Notes Based on Real Failed Attempts

**Athena**- is an interactive query service that enables you to analyze and query data located in S3 using SQL.

* Serverless
* Noting to provision
* Pay per query and per TB scanned
* No need to set up complex/transform/load (ETL) processes
* Works directly with data stored in S3

Use cases

* Can be used to query log files stored in S3, ELB logs, S3 access logs, CloudTrail logs
* Generate business reports on data stored in S3
* Analyze AWS cost and usage reports
* Run queries on click-stream data

**Quicksights**- A business analytics service you can use to build visualizations, perform ad hoc analysis, and get business insights from your data

How to make sure you do not use AWS provided DNS

Set the EnableDNSHostnames=False – this indicates whether the instances launched in the VPC get public DNS names

Set EnableDNSSupport=False – disables the amazon provided DNS server

**Cognito Sync Trigger**- an AWS service and client library that enables cross-device syncing of application related user data

**SES Port number**- Open port 587 which is SMTP

**ElastiSearch**- a managed service that makes it easy to deploy, operate, and scale ElastiSearch clusters in the AWS cloud.

**Securing ECS**- EC2 instances use IAM roles to access ECS. ECS tasks use an IAM role to access services and resources. Also you secure them with SG and NACLs

Write once Read many Glacier policy

For this specific question it will give you policy documents that you have to read. Look for ones that only allow uploads to glacier, and they don’t allow archive deletes. There should be an action of Glacier:DeleteArchive with the condition of NumericLessThan Glacier:ArchiveInDays that abides by the amount of days mandatory before deletes are allowed to occur.

**How to recover a deleted CMK?** So you can’t, its lost forever, except the question did not have that as one of the answers…

I think that this was a CMK with imported key material and I missed that. If you have a CMK that is deleted with imported key material then you can reupload that key material again. This is the only logical explanation I got for that specific question.

AWS Security Section 2 Notes

CIA is a security model in IT that stands for confidentiality, integrity, and availability.



Confidentiality = privacy

* Think about health care data. You’ve had a DNA test done that measures your disease risk.
* You want to keep this data private to yourself, but you might want to share it to your wife and children.
* Does not have to be absolutely secret to yourself.
* Data that you want to keep confidential, but you want to expose to 3rd parties where needed.
* Examples= Health, a bank statement

How to ensure confidentiality

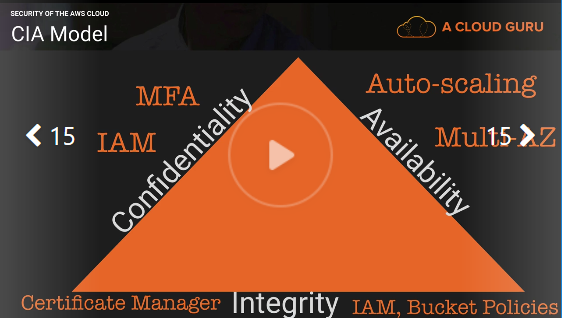
* Data encryption- encrypting data at REST and in Transit
* User ID’s and passwords
* 2 Factor Authentication
* AWS services- IAM, MFA, Bucket policies, ACL’s, Security Groups, encryption

Integrity = maintaining consistency, accuracy, and trustworthiness of your data over its entire lifecycle. Data can’t change in transit and ensure that data can not be altered by unauthorized people.

* File permissions
* User access controls
* Version control
* Example= Checksum- when you download a piece of software it comes with a Checksum. You can check the hash of that software with the checksum and if they match that means your data is integral.
* AWS services- amazon certificate manager, IAM, Bucket Policies, encryption, versioning, MFA delete

Availability= Keeping your systems available

* Redundancy
* Raided disks, HA clusters, multi AZ, multi regions, and design for failure
* AWS services- Auto-Scaling, Multi-AZ, Multi-Regions, Route 53 with health checks



AAA- Extends and compliments the CIA model

Authentication- When you log in to the AWS console the first thing you do is enter your user name and password. This is authenticated against IAM and checks whether there is a user with that user name and password.

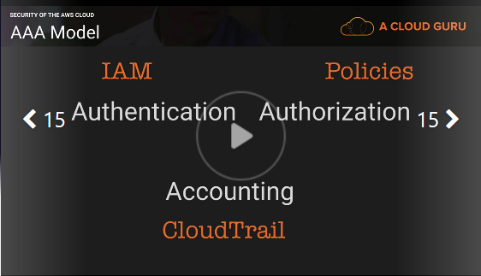
IAM

Authorization- The ability to use the console depends on how much permissions that you have.

Policies

Accounting- What is it that you are doing on the AWS platform?

Cloudtrail



The Security of AWS

Physical and environmental security

* Fire detection and suppression
* Power
* Climate and temperature
* Management
* Storage device decommissioning

Business Continuity Management

* Availability
* Incident response
* Companywide executive review
* communication

Network Security

* Secure Network Architecture
* Secure Access Points
* Transmission protection
* Amazon corporate segregation
* Fault tolerant design
* Network monitoring and protection

AWS access

* Account review and audit
* Background checks
* Credential policy

Secure design principles

* AWS development process follows secure software development best practices

Change management

* Software
* Infrastructure

Security of the Cloud- AWS responsibility

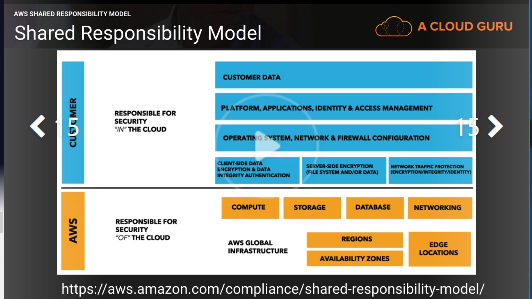
Security in the Cloud- your responsibility

AWS responsibility

* Global infrastructure
* Hardware, software, networking, and facilities
* “managed services”
* Hyper visors

Customer security responsibilities

* Infrastructure as a services (IAAS)
* Including updates and security patches
* Configuration of the AWS provided firewalls



The model changes for different service types:

* Infrastructure
* Container
* Abstracted

Infrastructure: This category includes compute services, such as amazon EC2, EBS, auto scaling, and amazon VPC. With these services, you can architect and build a cloud infrastructure using technologies similar to and largely compatible with on-premises solutions. You control the operating system, and you configure and operate any identity management system that provides access to the user layer of the virtualization stack.

EC2 you are responsible for;

* AMI’s
* Operating systems
* Applications
* Data in transit
* Data at rest
* Data stores
* Credentials
* Policies and configuration

Container services: Services in this category typically run on separate Amazon Ec2 or other infrastructure instances, but sometimes you don’t manage the operating system of the platform layer. AWS provides a managed service for these application “containers”. You are responsible for setting up and managing network controls, such as firewall rules, and for managing platform-level identity and access management separately from IAM. Examples of container services include RDS, EMR, and Elastic Beanstalk.

Abstracted Services: This category includes high-level storage, database, and messaging services, such as S3, Glacier, DynamoDB, SQS, SES. These services abstract the platform or management layer on which you can build and operate cloud applications. You access the endpoints of these abstracted services using AWS APIs, and AWS manages the underlying service components or the operating system on which they reside.

Security in AWS

Cloud controls

* Visibility
* Auditability
* Controllability
* Agility
* Automation
* Scale

Visibility

What assets do you have? (AWS Config)

Auditability

Do we comply with policies and regulations? (AWS CloudTrail)

Controllability

Is my data controlled? (AWS KMS and AWS CloudHSM)

Agility

How quickly can we adapt to changes? (AWS CloudFormation and AWS Elastic Beanstalk)

Automation

Are our processes repeatable? (AWS CodeDeploy and AWS OpsWorks)

Scale

In AWS the scale is on your side!

