**AWS Certified SysOps Administrator- Associates**

**Part 1: Monitoring and Reporting: -**

**CloudWatch:** - Amazon CloudWatch is a monitoring service to monitor your AWS resources, as well as the applications that you run on AWS.

**What can CloudWatch do?**

**CloudWatch Can monitor thigs like:**

* **Compute** 
  + Autoscaling Groups
  + Elastic Load Balancers
  + Route53 Health Checks
* **Storage & Content Delivery**
  + EBS Volumes
  + Storage Gateways
  + CloudFront
* **Database & Analytics**
  + Dynamo DB
  + Elastic Cache Node
  + RDS Instances
  + Elastic MapReduce Job Flows
  + RedShift
* **Others**
  + SNS Topics
  + SQS Queues
  + Opsworks
  + CloudWatch Logs
  + Estimated Charges on your Bill

**CloudWatch and EC2**

**Host Level Metrics Consist of:**

* CPU
* Network
* Disk
* Status Checks

RAM Utilization is a custom metric! By default, EC@ monitoring is 5 Minute intervals, unless you enable detailed monitoring which will then make it 1-minute interval.

How Long are CloudWatch Metrics Stored?

You can retrieve data using the GetMetricsStatistics API or by using third party tools offered by AWS partners.

You can store your log data in CloudWatch Logs for as long as you want. By default, CloudWatch Logs will store your log data indefinitely. You can change the retention for each Log Group at any time.

You can retrieve data from any terminated EC@ or ELB instance after its termination.

Metric Granularity?

Its depends on the AWS service. Many default metrics for many default services are 1 minute, but it can be 3 or 5 minute depending on the service.

**Exam Tips:- For Custom metrics the minimum granularity that you can have is 1 minute.**

**CloudWatch Alarms: -**

You can create an alarm to monitor any Amazon CloudWatch metric in your account. This can include EC2 CPU Utilization, Elastic Load Balancer Latency or even the charges on your AWS Bill.

You can set the appropriate thresholds in which to trigger the alarms and also set what actions should be taken if an alarm state is reached.

**CloudWatch Exam Tips: -**

**Host Level Metrics consist of:**

* CPU
* Network
* Disk
* Status Check

**RAM Utilization – is a custom metric.**

**Custom Metrics – minimum granularity is 1 minute.**

**Terminated Instances –** You can retrieve data from any terminated EC2 or ELB instance after its termination. CloudWatch Logs by default are stored indefinitely.

**Metric Granularity**

* **1 minute for detailed monitoring**
* **5 minutes for standard monitoring**

CloudWatch can be used on premise – Not restricted to just AWS resources. Can be on premise too. Just need to download and install the SSM agent and CloudWatch agent.

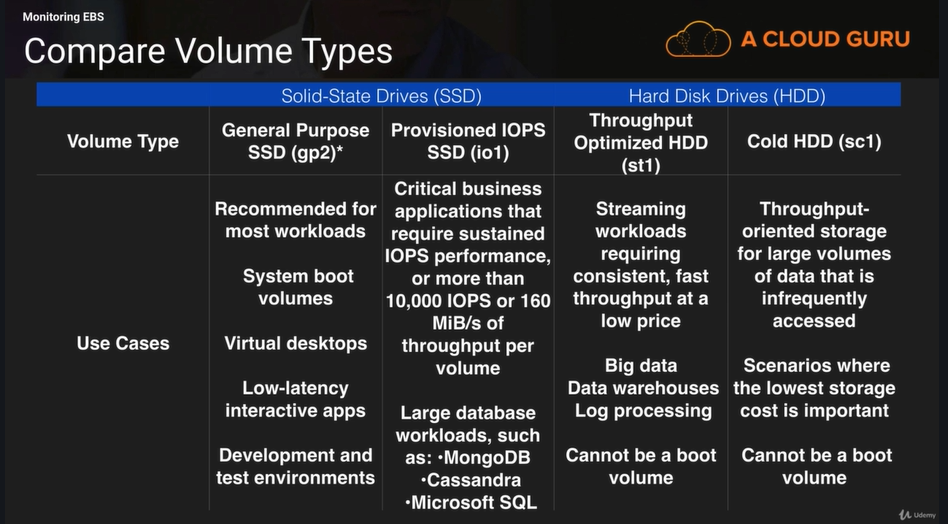
CloudWatch – Dashboards are multi-region and can display any widget to any region. To add the widget, change to the region that you need and then add the widget to the dashboard.

**Monitoring and Modifying EBS Volumes: -**

**EBS – Different Volume Types**

**4 Different Types of EBS Storage;**

* **General Purpose (SSD) – gp2**
* **Provisioned IOPS (SSD) – io1**
* **Throughput Optimized (HDD) – st1**
* **Cold (HDD) – sc1**





**IOPS & Volumes: -**

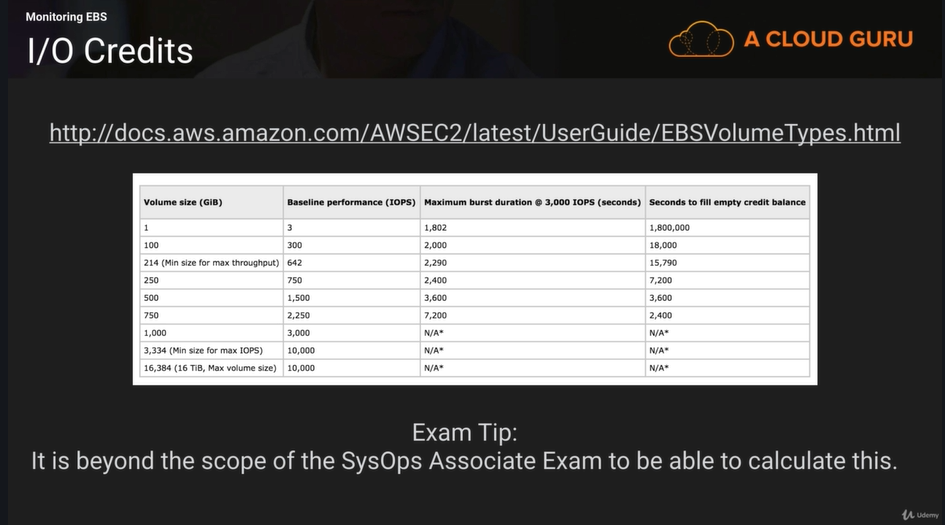
**General Purpose SSD volumes have a base of 3 IOPS per/GiB of volume size.**

* Maximum Volume size of 16,384 GiB
* Maximum IOPS size of 10,000 IOPS Total (after that you need to move to provisioned IOPS)

**I/O Credits**

When your volume requires more than the baseline performance I/O level, it simply uses I/O credits in the credit balance to burst to the required performance level, up to a maximum of 3,000 IOPS.

* Each volume receives an initial I/O credit balance of 5,400,000 I/O credits.
* This is enough to sustain the maximum burst performance of 3,000 IOPS for 30 minutes
* When you are not going over your provisioned IO level (i.e. bursting) you will be earning credits.

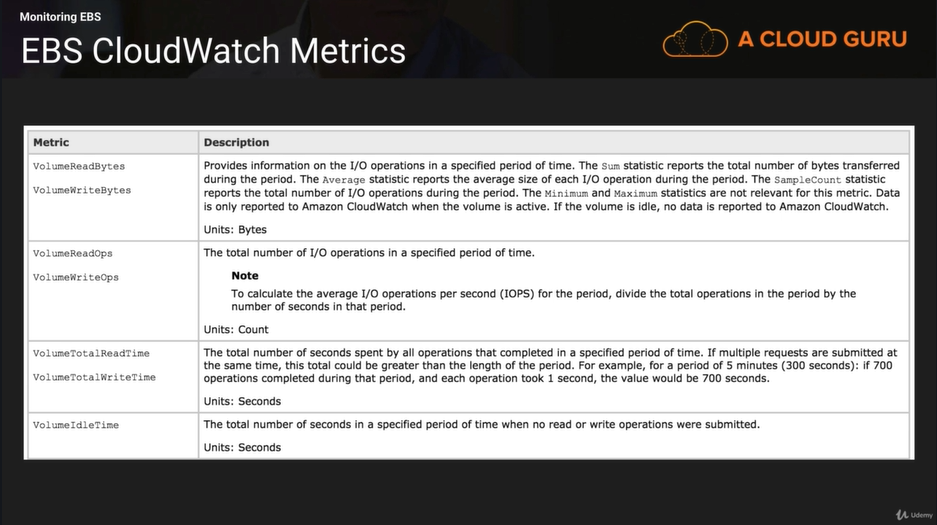


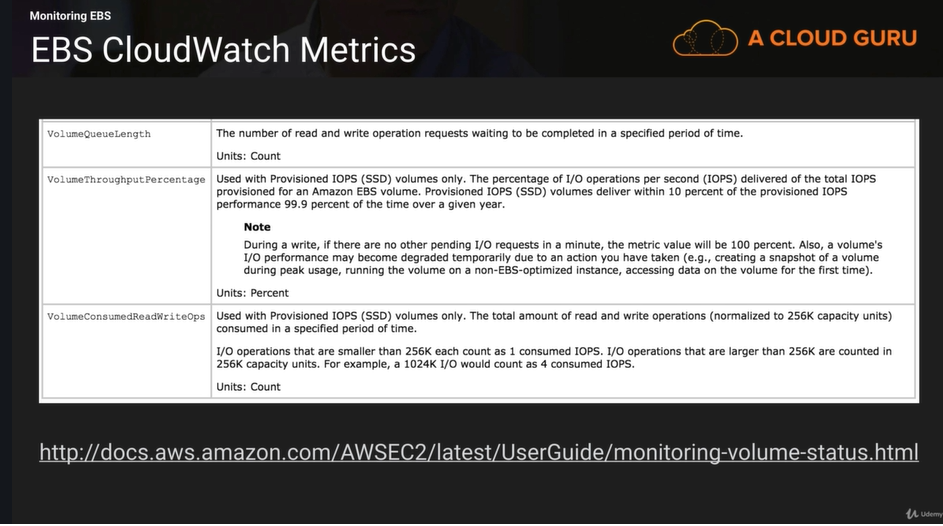
**Pre-Warming EBS Volumes: -**

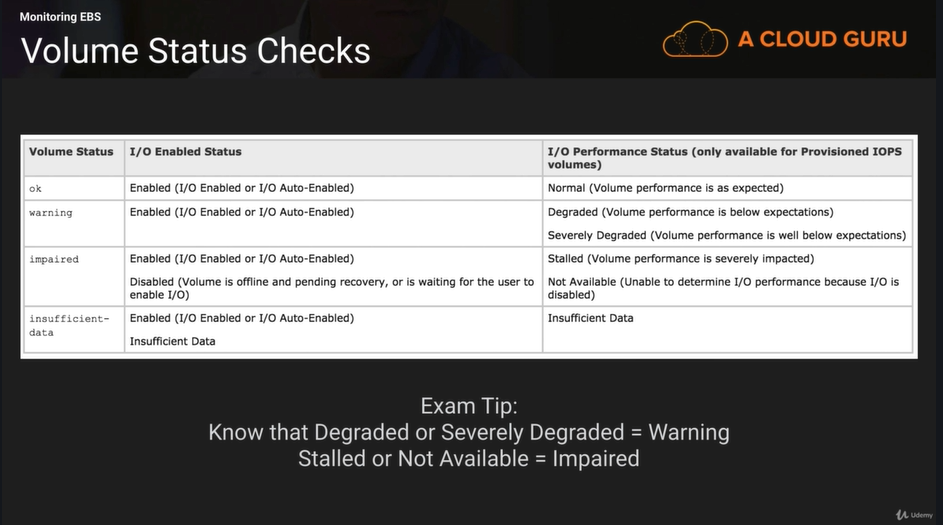
New EBS volumes receive their maximum performance the moment that they are available and do not require initialization (formerly known as pre-warming). However, storage blocks on volumes that were restored from snapshots must be initialized (pulled down from Amazon S3 and written to the volume) before you can access the block. This preliminary action takes time and can cause a significant increase in the latency of an I/O operation the first time each block is accessed. For most applications, amortizing this cost over the lifetime of the volume is acceptable.

Performance is restored after the data is accessed once.

You can avoid this performance hit in a production environment by reading from all the blocks on your volume before you use it; this process is called initialization. For a new volume created from a snapshot, you should read all the blocks that have data before using the volume.







**EBS Metrics with CloudWatch**

Volume Read Ops/Volume Write Ops = Total Number of IO Ops in a specific time period. So say 1000 in 1 minute = 1000/60 = IOPS.

Volume Queue Length = Number of read operations and write operation request waiting to be completed in a specific time period.

**Modifying EBS Volumes: -**

If your Amazon EBS volume is attached to a current generation EC2 instance type, you can increase its size, change its volume type, or (for an io1 volume) adjust its IOPS performance, all without detaching it. You can apply these changes to detached volumes as well.

* Issue the modification command (console or command line)
* Monitor the progress of the modification
* If the size of the volume was modified, extend the volume’s file system to take advantage of the increased storage capacity.

**Monitoring Elastic Load Balancer (ELB): -**

**3 Different Types of Elastic Load Balancers;**

* Application Load Balancer
* Network Load Balancer
* Classic Load Balancer

**ELB – Monitoring Types**

**4 Different Ways to Monitor Your Load Balancers;**

* CloudWatch Metrics
* Access Logs
* Request Tracing
* CloudTrail Logs

**CloudWatch Metrics: -**

Elastic Load Balancing publishes data points to Amazon CloudWatch for your load balancers and your targets. CloudWatch enables you to retrive statistics about those data points as an ordered set of time-series data, known as metrics. Think of a metric as a variable to monitor, and the data points as the values of that variable over time. For example, you can monitor the total number of healthy targets for a load balancer over a specified time period. Each data point has an associated time stamp and an optional unit of measurement.

