amazon web services

Masterclass

Advanced Security Best Practices on AWS


Ian Massingham — Technical Evangelist


ianmas@amazon.com


@IanMmmm

intel®

Masterclass

1. A technical deep dive that goes beyond the basics
2. Intended to educate you on how to get the best from AWS services
3. Show you how things work and how to get things done

Advanced Security Best Practices

Security is job zero at AWS

Built to satisfy the most security-sensitive organisations

Provides visibility, auditability, controllability & agility

Lower operational overhead that traditional IT

Increasing your Security Posture in the Cloud

AWS security approach

Size of AWS security team

Visibility into usage & resources

Broad Accreditations & Certifications

Security Benefits from Community Network Effect

Partner ecosystem

Customer ecosystem

Everyone benefits

Agenda

Sharing the Security Responsibility

Identity and Access Management with IAM

Defining virtual networks with Amazon VPC

Networking & Security for Amazon EC2 Instances

Working with Container and Abstracted Services

Encryption and Key Management in AWS

SHARING THE SECURITY RESPONSIBILITY

Shared Security Model

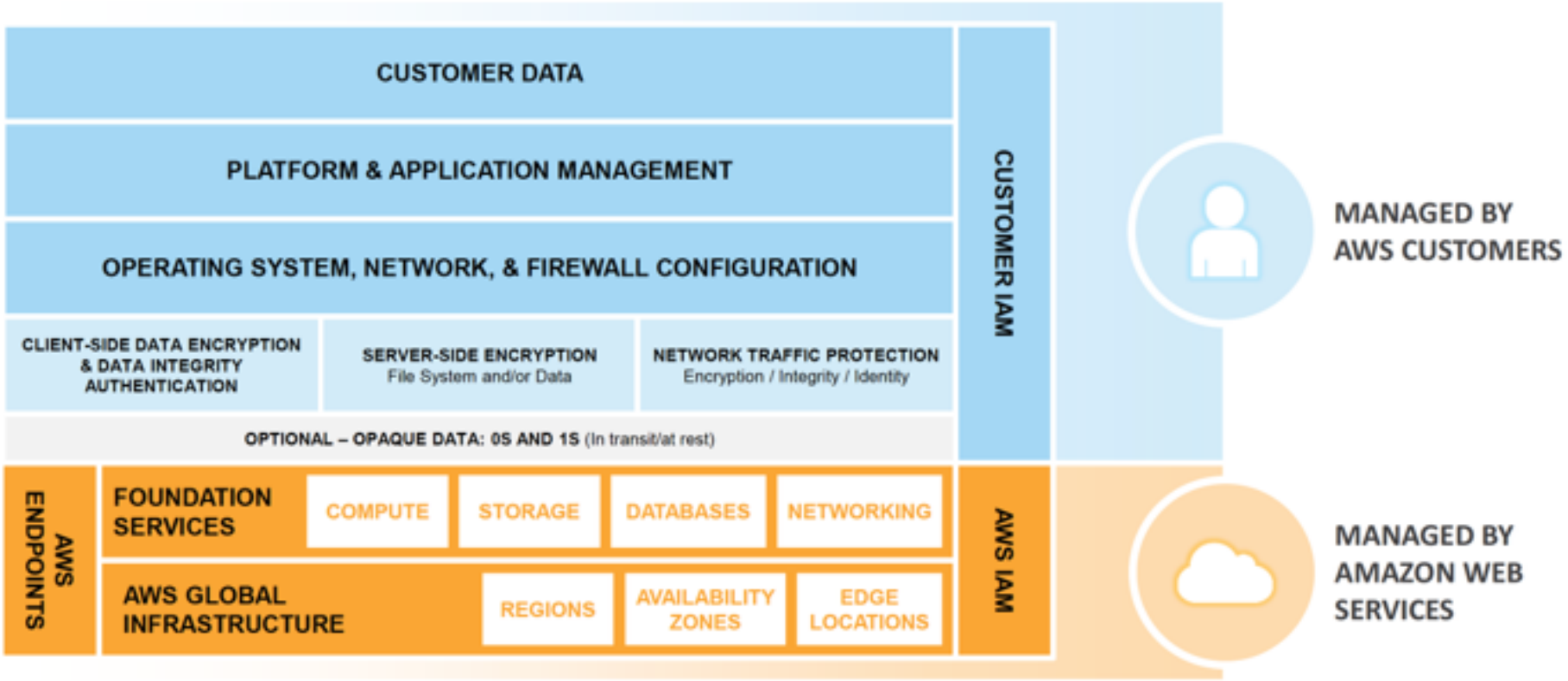
- Shared Responsibility
  - Let AWS do the heavy lifting
  - Focus on what's most valuable to your business

- AWS
  - Facility operations
  - Physical Security
  - Physical Infrastructure
  - Network Infrastructure
  - Virtualisation Infrastructure
  - Hardware lifecycle management

- Customer
  - Choice of Guest OS
  - Application Configuration Options
  - Account Management flexibility
  - Security Groups
  - ACLs
  - Identity Management

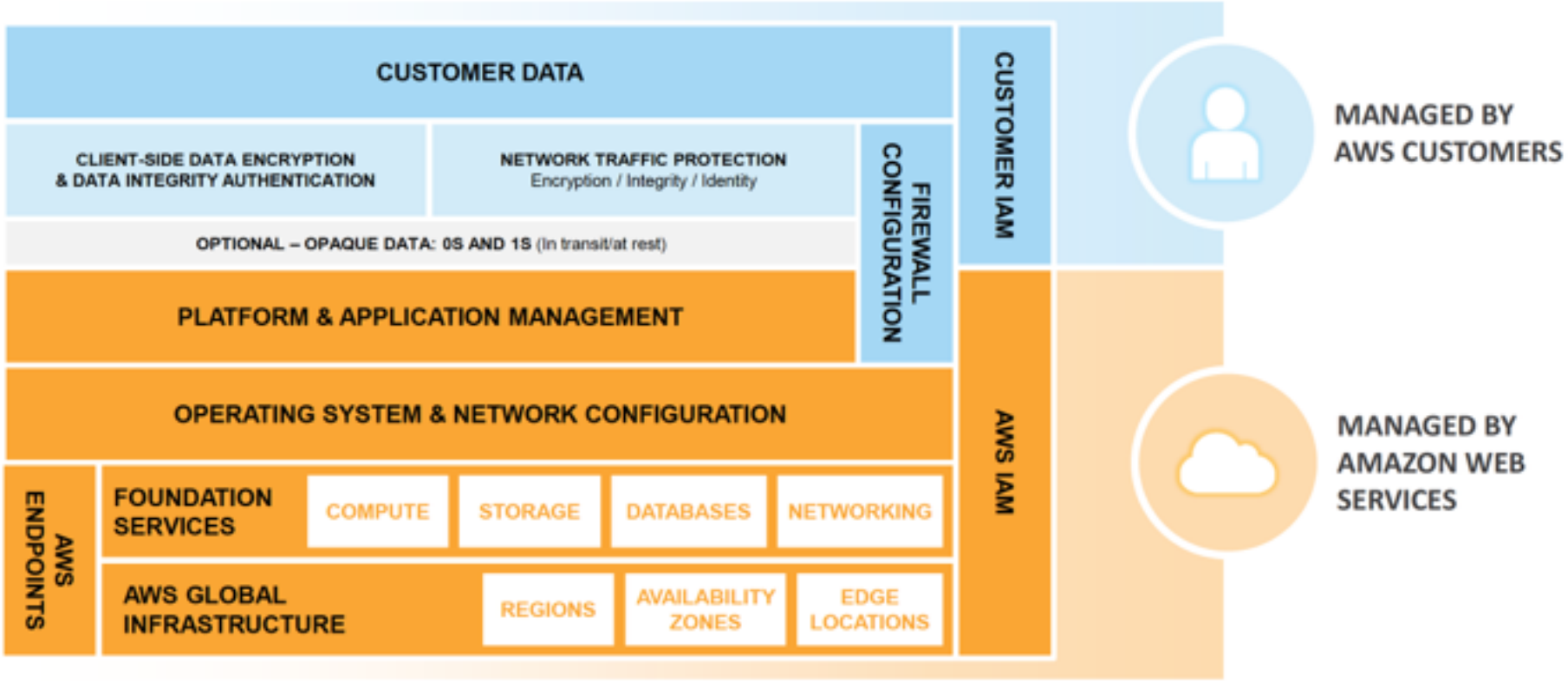
Shared Security Model: Infrastructure Services