Services that help with all control

AWS IAM AWS CloudWatch AWS Trusted Advisor

Summery

Non-repudiation- you can’t deny that you did something.

* Cloudwatch
* Cloudtrail
* IAM
* MFA

Why should we trust AWS?

Compliance programs

* Pci DSS
* ISO 27001
* HIPPA

AWS Security Specialty Section 3 Notes

IAM allows you to manage users and their level of access to the AWS Console.

What does IAM give you?

* Centralized control of your AWS account
* Shared access to your AWS account
* Granular permissions
* Identity federation
* MFA
* Provides temporary access for users/devices and services where necessary
* Allows you to set up your own password rotation policy
* Integrates with many different AWS services
* Supports PCI DSS compliance

**Users-** End users. (people)

**Groups-** A collection of users under one set of permissions

**Roles-** you create roles and can then assign them to AWS resources

**policies-** a document that defines one or more permissions

IAM is global

Root user scenario

1. Change the password
2. Delete root keys
3. Disable and reenable MFA
4. Evaluate existing users

IAM policies overview

**IAM policies**- sepecify what you are allowed to do with any AWS resources. They are global and apply to all areas of AWS. You can attach IAM policies to IAM users, groups, or roles. These users, groups, and roles are then subject to the permissions you define in the policy. In other words, IAM policies define what a principle can do in your environment.

Types of policies

* AWS Managed policies
* Customer managed policies
* Inline Policies

AWS Managed policies

* An AWS managed policy is a standalone policy that is created and administered by AWS
* Literally 100,000 or even millions of AWS accounts use these policies. They are the same applied across multiple accounts.
* They can change, but AWS is very careful as a small change could impact a lot of people as everyone has access to these policies.

Customer Managed policies

* Standalone policies that you admin in your own AWS account
* You can then attach the policies to multiple principle entities in your AWS account.
* When you attach a policy to a principle entity, you give the entity the permissions that are defined in the policy

Inline policy

* Are useful if you want to maintain a strict one-to-one relationship between a policy and the principle entity that it’s applied to
* Use it if you want to be sure that permissions in a policy are not inadvertently assigned to a principle entity other than the one they are intended for

What is the difference between the Root user, an Admin user, and a Power user?

* Root user has access to everything
* Admin has access to everything except billing
* Power user has access to everything except billing and IAM

Bucket policies

**Bucket policies-** S3 bucket policies are attached only to S3 buckets. S3 bucket policies specify what action are allowed or denied on the bucket. They can be broken down to a user level, so Alice can PUT but not DELETE and John can READ but not PUT. Bucket level only, S3 only.

Use cases

* You want a simple way to grant cross-account access to your S3 environment, without using IAM roles.
* Your IAM policies bump up against the size limit. (2kb for users, 5kb for groups, 10kb for roles). S3 supports bucket policies of up to 20kb.
* You prefer to keep access control policies in the S3 environment

A Bucket policy overwrites an IAM policy unless the IAM policy has an explicit deny. For example if we had a bucket policy allowing access and an IAM policy with nothing in it that user would be able to do everything the bucket policy was allowing.

At the end of the resource line stating your bucket you need to add a / followed by either a \* for all objects or followed by an object name. This is not talked about in the video so here is an example from my account.



I have 2 separate statements in a bucket policy highlighted by the red squares.

On the first statement the resource has the /\* which means allow access to every action stated for every file in the bucket.

The second statement has a deny effect on a specified file in my bucket called index.txt

This means everything is allowed in my bucket except for on that object in which I placed a deny.

**Principles-** when making a bucket you are asked to choose a principle that it effects and these are often IAM users, but can also be specific instances or DynamoDB tables and so on.

**Resources-** when using the policy generator your also required to choose a resource. This can be your overall bucket if you add the /\* or it can be an exact path name in which you want the statement to affect.

A bucket policy can extend extra access to IAM users who have no denies.

A bucket policy can also overwrite a root user

**Explicit deny-** an explicit deny is a deny in any policy and it overwrites all allows.

All policies are at a default deny state.

S3 ACL

**S3 ACL-** are a legacy access control mechanism that predates IAM. AWS recommend sticking to IAM policies and S3 bucket policies. However, if you need to apply policies on the objects themselves, then use S3 ACL’s. Bucket policies can only be applied at the bucket level where as S3 ACL’s can be applied to individual files.

Use cases

* If you need fine grained permissions on individual files/objects within S3
* Bucket policies are limited to 20kb in size, so consider using S3 ACL’s if you find that your bucket policy grows too large.

Setting up ACL’s

* You can go into a specific file and do basic grant read access/write access permissions
* If you want to do it at a user level you need to do it via the CLI or the API
* You need your AWS account ID and an owner canonical user id

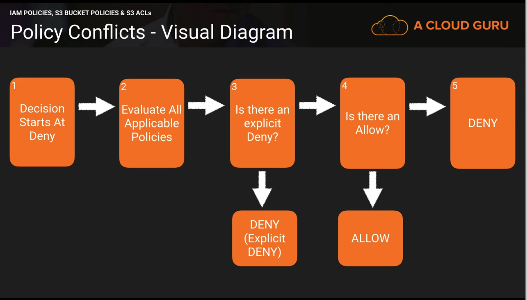
Confliction policies

Whenever an AWS principle (user, group, or role) issues a request to S3, the authorization decision depends on the union of all the IAM policies, S3 bucket policies, and S3 ACL’s that apply.

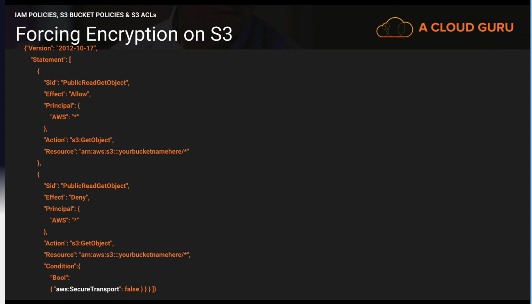
**Least privilege-** decisions in AWS always default to a deny meaning that the only way a user can do something on your account is because somewhere someone has given that user access to do so.

An explicit Deny always trumps an allow

Only if no method specifies a Deny and one or more methods specify an Allows will the request be allowed.



Forcing Encryption on S3



You need to have 2 statement. The first statement will allows access to get the object from S3 from the bucket you want.

The second statement will deny access only if the condition of Bool SecureTransport is false.

Cross Region Replication

**Cross Region Replication-** cross region replication replicates objects from one region to another

* By default, security in transit is used
* You can only replicate objects from a source bucket to only one destination bucket

Cross region replication requirements

* Versioning must be enables
* Both buckets must be in different AWS regions
* Amazon S3 must have permissions to replicate objects from that source bucket to the destination bucket on your behalf. Will create a role for you
* If the source bucket owner also owns the object, the bucket owner has full permissions to replicate the object. If not the object owner must grant the bucket owner the READ and READ\_ACP permissions via the object ACL

Cross region replications across accounts

* The IAM role must have permissions to replicate objects in the destination bucket
* In the replication configuration, you can optionally direct S3 to change the ownership of object replica to the AWS account that owns the destination bucket

What is replicated?

* Any new objects created after you add a replication configuration
* Unencrypted objects and SSE-S3 encrypted objects
* SSE-KMS encrypted objects if you grant permission
* Object metadata
* Any object ACL updates
* Any object tags
* Amazon S3 replicates only objects in the source bucket for which the bucket owner has permissions to read objects and read access control lists.

What is replicated? Deletes

* If you just use a delete marker, then the delete marker is replicates

What is not replicated?

* Anything create before CRR is turned on
* Objects create with SSE-C
* Objects with SSE-KMS unless you enable it
* Objects in the source bucket for which the bucket owner does not have permissions
* Deletes to a particular version of an object

Forcing S3 to use CloudFront

**Origin Domain Name-** The source from where your distribution is getting its content

**Origin Path-** any sub folder you want inside your bucket

**Restrict bucket access-** restricts users from accessing your S3 bucket through S3 URL’s and forces them to use CloudFront URL’s only.