**Access Logs: -**

Elastic Load Balancing provides access logs that capture detailed information about requests sent to your load balancer. Each log contains information such as the time the request was received, the client’s IP address, latencies, request paths and server responses. You can use these access logs to analyze traffic patterns and troubleshoot issues.

Access logging is an optional feature of Elastic Load Balancing that is disabled by default. After you enable access logging for your load balancer, Elastic Laod Balancing captures the logs and stores them in the Amazon S3 bucket that you specify as compressed files. You can disable access logging at any time.

**Access Logs – SUPER IMPORTANT**

Access Logs can store data where the EC2 instance has been deleted. For example, say you have a fleet of EC2 instances behind an auto scaling group. For some reason your application has a load of 5XX errors which is only reported by your end customers a couple of days after the event. If you aren’t storing the web server logs anywhere persistent, it is still possible to trace these 5XX errors using Access Logs which would be stored on S3.

**Request Tracing**

You can use request tracing to track HTTP request from clients to targets or other services. When the load balancer receives a request from a client, it adds or updates the X-Amzn-Trace-Id header before sending the request to the target. Any services or applications between the load balancer and the target can also add or update this header. **Available for Application Load Balancer Only.**

**CloudTrail**

You can use AWS CloudTrail to capture detailed information about the calls made to the Elastic Load Balancing API and store them as log files in Amazon S3. You can use these CloudTrail Logs to determine which calls were made, the source IP address where the call came from, who made the call, when the call was made, and so on.

**Monitoring Elastic Cache: -**

When it comes to monitoring our caching engines there are 4 important things to look at:

* CPU Utilization
* Swap Usage
* Evictions
* Concurrent Connections

**CPU Utilization: -**

**Memcached**

* Multi-Threaded
* Can handle loads of up to 90%. If it exceeds 90% and more to the cluster

**Redis**

* Not Multi-Threaded. To determine the point in which to scale, take 90 and divide by the number of cores.
* For example, suppose you are using a cache.m1.xlarge node, which has four cores. In this case, the threshold for CPU Utilization would be (90/4), or 22.5%

**Exam Tip: -**

**You will not have to calculate Redis CPU Utilization in the exam**

**What is SwapUsage?**

Put simply, swap usage is simply the amount of the Swap file that is used. The Swap File (or Paging File) is the amount of disk storage space reserved on disk if your computer runs out of RAM. Typically the size of the swap file = the size of the RAM. So if you have 4Gb of RAM, you will have a $GB Swap File.

**Swap Usage**

**Memcached**

* Should be around 0 most of the time and should not exceed 50MB.
* If this exceeds 50MB you should increase the

Memcached\_connections\_overhead parameter.

* The Memcached\_connections\_overhead defines the amount of memory to be reserved for Memcached connections and other miscellaneous overhead.

**What are Evictions?**

Think of evictions like tenants in an apartment building. There are number of empty apartments that slowly fill up with tenants. Eventually the apartment block is full, however more tenants need to be added.

An Eviction occurs when a new item is added, and old item must be removed due to lack of free space in the system.

**Evictions**

**Memcached**

* There is no recommended setting. Choose a threshold based off your application
* Either Scale UP (i.e. increase the memory of existing nodes) OR
* Scale Out (add more nodes)

**Redis**

* There is no recommended settings. Choose a threshold based off your application
* Only Scale Out (add read replicas)

**Exam Tip: -**

**This can be an exam question. Remember the different approaches between Memcached & Redis**

**Concurrent Connections:**

**Memcached & Redis**

* There is no recommended settings. Choose a threshold based off your application
* If there is a large and sustained spike in the number of concurrent connections this can either mean a large traffic spike OR your application is not releasing connections as it should be.

**Exam Tip:**

**This can be exam question. Remember to set an alarm on the number of concurrent connections for ElastiCache.**

**AWS Organizations: -**

AWS Organizations allows you to manage multiple AWS accounts at once. With Organizations, you can create groups of accounts and then apply policies to those groups.

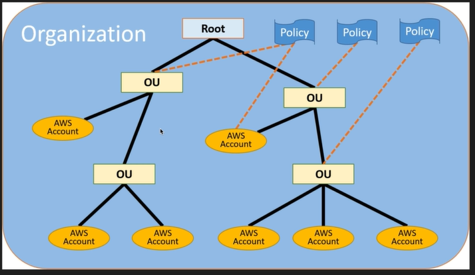
**AWS Organizations**

* Centrally Manage Policies Across Multiple AWS Accounts.
* Control Access To AWS Service
* Automate AWS Account Creation and Management
* Consolidate Billing Across Multiple AWS Accounts.

**What Does AWS Organizations do?**

**Central Managenment**

AWS Organizations allows you to manage multiple AWS accounts at once. You can create groups of accounts, and then attach policies to a group to ensur the correct policies are applied across the accounts. Organizations enables you to centrally manage policies across multiple accounts, without requiring custom scripts and manual processes.



**Automate AWS Account Creation**

You can use the AWS Organizations APIs to automate the creation and management of new AWS accounts. The Organizations APIs enable you to create new accounts programmatically, and to add the new accounts to a group. The policies attached to the group are automatically applied to the new account.

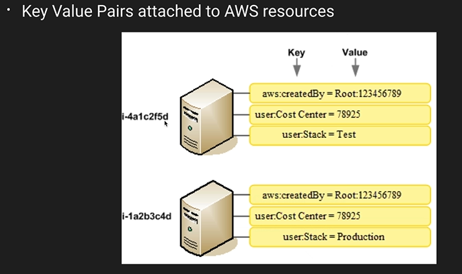
**Consolidated Billing**

AWS Organizations enables you to set up a single payment method for all the AWS accounts in your organization through consolidated billing. With consolidated billing, you can see a combined view of charges incurred by all your accounts, as well as take advantage of pricing benefits from aggregated usage, such as volume discounts for Amazon EC2 and Amazon S3.

**AWS Tagging and Resource Groups: -**

**What are Tags?**

* Key Value Pairs attached to AWS resources.
* Metadata (data about data)
* Tags can sometimes be inherited
  + Autoscaling, CloudFormation, and Elastic Beanstalk can create other resources



**What are Resource Groups?**

Resource groups make it easy to group your resources using the tags that are assigned to them. You can group resources that share one or more tags.

**Resource groups contain information such as;**

* Region
* Name
* Health Checks

**Specific information**

* For EC2 – Public & Private IP Addresses
* For ELB – Port Configurations
* For RDS – Database Engine etc.

**Exam Tips: -**

* **TAG Everything!**
* **Resource Groups are a way of grouping tags.**
* **You can use resource groups with AWS Systems Manager to automate tasks.**

**AWS Cost Explorer: -**

Cost Explorer is a tool that enables you to view and analyze your costs and usage. You can explorer your usage and costs using the main graph, the Cost Explorer cost and usage reports, or the Cost Explorer RI reports. You can view data for up to the last 13 months, forecast how much you’re likely to spend for the next three months, and get recommendations for what Reserved Instances to purchase. You can use Cost Explorer to identify areas that need further inquiry and see trends that you can use to understand your costs.

After you or AWS applies tags to your AWS resources (such as Amazon EC2 instances or Amazon S3 buckets) and you activate the tags in the Billing and Cost Management Console, AWS generates a cost allocation report as a comma-separated value (CSV file) with your usage and cost grouped by your active tags.

You can apply tags that represents business categories (such as cost centers, application names, or owners) to organize your costs across multiple services.

* **Use tags to tag your resources.**
* **Configure tags for cost centers (such as by department, employee id etc.)**
* **Activate cost allocation tags to track your costs by tags.**
* **Comes up frequently in the SysOps Associates Exam.**

**EC2 Pricing Models: -**

**On Demand: -**

* Users that want the low cost and flexibility of Amazon EC2 without any up-front payment or long-term commitment
* Applications with short term, spiky, or unpredictable workloads that cannot be interrupted.

**Reserved: -**

* Applications with steady state or predictable usage
* Applications that require reserved capacity
* Users able to make upfront payments to reduce their total computing costs even further
  + Standard RI’s (UP to 75% off on demand)
  + Convertible RI’s (UP to 54% off on demand) capacity to change the attributes of the RI as long as the exchange results in the creation of Reserved Instances of equal or greater value.
  + Scheduled RI’ s available to launch within the time windows you reserve. This option allows you to match your capacity reservation to a predictable recurring schedule that only requires a fraction of a day, a week, or a month.

**Spot Instances: -**

* Applications that have flexible start and end times
* Applications that are only feasible at very low compute prices
* Users with urgent computing needs for large amounts of additional capacity.

**Dedicated Hosts: -**

* Useful for regulatory requirements that may not support multi-tenant virtualization.
* Great for licensing which does not support multi-tenancy or cloud deployments.
* Can be purchased On-Demand (hourly)
* Can be purchased as a Reservation for up to 70% off the On-Demand price.

**Summary: -**

* **On Demand –** allow you to pay a fixed rate by the hour (or by the second) with no commitment.
* **Reserved –** Provide you with a capacity reservation and offer a significant discount on the hourly charge for an instance. 1 Year or 3 Year Terms.
* **Spot –** Enable you to bid whatever price you want for instance capacity, providing for even greater savings if your applications have flexible start and end times.
* **Dedicated Hosts** – physical EC2 server dedicated for your use. Dedicated Hosts can help you to reduce costs by allowing you to use existing server-bound software licenses.

1. **AWS Config: -**

“AWS Config is a fully managed service that provides you with an AWS resource inventory, configuration history, and configuration change notifications to enable security and governance.”

**Enables:**

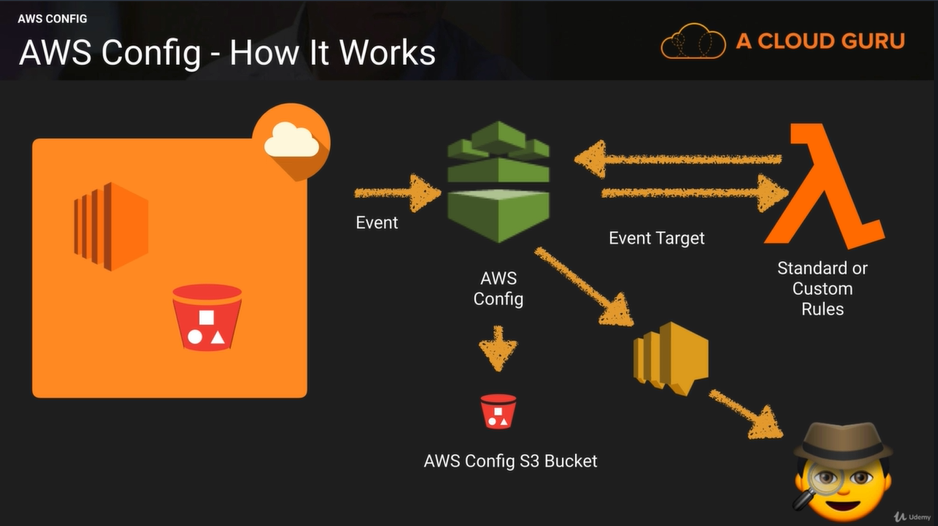
* Compliance Auditing
* Security Analysis
* Resource Tracking

**Provides:**

* Configuration snapshots and logs config changes of AWS resources
* Automated Compliance Checking

**Key Components:**

* Config Dashboard
* Config Rules
  + Managed
  + Custom
* Resources
* Settings



**Terminology:**

* Configuration Items
  + Point-in-time attributes of resource
* Configuration Snapshots
  + Collection of Config Items
* Configuration Stream
  + Stream of changed Config Items

**More Terminology:**

* Configuration History
  + 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 region
* Stores in S3
* Notifies SN

**What can we see:**

* Resource Type
* Resource ID
* Compliance
* Timeline
  + Configuration Details
  + Relationships
  + Changes
  + CloudTrail Events

**Compliance Checking:**

* Triger
  + Periodic
  + Configuration snapshot delivery
* Managed Rules
  + About 40 (at time recording)
  + Basic, but fundamental…

**Compliance Check:**

* Triger
  + Periodic
  + Configuration changes
* Managed Rules
  + About 40 (at time of recording)
  + Basic, but fundamental…

**Permissions needed for Config:**

* **AWS Config requires an IAM Role with**
  + 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-today use

**Exam Tips: -**

**Monitoring Config:**

* Use CloudTrail with Config deeper insight into resources.
* Use CloudTrail to monitor access to config, such as someone stopping the Config Recorder.