Such as Amazon EC2, Amazon EBS, and Amazon VPC



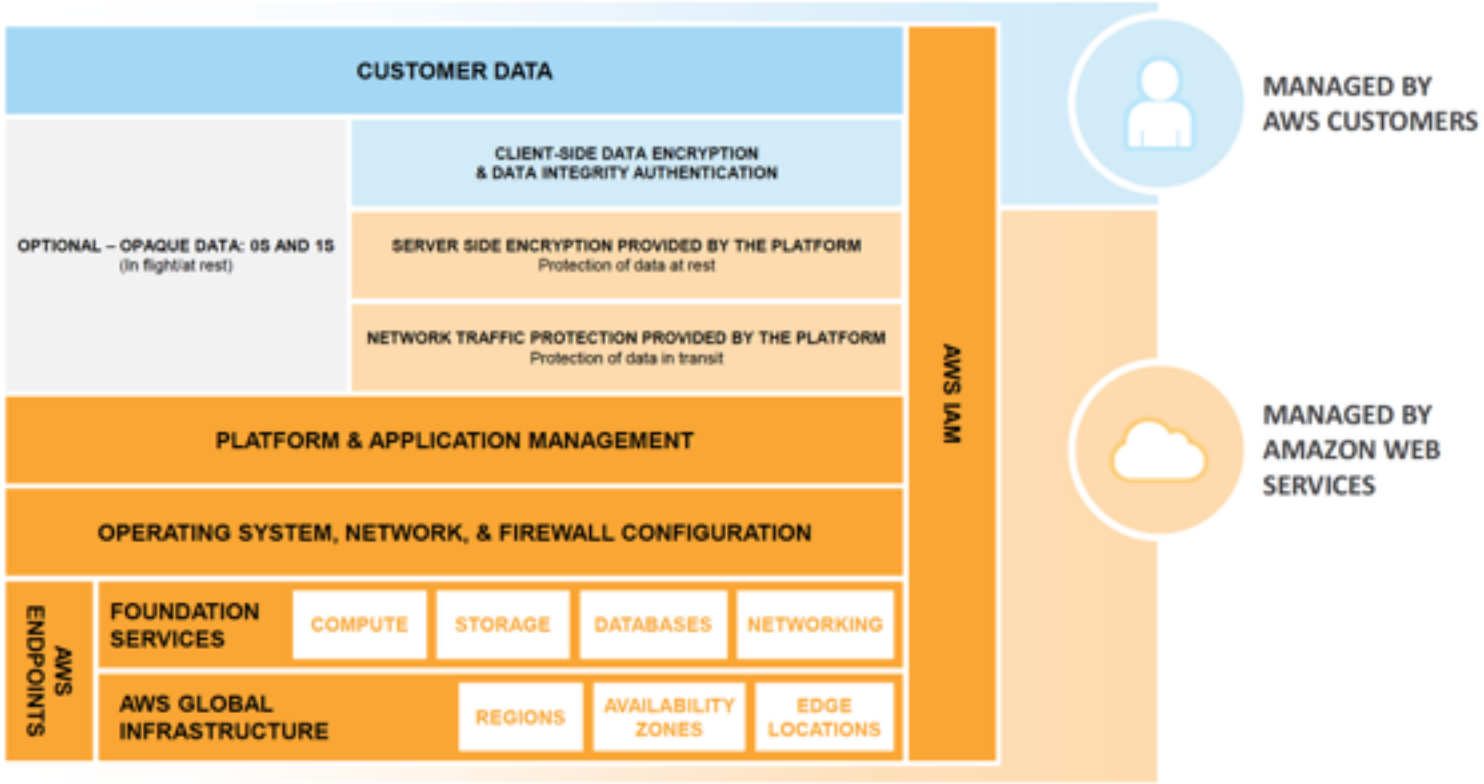
Shared Security Model: Container Services

Such as Amazon RDS and Amazon EMR



Shared Security Model: Abstracted Services

Such as Amazon S3 and Amazon DynamoDB



IDENTITY AND ACCESS MANAGEMENT WITH IAM

Users

Create individual users

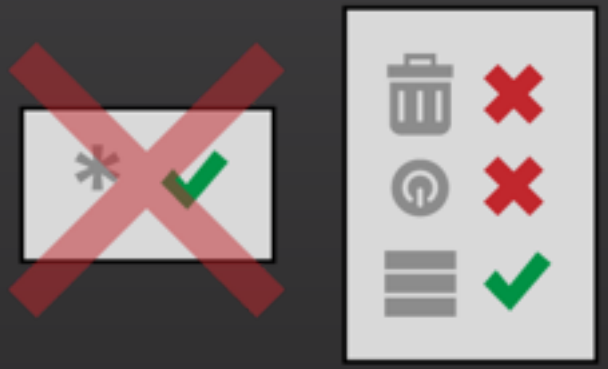
Create individual users

Benefits

- Unique credentials
- Individual credential rotation
- Individual permissions

How to get started

- Identify which IAM users you want to create
- Use the console, CLI or API to:
  - Create user
  - Assign credentials
  - Assign permissions



Permissions

Grant least privilege

Grant least privilege

Benefits

- Less chance of people making mistakes
- Easier to relax than tighten up
- More granular control
  - API and resource

How to get started

- Identify what permissions are required
- Password or access keys?
- Avoid assigning *:* policy
- Default Deny
- Use policy templates

IMPORTANT NOTE: Permissions do not apply to root!

Groups

Manage permissions with groups

Manage permissions with groups

Benefits

- Easier to assign the same permissions to multiple users
- Simpler to re-assign permissions based on change in responsibilities
- Only one change to update permissions for multiple users

How to get started

- Map permissions to a specific business function
- Assign users to that function
- Manage groups in the Group section of the IAM console

Conditions

Restrict privileged access further with conditions

Restrict privileged access further with conditions

Benefits

- Additional granularity when defining permissions
- Can be enabled for <u>any</u> AWS service API
- Minimizes chances of accidentally performing privileged actions

How to get started

- Use conditions where applicable
- Two types of conditions
  - AWS common
  - Service-specific

Restrict privileged access further with conditions

MFA

```
{
   "Statement":[{
   "Effect":"Allow",
   "Action":["ec2:TerminateInstances"],
   "Resource":["*"],
   "Condition":{
     "Null":{"aws:MultiFactorAuthAge":"false"}
   }
  }
 ]
}
```

Enables a user to terminate EC2 instances only if the user has authenticated with their MFA device.

SSL

```
{
   "Statement":[{
   "Effect":"Allow",
   "Action":"iam:*AccessKey*",
   "Resource”:"arn:aws:iam::123456789012:user/*",
   "Condition":{
      "Bool":{"aws:SecureTransport":"true"}
   }
  }
 ]
}
```

Enables a user to manage access keys for all IAM users only if the user is coming over SSL.

SourceIP

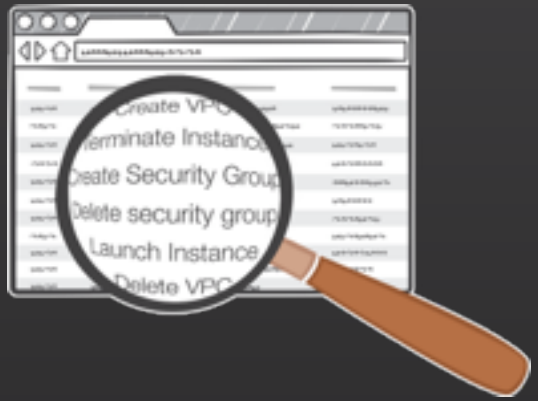
```
{
   "Statement":[{
   "Effect":"Allow",
   "Action":["ec2:TerminateInstances“],
   "Resource":["*“],
   "Condition":{
      "IpAddress":{"aws:SourceIP":"192.168.176.0/24"}
   }
  }
 ]
}
```

Enables a user to terminate EC2 instances only if the user is accessing Amazon EC2 from 192.168.176.0/24.

Tags

```
{
   "Statement":[{
   "Effect": "Allow",
   "Action":"ec2:TerminateInstances",
   "Resource": "*",
   "Condition":{
      "StringEquals":{"ec2:ResourceTag/Environment":"Dev"}
   }
  }
 ]
}
```

Enables a user to terminate EC2 instances only if the instance is tagged with “Environment=Dev”.



Auditing

Enable AWS CloudTrail to get logs of API calls

aws.amazon.com/cloudtrail

Enable AWS CloudTrail to get logs of API calls

Benefits

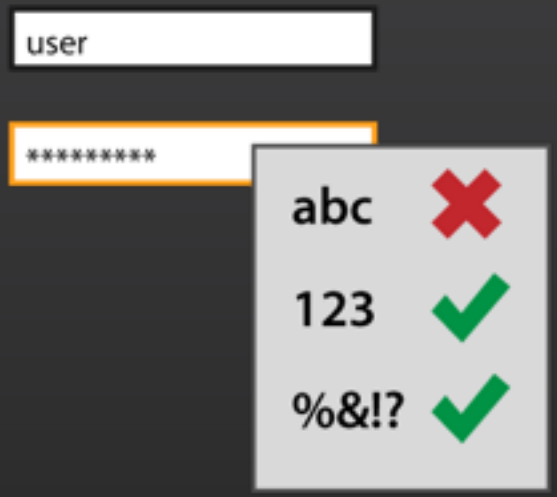
- Visibility into your user activity by recording AWS API calls to an Amazon S3 bucket

How to get started

- Set up an Amazon S3 bucket
- Enable AWS CloudTrail

Ensure the services you want are integrated with AWS CloudTrail

aws.amazon.com/cloudtrail



Passwords

Configure a strong password policy

Configure a strong password policy

Benefits

- Ensures your users and your data are protected

How to get started

- What is your company's password policy?
- You can configure
  - Password expiration
  - Password strength
    - Uppercase, lowercase, numbers, non-alphanumeric
  - Password re-use

IMPORTANT NOTE: Password policy does not apply to root!

Rotation

Rotate (or delete) security credentials regularly

Rotate/Delete security credentials regularly

Benefits

- Normal best practice

How to get started

- Use Credential Reports to identity credentials that should be rotated or deleted
- IAM console displays when password last used
- Grant IAM user permission to rotate credentials
- IAM roles for Amazon EC2 rotate credentials automatically

MFA

Enable multi-factor authentication for privileged users

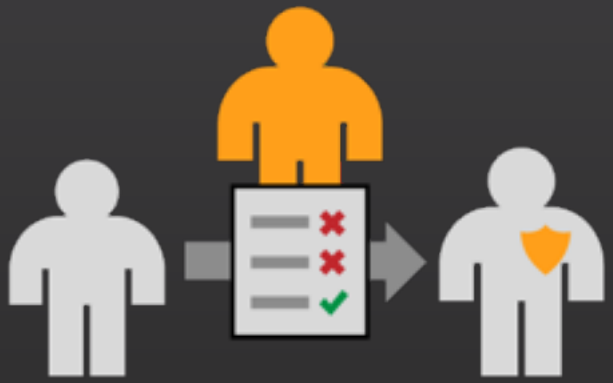
Enable MFA for privileged users

Benefits

- Supplements user name and password to require a one-time code during authentication

How to get started

- Choose type of MFA
  - Virtual MFA
  - Hardware
- Use IAM console to assign MFA device



Sharing

Use IAM roles to share access

http://docs.aws.amazon.com/STS/latest/APIReference/Welcome.html

Use IAM roles to share access

Benefits

- No need to share security credentials
- No need to store long term credentials
- Easy to break sharing relationship
- Use cases
  - Cross-account access
  - Intra-account delegation
  - Federation

How to get started

- Create a role
  - Specify who you trust
  - Describe what the role can do
- Share the name of the role
- Use ExternalID when sharing with a 3rd party

IMPORTANT NOTE: Never share credentials.