Steps to restrict bucket access after a distribution has been created

1. Click on the distribution and click distribution settings
2. Click on origins
3. Click origin and click edit
4. Change and restrict bucket access
5. Create or choose and origin access identity
6. Click grant permissions on bucket
7. Save

**Origin access identity-** a special user that forces people to use cloudfront URL’s to access your amazon S3 content

**Grant read permissions-** grants read permissions to your bucket automatically so that you do not need to manually change the bucket policy.

S3 Presigned URL’s

**Presigned URL’s-** typically done using the SDK’s, but you can also do it via the command line. Allows you to share objects inside S3 without changing permissions and allows for good confidentiality.

The command to do this is

aws s3 presign s3://bucketname/object

The default time for presigned URL’s is 1 hour. The max is 7 days.

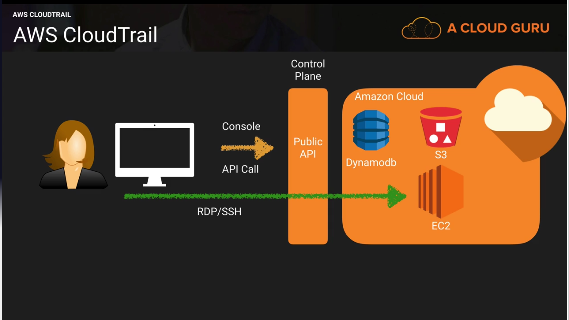
You can change the time by doing the –expires-in followed by the time you want in seconds. This looks like this.

aws s3 presign s3://bucketname/object –expires-in 300

this would change it to make it expire after 5 minutes.

AWS Security Specialty Section 4 Notes and Elaborations

CloudTrail- a web service that records AWS API calls for your account and delivers log files to you



Enables:

* After the fact incident investigation
* Near real-time intrusion detection
* Industry and regulatory compliance

Provides:

* Logs API call details (for supported service)

What is logged?

* Metadata around the API calls
* The identity of the API caller
* The time of the API call
* The source IP address of the API caller
* The request parameters
* The response elements returned by the service

CloudTrail Event Logs:

* Sent to an S3 bucket
* You manage the retention in S3 (life cycle management policies)
* Delivered every 5 (active) minutes with up to 15 minute delay
* Notifications available
* Can be aggregated across regions and across multiple accounts (use CRR for best practice)

Setup:

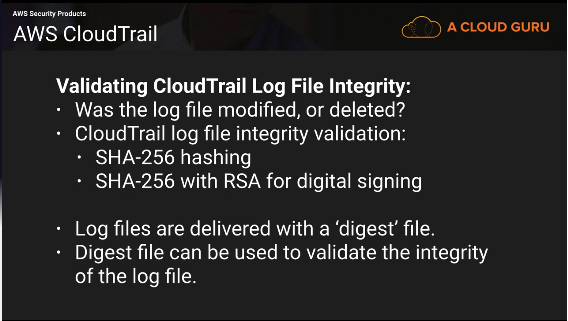
* Enabled by default (7 days)
* For longer you need to provision it in the console

**Management events-** operations that occur on your AWS account and resource, such as EC2 RunInstance API

* You can set up to log all events
* Read only events
* Write only events
* No events

**Digest File-** a way to validate if your cloudtrail logs are actually valid.

* Creates a hash for every log file that is delivered
* Hash is generated use public keys and private keys
* Amazon has access to the private keys
* Check if the hash is correct. If the hash matches then the log files have not been altered



CloudTrail Protecting Your Logs

Q) CloudTrail logs contain metadata not application data. Why should you consider security them?

A) CloudTrail logs may contain personal identifiable data such as usernames and even team memberships. Also, detailed configuration information such as DynamoDB table and key names may be stored. This information may provide valuable to an attacker and it is considered best practice to secure CloudTrail logs.

Q) How do we stop unauthorized access to log files?

A) Use IAM policies and S3 bucket policies to restrict access to the S3 bucket containing the log files.

Use SSE-S3 or SSE-KMS to encrypt the logs.

Q) How do we restrict log access to only employees with a security responsibility?

A) Place the employees who have a security role, into and IAM group with attached policies that enable access to the logs.

2 AWS CloudTrail policies that are AWS managed.

AWSCloudTrailFullAccess- Grant this policy to people who are going to be actually managing your AWS environment. Setting up CloudTrail and changing the policies of the buckets.

AWSCloudTrailReadOnlyAccess- Give this policy to auditors.

Q) How can we be notified that a log file has been created, and validate that it’s not been modified?

A) Configure SNS notifications and log file validation on the ‘Trail’. Develop a solution that when triggered by SNS will validate the logs using the provided digest file.

Set up a Lambda function that will compare digest files from a month ago and compare them with the CloudTrail logs they are associated with. If you compare files immediately then someone changes it you may not notice the change.

Q) How can we prevent logs from being deleted?

A) Restrict delete access with IAM and bucket policies. Configure S3 MFA delete. Calidate that logs have not been deleted using log file validation.

Q) How can we ensure that logs are retained for X years in accordance with our compliance standards.

A) By default, logs will be kept indefinitely. Use S3 object lifecycle management to remove the files after the required period of time, or move the files to AWS Glacier for more cost effective long term storage.

CloudWatch 101

**CloudWatch-** a monitoring service for AWS cloud resources and the applications you run on AWS

Enables:

* Resource utilization, operational performance monitoring (CPU, Disk, and custom metrics)
* Log aggregation and basic analysis

Provides:

* Real- time monitoring within AWS for resources and applications
* Hooks to event triggers

Key components:

* CloudWatch
* CloudWatch Logs
* CloudWatch Events

CloudWatch

* Real time
* Metrics
* Alarms
* Notifications
* Custom metrics (Can be on premise servers)

CloudWatch Logs

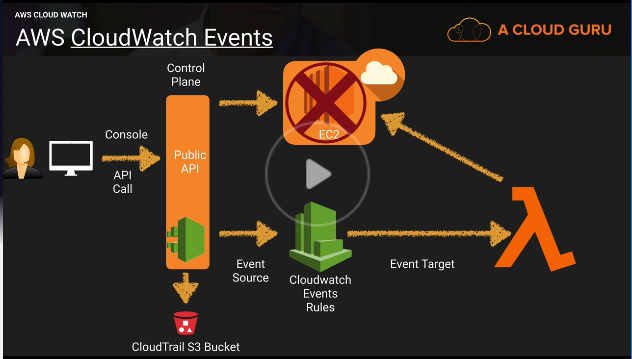
* Pushed from some AWS service (including cloudtrail)
* Pushed from your application/systems
* Metrics from log entry matches
* Stored indefinitely behind the scenes (not user S3)

Monitor HTTP response code in apache logs

Receive alarms for errors in kernel logs

Count exceptions in application logs

These are examples, but you can log just about anything in your application that you know how to script.

CloudWatch Events

* Near real-time stream of system events

Event types

* AWS resources state change (what happens when an instance gets stopped?)
* AWS CloudTrail (API Calls)
* Custom events (Code)
* Scheduled

Rules- match incoming events and route them to one or more targets

Targets- AWS Lambda, SNS topics, SQS queries, kinesis streams

Q) How do we control who can access CloudWatch and what they can do?

A) Use IAM policies to restrict access to CloudWatch and the actions they can perform.

However, remember that data is decoupled from its source, therefore you are not able to restrict access by the originating resource. (restrict access to CloudWatch and the originating source separately in your IAM policies)

Q) How are unauthorized users prevented from accessing CloudWatch?

A) Users need to be authenticated with AWS and have the appropriate permissions set via IAM policies to gain access.