**Part 2 – Deployment and Provisioning: -**

1. **EC2 Launch Issue: -**

**Common reasons why EC2 instances may fail to launch:**

* **InstanceLimitExceeded Error**
  + You have reached the limit on the number of instances you can launch in a Region
  + AWS sets default limits on the number of instances you can run on a per-region basis – 20 by default
  + You can request an increase on a per-region basis
* **InsufficientInstanceCapacity Error**
  + AWS does not currently have enough available On-Demand capacity to service your request**.**
* **Options to resolve:** 
  + Wait a few minutes and try again
  + Request fewer instances
  + Select a different instance type
  + Try purchasing Reserved Instances instead
  + Submit a new request without specifying the Availability Zone**.**

**EC2 Launch Issues Exam Tips: -**

**Remember the two common reasons for an instance failing to launch:**

* **InstanceLimitExceeded Error (you have exceeded the default limit for number of instances you can launch in the Region)**
* **InsufficientInstanceCapacity Error (AWS does not currently have enough available On-demand capacity to service your request)**

1. **EBS Volumes and IOPS: -**

**EBS Volumes:**

* EBS (Elastic Block Store) allows you to create storage volumes and attach them to your EC2 instances.
* Can be used to create a file system, run a database, run an operating system, etc.
* We are going to focus on SSD backed storage – used for Operating Systems and databases which can be I/O intensive

**EBS SSD IOPS**

* **2 different variants of SSD:**
  + GP2 – General Purpose – boot volumes
  + IO1 – Provisioned IOPS – I/O intensive, NoSQL / Relational Databases, latency sensitive workloads
* IOPS (Input/ Output Operations per second) used to benchmark performance for SSD volumes
* IOPS capability is dependent on the size of your volume:
  + GP2 volumes: (minimum 100 IOPS) 3 IOPS/GB up to a maximum of 16,000 IOPS
  + IO1 volumes: 50 IOPS/GB to a maximum of 64,000 IOPS

**Hitting the IOPS Limit of your Volume**

**What happens if you are using gp2 and your workload exceeds the IOPS limit of the gp2 volume you have provisioned?**

* You will start to get your I/O request queuing
* Depending on your applications sensitivity to IOPS and latency. You may see your application becoming slow.

**What can you do about it?**

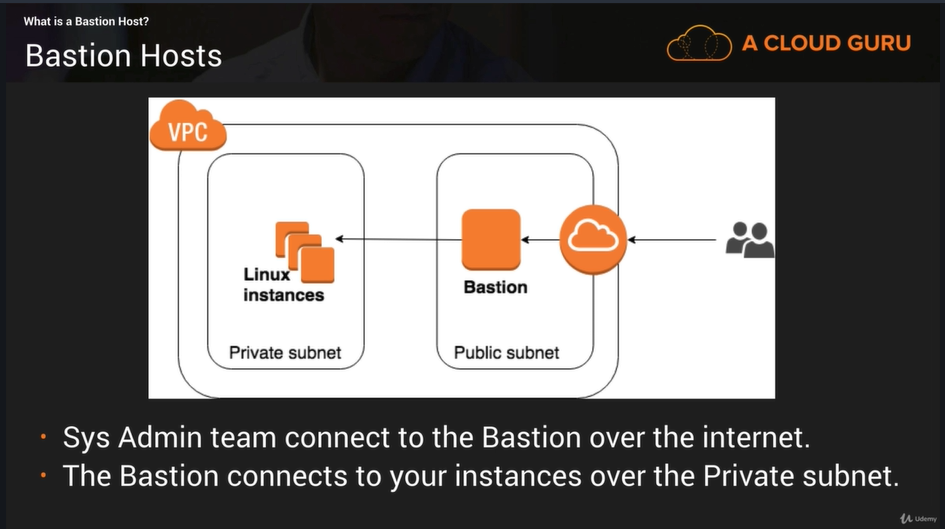
* **2 approaches to address hitting the IOPS limit:**
  + For gp2, you can increase the size of your volume – but if your volume is already 5.2TB or more, you will have already reached the 16,000 IOPS limit for gp2 volumes
  + If you need more than 16,000 IOPS, you will need to change your storage class to Provisioned IOPS

**Exam Tips:**

* IOPS (Input / Output Operations per second) used to benchmark performance for SSD volumes.
* IOPS is dependent on the size of your volume.
* If your workload is hitting the IOPS limit for your volume:
  + Increase the volume size – (only works if your gp2 volume is < 5.2TB)
  + Change to Provisioned IOPS if your gp2 volume is 5.2TB or grater, or you need more than 16k IOPS

1. **AWS Bastion Hosts: -**

* A Bastion Host is a host located in your Public Subnet.
* Allows you to connect to your EC2 instances using SSH or RDP
* You can log-in to the Bastion host over the internet, form your desktop.
* You then use the Bastion host to initiate an SSH / RDP session over the Private subnet to your EC2 instances in the Private subnet.
* This allows you to safely administer your EC2 instances without exposing them to the internet.



**Elastic Load Balancer: -**

**Pre-Warming Your Load Balancers:**

Imagine you are running a busy e-commerce website, and your marketing team is planning to announce a huge Black Friday sale.

Your sales director estimate that this will increase the traffic to your website by up to 10x once the sale is announced next week.

Pre-Warming will configure the ELB to the appropriate level of capacity, based on the traffic that you expect.

**AWS will need to know:**

1. **Start and end dates**
2. **Expected request rate per second**
3. **Total size of a typical request.**

**Load Balancers and Static IP Addresses:**

Application Load Balancers scale automatically to adapt to your workload.

However, this has the effect of changing the IP address which your clients connect to as new ALBs are brought into service.

Network Load Balancers solve this by creating static IP address in each subnet you enable so that keeps firewall rules simple – clients only need to enable access to a single IP address per subnet.

You don’t have to choose one or the other, you can get the benefit of both, by putting an ALB behind a NLB.

**ELB Exam Tips:**

* 3 Types of Load Balancers:
  + Application Load Balancers
  + Network Load Balancers
  + Classic Load Balancers
* Per-Warm Your Elastic Load Balancer if you expect a sudden and significant increase in traffic to your application.
* Static IP addresses can be provided by a Network Load Balancer, 1 per subnet.

**ELB Error Messages: -**

**Load Balancer Errors 4XX and 5XX**

* Classic & Application Load Balancers – by default the successful response code is 200
* Unsuccessful request will generate either a 4XX or 5XX error message
* 4XX message indicates that something has gone wrong on the client side
* 5XX messages relate to server-side errors.

**Example Error Codes – Client-Side Errors**

* 400 – Bad / Malformed request – e.g. the header is malformed
* 401 – Unauthorized – user access denied
* 403 - Forbidden – request is blocked by WAF access control list
* 460 – Load Balancer received an X-Forwarded-For request header with >30 IP addresses – similar to a malformed request

**Example Error Codes – Server-Side Errors**

* 500 – Internal server error – e.g. error on the load balancer
* 502 – Bad gateway – e.g. application server closed the connection or sent back a malformed response
* 503 – Service unavailable – no registered targets.
* 504 – Gateway timeout – e.g. application is not responding – problem with your web server, application server or database
* 561 – Unauthorized – received an error code from the D provider when trying to authenticate a user

**Exam Tips: -**

* 4XX – Client-side errors – Problem with the client request, identify the problem, fix the request and retry
* 5XX – Server-side errors – identify and fix the problem with your webserver / application / database / load balancer

**ELB CloudWatch Metrics: -**

* Elastic Load Balancers publish metrics to CloudWatch for the load balancer itself and for the back-end instances.
* Helps to verify that your system is performing as expected.
* You can create a CloudWatch alarm to perform a specific action – e.g. send an email when a metric reaches a pre-defined limit / range.
* Metrics are gathered at 60 second intervals.

**ELB CloudWatch Metrics – Overall Health**

* BackendConnectionErrors – number of unsuccessful connections to backend instances.
* HealthyHostCount – number of healthy instances registered
* UnHealthyHostCount – number of unhealthy instances
* HTTPCode\_Backend\_2XX, 3XX, 4XX, 5XX

**ELB CloudWatch Metrics – Performance Metrics**

* Latency – number of seconds taken for registered instance to respond / connect.
* RequestCount – number of requests completed / connections made during the specified interval (1 or 5 mins)
* SurgeQueueLength – number of pending requests, max queue size is 1024, additional requests will be rejected (Classic only)
* SpilloverCount – number of requests rejected because the surge queue is full (Classic Only)

**ELB CloudWatch Metrics – Exam Tips**

* Load Balancer Metrics are published to CloudWatch.
* You can create a CloudWatch alarm to send you a notification if a certain metric reaches a user-defined limit
* Remember the different types of metric that CloudWatch can monitor for
  + Metric for general health – e.g. HealthyHostCount / UnHealthyHostCount, HTTPCode\_Backend\_2XX
  + Metrics for performance – e.g. Latency, RequestCount, SurgeQueueLength, SpilloverCount (classic only) – high numbers can indicate a performance issue, need to scale infrastructure etc.

**AWS System Manager: -**

* System Manager (SSM) is a management tool which gives you visibility and control over your AWS infrastructure.
* Integrates with CloudWatch allowing you view your dashboards, view operational data & detect problems.
* Includes Run Command which automates operational tasks across resources – e.g. security patching, package installs.
* Organize your inventory, grouping resources together by application or environment – including on-premises systems.

**Run Command: -**

* Allows you to run pre-defined commands on one or more EC2 instances
* Stop, Restart, Terminate, Re-size instance
* Attach / Detach EBS volumes
* Create snapshots, Backup DynamoDB tables
* Apply patches and updates
* Run an Ansible playbook
* Run a shell script

**Exam Tips: -**

* Remember. System Manager is used to give visibility and control over your AWS infrastructure.
* Integrates with CloudWatch dashboards
* Allows you to organize your inventory and logically group resources together.
* Run Command enables to you to perform common operational tasks on group of instances simultaneously without needing log in to each one.

**Part 4 High Availability: -**

1. **Elasticity and Scalability: -**

**What is Elasticity?**

Think of elasticity as a rubber band. Elasticity allows you to stretch out and retract back your infrastructure, based on your demand.

Under this model you only pay for what you need.

Elasticity is used during a short time period, such as hours or days.

**What is Scalability?**

Scalability is used to talk about building out the infrastructure to meet your demands long term.

Scalability is used over longer time periods, such as weeks, days, months and years.

**AWS Services – Scalability vs Elasticity**

**EC2**

* **Scalability** – Increase Instance Sizes as required, using reserved instances
* **Elasticity –** Increase the number of EC2 instances, based on autoscaling

**DynamoDB**

* **Scalability –** Unlimited amount of storage
* **Elasticity –** Increase additional IOPS for additional spikes in traffic. Decrease that IOPS after the spike.

**RDS**

* **Scalability –** Increase instance size, e.g. from small to medium
* **Elasticity –** Not very elastic, can’t scale RDS on demand

**Aurora**

* **Scalability –** Modify the instance type
* **Elasticity –** Aurora Serverless

**Exam Tips:**

* **Elasticity –** Scale with Demand (Short Term)
* **Scalability –** Scale Out Infrastructure (Long Term)

1. **RDS and Multi-AZ Failover: -**

Multi AZ keeps a copy of your production database in a separate Availability Zone in case of a failure or disaster. AWS manage the failure from one AZ to Another automatically.

**It is not primarily used for improving performance. For performance improvement, you need Read Replicas.**

Multi-AZ deployments for the MySQL, Oracle, and PostgreSQL engines utilize synchronous physical replication to keep data on the standby up to date with the primary.

Multi-AZ deployments for the SQL Server engine use synchronous logical replication to achieve the same result, employing SQL Server-native Mirroring technology.

Both approaches safeguard your data in the event of a DB Instance failure or loss of an Availability Zone.

**RDS Multi-AZ Failover Advantages**

* High availability
* Backups are taken from secondary which avoids I/O suspension to the primary
* Restores are taken from secondary which avoids I/O suspensions to the primary

Exam Tip: you can force a failover from one AZ to another by rebooting your instance. This can be done through the AWS Management console or by using RebootDBInstance API call.

**Exam Tip:**

* RDS Multi-AZ Failover IS NOT A SCALING SOLUTION
* Amazon handles the failover for you. Done by updating the private DNS for the database endpoint.
* Backups & Restores are taken from the secondary Multi-AZ instance
* Read Replica’s are used to scale
* You can force a failover from one AZ to another by rebooting your instance. This can be done through the AWS Management console or by using RebootDBInstance API call.

**RDS - Read Replica: -**

Read Replicas make it easy to take advantage of supported engines built-in replication functionality to elastically scale out beyond the capacity constraints of a single DB Instance for read-heavy database workloads**.**

**When you use Read Replica’s?**

* Scaling beyond the computer or I/O capacity of a single DB Instance for read-heavy database workloads. This excess read traffic can be directed to one or more Read Replicas.
* Serving read traffic while the source DB Instance is unavailable. If your source DB Instance cannot take I/O request. (e.g. due to I/O suspension for backups or scheduled maintenance), you can direct read traffic to your Read Replica(s)
* Business reporting or data warehousing scenarios; you may want business reporting queries to run against a Read Replica, rather than your primary, production DB Instance.

**Supported Versions:**

* **MySQL**
* **PostgreSQL**
* **MariaDB**
  + For all 3 Amazon uses these engines native asynchronous replication to update the read replica
* **Aurora**
  + Aurora employs an SSD-backed virtualized storage layer purpose-built for database workloads. Amazon Aurora replicas share the same underlying storage as the source as the source instance, lowering costs and avoiding the need to copy data to the replica nodes.

**Creating Read Replicas:**

When creating a new Read Replica, AWS will take a snapshot of your database.

**If Multi-AZ is not enabled:**

* This snapshot will be of your primary database and can cause brief I/O suspension for around 1 minute.

**If Multi-AZ is enabled:**

* The snapshot will be of your secondary database and you will not experience any performance hits on your primary database.

**Connecting to Read Replica: -**

When a new read replica is created you will be able to connect to it using a new end point DNS address.

**Read Replica’s Can be Promoted:**

You can promote a read replica to its own standalone database. Doing this will break the replication link between the primary and secondary.