Roles

Use IAM roles for Amazon EC2 instances

http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/iam-roles-for-amazon-ec2.html

Use IAM roles for Amazon EC2 instances

Benefits

- Easy to manage access keys on EC2 instances
- Automatic key rotation
- Assign least privilege to the application
- AWS SDKs fully integrated
- AWS CLI fully integrated

How to get started

- Create an IAM role
- Assign permissions to role
- Launch instances w / role
- If not using SDKs, sign all requests to AWS services with the role's temporary credentials

Root

Reduce or remove use of root

Reduce or remove use of root

Benefits

- Reduce potential for misuse of credentials

How to get started

- Security Credentials Page
  - Delete access keys
  - Activate an MFA device
- Ensure you have set a “strong” password

aws.amazon.com/iam

DEFINING VIRTUAL NETWORKS WITH AMAZON VPC

Amazon VPC

A virtual network in your own **logically isolated area** within the AWS cloud populated by infrastructure, platform, and application services that share common **security** and **interconnection**

aws.amazon.com/vpc/

VPC Networking

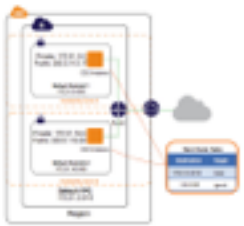
- Elastic Network Interface (ENI)
- Subnet
- Network Access Control List (NACL)
- Route Table
- Internet Gateway
- Virtual Private Gateway
- Route 53 Private Hosted Zone

VPC Network Topology

A VPC can span multiple AZs, but each subnet must reside entirely within one AZ

Use at least 2 subnets in different AZs for each layer of your network

Control of subnets and routing tables

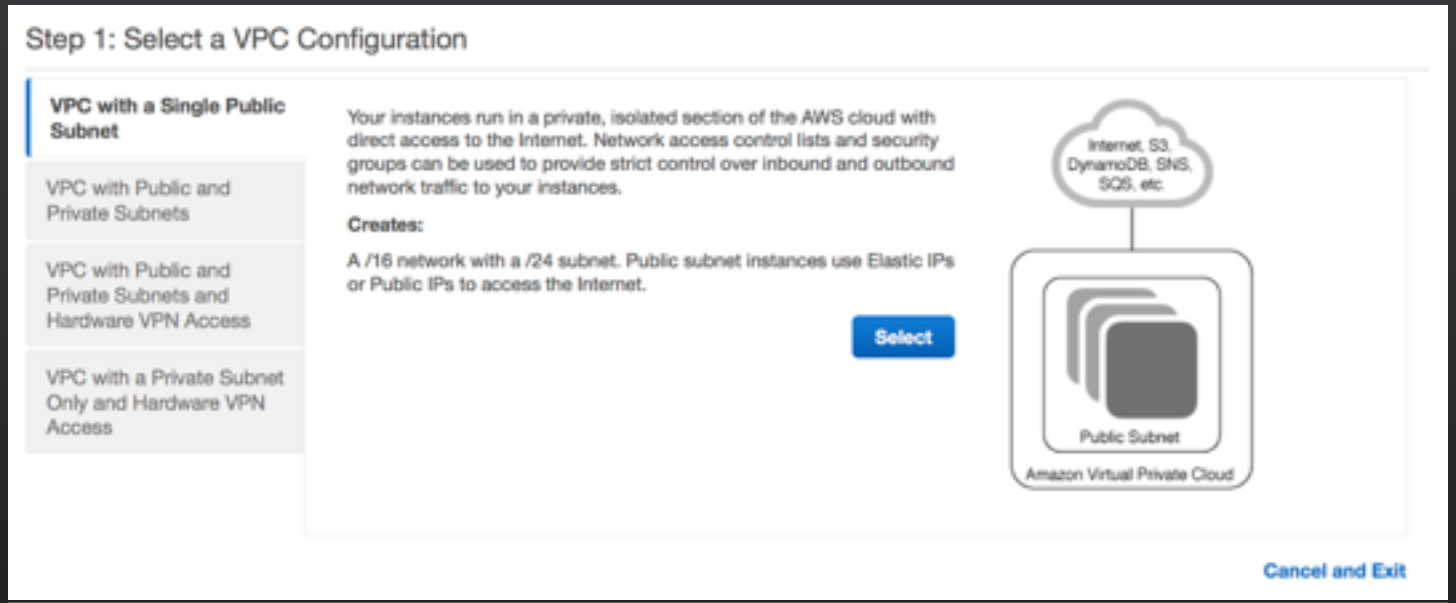


Sample VPC
with
2 Public Subnets

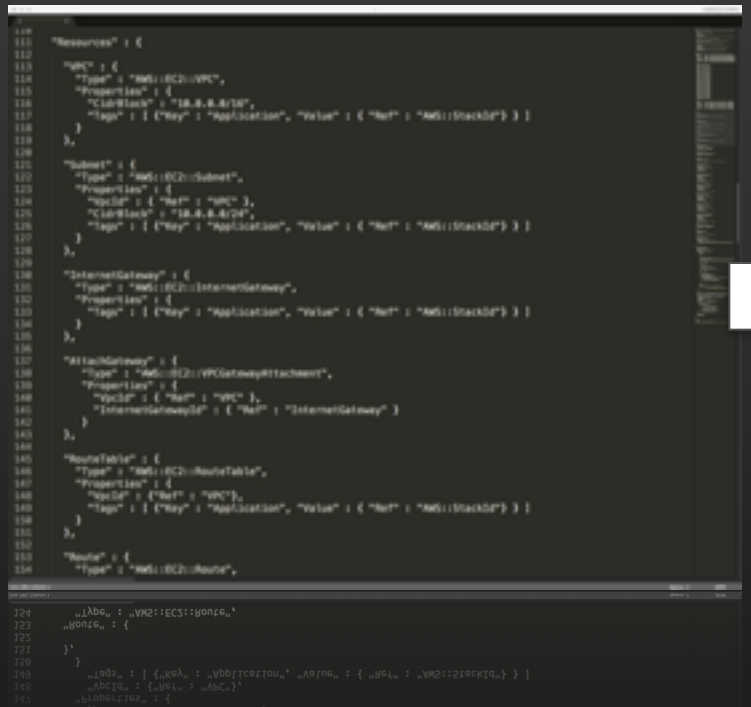
Sample VPC
with
1 Public Subnet,
2 Private Subnets,
1 of which
can route
through the VPN