Config 101

**Config-** A fully managed service that provides you with an AWS resource inventory, configuration history, and configuration change notifications to enable security and governance

Enables:

* Compliance auditing
* Security analysis
* Resource tracking

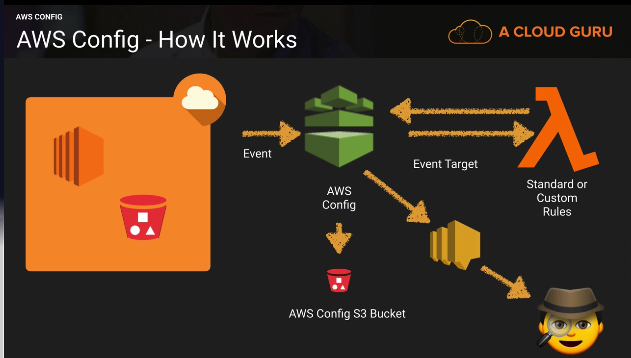
Provides:

* Configuration snapshots and logs config changes of AWS resources
* Automated compliance checking

Key components:

* Config dashboard
* Config rules (managed and custom)
* Resources
* Settings

You cannot just turn it on for all regions with one click of the button



Key Terminology:

Configuration Items- point in time attributes of a resource

Configuration snapshots- a collection of config items

Configuration stream- stream of changed config items

Configuration history- a collection of config items for a resource over time

Configuration recorder- the configuration of config that records and stores config items

Recorder setup:

* Logs config for account in a region
* Stores in S3
* Notifies SNS

What can we see:

* Resource Type
* Resource ID
* Compliance
* Timeline
  + Configuration details
  + Relationships
  + Changes
  + Cloudtrail events

Compliance checks:

* Trigger
  + Periodic
  + Configuration snapshot delivery

Managed rules

There are 60 or so right not (the video says 40)

Managed rules are basic, but fundamental.

Permissions needed for config:

* AWS config requires an IAM role with (console will optionally create this for you)
  + Read only permissions to the recorded resources
  + Write access to S3 logging bucket
  + Publish access to SNS

Restrict Access:

* Users need to be authenticated with AWS and have the appropriate permissions set via IAM policies to gain access
* Only admins needing to set up and manage config require full access
* Provide read only permissions for config day-to-day use

Monitoring config:

* Use cloudtrail with config to provide deeper insight into resources
* Use cloudtrail to monitor access to config such as someone stopping the config recorder

Set up an alert if the root user logs in

1. Turn on Cloudtrail and CloudWatch logs integration
2. Create a metric filter
3. Assign a metric
4. Create an alarm
5. Push it out to an SNS topic
6. Look up the event and take corrective action

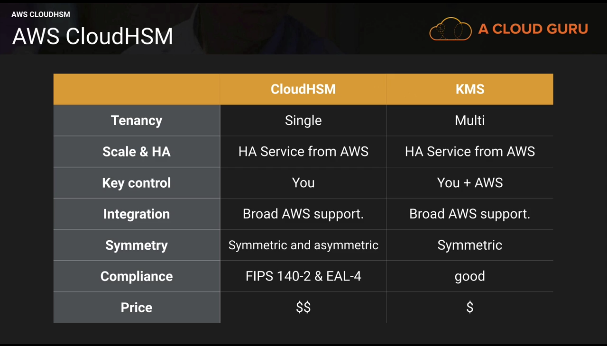
CloudHSM- A service that helps you meet corporate, contractual and regulatory compliance requirements for data security by using dedicated Hardware Security Module appliances within the AWS cloud

Enables:

* Control of data
* Evidence of control
* Meet tough compliance controls

Provides

* Secure key storage
* Cryptographic operations
* Tamper resistance Hardware Security Module



Key control:

* AWS does not have access to your keys
* Separation of duties and role based access control is part of the design of the HSM
* AWS can only administer the appliance, not the HSM partitions where the keys are stored
* AWS can (but probably won’t) destroy your keys. But otherwise they have no access

Tampering:

* If the CloudHSM detects physical tampering the keys will be destroyed
* If the CloudHSM detects five unsuccessful attempts to access an HSM partition as Crypto Officer the HSM appliance erases itself
* If the CloudHSM detects five unsuccessful attempts to access an HSM with CryptoUser credentials, the user will be locked and must be unlocked by a crypto officer

Monitoring:

Use CloudTrail to monitor your API calls to your HSM to see who is doing what.

Inspector and Trusted Advisor

Inspector- automated security assessment service that helps improve the security and compliance of applications deployed on AWS. Inspector automatically assesses applications for vulnerabilities or deviations from best practices. After performing an assessment, inspector produces a detailed list of security findings prioritized by level of severity. These findings can be reviewed directly or as part of detailed assessment reports which are available via the amazon inspector console or API.

How does it work?

* Create “assessment target”- the instances you want inspector to run on
* Install agents on EC2 instances
* Create “Assessment template”- the type of rules you want to run and for how long
* Perform “Assessment run”
* Review “Findings” against “rules”

Create a new role- gives inspector read only access of your instances in your account

Uses tags to define which instances inspector will check

Install the agent on your instances

Define an assessment target by using your tags

Define an assessment template

Rule packages

* Security best practices 1.0
* Runtime Behavior analysis 1.0
* Common vulnerabilities and exposures 1.1
* CIS operating system security configuration benchmark 1.0

Duration

* 15 minutes
* 1 hour
* 8 hours
* 12 hours
* 24 hours

Severity levels

* High
* Medium
* Low
* Informational

After a run you can download the findings report or a full report in the form of HTML of a PDF

You can run all the rules packages at the same time

AWS Trusted Advisor- an online resource to help you reduce cost, increase performance, and improve security by optimizing your AWS environment. Advisor will advise you on cost optimization, performance, security, fault tolerance, and not stated in the video but service limits for free tier account.

By default, if you do not get full trusted advisor and will only get a few security checks, one performance issue, and all service limit checks. To unlock full trusted advisor, you need to have a business or enterprise AWS account.

Core checks and recommendation.

The main default check is security groups that are opened to the world.

Inspector will be needed if you want to check security on your instances, but if it is trusted advisor security it will usually ask about Security Groups or MFA. Also remember that trusted advisor is more than just security.

Inspector will:

* Monitor the network, file system, and process activity within the specified target
* Compare what it ‘sees’ to security rules
* Report on security issues observed within target during run
* Report findings and advise remediation

It will not:

* Relive you of your responsibility under the shared responsibility model
* Perform miracles

Logging with AWS

Services:

* AWS CloudTrail- logs API calls- after the fact investigation
* AWS Config- logs configuration changes- point in time logging tool
* VPC Flow Logs- logs ENI traffic in and out of your VPC
* AWS CloudWatch Logs- logs anything on your EC2 instances or on prim servers that you can code in python. Usually around the application. You can also send AWS services like cloudtrail to cloudwatch logs. Logs performance of your AWS assets

Prevent unauthorized access:

* IAM users, groups, roles, and policies
* Amazon S3 bucket policies
* MFA

Ensure role-based access:

* IAM users, groups, roles and policies- security people should be able to look at your logs
* Amazon S3 bucket policies

Alerts when logs are created or fail:

* CloudTrail notifications
* AWS Config Rules

Alerts are specific, but don’t divulge detail:

* CloudTrail SNS notification only point to log file location

Log changes to system components:

* AWS Config Rules
* CloudTrail

Controls exist to prevent modification to logs

* IAM and S3 control and policies
* CloudTrail log file validation
* CloudTrail log file encryption

Storage of log files

* Logs are stored for at least one year
* Store logs for an org-defined period of time
* Store logs Realtime for resiliency
* Lifecycle policies

AWS Security Specialty Section 5 Notes

KMS- a managed service that makes it easy for you to create and control the encryption keys used to encrypt your data, and uses HSMs to protect the security of your keys.

IAM is global- you can use any user in any region

KMS is regional- a key can only be used in the region you created it in.