**Exam Tips:**

* You can have up to 5 read replicas for MySQL, PostgreSQL & MariaDB
* You can have read replicas in different REGION for all engines
* Replication is Asynchronous only, not synchronous
* Read Replica’s can be built off Multi-AZ databases
* Read Replica themselves can now be Multi-AZ
* You can have Read Replica’s of Read Replica’s beware of latency
* DB Snapshots and Automated backups cannot be taken of Read Replicas
* Key Metric too look for is REPLOCA LAG
* **KNOW THE DIFFRENCE BETWEEN READ REPLICAS AND MULTI-AZ!**

**Which Version of RDS: -**

* Using AWS Console
* Using CLI – aws rds describe-db-instances --region

**Steps to Encrypt RDS Snapshots:**

* Take a snapshot of existing RDS instance.
* Copy the snapshot to the same/different region
* Encrypt the copy during the copy process.
* Restore the snap.

**How to Share Encrypted Snapshots Between Accounts: -**

You can share DB snapshots that have been encrypted “at rest” using the AES-256 encryption algorithm. To do this:

* Create a CUSTOM KMS Encryption key.
* Create an RDS snapshot using the custom key.
* Share the CUSTOM AWS KMS encryption key that was used to encrypt the snapshot.
* Use the AWS Management Console, AWS CLI, or Amazon RDS API to share the encrypted snapshot with the other accounts.

**Restrictions:**

**These restrictions apply to sharing encrypted snapshots:**

* You can’t share Oracle or Microsoft SQL Server snapshots that are encrypted using Transparent Data Encryption (TDE)
* You can’t share a snapshot that has been encrypted using the default AWS KMS encryption key of the AWS account that shared the snapshot.

**Exam Tips:**

**You can share DB snapshots that have been encrypted “at rest” using the AES-256 encryption algorithm.**

* Share the CUSTOM AWS KMS encryption key that was used encrypt the snapshot.
* Use the AWS Management Console, AWS CLI, or Amazon RDS API to share the encrypted snapshot with the other accounts.
* You can’t share a snapshot that has been encrypted using the default AWS KMS encryption key of the AWS account that shared the snapshot.

**Maintenance Windows:**

**Services with maintenance windows include:**

* RDS
* ElastiCache
* RedShift
* DynamoDB DAX
* Neptune
* Amazon DocumentDB

**Services without maintenance Window**

* EC2
* Lambda
* Amazon QLDB

**What is ElastiCache:**

ElastiCache is a web service that makes it easy to deploy, operate, and scale an in-memory cache in the cloud. The service improves the performance of web applications by allowing you to retrieve information from fast, managed, in-memory caches, instead of relying entirely on slower disk-based databases.

Amazon ElastiCache can be used to significantly improve latency and throughput for many read-heavy applications workloads (such as social networking, gaming, media sharing and Q&A portals) or compute-intensive workloads (such as recommendation engine).

Caching improves application performance by storing critical pieces of data in memory for low-latency access. Cached information may include the results of I/O-intensive database queries or the results of computationally intensive calculations.

**Types of ElastiCache:**

* **Memcached:** 
  + A widely adopted memory object caching system. ElastiCache is protocol compliant with Memcached, so popular tools that you use today with existing Memcached environments will work seamlessly with the service.
* **Redis:**
  + A popular open-source in-memory key-value store that supports data structure such as sorted sets and lists. ElastiCache supports Master / Slave replication and Multi-AZ which can be used to achieve cross AZ redundancy.

**Exam Tips:**

**When it comes to monitoring our caching engines there are 4 important things to look at:**

* CPU Utilization
* Swap Usage
* Evictions
* Concurrent Connections

Typically, you will be given a scenario where a database is under a lot of stress/load. You may be asked which service you should use to alleviate this.

ElastiCache is a good choice if your database is particularly read-heavy and not prone to frequent changing.

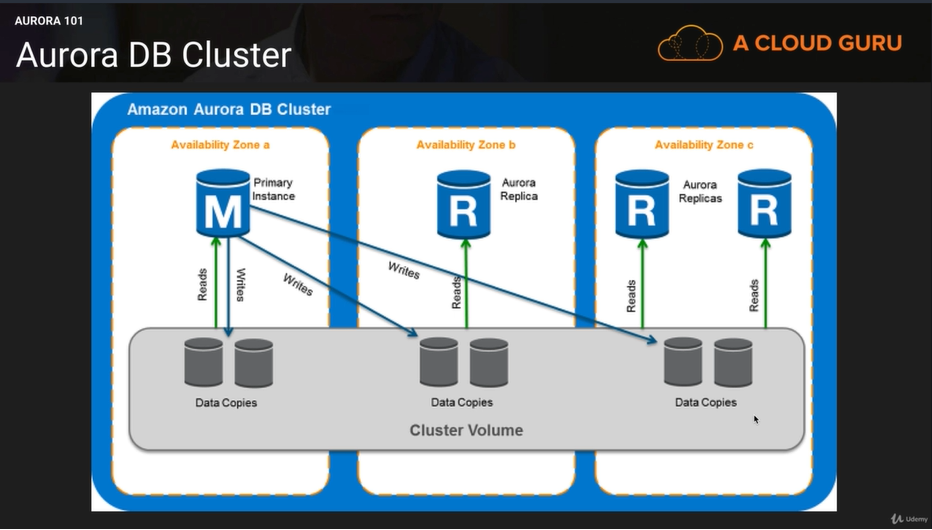
Redshift is a good answer if the reason your database is feeling stress is because management keep running OLAP transactions on it etc.

**What is Aurora?**

Amazon Aurora is a MySQL/PostgreSQL compatible, relational database engine that combines the speed and availability of high-end commercial databases with the simplicity and cost-effectiveness of open source databases. Amazon Aurora provides up to five times better performance than MySQL (and three times better performance than PostgreSQL) at a price point one tenth that of a commercial database while delivering similar performance and availability.

**Aurora Scaling: -**

* Start with 10GB, Scale in 10GB increments to 64TB (Storage Autoscaling)
* Compute resources can scale up to 64vCPUs and 488GiB of Memory
* 2 copies of your data is contained in each availability zone, with minimum of 3 availability zones. 6 copies of your data.
* Aurora is designed to transparently handle the loss of up to two copies of data without affecting database write availability and up to three copies without affecting read availability.
* Aurora storage is also self-healing. Data blocks and disk are continuously scanned for errors and repaired automatically.



**Aurora Replicas:**

**Two types of Replicas are available:**

* Aurora Replicas (currently 15)
* MySQL Read Replicas (currently 15)

**Aurora 100% CPU Utilization?**

* Is it Writes causing the issue? If so Scale Up (increase instance size)
* Is it Reads causing the issue? If so Scale Out (increase the number of read replicas).

**Aurora Serverless: -**

Aurora Serverless is an on-demand, auto-scaling configuration for Aurora (MySQL-compatible edition) where the database will automatically start-up, shut down, and scale up or down capacity based on your application’s needs.

You pay on a per-second basis for the database capacity you use when the database is active, and you can migrate between standard and serverless configurations with a few clicks in the Amazon RDS Management Console.

**Aurora Exam Tips: -**

**Aurora Comes in Two Flavors**

* Aurora
* Aurora Serverless

**Redundancy –** Two copies of your data is contained in 3 separate Availability Zones with a total of 6 copies.

**Storage is Self-Healing** – Data blocks and disks are continuously scanned for errors and repaired automatically.

**Aurora –** Encryption at rest is turned on by default. Once Encryption is turned on, all read replicas will be encrypted.

**Failover –** Failover is defined by Tiers. The lower the tier the higher the priority with Tier 0 being the highest priority available.

**Cross Region Replicas –** Creating a new cross region replica will also create a new Aurora cluster in the target region. If the replication is disrupted, you will have to set up again. It is recommended that you select “Multi-AZ Deployment” to ensure high availability for the target cluster.

**Instances not launching in to Autoscaling Groups?**

Below is a list of things to look for if your instances are not launching in to an autoscaling groups;

* Associated Key Pair does not exist
* Security group does not exist
* Autoscaling config is not working correctly
* Autoscaling group not found
* Instance type specified is not supported in the AZ
* AZ is no longer supported
* Invalid EBS device mapping
* Autoscaling service is not enabled on your account
* Attempting to attach and EBS block devoice to an instance-store AMI

**AWS CloudFront**

**CloudFront – Key Terminology**

* Edge Location – This is the location where content will be cached. This is separate to an AWS Region/AZ
* Origin – This is the origin of all the files that the CDN will distribute. This can be an S3 Bucket, an EC2 Instance, an Elastic Load Balancer, or Route53.
* Distribution – This is name given the CDN which consists of a collection of Edge Locations.

**CloudFront in Action: -**

Amazon CloudFront can be used to deliver your entire website, including dynamic, static, streaming, and interactive content using a global network of edge locations. Request for your content are automatically routed to the nearest edge location, so content is delivered with the best possible performance.

**Purpose of CloudFront: -**

One of the purpose of using CloudFront is to reduce the number of request that your origin server must respond to directly. This reduce the load on your origin server and reduce latency because more objects are served from CloudFront edge locations, which are closer to your users.

**Cache Hit Ratios: -**

The more request that CloudFront is able to serve from edge locations, the better it works.

The ratio of request served from edge locations (rather than the origin) is know as the cache hit ratio. The more requests from edge locations, the better the performance.

You can view the percentage of viewer request that are hits, misses, and errors in the CloudFront console.

**Maximize Cache Hit Ratio: -**

**The following strategies can maximize you cache hit ratios**

* Specifying How Long CloudFront Caches Your Objects
* Caching Based on Query String Parameter
* Caching Based on Cookie Headers
* Remove Accept-Encoding Header When Compression is Not Needed
* Serving Media Content by Using HTTP

**Specifying How Long CloudFront Caches Your Objects: -**

To increase your cache hit ratio, you can configure your origin to add a Cache-Control max-age directive to your objects and specify the longest practical value for max-age. The shorter the cache duration, the more frequently CloudFront forwards another request to your origin to determine whether the objects have changed and, if so, to get the latest version.

**Caching Based on Query String Parameters:**

**Query String Parameters**

<http://www.example.com?id=a1>

http://www.example.com?id=A1

http://www.example.com?Id=a1

<http://www.example.com?ID=a1>

Query String Parameter is CASE SENSITIVE. Ensure your application uses consistent variables.

**Caching Based on Cookie Values:**

Create separate cache behaviors for static and dynamic content and configure CloudFront to forward cookies to your origin only for dynamic content.

For example, suppose you have just one cache behavior for your distribution and that you’re using the distribution both for dynamic content, such as .js files. And for .css files that rarely change. CloudFront caches separate versions of your .css files based on cookie values, so each CloudFront edge location forwards a request to your origin for every new cookie value or combination of cookie values.

If you create a cache behavior for which the path pattern is \*.css and for which CloudFront doesn’t cache based on cookie values, then CloudFront forwards requests for .css files to your origin only for the first request that an edge location receives for a given .css file and for the first request after a .css file expires.

**Caching Based on Request Header: -**

If you configure CloudFront to cache based on request headers, you can improve caching if you Configure CloudFront to forwarding and caching based on all headers.

Also try to avoid caching based on request headers that have large numbers of unique values.

By default when CloudFront receives a request, it checks the value of the Accept-Encoding header. If the value of the header contains gzip, then CloudFront adds the header and value gzip-Accept-Encoding: gzip-to the cache key, and then forwards it to the origin. This behavior ensures that CloudFront server either an object or a compressed version of the object, based on the value of the Accept-Encoding header.

If compression is not enabled-because the origin doesn’t support it, CloudFront doesn’t support it, or content is not compressible – you can increase the cache hit ratio by specifying different behavior.

**Serving Media Content by Using HTTP: -**

You can use CloudFront to deliver on-demand video or live streaming video using any HTTP origin. One way you can set up video workflows in the cloud is by using CloudFront together with AWS Media Services.

**Section 5: - Storage and Data Management**

**What is S3?**

S3 provides developers and IT teams with secure, durable, highly scalable object storage. Amazon S3 is easy to use, with a simple web services interface to store and retrieve any amount of data from anywhere on the web.

* S3 is a safe place to store your files.
* It is object-based storage
* The data is spread across multiple devices and facilities.

**S3 – The Basics**

* S3 is Object-based – i.e. allows you to upload files.
* Files can be from 0 Bytes to 5TB.
* There is unlimited storage.
* Files are stored in Bucket (similar to a folder)
* S3 is a universal namespace. That is, names must be unique globally.
* <https://s3-eu-west-1.amazonaws.com/test-bucket>
* When you upload a files to S3, you will receive a HTTP 200 code if the upload was successful.