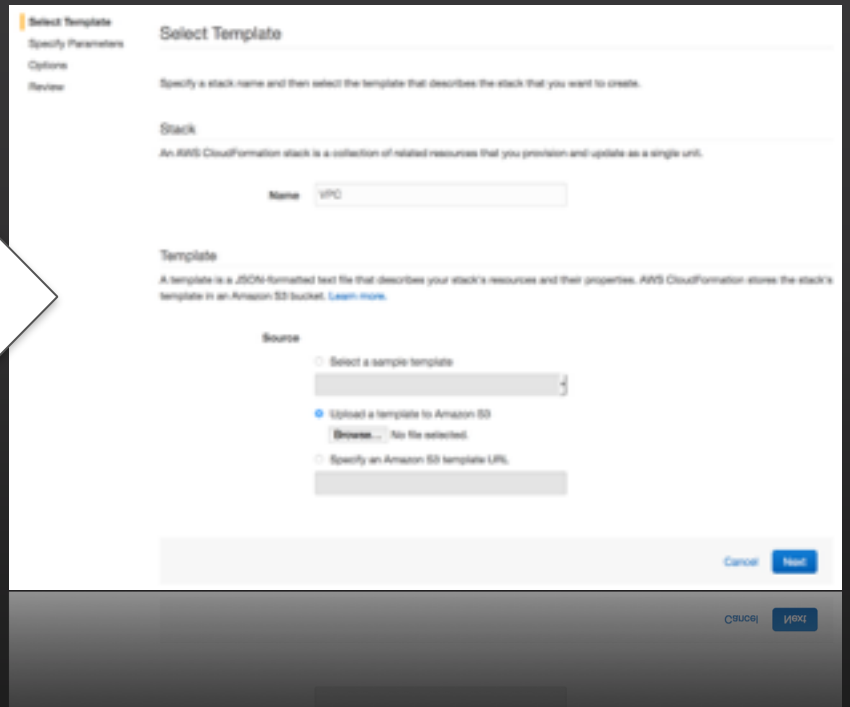
Sample
VPN
CloudHub

VPC Creation with the VPC Wizard



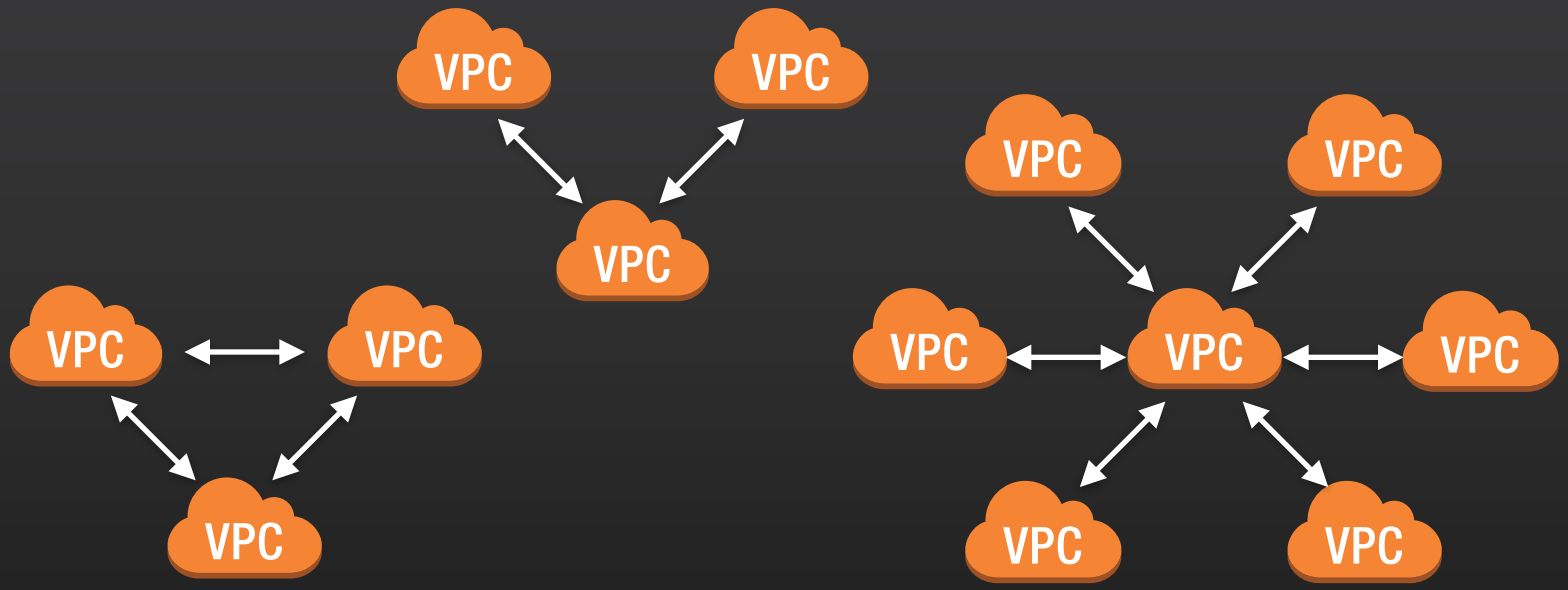
VPC Creation with AWS CloudFormation





VPC Peering

A networking connection between two VPCs



docs.aws.amazon.com/AmazonVPC/latest/UserGuide/vpc-peering.html

Using Network Access Control Lists

An optional layer of security that acts as a firewall for controlling traffic in and out of a subnet

You might set up network ACLs with rules similar to your security groups in order to add an additional layer of security to your VPC

http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_ACLs.html

Default Network ACL

Inbound				
Rule #	**Source IP**	**Protocol**	**Port**	**Allow/Deny**
100	0.0.0.0/0	All	All	ALLOW
*	0.0.0.0/0	All	All	DENY
Outbound				
Rule #	**Dest IP**	**Protocol**	**Port**	**Allow/Deny**
100	0.0.0.0/0	all	all	ALLOW
*	0.0.0.0/0	all	all	DENY

http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_ACLs.html

Whitelisting with NACLs

Inbound					
Rule #	Source IP	Protocol	Port	Allow/Deny	Comments
100	0.0.0.0/0	TCP	80	ALLOW	Allows inbound HTTP traffic from anywhere.
110	0.0.0.0/0	TCP	443	ALLOW	Allows inbound HTTPS traffic from anywhere.
120	192.0.2.0/24	TCP	22	ALLOW	Allows inbound SSH traffic from your home network's public IP address range (over the Internet gateway).
130	192.0.2.0/24	TCP	3389	ALLOW	Allows inbound RDP traffic to the web servers from your home network's public IP address range (over the Internet gateway).
140	0.0.0.0/0	TCP	49152-65535	ALLOW	Allows inbound return traffic from the Internet (that is, for requests that originate in the subnet). For more information about how to select the appropriate ephemeral port range, see Ephemeral Ports.

http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_ACLs.html

Blacklisting with NACLs

acl-96a756f3

Summary | **Inbound Rules** | Outbound Rules | Subnet Associations | Tags

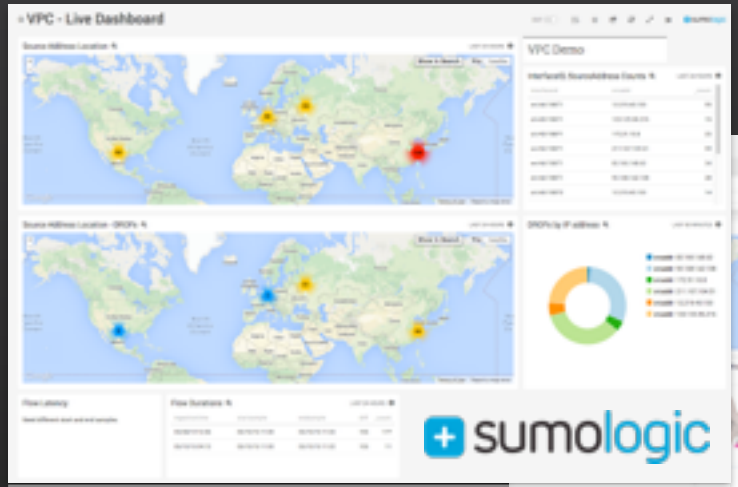
Allows inbound traffic. Because network ACLs are stateless, you must create inbound and outbound rules.

Edit

Rule #	Type	Protocol	Port Range	Source	Allow / Deny
50	SSH (22)	TCP (6)	22	0.0.0.0/0	DENY
100	ALL Traffic	ALL	ALL	0.0.0.0/0	ALLOW
*	ALL Traffic	ALL	ALL	0.0.0.0/0	DENY

http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_ACLs.html

VPC Flow Logs



https://aws.amazon.com/blogs/aws/vpc-flow-logs-log-and-view-network-traffic-flows/

DEMO: CREATING A VPC

NETWORKING AND SECURITY FOR AMAZON EC2 INSTANCES

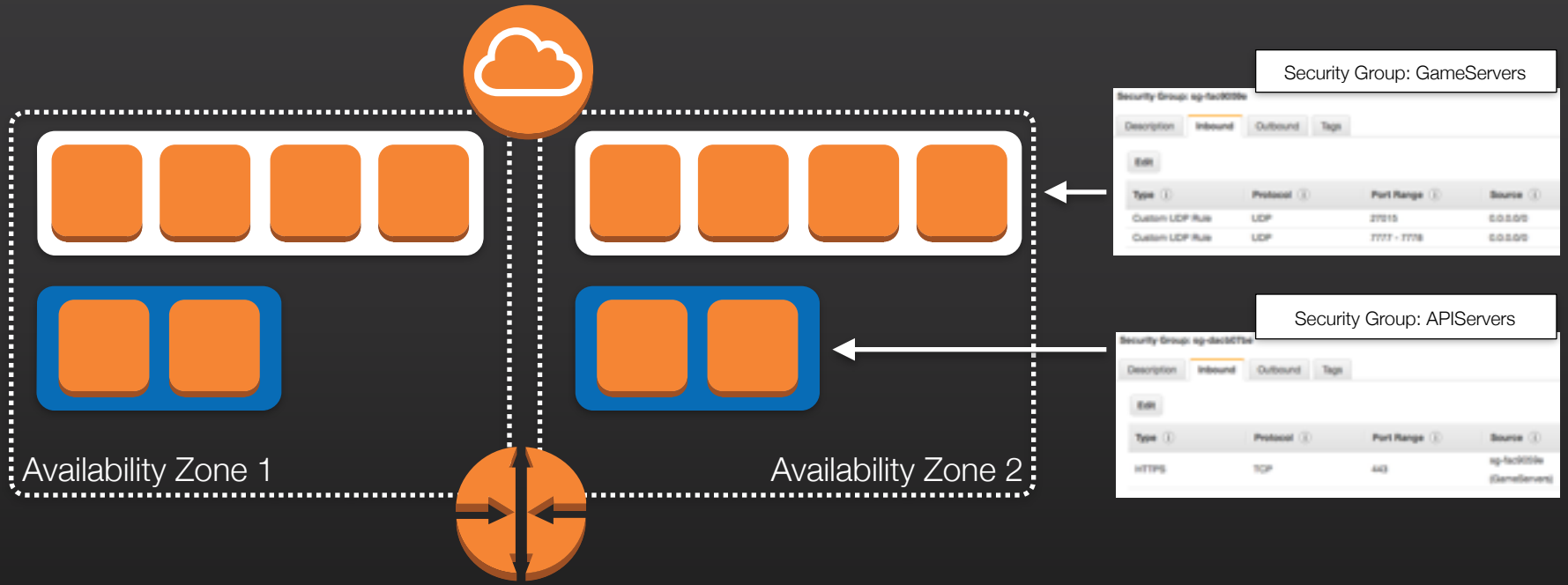
Amazon EC2 Security Groups

A security group acts as a virtual firewall that controls the traffic for one or more instances.

You add rules to each security group that allow traffic to or from its associated instances.