3 Users

1st user= system admin= policy is administrator access

2nd user= financial controller= S3FullAdminAccess and ReadOnlyAccess to the console. He gets a KMS key.

3rd user= account team but not CFO= S3FullAdminAccess and ReadOnlyAccess to the console

Key material orgins

* KMS= AWS provides the key material
* External= You provide the key material

Key Administrators- defines which user or role can admin the key through the KMS API

Key Deletion- you can allow key admins to delete the key, but if you delete the key and the key is encrypted you will lose that data.

Key Usage- the IAM user and roles that can use the key to encrypt and decrypt the data

You can give external AWS accounts the ability to use your key

* Just add the account ID you want to give access too

2nd user uploads the 3 files using the KMS key that is made for the 2nd user.

What if you make one object public?

Because it is encrypted people cannot view it over the web without the KMS key

What happens if we change that encryption to SSE-S3?

You can now access it over the web because amazon has the key.

In conclusion KMS has an added layer of security.

If you went to click open and or download the object that is encrypted in S3 with the user who has the KMS key you would be able to open or download it.

3rd user tries to access the same files without the KMS key.

The 1 file with SSE-S3 will still be accessible by this user, but the files with the KMS key cannot be accessed without the KMS key usage permissions. Same with open or downloading the file even though this user has full S3 access.

1st user with admin access but no KMS key usage permissions

The file with KMS will get denied if you try to access it thru S3 URL’s because you are trying to access it as a random user, but if you try to open or download it you can decrypt it without the key usage permissions

* Be careful who you give admin access too
* Use least privilege
* Admins can also change themselves to a key administrator or a key user if they want

1st user with system administrator policy access

You can no longer add yourself to a key as an administrator or a user. You can only add roles

If you go back to S3 you can also no longer open or download the KMS file.

SysAdmin access takes away some important privileges in terms of KMS

What happens if the 2nd user leaves the company and decides to delete the encryption key?

* You can only schedule a key deletion- deleting a key makes all data under that encryption key unrecoverable and it forces a minimum waiting period of 7 days and a max of 30
* You can also disable it- if you disable it that means any data encrypted with it will not be viewable until your enable it as the data stays encryption but the key is disabled. No matter which user
* If you let the key delete any data encrypted is unrecoverable

KMS Part 3

Create a new key with External Key Material Origin and check the box saying I understand the Security, Availability, and durability implications of using an imported key.

The key has been created but you cannot use it until you import key material.

* Download wrapping key and import token you are going to use to generate that key material
* Wrapping algorithm= RSAES\_OAEP\_SHA\_1 and download it
* Wrapping key and import token expire after 24 hours
* Click I am ready to upload my key material
* Download and unzip openSSL. Move your import parameter into your openSSL directory
* 3 file wrapping key, import token, and read me

Ready to upload:

* Upload the encrypted key material you generated
* Upload the import token
* Then you need to choose an expiration option and that can either be a date you specify or you can make it so key material does not expire
* Then click finish

Can you import and generate a new CMK with the same wrapping algorithm and the import token?

No, you cannot use somebody else’s import token to generate a new key or the same encrypted key material.

You cannot enable automatic key rotation for a CMK with imported key material, however you can manual rotate a CMK with imported material. To do so create a new CMK then change the CMK identifier to your new CMK you just created.

Ciphertext are not portable between CMKs. Data cannot be encrypted with a separate CMK only the CMK it is associated with.

If you use SSE-S3 it will generate a new key in KMS, but you cannot do anything with that key because AWS manage it.

With external key material origin, you can delete the key material immediately and bypass the 7 day wait, and that data will become unusable immediately. You will still need 7 days to delete the customer master key from the KMS screen.

KMS Part 4

KMS integrates with EBS, S3, Amazon Redshift, Amazon Elastic Transcoder, Amazon WorkMail, Amazon RDS, and other to make it simple to encrypt your data with encryption keys that you manage.

The customer master key:

* Alias
* Creation date
* Description
* Key state
* Key material (customer provided or AWS provided)
* Can never be exported

AWS managed CMK for each service that is integrated with AWS KMS

Customer-managed CMK that you generate by providing AWS with key material

Setup a customer master key:

* Create Alias and Description
* Choose material option
* Define key administrator permissions
  + IAM user/Roles that can admin the key through the KMS API
* Define key usage permissions
  + IAM users/roles that can use the key to encrypt and decrypt data

Why import your own key material?

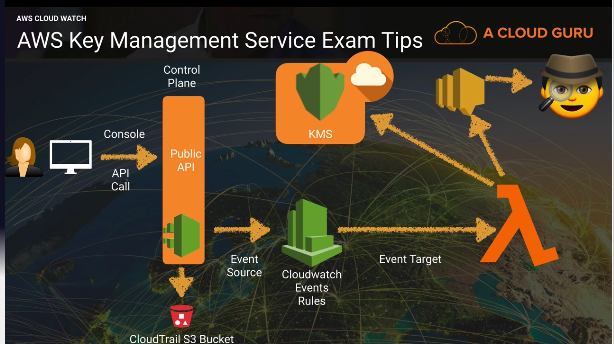
* Prove that randomness meets your requirements (Compliance)
* Extend your existing processes to AWS
* To be able to delete key material without a 7-30 day wait. Then be able to import them again
* To be resilient to AWS failure by storing keys outside AWS

How to import your own key material:

* Create a CMK with no key material
* Download a public key (wrapping key) and import token
* Encrypt the key material
* Import the key material

Considerations for imported key material:

* Availability and durability is different
* Secure key generation is up to you
* No automatic rotation
* Ciphertext are not portable between CMKs



Event driven security on KMS. Have cloudwatch events monitor KMS API via cloudtrail and if someone were to disable a key have that trigger an event. Send that event to the event target of lambda and have lambda reenable that key. Also have lambda send an SNS notification to your sysadmin.



Use config with KMS to monitor if anyone changes the key admin or usage permissions on a key or maybe to monitor any new key provisions. Have config send an SNS notification to your sysadmin.

Using KMS with EBS

By default, root device volumes are not encrypted

If you add additional volumes you can encrypt those, but those additional volumes you encrypt will be treated like SSE-S3 but for EBS. Fully managed by AWS.

Once you provision a volume using KMS that volume will always be linked to that key

Create a snapshot of the root device volume

Create an image of the snapshot

AMI options:

* You can modify the image permissions
* By default, all images you take will be private, but you can change them to public or share them with specific AWS accounts by providing the AWS account number

Copy the AMI and copy it to the exact same region

Check the encrypt target EBS snapshot and select your own KMS CMK

If you copy your AMI to a different region you need to make sure that the key you want to use has to be in the destination region.

You can use KMS to encrypt EBS volumes, but you cannot use KMS to generate a public key/private key to log in to EC2.

You can import public keys into EC2 key pairs, but we cannot use EC2 key pairs to encrypt EBS volumes, you must use KMS or 3rd party application/tools.

EC2 and key pairs

If you cd .ssh you will be able to go to the hidden directory of .ssh. this directory contains the authorized\_keys file. If you cat that file you will be able to see the public keys that are stored in that instance.

You can also get this using the meta data

Curl <http://169.254.169.254/latest/meta-data/public-keys/0/openssh-key/>

Add a new public key to an instance.

Add a full access S3 role for your instance

Ssh-keygen -t rsa

Enter a file name for it. This will generate 2 files mynewkp and mynewkp.pub.

You need to add the mynewkp.pub to the authorized\_keys file

Cat mynewkp.pub >> .ssh/authorized\_keys

The private key is the other file without the .pub

Copy they private key to an S3 bucket

Aws s3 cp mynewkp s3://bucketname

Download it from your bucket

Can you connect to your instance using that new key?

Yes you can

EC2 Key Pairs part 2

What happens if you delete all your key pairs in the AWS console? Will you still be able to access your instance?