**Data Consistency Model for S3**

* Read after Write consistency for PUTS of new Objects
* Eventual Consistency for overwrite PUTS and DELETES (can take some time to propagate)

**S3 is A Simple Key-value Store**

**S3 is Object based. Objects consist of the following:**

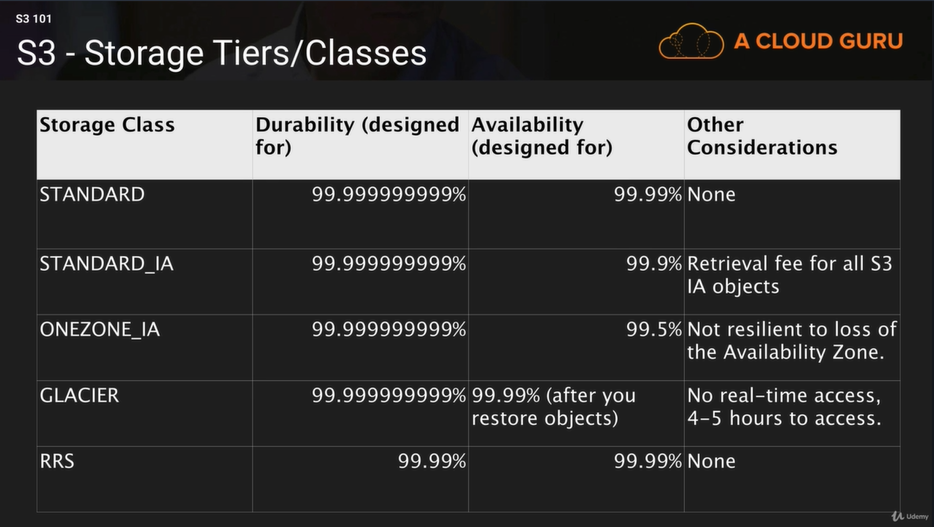
* Key (This is simply the name of the object)
* Value (This is simply the data, which is made up of a sequence of bytes).
* Version ID (Important for versioning)
* Metadata (Data about data you are storing)
* Sub resources – bucket-specific configuration:
  + Bucket Policies, Access Control Lists,
  + Cross Origin Resource Sharing (CORS)
  + Transfer Acceleration

**S3 – The Basics**

* Built for 99.99% availability for the S3 platform
* Amazon Guarantee 99.9% availability
* Amazon guarantees 99.999999999% durability for S3 information. (Remember 11 x 9s)
* Tired Storage Available
* Lifecycle Management
* Versioning
* Encryption
* Secure your data – Access Control Lists and Bucket Policies

**S3 – Storage Tiers/Classes: -**

* S3: 99.99% availability, 99.999999999% durability, stored redundantly across multiple devices in multiple facilities, and is designed to sustain the loss of 2 facilities concurrently.
* S3 – IA (Infrequently Accessed): For data that is accessed less frequently but requires rapid access when needed. Lower fee than S3, but you are charged a retrial fee
* S3 – One Zone IA: same as IA however data is stored in single Availability Zone only, still 99.999999999% durability, but only 99.5% availability. Cost is 20% less than regular S3-IA.
* Reduced Redundancy Storage: Designed to provide 99.99% durability and 99.99% availability of objects over a given year. Used for data that can be recreated if lost, e.g. thumbnails. (Starting to disappear from AWS documentation but may still feature in exam)
* Glacier: Very cheap but used for archival only. Optimized for data that is infrequently accessed and it take 3-5 hours to restore from glacier.



**S3 – intelligent Tiering: -**

* Unknown or unpredictable access patterns
* 2 tiers-frequent and infrequent access
* Automatically moves your data to most cost-effective tier based on how frequently you access each object
* 99.999999999% durability
* 99.9% availability given a year
* Optimize cost
* No fees for accessing your data but a small monthly fee for monitoring/automation $0.0025 per 1,000 objects

**S3 – Charges: -**

**Charged for:**

* Storage per GB
* Requests (Get, Put, Copy, etc.)
* Storage Management Pricing
  + Data transferred out of S3
* Transfer Acceleration
  + Use of CloudFront to optimize transfers

**Exam Tips: -**

* Remember that S3 is Object-based: i.e. allows you to upload files. Object-based storage only (for files.)
* Not suitable to install an operating system or running a database on.
* Files can be from 0 bytes to 5 TB
* There is unlimited storage.
* Files are stored in Buckets
* S3 is a universal namespace. That is, names must be unique globally.
* <https://s3-eu-west-1.amazonaws.com/test-bucket>
* **S3 Storage Classes/ Tires**
  + S3 (durable, immediately available, frequently accessed)
  + S3 – IA (durable, immediately, available, infrequently accessed)
  + S3 – One Zone IA: Same as IA. However, data is stored in a single Availability Zone only
  + S3 – Reduced Redundancy Storage (data that is easily reproducible, such as thumbnails, etc.)
  + Glacier – Archived data, where you can wait 3 -5 hours before accessing.
* **Remember the core fundamentals of an S3 object:** 
  + Key (name)
  + Value (data)
  + Version Id
  + Metadata
  + Sub resources – bucket specific configurations:
    - Bucket policies, Access Control Lists,
    - Cross Origin Resource Sharing (CORS)
    - Transfer accelerations
* Successful uploads will generate a HTTP 200 status code. When you use the CLI or API
* Make sur you read the S3 FAQ: <https://aws.amazon.com/s3/faqs>

**S3 Lifecycle Policies**

* You can use S3 Lifecycle Management to manage your objects so that they are stored using the most cost effective S3 option throughput their lifecycle
* You can configure lifecycle rules to tell S3 to transition objects to less expensive storage classes, archive them or delete them.
* Best for objects which have a well-defined lifecycle. E.g. log files. Which may not be useful once they reach a certain age.
* Transition objects to an IA storage class 90 days after you created them – e.g. transaction log files.
* Archive objects to Glacier 1 year after creating them.
* Configure objects to expire 1 year after creating them – S3 will auto delete expired objects on your behalf.
* E.g. buckets with Server Access Logging can accumulate many log files over time.
* S3 Lifecycle policies are used to ensure you are using the most cost effective option to store your objects in S3
* Lifecycle rules are based on objects creation date.
* S3 can transaction your objects to Infrequently Accessed Storage or to Glacier based on the rules you configure.
* You can also set an expiry date for objects you want S3 to delete after a certain time period has elapsed.

**MFA Delete & S3 Versioning: -**

* S3 Versioning enables you to revert to older versions of S3 objects.
* Multiple versions of an objects are stored in the same bucket.
* Versioning also protects you from accidental / malicious deletes.
* With versioning enabled, a DELETE action doesn’t delete the objects version, but applies a delete marker instead.
* To permanently delete, provide the object.
* MFA Delete provides an additional layer of protection to S3 Versioning
* Once enabled, MFA Delete will enforce 2 things:
  + You will need a valid code from your MFA device in order to permanently delete n object version
  + MFA also needed to suspend / reactivate versioning on an S3 bucket.

**S3 MFA Delete Exam Tips:**

* Use MFA Delete to protect against accidental or malicious deletions of your version-controlled S3 buckets.
* Remember the 2 things MFA Delete enforces:
  + You need a valid code from your MFA device to enable permanent deletion of an S3 object,
  + AND to suspend or reactivate versioning on the S3 bucket.

**Encryption**

* **In Transit:** 
  + **SSL/TLS**
* **At Rest:**
  + **Server Side Encryption:**
    - **S3 Managed Keys – SSE-S3**
    - **AWS Key Management Service, Managed Keys, SSE-KMS**
    - **Server Side Encryption with Customer Provided Keys -SSE-C**
* **Client Side Encryption**

**Enforcing Encryption on S3 Buckets: -**

* Every time a file is uploaded to S3, a PUT request is initiated.
* This is what a PUT request looks like:

**PUT / myFile HTTP/1.1**

**Host: myBucket.s3.amazonaws.com**

**Date: Web, 25 Apr 2018 09:50:00 GMT**

**Authorization: authorization string**

**Content-Length: 27364**

**x-amz-meta-author:Suraj**

**Expect: 100-continue**

**[27364 bytes of object data]**

**Enforcing Encryption on S3 Buckets**

* If the files is to be encrypted at upload time, the x-amz-server-side-encryption parameter will be included in the request header
* Two options are currently available:

x-amz-server-side-encryption: AES256 (SSE-S3 – S3 managed keys)

x-amz-server-side-encryption: ams:kms (SSE-KMS managed keys)

* When this parameter is included in the header of the PUT request, it tells S3 to encrypt the object at the time of upload, using the specified encryption method.
* You can enforce the use of Server Side Encryption by using a Bucket Policy which denies any S3 PUT request which doesn’t include the x-amz-server-side-encryption parameter in the request header.





**EC2 EBS Volume Types: -**

**Confusion: -**

There is lot of confusion between instance store volumes and EBS volumes in the AWS community and you need to have a good understanding of the differences between the two. Let’s start with volumes. There are two types of volumes:

* Root Volume (this is where your operating system is installed)
* Additional Volumes (this can be your D:\ E:\ F:\ or /dev/sdb, /dev/sdc. /dev/sdd etc.)

**Root Volume Sizes: -**

* Root device volumes can either be EBS volumes or instance Store Volumes
* An Instance store root device volume’s maximum size is 10GB
* EBS root device volume can be up to 1 or 2TB depending on the OS

**Terminating an Instance -EBS**

**EC2 Instances can be terminated:**

* EBS root device volumes are terminated by DEFAULT when the EC2 instance is terminated. You can stop this be unselecting the “Delete on Termination” option when creating the instance or by setting the deleteontermination flag using the command line.
* Other EBS volumes attached to the instance are preserved however, if you delete the instance.

**Terminating an Instance – Instance Store**

**EC2 instances can be terminated:**

* Instance store device root volumes are terminated by DEFAULT when the EC2 instance is terminated. You cannot stop this.
* Other instance store volumes will be deleted on termination automatically.
* Other EBS volumes attached to the EC2 instance will persist automatically

**Stopping an Instance: -**

* EBS backed instances can be stopped
* Instance Store backed instances CANNOT be stopped. Only rebooted or terminated.

**Instance Store Data: -**

The Data in an instance store persists only during the lifetime of its associated instance. If an instance reboots (intentionally or unintentionally), data in the instance store persists. However, data on instance store volumes is lost under the following circumstances:

* **Failure of n underlying drive**
* **Stopping an Amazon EBS-backed instance**
* **Terminating an instance**

**Instance Store: -**

Therefore, do not rely on instance store volumes for your data safe by using a replication strategy across multiple instances, storing data in Amazon S3, or using Amazon EBS volumes**.**

**Comparison**

|  |  |  |
| --- | --- | --- |
| **Characteristics** | **Amazon EBS-Backed** | **Amazon Instance Store-Backed** |
| Boot time | Usually less than 1 minute | Usually less than 5 minutes |
| Size Limit | 1 TiB | 10 GiB |
| Root Device Volume | Amazon EBS volume | Instance Store Volume |
| Data Persistence | By default, the root volume is deleted when the instance terminates. Data on any other Amazon EBS volumes persists after instance termination by default. | Data on any instance store volumes persists only during the life of the instance. |
| Upgrading | The instance type, kernel, RAM, disk, and user data can be changed while the instance is stopped. | Instance attributes are fixed for the life of an instance. |
| Chagres | You’re charged for instance usage, Amazon EBS volume usage, and storing your AMI as an Amazon EBS snapshot | You’re charged for instance usage and soring your AMI in Amazon S3 |
| AMI Creation/bundling | Use a single command/call | Require Installation and use of AMI tools |
| Stopped State | Can be placed in stopped state where instance is not running, but the root volumes is persisted in Amazon EBS | Can’t be in stopped state; instances are running or terminated. |

**Exam Tips: -**

* ‘Delete on Termination’ is the default for all EBS root device volumes. You can set this to false however but only at instance creation time.
* Additional volumes will persist automatically. You need to delete these manually when you delete an instance.
* Instance Store is Known as ephemeral storage, meaning that data will not persist after an instance is deleted. You cannot set this to false, data will always be deleted when that instance disappears

**Volumes & Snapshots**

* Volumes exist on EBS:
  + Virtual Hard Disk
* Snapshot exist on S3
* Snapshots are point in time copies of volumes.
* Snapshots are incremental – this means that only the blocks that have changed since your last snapshot are moved to S3.
* If this is your first snapshot, it may take some time to create.

**Snapshots of Root Devices Volumes: -**

* To create a snapshot for Amazon EBS volumes that serve as root devices, you can take a snap while the instance is running.
* You can create AMI’s from both Images and Snapshots
* You can change EBS volume sizes on the fly, including changing the size and storage type.
* Volume will ALWAYS be in the same availability zone as the EC2 instance.
* To move an EC2 volume from one AZ/Region to another, take a snap or an image of it them copy it to the new AZ/Region

**Volumes vs Snapshots – Security**

* Snapshots of encrypted volumes are encrypted automatically.
* Volumes restored from encrypted snapshots are encrypted automatically
* You can share snapshots, but only if they are unencrypted.
  + These snapshots can be shared with other AWS accounts or made public.

**Exam Tips: -**

* ‘Delete on Termination’ is the default for all EBS device volumes. You can set this to false however but only at instance creation time.
* Additional volumes will persist automatically. You need to delete these manually when you delete an instance.
* Instance Store is known as ephemeral storage, meaning that data will not persist after an instance is deleted. You cannot set this to false, data will always be deleted when that instance disappears

**Encryption and Down Time: -**

* For most AWS resources, encryption can only enable at creation.
* EFS (Elastic File System) If you want to encrypt an EFS filesystem that already exists, you will need to create a new encrypted EFS and migrate your data.
* RDS (Relational Database) If you want to encrypt an existing RDS, you will need to create a new encrypted database and migrate your data.
* **EBS Volumes – Encryption must be selected at creation time:** 
  + You cannot encrypt an unencrypted volume or unencrypt an encrypted volume.
  + You can migrate data between encrypted and unencrypted volumes. (e.g. use Rsync or Robocopy)
  + If you want to encrypt an existing volume, you can create snapshot, copy the snapshot and apply encryption at the same time to give an encrypted snapshot. Then restore the encrypted snapshot to new encrypted volume.
* **S3 in much more flexible:**
  + S3 Bucket – you can enable encryption on your S3 bucket at any time.
  + S3 Objects – you can enable encryption on individual S3 objects at any time.