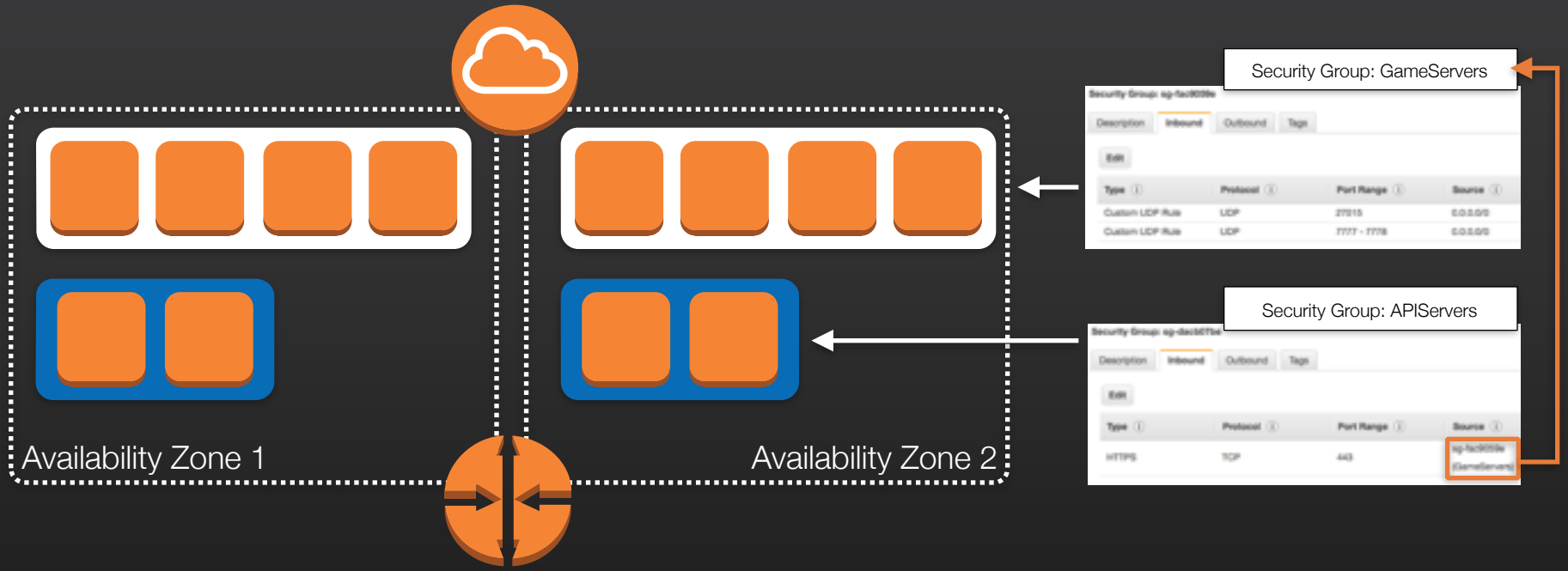
aws.amazon.com/ec2/

Amazon EC2 Security Groups



aws.amazon.com/ec2/

Amazon EC2 Security Groups



aws.amazon.com/ec2/

DEMO: WORKING WITH SECURITY GROUPS

Creating Security Groups

AWS CLI

```
$ aws ec2 create-security-group --group-name GameServers --
description “Game Server Fleet SG“ --vpc-id vpc-21b05a44
```

```
{
      “GroupId” : “sg-fac9059e”
}
```

http://docs.aws.amazon.com/cli/latest/reference/ec2/create-security-group.html

Authorising Security Group Ingress/Egress

AWS CLI

```
$ aws ec2 authorize-security-group-ingress --group-id sg-fac9059e
--protocol udp --port 27016 --cidr 0.0.0.0/0

$ aws ec2 describe-security-groups --filters Name=group-
name,Values=GameServers
```

```
SECURITYGROUPS      Sample sg-fac9059e  GameServers  650160225048 vpc-21b05a44
IPPERMISSIONS       27015  udp    27015
IPRANGES            0.0.0.0/0
IPPERMISSIONS       27016  udp    27016
IPRANGES            0.0.0.0/0
IPPERMISSIONS       7777   udp    7778
IPRANGES            0.0.0.0/0
IPPERMISSIONSEGRESS       -1
IPRANGES            0.0.0.0/0
```

http://docs.aws.amazon.com/cli/latest/reference/ec2/authorize-security-group-ingress.html

Describing Security Groups

AWS CLI

```
$ aws ec2 describe-security-groups --filters Name=group-name,Values=GameServers --output text
```

```
SECURITYGROUPS      Sample sg-fac9059e  GameServers  650160225048 vpc-21b05a44
IPPERMISSIONS       27015  udp    27015
IPRANGES            0.0.0.0/0
IPPERMISSIONS       27016  udp    27016
IPRANGES            0.0.0.0/0
IPPERMISSIONS       7777   udp    7778
IPRANGES            0.0.0.0/0
IPPERMISSIONSEGRESS       -1
IPRANGES            0.0.0.0/0
```

http://docs.aws.amazon.com/cli/latest/reference/ec2/describe-security-groups.html

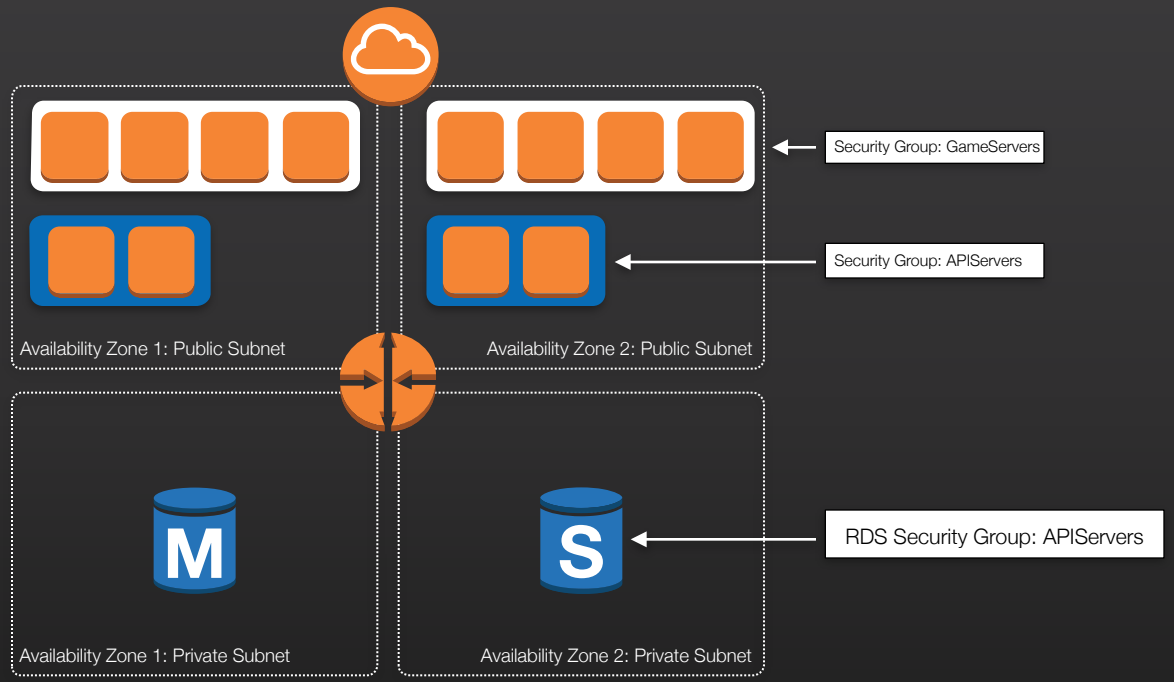
WORKING WITH CONTAINER & ABSTRACTED SERVICES

Container Services

Amazon RDS

Amazon EMR

Amazon RDS Security Groups



STORING SECRETS FOR ACCESS TO CONTAINER SERVICES

AWS SECURITY BLOG

How to Create a Policy That Whitelists Access to Sensitive Amazon S3 Buckets

amazon web services | Security Blog

Stay up to date on security and compliance in AWS

How to Create a Policy That Whitelists Access to Sensitive Amazon S3 Buckets

September 14, 2015 | Matt Bretan | How-to guides | Amazon S3 | NotPrincipal element | Principal element | Whitelisting

When it comes to securing access to your Amazon S3 buckets, AWS provides various options. You can utilize access control lists (ACLs), AWS Identity and Access Management (IAM) user policies, and S3 access policies. Even within S3 access policies, you have options to consider. You can use the Principal element, which allows you to utilize the default-deny capabilities of the policy language to grant access to, for example, a list of AWS accounts. There is also an often-overlooked "sibling" to the `Principal` element, the NotPrincipal element, which enables more-granular whitelisting. The `NotPrincipal` element allows you to ensure explicitly that no one—*except* a few select users—has access to a specific resource.

In this blog post, I will demonstrate how to create an S3 access policy that uses the `NotPrincipal` element to whitelist access to sensitive S3 buckets.

The Principal element

Before, I dive into a use case that will show the `NotPrincipal` element at work, I will first explain the `Principal` element.

The `Principal` element specifies the user, account, service, or other entity that is allowed or denied access to a resource. It is used in the trust policies for IAM roles and in resource-based policies—that is, in policies that can be attached directly to a resource, such as an S3 bucket or an Amazon SQS queue.

The `Principal` element is *not* used in policies that you attach to IAM users and groups. Similarly, in the access policy for an IAM role, you do not specify a principal. In those cases, the principal is implicitly the user that the policy is attached to (for IAM users) or the user who assumes the role (for role access policies). If the policy is attached to an IAM group, the principal is the member of the group who is making the request.

How to use the NotPrincipal element

The `NotPrincipal` element lets you specify an exception to a list of principals. For example, you can use this element to allow all AWS accounts except a specific account to access a resource. Conversely, you can deny access to all principals except the one named in the `NotPrincipal` element. As with the `Principal` element, you specify the user or account that should be allowed or denied permission. The difference is that the `NotPrincipal` element applies to everyone *except* that person or account. When

AWS

How to C
Access to

```
{
    "Sid": "ListRelevantDirectories20150907",
    "Effect": "Deny",
    "NotPrincipal": {
        "AWS": [
            "arn:aws:iam::123456789012:role/CredMgr",
            "arn:aws:iam::123456789012:role/CredUsr",
            "arn:aws:sts::123456789012:assumed-role/CredMgr/Mgr1",
            "arn:aws:sts::123456789012:assumed-role/CredUsr/User1",
            "arn:aws:sts::123456789012:assumed-role/CredUsr/User2"
        ]
    },
    "Action": [
        "s3:ListBucket"
    ],
    "Resource": "arn:aws:s3:::CredentialBucket"
}
```

amazon web services | Security Blog

ance in AWS

hitelists Access to

Amazon S3 | NotPrincipal element |

S provides various options. You can
ment IAM user policies, and S3 access
ier. You can use the Principal element,
y language to grant access to, for
"sibling" to the Principal element,
The NotPrincipal element allows
s access to a specific resource.

icy that uses the NotPrincipal

ment at work, I will first explain the

her entity that is allowed or denied
d in resource-based policies—that is, in
bucket or an Amazon SQS queue.