Yes, you can. When you delete it in the console it does not affect your EC2 instances

Nothing will change in the meta data or the authorized\_keys file.

Can you add a new public key to your instance if you can no longer access an instance?

Create an AMI of you instance.

Launch that AMI instance and add a new key pair.

Will this append the new key on the authorized keys file or will it overwrite the whole file?

It will append the key and keep the existing keys. You should clean up this file so you can only use the brand new private key.

AWS Marketplace Security Products

You can purchase 3rd party AMI’s from the marketplace that are preconfigured

* You sort by OS
* Sort by billing
  + Free
  + Monthly
  + Annual
  + Bring your own license

Types of products:

* Firewalls
* Kali Linux
* CIS Red Hat Linux 7 benchmark

AWS WAF and Shield

WAF- a web application firewall that lets you monitor the HTTPS and HTTPS requests that are forwarded to Amazon cloudFront or an application load balancer. AWS WAF also lets you control access to your content.

You can configure conditions such as what ip addresses are allowed to make this request or what query string parameters need to be passed for the request to be allowed and then the application load balancer or cloudfront will either allow this content to be received or to give a HTTP 403 status code.

At its most basic level, AWS WAF allows 3 different behaviors:

* Allows all requests except the ones that you specify
* Block all requests except the ones that you specify
* Count the requests that match the properties that you specify

Additional protection against web attacks using conditions that you specify:

* IP addresses that requests originate from
* Country that requests originate from
* Values in request headers
* String that appear in request, either specific string or string that match regular expressions
* Presence of sql code that is likely to be malicious
* Presence of a script that is likely to be malicious

WAF integration:

* Cloudfronts distributions (global)
* Application load balancers (regional)

Associating a WAF with a Cloudfront distribution:

1. Go to your WAF and click rules
2. Click add association
3. Find the resource you want to add and click add

Manual IP block set- you can add IP addresses or IP addresses ranges with this option. You can have /8 /16 /24 /32

When you associate your cloudfront distribution with your WAF it redeploys the distribution so this can cause delays and may take 15 to 20 minutes to become active.

You can use WAF to protect web sites not hosted in AWS via CloudFront. Cloudfront supports custom origins outside of AWS.

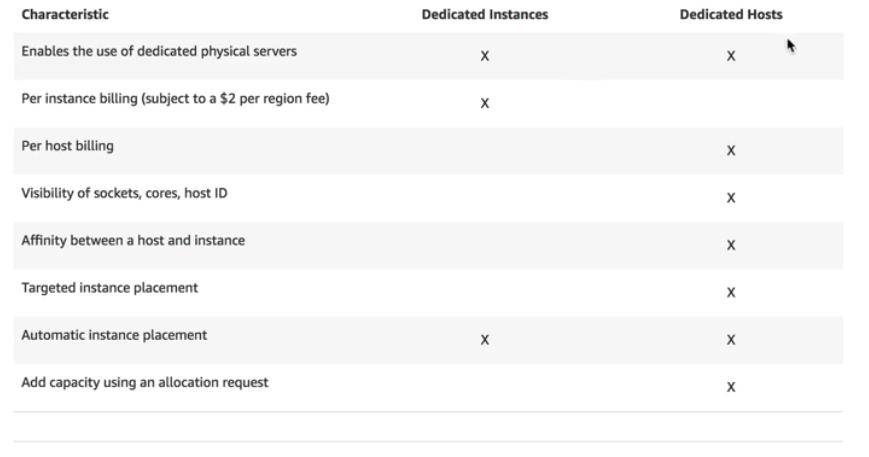
Dedicated Instances vs Dedicated Hosts

Dedicated Instance- Amazon EC2 instances that run in a VPC on hardware that is dedicated to a single customer. Your dedicated instances are physically isolated at the host hardware level from instances that belong to other AWS accounts.

Dedicated instances may share hardware with other instances from the same AWS account that are not dedicated instances.

Pay for dedicated instances on-demand, save up to 70% by purchasing reserved instances, or save up to 90% by purchasing spot instances.

Dedicated hosts- the main difference is that dedicated hosts give you additional visibility and control over how instances are placed on a physical server, and you can consistently deploy your instance to the same physical server over time. As a result, dedicated hosts enable you to use your existing server-bound software licenses and address corporate compliance and regulatory requirements.



Both have dedicated hardware to you. Single tenancy.

Dedicated instances are charged by the instance, dedicated hosts are charged by the host.

If you have specific regulatory requirements or licensing conditions, choose dedicated hosts.

Dedicated instances may share the same hardware with other AWS instances from the same account that are not dedicated

Dedicated hosts give you much better visibility in to things like sockets, cores, and host id.

AWS Hypervisors

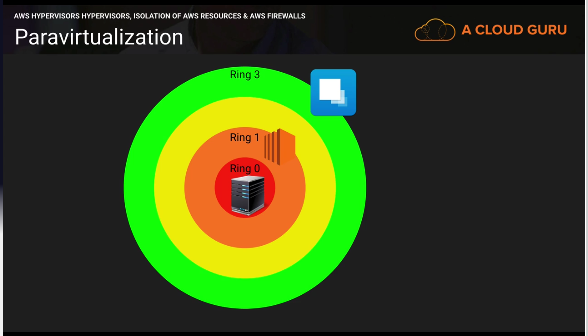
Hypervisor- a computer software, firmware or hardware that creates and runs virtual machines. A computer on which a hypervisor runs one or more virtual machines is called a host machine, and each virtual machine is called a guest machine.

EC2 currently runs on Xen Hypervisors. Xen hypervisors can have guest operating systems running either as Paravirtualization (PV) or using Hardware Virtual Machine (HVM).

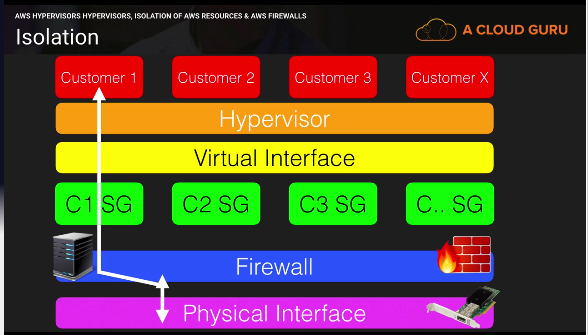
**HVM-** are fully virtualized. The VMs on top of the hypervisors are not aware that they are sharing processing time with other VMs.

**PV-** is a lighter form of virtualization and it used to be quicker (not anymore so use HVM whenever possible)

Windows instances can only be HVM where as Linux can be both PV and HVM

Paravirtualized guests rely on the hypervisor to provide support for operations that normally require privileged access, the guest OS has no elevated access to the CPU. The CPI provides four separate privilege modes: 0-3, called rings. Ring 0 is the most privileged and 3 is the least. The host OS executes in ring 0. However, rather than executing in ring 0 as most OS do, the guest OS runs in a lesser-privileged ring 1 and applications run on ring 3.

Isolation



Customers are completely isolated from one another.

Hypervisor Access:

Administrators with a business need to access the management plane are required to use MFA to gain access to purpose-build administration hosts. These administrative hosts are systems that are specifically designed, build, configured, and hardened to protect the management plane of the cloud. All such access is logged and audited. When an employee no longer has a business need to access the management plan the privileges and access to these hosts and relevant systems can be revoked.

Guest EC2 Access:

Virtual instances are completely controlled by you, the customer. You have full root access or administrative control over accounts, services, and applications. AWS does not have any access rights to your instances or the guest OS.

Memory Scrubbing:

EBS automatically resets every block of storage used by the customer, so that one customers data is never unintentionally exposed to another. Also memory allocated to guests is scrubbed by the hypervisors when it is unallocated to a guest. The memory is not returned to the pool of free memory available for new allocations until memory scrubbing is complete.