**Exam Tips: -**

* Remember that for the majority of services, you will need to enable encryption at creation time:
  + EFS
  + RDS
  + EBS Volumes
  + To add encryption later will involve migrating your data in some way; you may wish to stop your applications at this time.
* S3 has greater flexibility, and you can enable for S3 Bucket or objects at any time and without disrupting your applications.

**KMS & CloudHSM: -**

* Both are allow you to generate, store and manage cryptographic keys used to protect your data in AWS.
* HSMs (Hardware Security Modules) are used to protect the confidentially of your keys
* Both offer a high level of security
* KMS – Shared hardware, multi-tenant managed service
* Allows you to generate, store and manage your encryption keys
* Suitable for applications for which multi-tenancy is not an issue
* Free-tier eligible
* Encrypt data stored in AWS, including EBS Volumes, S3, RDS, DynamoDB etc.
* Cloud HSM – Dedicated HSM (Hardware Security Module) instance, hardware is not shared with other tenants no Free-Tier
* Allows you to generate, store and manage your encryption keys
* HSM is under your exclusive control within your own VPC.

**KMS vs CloudHSM: -**

* FIPS 140-2 Level 3 compliance (US Government standard for HSMs) – includes tamper-evident physical security mechanisms
* Suitable for applications which have a contractual or regulatory requirement for dedicated hardware managing cryptographic keys.
* Use cases include database encryption, Digital Rights Management (DRM), Public Key Infrastructure (PKI), authentication and authorization, document signing, and transaction processing.



**Exam Tips: -**

* Both KMS and CloudHSM enable you to generate, store and manage your own encryption keys to encrypt data stored in AWS
* KMS is multi-tenancy and good for use cases which do not require dedicated hardware
* If your application has a requirement for dedicated hardware for managing keys, use CloudHSM

**Amazon** **AMIs: -**

An AMI (Amazon Machine Image) provides all the information needed to launch an EC2 instance.

* Template for the root volume, e.g. Operating System, Applications
* Launch permissions – defining which AWS accounts can use the AMI to launch instances
* Block device mapping to specify EBS volumes to attach to the instance at launch time.
* AWS provides a selection of default AMIs
* You can also create your own custom AMIs:
  + Launch an instance from an existing AMI
  + Connect to your instance and customize it (install apps, copy data etc)
  + Create a custom AMI image from your instance
  + Your AMI must be registered before it can be used to launch an instance

**AMIs are Regional: -**

* AMI are registered on a per-region basis.
* If you have registered your AMI in us-east-1 and you want to use it to launch instances in eu-west-1, you need to copy it to eu-west-1 in order to use it
* If you have created an AMI and cannot find it when attempting to launch an instance in a particular region, it may be because you haven’t copied the AMI to the region.

**AMIs – Exam Tips: -**

* AMIs provide a template for launching EC2 instances.
* You can create your own AMI from customized EC2 instance
* AMIs are region-bound, so if you are attempting to launch an instance in a new region using a custom AMI, make sure you have copied your AMI to the new destination region

**Sharing AMIs: -**

* After creating an AMI, you can either keep it private, share it with a specified list of AWS accounts, make it publicly available or even sell your AMI to another AWS users.
* The sharing account still has control an ownership of the AMI and is still charged for storage of the AMI within their AWS account.

**Copying AMIs: -**

* The owner of the source AMI must grant you read permissions for the storage that backs the AMI (EBS snapshot or S3)
* If you copy an AMI that was shared with you, you are then the owner of the copy and will be charged for storage of the target AMI in the destination region.

**Copying AMIs – Limitations**

* You cannot directly copy an encrypted AMI shared by another account
  + Copy the snapshot and re-encrypt using your own key
  + The sharing account must also share with you the underlying snapshot and encryption key used to create the AMI
  + You’ll own the copied snapshot and can register it as a new AMI

**Copying AMIs – Limitations**

* You cannot directly copy an AMI with an associated billingProducts code (applies to windows, RedHat and Amis from AWS Marketplace.)
* BillingProducts code is used to bill for the use of an AMI e.g. where a small fee is included to cover the Windows Server or SQL Server License
* Launch an EC2 instance using the shared AMI and create an AMI from the instance.

**Sharing AWS AMIs – Exam Tips**

* AMIs can be shared and copied between user accounts
* Remember the 2 restrictions:
  + Encrypted AMIs
  + Copy the underlying snapshot, re-encrypt using your own key and create a mew AMI from the snapshot
  + AMIs with an associated billingProducts code (e.g. Windows AMIs, RedHat, AWS Marketplace AMIs)
  + Launch an EC2 instance using the shared AMI and crate an AMI from the instance.

**Snowball: -**

* Snowball is a physical device used for transporting many terabytes or petabytes of data into and out of AWS.
* Makes large scale data transfer fast, easy and secure
* Tamper-resistant enclosure
* 256-bit encryption
* Region specific, not for transporting data from one region to another
* Connect the device to your local network
* The Snowball client automatically encrypts and copies the data you select onto to the Snowball.
* When the transfer is complete, the device is shipped back to Amazon.

**When to Use Snowball: -**

* When you have many TB or PB of data to upload
* You don’t want to expensive upgrades to your network for a one-off data transfer
* If you frequently experience backlogs of data
* You’re in a physically isolated environment, high-bandwidth internet is not available or is cost-prohibitive.
* If it takes more than a week to upload your data.

**What is Snowball Edge?**

* Each Snowball Ede is a 100TB device, which also feature onboard compute power which can be clustered to act as a single storage and compute pool.
* Designed to undertake local processing / edge computing, as well as data transfer.
* S3-compatible endpoint, supports NFS, and can also run Lambda functions as data is copied to the device.
* S3 buckets and Lambda functions come pre-configured on the device.

**Snowball vs Snowball Edge: -**

* Snowball is a data transport solution only.
* The Snowball Edge adds additional capability to run simple computing functions on the device.
* Use Snowball Edge for use cases that require local processing before returning the data to AWS.
* E.g. Performing data analysis before writing the processed data to an S3 bucket on the Snowball Edge.

**Snowball & Snowball Edge – Exam Tips: -**

**Remember the difference:**

* Snowball is for Data transfer only
* Snowball Edge provides Edge Computing in addition to data transfer.
* If you have 100s of TB upload or your data is taking a few days to upload – you need Snowball.
* Use Snowball Edge if you need to process the data locally before returning the device to AWS.

**Storage Gateway: -**

Storage Gateway consists of an on-premises software appliance which connects with AWS cloud-based storage to give you a seamless and secure integration between your on-premises IT environment and AWS.

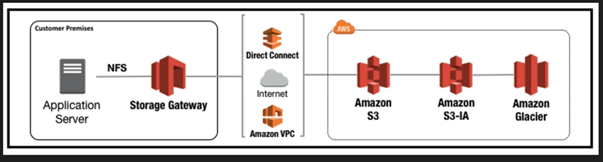
* Storage Gateway Virtual Appliance is installed your data center
* Supports VMware ESXi of Microsoft Hyper-V
* On-premises systems seamlessly integrate with AWS storage – e.g. S3

**Types of Storage Gateway: -**

* **File Gateway – NFS / SMB**
* **Volume Gateway (iSCSI)**
  + **Stored Volumes**
  + **Cached Volumes**
* **Tape Gateway (VTL)**

File Gateway: -

* Files stored as objects in your S3 buckets
* Accessed using NFS or SMB mount point
* To your on-premises systems this appears like a file system mount backed by S3
* All the benefits of S3: bucket policies, S3 versioning, lifecycle management, replication etc.
* Low-cost alternative to on-premises storage



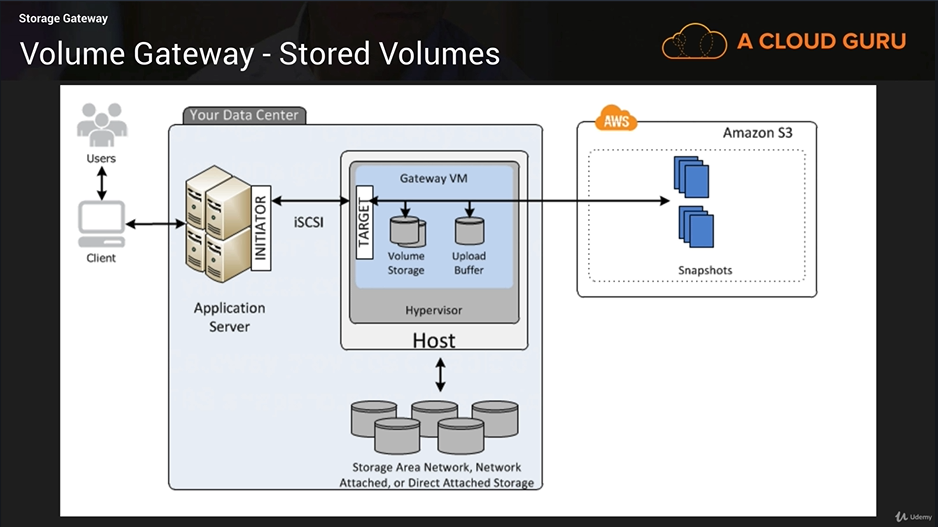
* On-premises servers access files stored in S3 / Glacier
* NFS provided by the Storage Gateway appliance

**Volume Gateway: -**

* Volume Gateway provides cloud backed storage which is accesses using iSCSI protocol
* **Types of Volume Gateway: -**
  + **Gateway Stored Volumes –** Store your all data locally and only backup to AWS
  + **Gateway Cache Volumes –** Use S3 as your primary storage and cache frequently accesses data in your Storage Gateway.

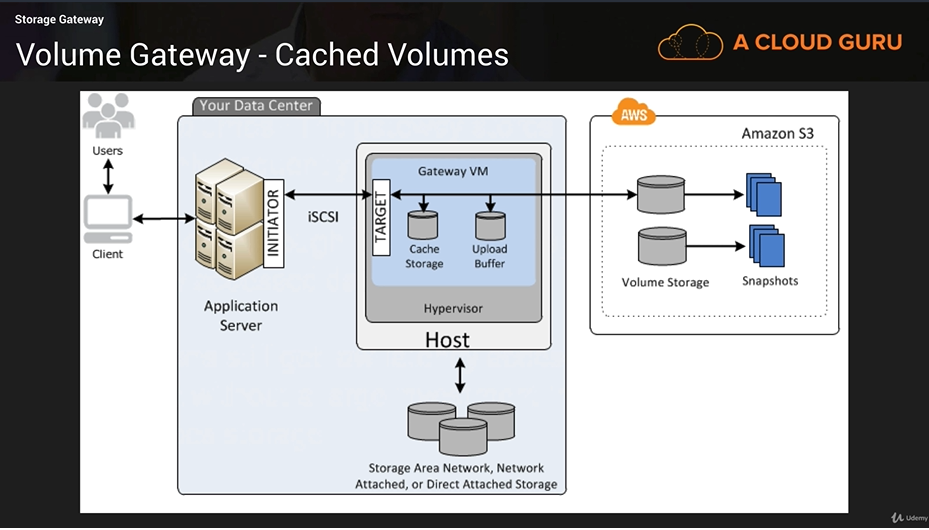
**Volume Gateway – Stored Volumes: -**

* Stored Volumes – The gateway stores all your data locally, so your applications get low latency access to the entire dataset
* You need your own storage infrastructure as all data is stored locally in your data center
* Volume Gateway provides durable off-site async backups in the form of EBS snapshots which are stored in S3



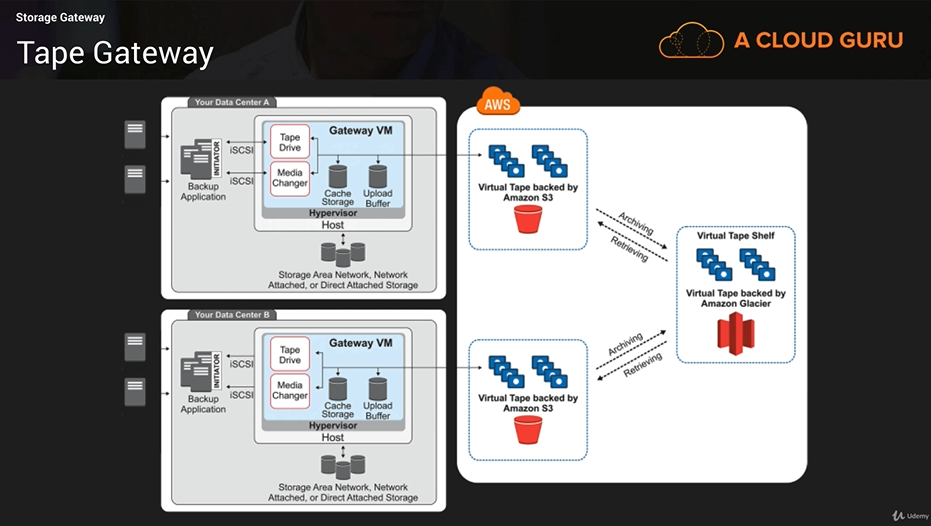
**Volume Gateway – Cached Volumes: -**

* **Cached Volumes** – the gateway stores all your data in S3 and caches only frequently accessed data locally
* You need only enough local storage capacity to store the frequently accessed data
* Applications still get low-latency access to frequently used data without a large investment in on-premises storage



**Tape Gateway (VTL)**

* Tape Gateway is a Virtual Tape Library which provides cost effective data archiving in the cloud using Glacier
* You don’t need to invest in your own tape backup infrastructure
* Integrates with existing tape backup infrastructure – NetBackup,
* Backup Exec, Veeam etc. which connect to the VTL using iSCSI
* Data is stored on virtual tapes which are stored in Glacier and accessed using the VTL