AM users and groups. Similarly, in the
e cases, the principal is implicitly the
sumes the role (for role access policies).
r of the group who is making the

The NotPrincipal element lets you specify an exception to a list of principals. For example, you can use this element to allow all AWS accounts except a specific account to access a resource. Conversely, you can deny access to all principals except the one named in the NotPrincipal element. As with the Principal element, you specify the user or account that should be allowed or denied permission. The difference is that the NotPrincipal element applies to everyone except that person or account. When

Abstracted Services

Amazon S3

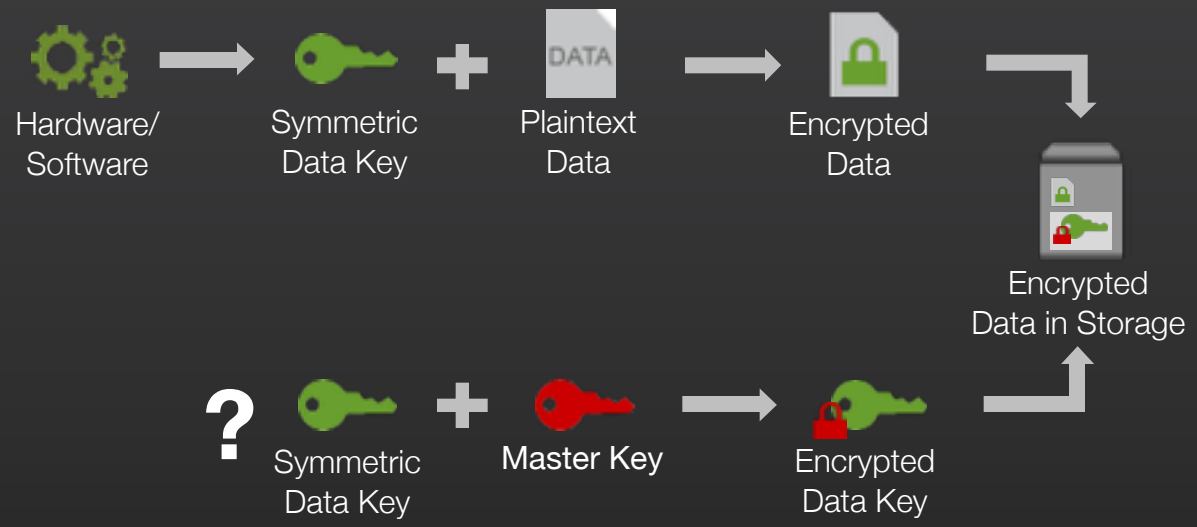
Amazon DynamoDB

USE IAM ROLES TO PASS ACCESS CREDENTIALS TO AN INSTANCE

DEMO: WORKING WITH IAM ROLES

ENCRYPTION AND KEY MANAGEMENT IN AWS

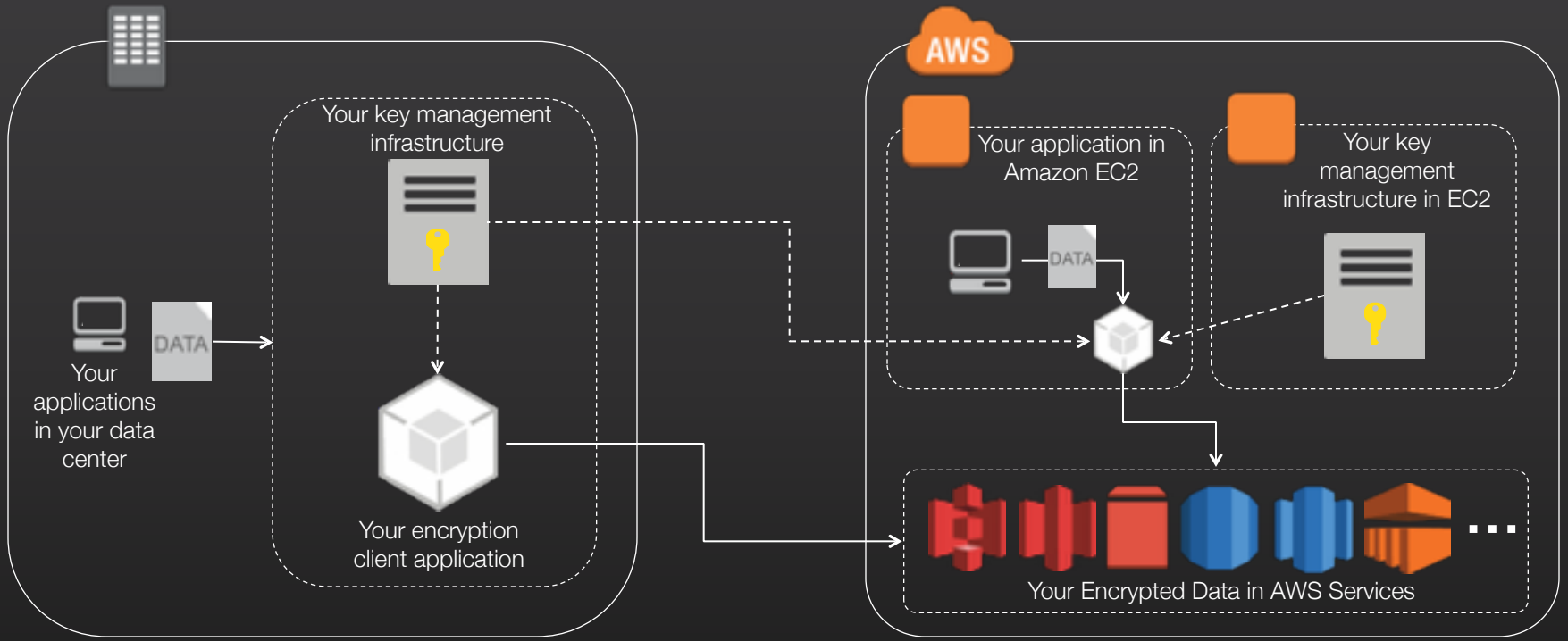
Encryption Primer



Key Hierarchy

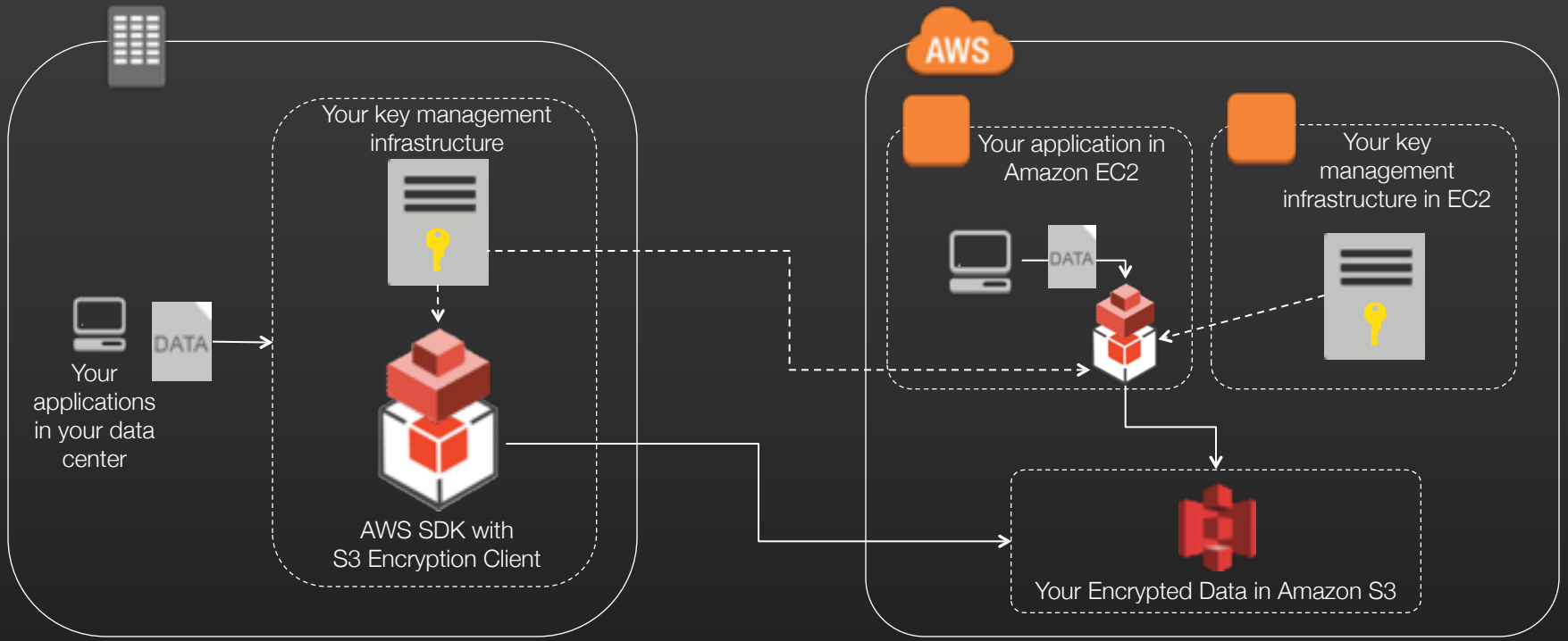
DIY Key Management in AWS

Encrypt data client-side and send ciphertext to AWS storage services



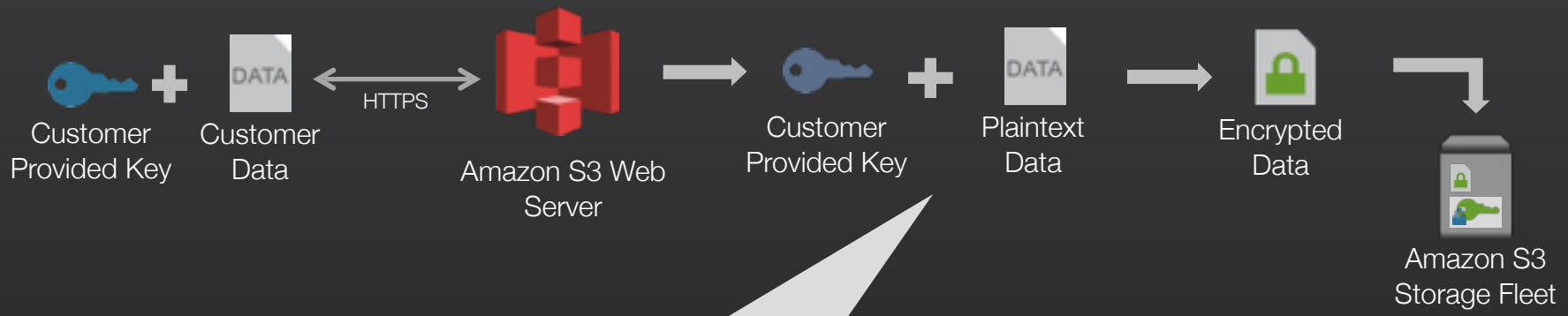
DIY Key Management in AWS

Amazon S3 Encryption Client in AWS SDKs



DIY Key Management in AWS

Amazon S3 Server-Side Encryption with Customer-Provided Keys



- Key is used at Amazon S3 webserver, then deleted
- Customer must provide same key when downloading to allow Amazon S3 to decrypt data

AWS Key Management Service

- A managed service that makes it easy for you to create, control, rotate, and use your encryption keys
- Integrated with AWS SDKs and AWS services including Amazon EBS, Amazon S3, and Amazon Redshift
- Integrated with AWS CloudTrail to provide auditable logs to help your regulatory and compliance activities

AWS Key Management Service

Integrated with AWS IAM Console

Services Edit

Dashboard

Details

Groups

Users

Roles

Identity Providers

Password Policy

Credential Report

Encryption Keys

Create Key | Key Actions

Filter: US East (N. Virginia) Search

Alias	Key ID	Status
HighlyConfidentialData	[redacted]-4b59-ae60-910bc8011638	Enabled
CriticalData	[redacted]4226-ac1c-ca8a1a92204f	Enabled
ApplicationXYZ	[redacted]-42f8-9c27-853558d4f8af	Enabled
aws/redshift	[redacted]93b-8252-67095b5e3d5e	Enabled
aws/ebs	[redacted]-4aa6-889f-db95f02123b0	Enabled
aws/s3	[redacted]-4f4e-95c8-801a16e13921	Enabled

AWS Key Management Service

Integrated with Amazon EBS

Create Volume

Type	General Purpose (SSD)
Size (GiB)	100 (Min: 1GiB, Max: 1024GiB)
IOPS	300 / 3000 (3000 IOPS bursts and baseline of 3 IOPS per GB)
Availability Zone	us-east-1b
Snapshot ID	Search (case-insensitive)
Encryption	☑ Encrypt this volume
Master Key	CriticalData

Key Details

Description	This key protects critical data in my account
Account	This account ([redacted])
KMS Key ID	[redacted]-a0ec-33d40cacf295

Cancel Create

AWS Key Management Service

Integrated with Amazon S3

Set Details Cancel

Upload to: All Buckets / critical-data

Details: Set additional details for all of the objects you upload. You can choose between Standard Storage and Reduced Redundancy Storage. You can also choose whether or not to encrypt your files.

☐ Use Reduced Redundancy Storage

☑ Use Server Side Encryption Learn more

○ **Use the Amazon S3 service master key**
S3 will decrypt the object for anyone with permission to access this object.

◉ **Use an AWS Key Management Service master key**
S3 will decrypt the object for anyone with permission to access this object and permission to use the master key.

Master Key: CriticalData

Only keys in the same region as this bucket are available for encrypting objects in this bucket.

Description: Protects critical data in my applications

Account: [redacted] (this account)

Key ID: [redacted]6-ac1c-ca8a1a92204f

< Select Files | Set Permissions > | Start Upload | Cancel

AWS Key Management Service

Integrated with Amazon Redshift