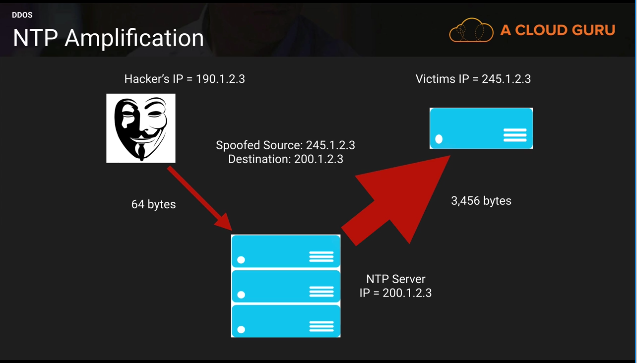
AWS Security Specialty Section 8 notes

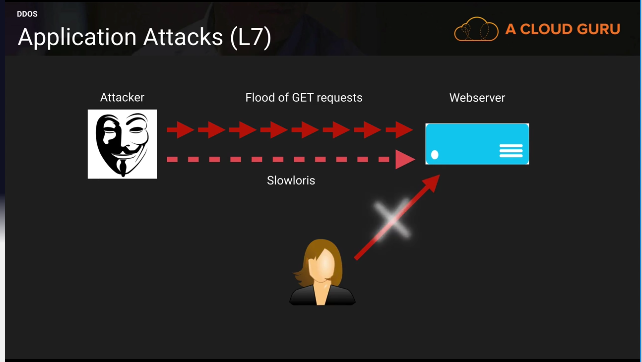
DDOS Overview

DDoS- A distributed Denial of Service attack is an attack that attempts to make your website or application unavailable to your end users.

This can be achieved by multiple mechanisms, such as packet floods, by using a combination of reflection and amplification techniques, or by using large botnets.

Amplification/Reflection Attacks- can include things such as NTP, SSDP, DNS, Chargen, SNMP attacks, and more. It is where an attacker may send a third-party server a request using a spoofed IP address. That server will then respond to that request with a greater payload than initially requested to the spoofed IP address.

This means that if the attacker sends a packet with a spoofed IP address of 64bytes the NTP server would respond with up to 3,456 bytes of traffic. Attackers can co-ordinate this and use multiple NTP servers a second to send legitimate NTP traffic to the target.



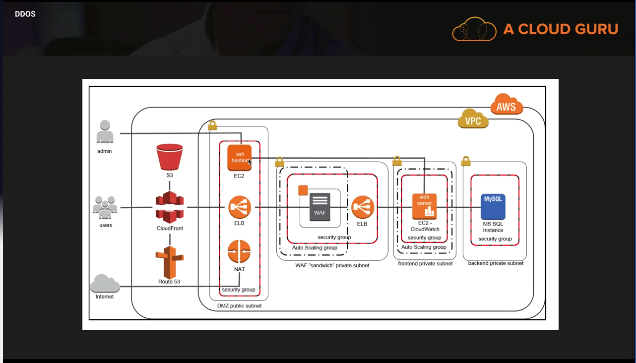
How to mitigate DDos attacks?

* Minimize the attack surface area
* Be ready to scale to absorb the attack
* Safeguard exposed resources
* Learn normal behavior
* Create a plan for attacks

Minimize the attack surface area

Some production environments have multiple entry points in to them. Perhaps they allow direct SSH or RDP access to their web servers/application and DB server for management.

This can be minimized by using a Bastion/Jump Box that only allows access to specific white listed IP addresses to these bastion servers and move the web, application, and DB servers to private subnets. By minimizing the attack surface area, you are limiting your exposure to just a few hardened entry points.



Be ready to scale to absorb the attack.

The key strategy behind a DDoS attack Is to bring your infrastructure to a breaking point. This strategy assumes one thing: that you cant scale to meet the attack.

The easiest way to defeat this strategy is to design your infrastructure to scale as, and when it is needed.

You can scale both horizontally and vertically.

Scaling Benefits:

* The attack is spread over a larger area
* Attackers then have to counter attack, taking up more of their resources
* Scaling buys you time to analyze the attack and to respond with the appropriate countermeasures
* Scaling has the added benefit of providing you with additional levels of redundancy

Safeguard Exposed Resources

In situations where you cannot eliminate internet entry points to your applications, you will need to take additional measures to restrict access and protect those entry points without interrupting legitimate end user traffic.

3 resources that can provide this control and flexibility are CloudFront, Route 53, and WAFs.

CloudFront:

* Geo Restriction/Blocking- Restrict access to users in specific countries (using whitelists or blacklists)
* Origin Access Identity- Restrict Access to your S3 bucket so that people can only access S3 using CloudFront URLs.

Route53:

* Alias Record Sets- you can use these to immediately redirect your traffic to an Amazon cloudFront distribution, or to a different ELB with higher capacity EC2 instances running WAFs or your own security tools. No DNS change, and no need to worry about propagation.
* Private DNS- Allows you to manage internal DNS names for you application resources (web servers, application servers, DBs) without exposing this information to the public internet

WAFs: DDoS attacks that happen at the application layer commonly target web applications with lower volumes of traffic compared to infrastructure attacks. To mitigate these types of attacks, you will want to include a WAF as part of your infrastructure.

* New AWS WAF service
* Find WAF’s in the AWS market place for specific needs

Learn normal behavior:

* Be aware of normal and unusual behavior
  + Know the different types of traffic and what normal levels of this traffic should be
  + Understand expected and unexpected resource spikes

What are the benefits?

* Allows you to spot abnormalities fast
* You can create alarms to alert you of abnormal behavior
* Helps you to collect forensic data to understand the attack

Create a plan for attacks:

* Having a plan in place before an attack ensures that
  + You’ve validated the design of your architecture
  + You understand the costs for your increased resiliency and already know what techniques to employ when you come under attack.
  + You know who to contact when an attack happens

AWS Shield:

* Free service that protects all AWS customers on ELB’s, CloudFront, and Route 53
* Protects against SYN/UDP floods, reflection attacks, and other layer3/layer 4 attacks
* Advanced provides enhanced protections for your apps running on ELB’, CloudFront, and Route 53 against larger more sophisticated attacks for only 3000 per month.
* Shield for layer 3 and layer 4 attacks. WAF’s for layer 7 attacks.

AWS Shield Advanced provides:

* Always on, flow-based monitoring of network traffic and active applications monitoring to provide near real time notification of DDoS attacks
* DDoS response team 24x7 to manage and mitigate applications layer DDoS attacks.
* Protects your AWS bill against higher fees due to ELB, CloudFront, and Route 53 usage spikes during a DDoS attack.

Technologies you use to mitigate a DDoS attack:

CloudFront, Route 53, ELBs, WAFs, Autoscaling, CloudWatch

WAF Integration

WAF integrates with both Application Load Balancers and CloudFront. It does not integrate with EC2 directly, nor Route 53 or any other services.

EC2 has been hacked. What should you do?

What steps should you take?

1. Stope the instance immediately
2. Take a snapshot of the EBS volume
3. Terminate the instance (ryan states in the video)
4. Deploy the instance in to a totally isolated environment. Isolated VPC, no internet access- ideally into a private subnet. Also monitor using VPC Flow Logs
5. Access the instance using an isolated forensic workstation. Wireshark or kali linux to investigate.
6. Read through the logs to figure out how it happened. (Windows Event Logs or Linux logs)

I’ve leaked my keys on GitHub accidentally

1. Go to IAM and find the user whose keys have been leaked
2. Navigate to the security credentials
3. First make it inactive
4. Make a new key
5. Then test the new key and delete the old one

For the root user to the same thing, but the keys are located under my security credentials. Also remember it is best practice to just delete your root keys as soon as you make your AWS account.