**Storage Gateway – Exam Tips: -**

* **File Gateway** – Flat files stored on S3, accessed using NFS or SMB
* **Volume Gateway** – 2 Types:
  + **Stored Volumes** – Entire dataset stored on-site, backed-up to S3 as EBS Snapshots
  + **Cached Volumes** – Entire dataset stored in S3, only frequently accessed data cached on-site
* **Tape Gateway – VTL**
  + Used for archiving your backups to Glacier
  + Can be used with or without your own backup application

**Athena: -**

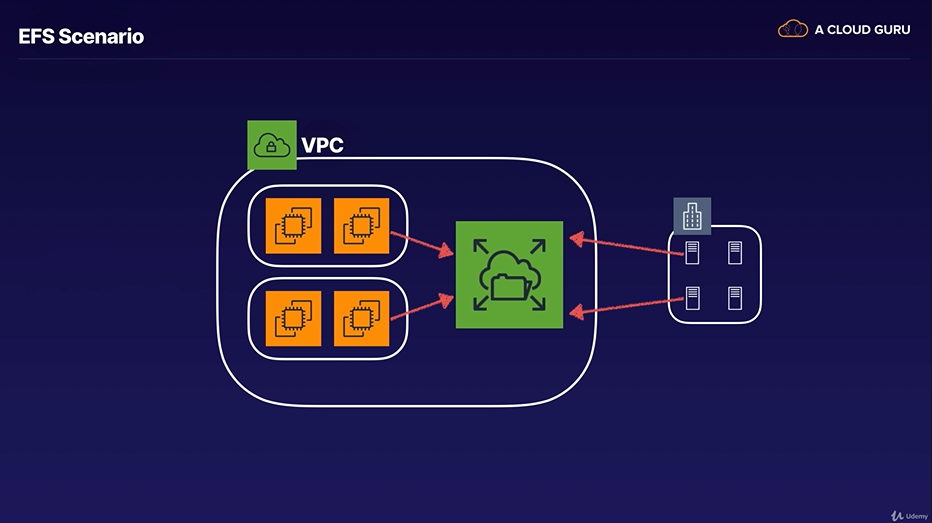
* Athena is an interactive query service that enables you to analyse and query data located in S3 using standard SQL.
* Serverless, nothing to provision, pay per query / per TB scanned
* No need to set up complex Extract/Transform/Load (ETL) processes
* Works directly with data stored S3.

**Athena Use cases: -**

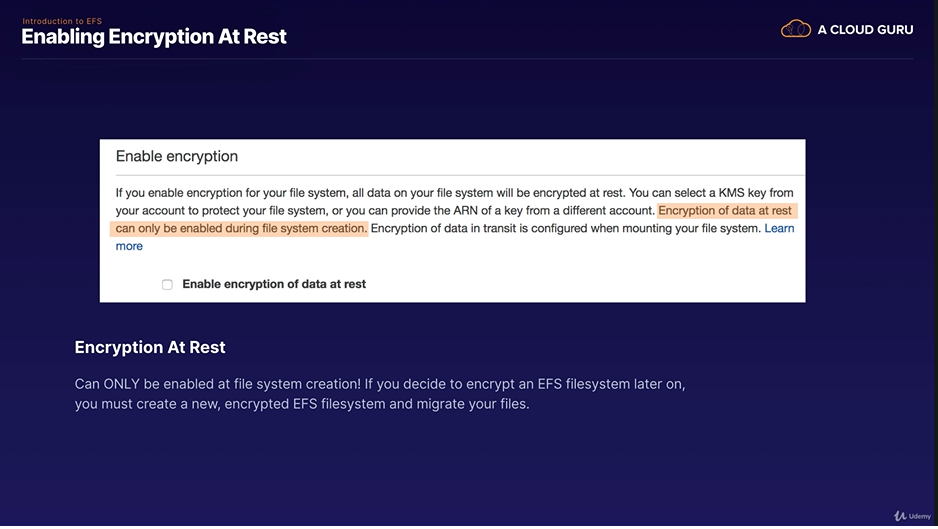
* Can be used to query log files stored in S3, e.g. ELB logs, S3 access logs, etc.
* Generate business reports on data stored in S3
* Analyze AWS cost and Usage reports
* Run queries on click-stream data

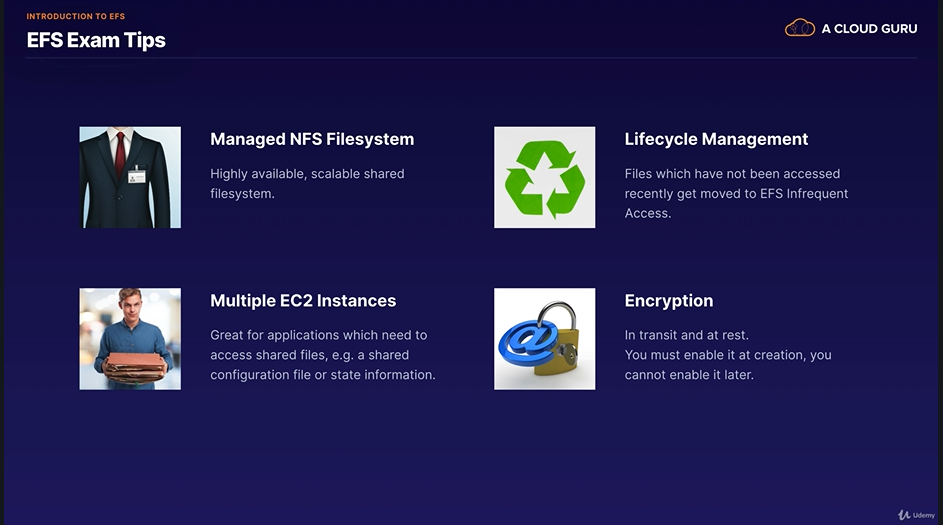
**Athena – Exam Tips: -**

* Athena is an interactive query service.
* Allows you to query data located in S3 using standard SQL
* Serverless









**Part 6 Security: -**

**Compliance on AWS: -**

**ISO 27001:2005/10/13**

**“**ISO/IEC 27001:2005 specifies the requirements for establishing, implementing, operating, monitoring, reviewing, maintaining and improving a documented information Security Management System within the context of the organization’s overall business risks.”

**FedRAMP: -**

“The Federal Risk and Authorization Management Program, or FedRAMP, is a government-wide program that provides a standardized approach to security assessment, authorization, and continuous monitoring for cloud products and services.”

**HIPAA: -**

“HIPPA is the federal Health Insurance Portability and Accountability Act of 1996. The primary goal of the law is to make it easier for people to keep health insurance, protect the confidentiality and security of healthcare information and help the healthcare industry control administrative costs.”

**PCI DSS v3.2**

“The Payment Card Industry Data Security Standard (PCI DSS) is a widely accepted set of policies and procedures intended to optimize the security of credit, debit and cash card transactions and protect cardholders against misuse of their personal information.”

**Build and Maintain a Secure Network and Systems**

* **Requirement 1:** Install and maintain a firewall configuration to protect cardholder data
* **Requirement 2:** Do not use vendor-supplied defaults for system passwords and other security parameters

**Protect Cardholder Data**

* **Requirement 3:** protect stored cardholder data
* **Requirement 4:** Encrypt transmission of cardholder data across open, public networks

**Maintain Vulnerability Management Program**

* **Requirement 5:** Protect all systems against malware and regularly update anti-virus software or programs
* **Requirement 6:** Develop and maintain secure systems and applications

**Implement Strong Access Control Measures**

* **Requirement 7:** Restrict access to cardholder data by business need to know
* **Requirement 8:** Identify and authenticate access to systems components
* **Requirement 9:** Restrict physical access to cardholder data

**Regularly Monitor and Test Networks**

* **Requirement 10:** Track and monitor all access to network resources and cardholder data
* **Requirement 11:** Regularly test security systems and processes.

**Maintain an Information Security Policy**

* **Requirement 12:** Maintain a policy that addresses information security for all personnel.

**Other frameworks:**

* **SAS70**
  + **Statement on Auditing Standards No. 70**
* **SOC1**
  + **Service Organization Controls – accounting standards**
* **FISMA**
  + **Federal Information Security Modernization Act**

**Other frameworks:**

FIPS 140-2 is a U.S. government computer security standard used to approve cryptographic modules. Rated from Level 1 to Level 4, with being the highest security. Cloud HSM meets the level 3 standard.

**DDoS Attacks : -**

**What is a DDoS Attack?**

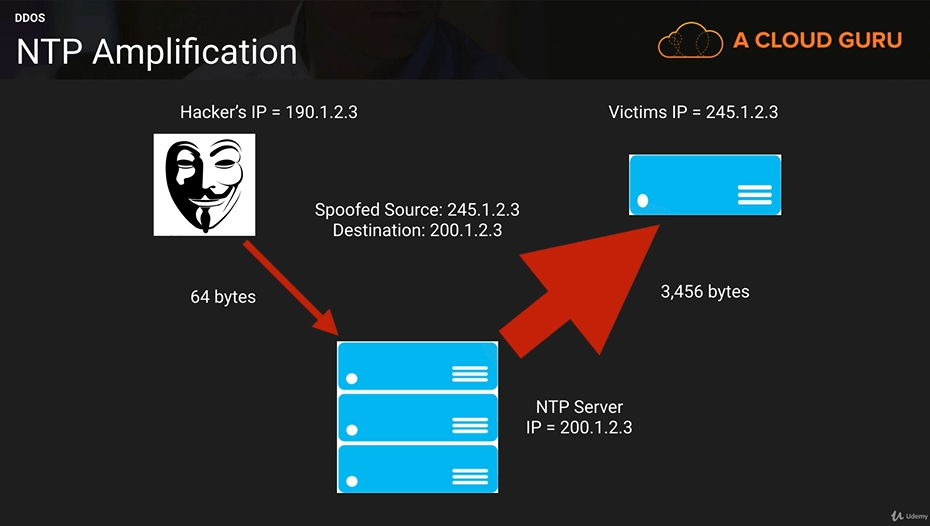
A Distributed Denial of Service (DDoS) attack is an attack that attempts to make our website or application unavailable to your end users.

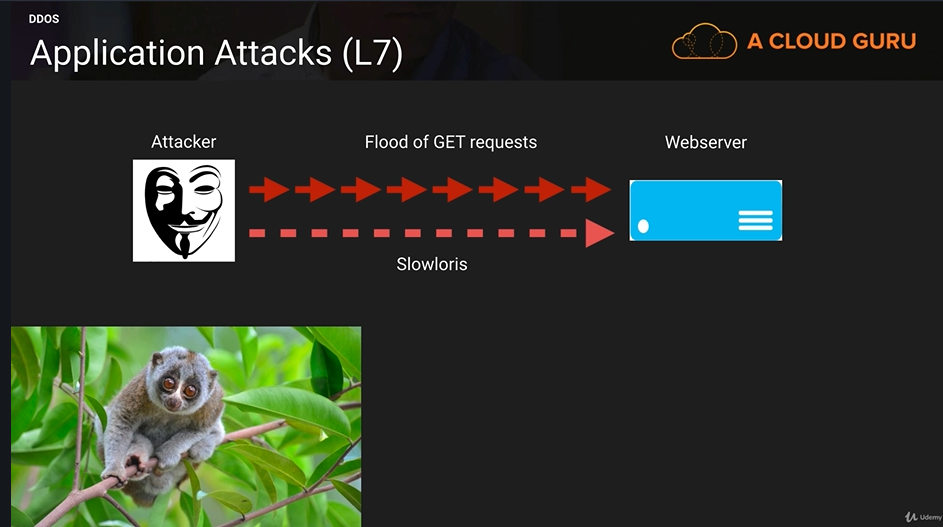
This can be achieved by multiple mechanisms, such as large packet floods, by using a combination of reflection and amplification techniques, or by using large botnets.

**Amplification/Reflection Attacks: -**

Amplification/Reflection attacks can include things such as NTP, SSDP, DNS, Chargen, SNMP attacks, etc. and is where an attacker may send a third party server (such as an NTP Server) a request using a spoofed IP address. That server will then respond to that request with payload than initial request (usually within the region of 28 x 54 times larger than the request) to the spoofed IP address.

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





**How to Mitigate DDoS?**

* Minimize the Attack Surface Area
* Be ready to Scale to Absorb the Attack
* Safeguard Exposed Resources
* Learn Normal Behavior
* Create a Plan for Attacks

**AWS Shield**

* Free service that protects all AWS customers on Elastic Load Balancing (ELB), Amazon CloudFront and Route 53
* Protects against SYN/UDP Floods, Reflection attacks, and other layer 3/layer 4 attacks.
* Advanced provides enhanced protections for your applications running on Elastic Load Balancing (ELB), Amazon CloudFront and Route 53 against larger and more sophisticated attack. $3000 per month.

**AWS Shied Advanced Provides**

* Always – on, flow-based monitoring of network traffic and active application monitoring to provide near real-time notifications of DDoS attacks.
* DDoS Response Team (DRT) 24X7 to manage and mitigate application layer DDoS attacks.
* Protects your AWS bill against higher fees due to Elastic Load Balancing (ELB), Amazon CloudFront and Amazon Route 53 usage spikes during a DDoS attack.
* $3000 a month

**DDoS Exam Tips: -**

**Remember the technologies you can use to mitigate a DDoS attack:**

* **CloudFront**
* **Route53**
* **ELB’s**
* **WAFs**
* **Autoscaling (Use for both WAFs and Web Servers)**
* **CloudWatch**

**AWS Hypervisor: -**

A hypervisor or virtual machine monitor (VMM) is 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).