CLUSTER DETAILS | NODE CONFIGURATION | ADDITIONAL CONFIGURATION | REVIEW

Provide the optional additional configuration details below.

Cluster Parameter Group: default.redshift-1.0 — Parameter group to associate with this cluster.

Encrypt Database: None / KMS / HSM — Learn more about database encryption

Master Key: CriticalData

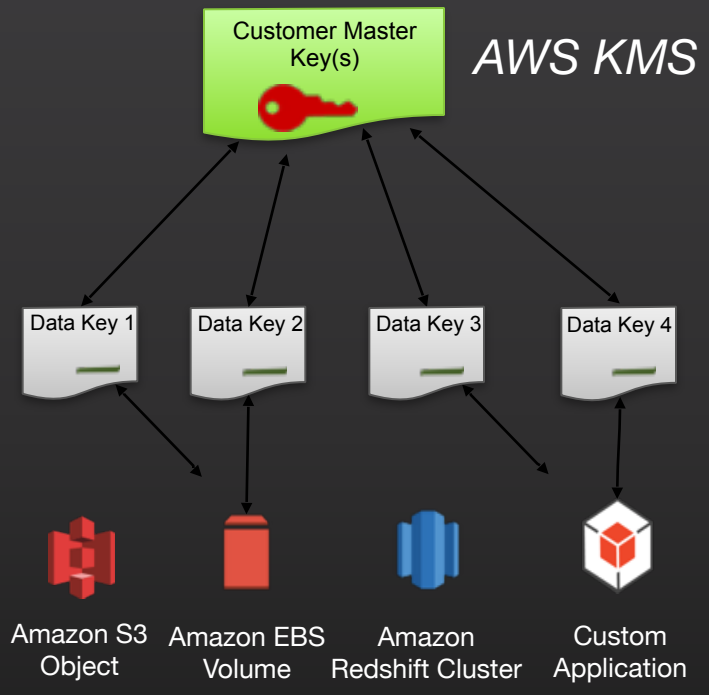
Description: Protects critical data in my applications

Account: This account ([redacted])

KMS Key ID: [redacted]-4226-ac1c-ca8a1a92204f

How AWS Services Integrate with AWS Key Management Service

- Two-tiered key hierarchy using envelope encryption
- Unique data key encrypt customer data
- AWS KMS master keys encrypt data keys
- Benefits of envelope encryption:
  - Limits risk of a compromised data key
  - Better performance for encrypting large data
  - Easier to manage a small number of master keys than millions of data keys



AWS Key Management Service

Providing security for your keys

- Plaintext keys are never stored in persistent memory on runtime systems
- Automatically rotate your keys for you
- Separation of duties between systems that use master keys and data keys
- Multi-party controls for all maintenance on systems that use your master keys
- See public white papers and Service Organization Control (SOC 1) compliance package

RESOURCES YOU CAN USE TO LEARN MORE

aws.amazon.com/security/

Cloud Security Tools

DDoS Mitigation

Learn about how to use AWS technologies like autoscaling, Amazon CloudFront and Amazon Route 53 to mitigate Distributed Denial of Service attacks. **Learn more »**

More Secure in the Cloud

This IDC paper outlines the factors to consider, and the controls you have with AWS that can make your cloud deployment more secure than your on-premises deployment. **Download now »**

AWS Security Blog

NIST Compliance in the AWS Cloud

How to Help Prepare for DDoS Attacks by Reducing Your Attack Surface

New Australian IRAP FAQ and Hub Page

Organize Your Permissions by Using Separate Managed Policies

Security Whitepapers

- Introduction to AWS Security
- Security at Scale: Governance in AWS
- Security at Scale: Logging in AWS
- AWS Security Best Practices
- Securing Data at Rest with Encryption
- AWS Security Whitepaper

Security Videos

- re:Invent 2014 - AWS Security Keynote Address
- Architecting for Greater Security on AWS
- Understanding AWS Security
- VPC: A Day in the Life of a Billion Packets
- Intrusion Detection in the Cloud
- IAM Best Practices
- Architecting for End-to-End Security in the Enterprise
- Encryption and Key Management in AWS
- Incident Response in the Cloud

Online Documenatation

- EC2 Security and Networking
- Security in Your Virtual Private Cloud (VPC)
- Networking in Your VPC
- AWS Identity and Access Management (IAM)
- Multi-Factor Authentication (MFA)
- Amazon S3 Bucket Logging
- Customer Penetration Testing on AWS

AWS Technical Documentation

Amazon Virtual Private Cloud

User Guide (API Version 2015-04-15)

Security in Your VPC

Amazon VPC provides two features that you can use to increase security for your VPC:

- Security groups—Act as a firewall for associated Amazon EC2 instances, controlling both inbound and outbound traffic at the instance level
- Network access control lists (ACLs)—Act as a firewall for associated subnets, controlling both inbound and outbound traffic at the subnet level

When you launch an instance in a VPC, you can associate one or more security groups that you've created. Each instance in your VPC could belong to a different set of security groups. If you don't specify a security group when you launch an instance, the instance automatically belongs to the default security group for the VPC. For more information about security groups, see Security Groups for Your VPC

You can secure your VPC instances using only security groups; however, you can add network ACLs as a second layer of defense. For more information about network ACLs, see Network ACLs.