Reading CloudTrail Logs

Every API call will be logged into CloudTrail

CloudTrail logs are in JSON or key value pairs

Pen Testing AWS Market Place

Whenever you want to do pen testing you always need permission. Just a little note if you actually read this, there was a whiz labs question that stated there are now 2 ways to do pen testing and the second way is to use pre-approve tools in the AWS market place. Obviously on the test as it may not have been updated always choose the older way, but if you have a question that says choose 2 well now you know.

Allowed resources:

* EC2
* RDS
* Aurora
* CloudFront
* API Gateway
* Lambda
* LightSail
* DNS zone walking

RDS instance types:

The policy does not permit testing on m1.small, t1.micro, or t2.nano EC2 instances types. Same for EC2 instance types.

Other simulated events:

* Security simulations or security game days
* Support simulations or support game days
* War game simulations
* White cards
* Red team and blue team testing
* Disaster recovery simulations
* Other simulated events

Pen testing tools in the AWS marketplace

Kali Linux- gold standard for pen testing

AWS Certificate Manager

Need a Registered Route 53 Domain name to use Certificate Manager

You can extend your domain name for up to 9 years, but it will cost you money. You can also turn on auto renew and amazon will automatically renew the domain for you granted that you pay for it.

Certificate Manager- Makes it easy to provision, manage, deploy, and renew SSL/TLS certificates on the AWS platform.

* You can import your own SSL certificates into ACM
* You can never export an AWS certificate out of AWS

Validation method:

* DNS validation- choose this option if you have or can obtain permissions to modify the DNS configuration for the domains in your certificate request
* Email validation- choose this option if you do not have permissions or cannot obtain permissions to modify the DNS configuration for the domain in you certificate request

To validate:

* You need to add the CNAME record to your Route 53 configuration
* You can also choose create record in Route 53 if you want amazon to update your DNS configuration for you

Steps:

1. Go to your Route 53 and click create a record set
2. Choose CNAME and copy the name and paste it into the name text box
3. Copy the value and past it into the value of your record
4. Click create
5. Click continue on your ACM page and it will update yourself

ACM automatically renews unless:

* It does not auto renew imported certificates
* It does not auto renew certificates associated with Route 53 private hosted zones

ACM does not have to be used through Route 53. You can use it for like go daddy URLs and those will automatically renew.

ACM services that use it:

CloudFront and Application load balancers

To associate it with cloudfront:

* Go to edit your distribution
* Click custom SSL certificates
* And click on your certificate and save it

To associate it with ALB:

* Click on listeners and choose HTTPS
* Click next
* Choose a certificate from ACM and apply the certification name

If you want to apply them to both you need to do it separately. It still uses the same certificate, but it is just used at separate locations

You cannot export ACM certificates due to the fact that they are free, but you have to use them on AWS services essentially making them not free.

Perfect Forward Secrecy and ALBs

Perfect Forward Secrecy- a property of secure communication protocols in which compromises of long-term keys do not compromise past session keys.

* Old- all of your traffic to and from a load balancer would be recorded and you could take a compromised key and you could go in and decrypt old traffic to and from both points
* New- even if a key is compromised a year in the future people won’t be able to go back and decrypt that old traffic

If a private key is compromised in the future and they recorded all the traffic to and from the load balancer, they won’t be able to use that private key to decrypt later traffic in the future.

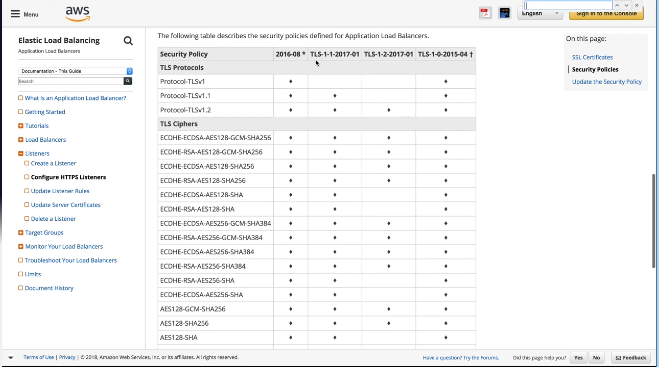
Securing your load balancer with perfect forward secrecy:

* Select Application load balancer
* Choose HTTPS
* Click next and choose select security policy
* Choose the policy with 2016-08

The TLS cipher that needs to be enables on the load balancer is the ECDHE cipher.

HTTP=80 HTTPS=443

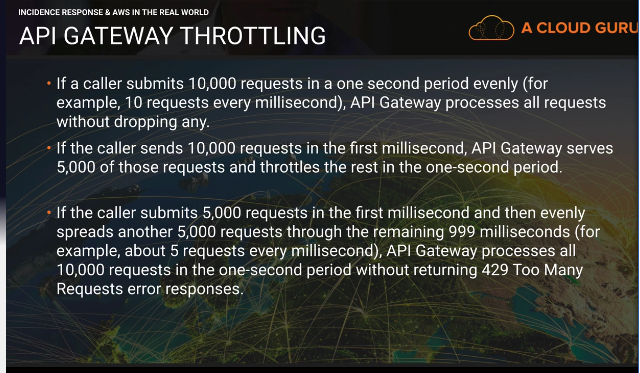
SSL/TLS=HTTPS



API Gateway – Throttling and Caching

API Gateway Throttling:

* To prevent your API from being overwhelmed by too many requests, amazon API gateway throttles request to you API
* When request submission exceeds the steady-state request rate and burst limits, API Gateway fails the limit-exceeding requests and returns 429 Too many requests error responses to the client
* By default, API gateway limits the steady-state request rate to 10,000 requests per second
* It limits the burst to 5,000 requests across all APIs within an AWS account.
* DDoS mitigation technique



API gateway is throttled by default

How to raise it:

* Raise a ticket with AWS to raise the rate limit and burst limit

API Gateway Caching:

What if you are getting the same request 10,000 times a second?

* You can enable API caching in Amazon API Gateway to cache your endpoints response
* With caching, you can reduce the number of calls made to your endpoint and also improve the latency of the requests to your API.
* When you enable caching for a stage, API Gateway caches responses from your endpoint for a specified time-to-live period, in seconds
* API gateway then responds to the request by looking up the endpoint response from the cache instead of making a request to your endpoint. The default TTL value for API caching is 300 seconds
* The max Is 3600 seconds
* TTL=0 means caching is disabled.

AWS Systems Manager Parameter Store

Systems Manager- A way to control your EC2 fleet at scale

Parameter Store- a way to store confidential information such as users, passwords, license keys etc. this information can be passed to EC2 as a bootstrap script, while maintaining the confidentiality of the information.

How to use it:

1. Create a parameter
2. Store sensitive data
3. Access parameters across services

Create parameter:

* Choose name- name has to be unique as this is what your resources will ref

Types:

* String- enter any data
* String list- enter multiple values separated with a comma
* Secure string- encrypts the data using KMS. Used to store sensitive information. Other system admins can ref the parameter, but won’t be able to see the data.

Services that can ref the parameter:

* EC2
* CloudFormation
* Lambda
* EC2 Run Command

AWS System Manager Run Command

EC2 Run Command

* You work as a systems administrator managing a large number of EC2 instances and on-premise systems
* You would like to automate common admin tasks and ad hoc configuring changes e.g. installing applications, applying the latest patches, joining new instances to a windows domain without having to login to each instance

What do you need to make it work?

* You need to create a role and you need to apply that role to the EC2 instances you want it to run on
* EC2 role for simple system manager
* The SSM agent needs to be installed on all your managed instances

How can you base what instances it runs on?

* You can base it off tags
* You can manually select the instances you want

Commands can be issued using AWS console, AWS CLI, AWS Tools for Windows Powershell, Systems Manager API or Amazon SDKs

You can use this service with your on-premise systems as well as EC2 instances

The commands and parameters are defined in a system manager document

Compliance in AWS

PCI DSS

ISO 27001

HIPPA