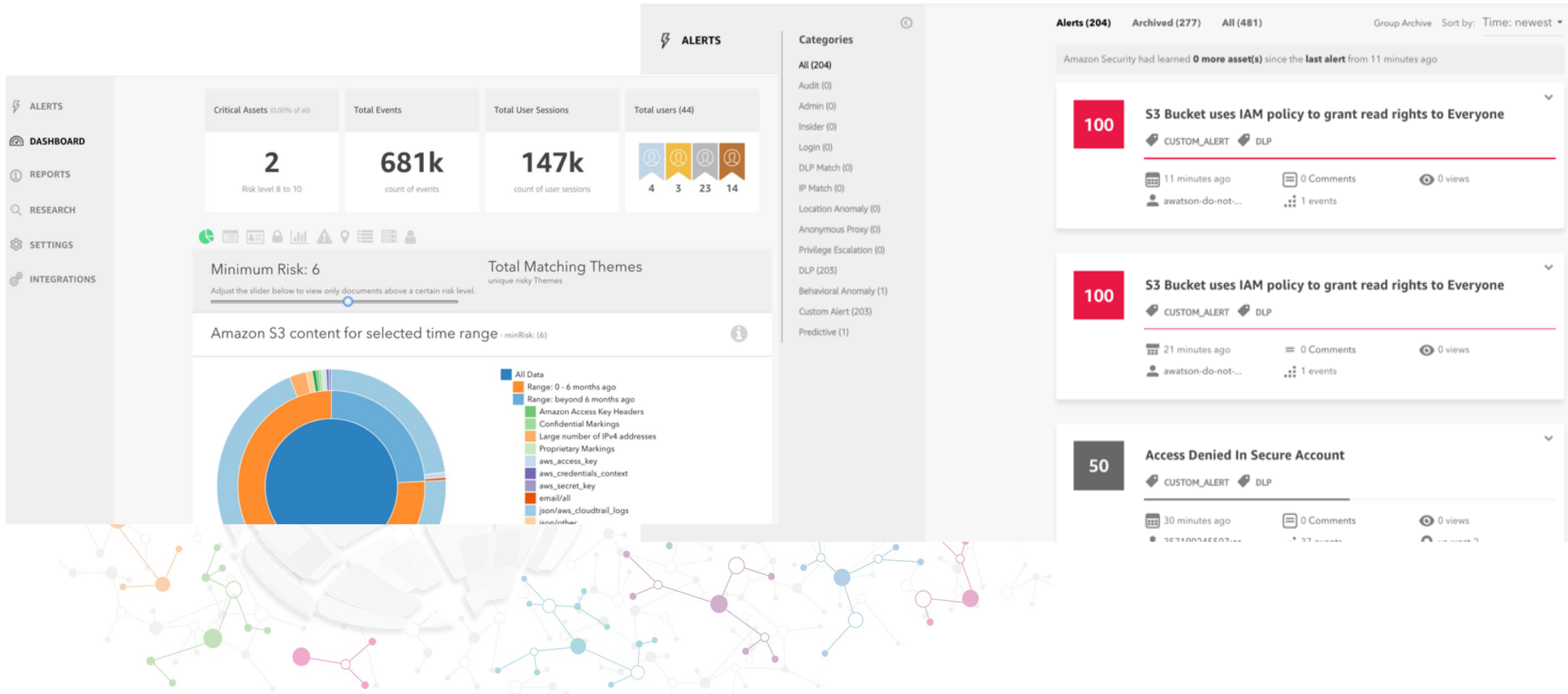
View

## Security Checks

-  **Security Groups - Specific Ports Unrestricted** Updated: Dec 22, 2014 6:32 AM  
- Checks security groups for rules that allow unrestricted access (0.0.0.0/0) to specific ports.  
44 of 124 security group rules allow unrestricted access to a specific port.
-  **Security Groups - Unrestricted Access** Updated: Dec 22, 2014 6:24 AM  
- Checks security groups for rules that allow unrestricted access to a resource.  
47 of 124 security group rules have a source IP address with a /0 suffix. 1 items have been excluded.
-  **Amazon S3 Bucket Permissions** Updated: Dec 22, 2014 6:24 AM  
- Checks buckets in Amazon Simple Storage Service (Amazon S3) that have open access permissions.

# Amazon Macie

- Leverage Amazon Macie to help prevent data loss in AWS.



Thank You!