You can use AWS Identity and Access Management to control who in your organization has permission to create and manage security groups and network ACLs. For example, you can give only your network administrators that permission, but not personnel who only need to launch instances. For more information, see Controlling Access to Amazon VPC Resources.

Amazon security groups and network ACLs don't filter traffic to or from link-local addresses (`169.254.0.0/16`) or AWS reserved addresses (the first four IP addresses and the last one in each subnet). These addresses support the services: Domain Name Services (DNS), Dynamic Host Configuration Protocol (DHCP), Amazon EC2 instance metadata, Key Management Server (KMS—license management for Windows instances), and routing in the subnet. You can implement additional firewall solutions in your instances to block network communication with link-local addresses.

Comparison of Security Groups and Network ACLs

The following table summarizes the basic differences between security groups and network ACLs.

Security Group	Network ACL
Operates at the instance level (first layer of defense)	Operates at the subnet level (second layer of defense)
Supports allow rules only	Supports allow rules and deny rules
Is stateful: Return traffic is automatically allowed, regardless of any rules	Is stateless: Return traffic must be explicitly allowed by rules
We evaluate all rules before deciding whether to	We process rules in number order when deciding

blogs.aws.amazon.com/security

Organize Your Permissions by Using Separate Managed Policies

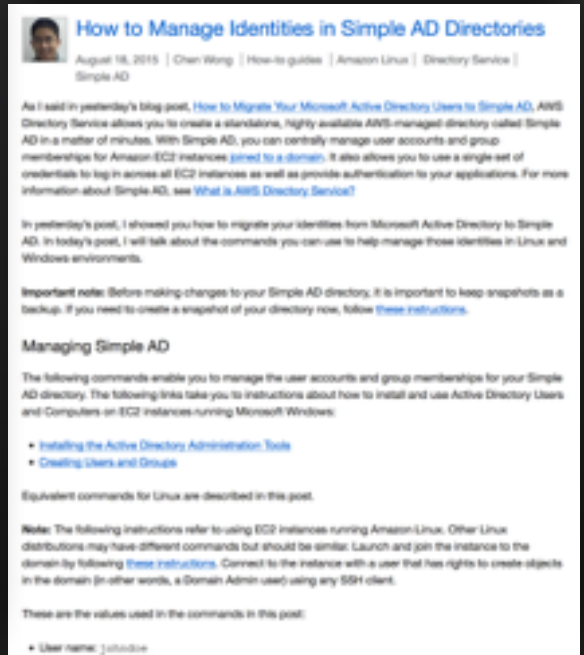
August 20, 2015 | Brigid Johnson | Announcements | How-to guides | console | IAM | Policies

This year we released managed policies to enable you to create a set of stand-alone policies that you can attach to multiple IAM entities (users, groups, and roles) in your AWS account. Since that release, we have heard from many of you that you'd prefer to mix and match policies instead of just using one universal policy. For example, instead of creating one policy to grant access to multiple services, you might want to attach a separate policy for each service. In order to facilitate the flexibility to logically separate policies, you can now attach 10 managed policies to each entity. This allows for an easier understanding of permissions by looking at the list of policies attached to each entity.

Let's walk through an example use case. Imagine you have a database administrator with an IAM user named Alice that needs full access to Amazon DynamoDB, Amazon Relational Database Service (RDS), Amazon Redshift, and Amazon ElastiCache. Additionally, she also needs read-only access to Amazon Simple Storage Service (S3) and Amazon Glacier. To grant these permissions to Alice, we'll use AWS managed policies (policies created and maintained by AWS that can be used to grant common types of access). We'll attach the following AWS managed policies to Alice:

- **AmazonDynamoDBFullAccess**
- **AmazonRDSFullAccess**
- **AmazonRedshiftFullAccess**
- **AmazonElastiCacheFullAccess**
- **AmazonS3ReadOnlyAccess**
- **AmazonGlacierReadOnlyAccess**

To attach these six policies to Alice, click **Users** in the left pane of the console.



How to Manage Identities in Simple AD Directories

August 18, 2015 | Chen Wong | How-to guides | Amazon Linux | Directory Service | Simple AD

As I said in yesterday's blog post, How to Migrate Your Microsoft Active Directory Users to Simple AD, AWS Directory Service allows you to create a standalone, highly available AWS-managed directory called Simple AD in a matter of minutes. With Simple AD, you can centrally manage user accounts and group memberships for Amazon EC2 instances joined to a domain. It also allows you to use a single set of credentials to log in across all EC2 instances as well as provide authentication to your applications. For more information about Simple AD, see What is AWS Directory Service?

In yesterday's post, I showed you how to migrate your identities from Microsoft Active Directory to Simple AD. In today's post, I will talk about the commands you can use to help manage those identities in Linux and Windows environments.

Important note: Before making changes to your Simple AD directory, it is important to keep snapshots as a backup. If you need to create a snapshot of your directory now, follow these instructions.

Managing Simple AD

The following commands enable you to manage the user accounts and group memberships for your Simple AD directory. The following links take you to instructions about how to install and use Active Directory Users and Computers on EC2 instances running Microsoft Windows:

- Installing the Active Directory Administration Tools
- Creating Users and Groups

Equivalent commands for Linux are described in this post.

Note: The following instructions refer to using EC2 instances running Amazon Linux. Other Linux distributions may have different commands but should be similar. Launch and join the instance to the domain by following these instructions. Connect to the instance with a user that has rights to create objects in the domain (in other words, a Domain Admin user) using any SSH client.

These are the values used in the commands in this post:

- User name: jstudio

How to Address the PCI DSS Requirements for Data Encryption in Transit Using Amazon VPC

July 23, 2015 | Balaji Palanisamy | Compliance | Encryption | Amazon VPC | PCI DSS

The PCI requirements for encryption for data in transit are different for private networks than they are for public networks. When correctly designed, Amazon Virtual Private Cloud (Amazon VPC), a logically isolated portion of the AWS infrastructure that allows you to extend your existing data center network to the cloud, can be considered a private network, as qualified by the Payment Card Industry Data Security Standards (PCI DSS).

In this blog post, I will review the importance of understanding the logical isolation provided by Amazon VPC and then review some of the key points to consider when designing for PCI workloads that need to transmit sensitive data within or outside the AWS infrastructure. I will also demonstrate how you can use the native isolation provided by Amazon VPC for additional security.

Amazon VPC is the architectural construct of choice for AWS customers deploying workloads that are in scope for a PCI DSS assessment. Within Amazon VPC, Amazon EC2 instances must have an Internet gateway or a virtual private gateway in order to communicate with hosts outside Amazon VPC. Additionally, AWS-designed Layer 2 networking features include the mapping service, which performs checks to ensure that even packets with malformed or modified addresses cannot hop across Amazon VPC boundaries. Network access-control lists (NACLs) and security groups may be used to filter inbound and outbound traffic to hosts within Amazon VPC. These controls make it difficult for data to be intercepted or diverted while in transit, and demonstrate the private nature of Amazon VPC.

Encryption of sensitive data in motion is addressed in PCI DSS version 3.1 via Requirement 4 and its corresponding subrequirements. The DSS is clear that the requirements apply to the transmission of payment card data across "open, public networks" that are susceptible to unauthorized access. The PCI DSS and the PCI Glossary describe public networks as network transport providers that connect an organization's networks to each other over a wide area network (WAN), to the Internet, or to partner networks—and not software-defined cloud constructs such as Amazon VPC.

Typically, such public networks exhibit managed ingress and egress points that act as gateways to a shared network, with the provider managing the routing within the shared network. It is also possible that the ingress and egress points may be represented by dedicated physical hardware called the customer-premises equipment (CPE). On the other hand, the software-defined Amazon VPC abstracts any underlying hardware and allows for logical isolation. Additionally, PCI DSS testing procedures such as 4.1.c require the PCI Qualified Security Assessor (QSA) to "observe a sample of inbound and outbound transmissions as they

AWS Security White Papers

Introduction to AWS Security

Security at Scale: Governance in AWS

Security at Scale: Logging in AWS

AWS Security Best Practices

Securing Data at Rest with Encryption

AWS Security Whitepaper

aws.amazon.com/iam

aws.amazon.com/vpc

aws.amazon.com/kms

aws.amazon.com/config

aws.amazon.com/cloudtrail

aws.amazon.com/cloudhsm

aws.amazon.com/cloudwatch

aws.amazon.com/trustedadvisor

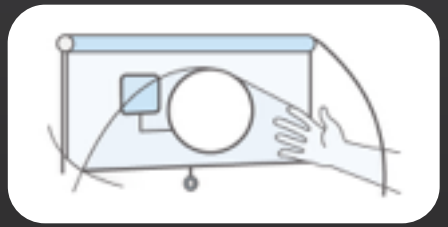
AWS Training & Certification

Self-Paced Labs

Try products, gain new skills, and get hands-on practice working with AWS technologies

aws.amazon.com/training/self-paced-labs

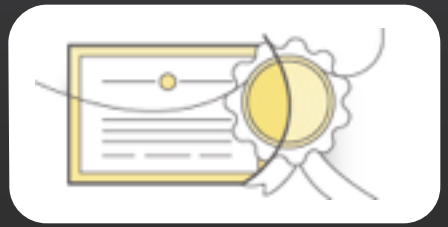
Training



Build technical expertise to design and operate scalable, efficient applications on AWS

aws.amazon.com/training

Certification



Validate your proven skills and expertise with the AWS platform

aws.amazon.com/certification

Follow us for more events & webinars

Ian Massingham — Technical Evangelist

@IanMmmm

@AWS_UKI for local AWS events & news

@AWScloud for Global AWS News & Announcements