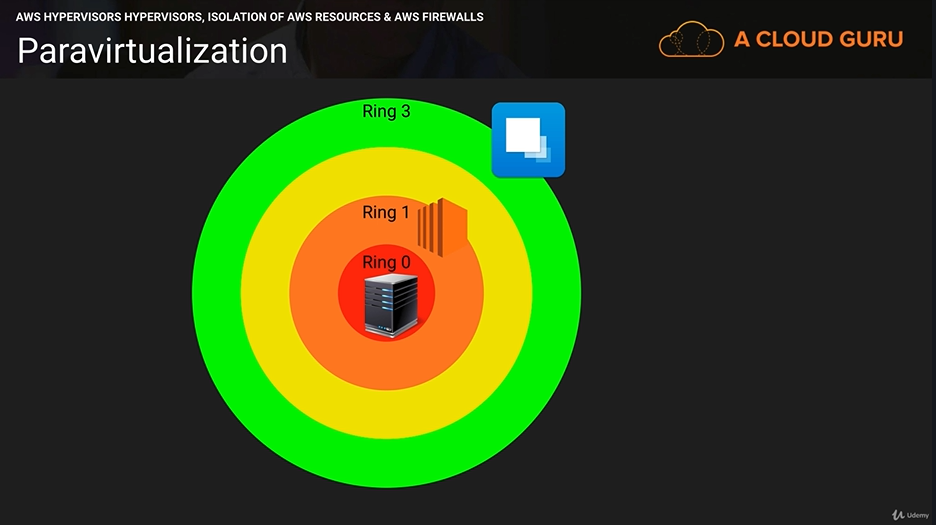
HVM guests are fully virtualized. The VMs on top of the Hypervisors are not aware that they are sharing processing time with other VMs.

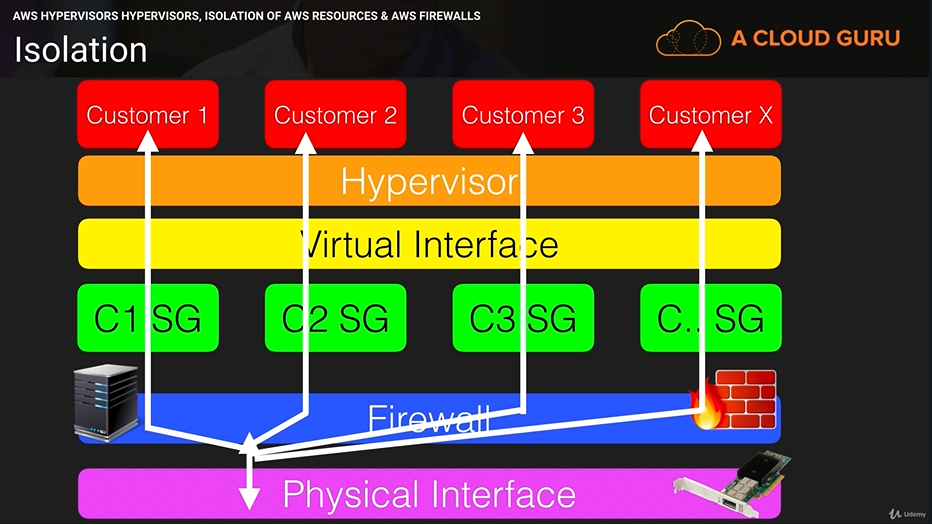
PV is a lighter from of virtualization and it used to be quicker.

However, this performance gap has now closed and Amazon now recommend using HVM over PV where possible. It’s also worth noting that Windows EC2 instances can only be HVM whereas Linux can be both PV and HVM.

**PV**

Para-virtualized 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 CPU provides four separate privilege modes: 0-3, called rings. Ring o is the most privileged and 3 the least. The host OS executes in Ring 0. However, rather than executing in Ring 0 as most operating systems do, the guest OS runs in a lesser-privileged Ring 1 and applications in the least privileged Ring 3





**Hypervisor Access: -**

Administrator with a business need to access the management plane are required to use multifactor authentication to gain access to purpose-built administration hosts. These administrative hosts are systems that are specifically designed, built, 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 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 customer’s data is never unintentionally exposed to another. Also memory allocated to guests is scrubbed (set to zero) by the hypervisor when it is unallocated to a guest. The memory is not returned to the pool of free memory available for new allocations until the memory scrubbing is complete.

**AWS Hypervisors Exam Tips: -**

* Choose HVM over PV where possible
* PV is isolated by layers, Guest OS sits on Layer 1, Applications Layer 3
* Only AWS Administrators have access to hypervisors
* AWS staff do not have access to EC2, that is your responsibility as a customer.
* All storage memory and RAM memory is scrubbed before it’s delivered to you.

**Dedicated Instances vs Dedicated Hosts: -**

**What are EC2 Dedicated Instances?**

Dedicated Instances are Amazon EC2 instances that run in a VPC on hardware that’s dedicated to 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 accounts 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.

**What are EC2 Dedicated Hosts?**

You can use Dedicated Hosts and Dedicated instances to launch Amazon EC2 instances on physical servers that are dedicated for your use. An important difference between a Dedicated Host and Dedicated instance is that a Dedicated Host and Dedicated instance is that a Dedicated Host gives you additional visibility and control over how instances are placed on a physical server, and you can consistently deploy your instances 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.

**Dedicated Instance vs Dedicated Hots – Exam Tips: -**

* Both dedicated instances and dedicated hosts have dedicated hardware
* Dedicated instances are charged by the instance, dedicated hosts are charge 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 accounts that are not dedicated.
* Dedicated hosts give you much better visibility into things like sockets, cores and host id.

**S3 Bucket pre-signed URL: -**

**Pre-signed URL’s**

* You can access objects using pre-signed URL’s
* Typically, these are done via the SDK but can also be done using the CLI
* They exist for a certain length of time in seconds. Default is 1 hour.
* You can change this using “—expires-in” followed by the number of seconds

**AWS Inspector vs Trusted Advisor: -**

**What is AWS Inspector: -**

Amazon Inspector is an automated security assessment service that helps improve the security and compliance of applications deployed on AWS. Amazon Inspector automatically assesses applications for vulnerabilities or deviations from best practices. After performing an assessment, Amazon 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.

**What is AWS Trusted Advisor: -**

An online resource to help you reduce cost, increase performance, and improve security by optimizing your AWS environment.

Advisor will advice you on Cost Optimization, Performance, Security, Fault Tolerance.

* **Core Checks and Recommendations**
* **Full Trusted Advisor – Business and Enterprise Companies Only**

**Exam Tips: -**

**How does it work?**

* Create “Assessment target”
* Install agents on EC2 instances
* Create “Assessment template”
* Perform “Assessment Run”
* Review “Findings” against “Rules”

**Rules Packages:**

* Common Vulnerabilities and Exposures
* CIS Operating System Security Configuration Benchmarks
* Security Best Practices
* Runtime Behavior Analysis

**Severity Levels for Rules in Amazon Inspector**

* High
* Medium
* Low
* Informational

**It 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

**Trusted Advisor: -**

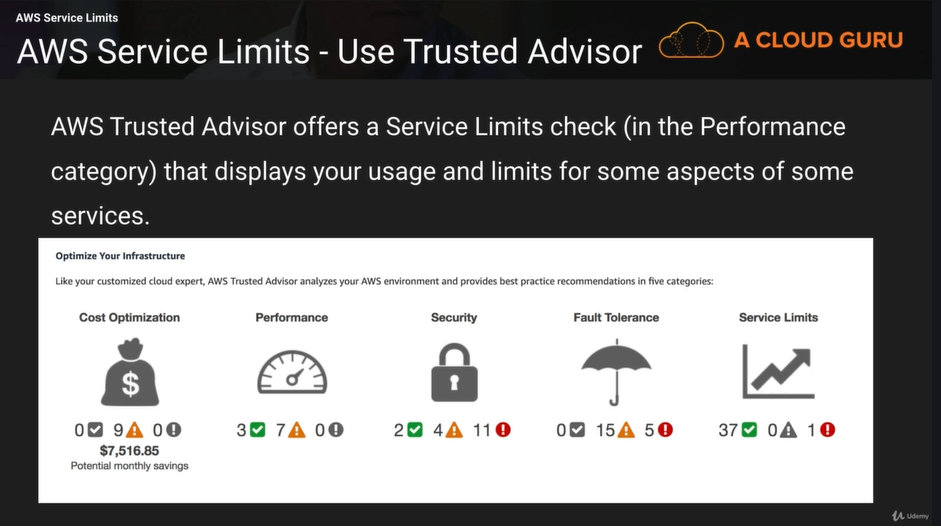
* Cost Optimization
* Availability
* Performance

**AS WELL AS Security.**

**AWS Service Limit: -**

Every AWS account has service limits by default. A service limit is simply the number of services you are able to provision on a per-account basis.

For example, you can only provision so many EC2 instances (depending on instance type and region). To view service limits, check out aws\_service\_limits.html



**Exam Tips: -**

Know that all AWS Services have service limits. These are unique on a per-region basis. You can always check your service limits by using AWS Trusted Advisor.

**Security Token Service: -**

Grants users limited and temporary access to AWS resources. Users can come from three sources:

* **Federation (Typically Active Directory)**
  + Uses Security Assertion Markup Language (SAML)
  + Grants temporary access based off the users Active Directory credentials, does not need to be a user in IAM
  + Single sign on allows users to log in to AWS console without assigning IAM credentials
* **Federation with Mobile Apps**
  + Use Facebook/Amazon/Google or other OpenID providers to log in.
* **Cross Account Access**
  + Let’s users from one AWS account access resources in another

**Understanding Key Terms:**

* Federation: combining or joining a list of users in one domain (such as IAM) with a list of users in another domain (such as Active Directory, Facebook etc.)
* Identity Broker: a service that allows you to take an identity from point A and join it (federate it) to point B
* Identity Store: Services like Active Directory, Facebook, Google etc.
* Identities – a user of a service like Facebook etc.

**AWS WAF: -**

AWS WAF is a web application firewall that lets you monitor the HTTP and HTTPS requests that are forwarded to Amazon CloudFront or an Application Load Balancer or to API Gateway. 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

**What is AWS WAF?**

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

* Allow 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

More protection against web attacks using conditions that you specify: you can define conditions by using characteristics of web requests such as:

* IP addresses that requests originate from
* Country that request originate from
* Values in request headers
* Strings that appear in requests, either specific strings or string that match regular expression (regex) patterns.
* Length of requests.
* Presence of SQL code that is likely to be malicious (or ‘SQL injection’).
* Presence of a script that is likely to be malicious (known as cross-site scripting)

**Which Services Does it Integrate with?**

**WAF Integrates with the following services:**

* Application Load Balancers
* CloudFront
* API Gateway

**It DOES NOT integrate with:**

* Classic Load Balancers
* Network Load Balancers

**Exam Tips: -**

AWS WAF is a web application firewall that lets you monitor the HTTP and HTTPS requests that are forwarded to Amazon CloudFront or an Application Load Balancer or to API Gateway. AWS WAF also lets you control access to your content.

**It does not integrate with Classic or Network Load Balancers.**

**Shared Responsibility Model: -**

While AWS manages security of the cloud, security in the cloud is the responsibility of the customer. Customers retain control of what security they choose to implement to protect their own content, platform, applications, systems and networks, no differently than they would in an on-site datacenter.

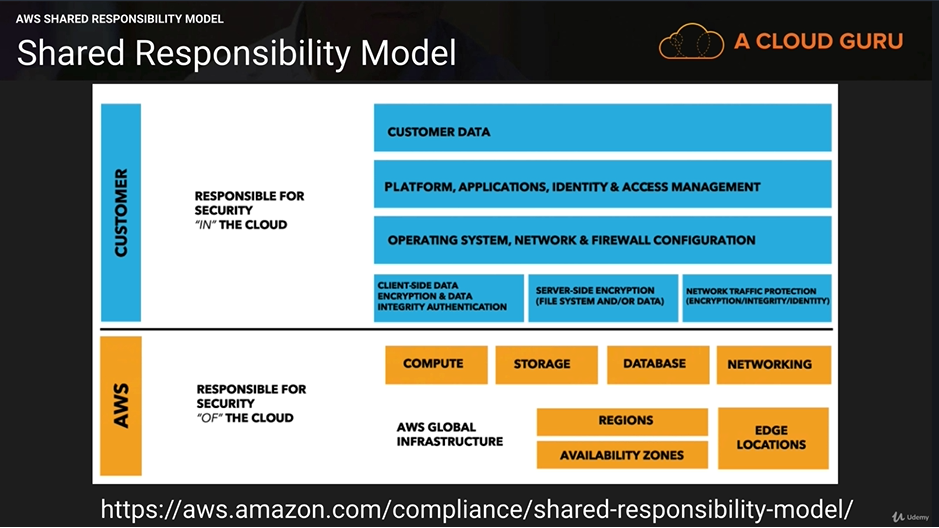
**AWS Security Responsibilities**

* Global Infrastructure
* Hardware, Software, Networking and Facilities
* “Managed Services”

**Customer Security Responsibilities: -**

* Infrastructure as a Service (IaaS)
* Including updates and security patches
* Configuration of the AWS-provided firewall

“These are basically the same security tasks that you’re used to performing no matter where your servers are located.”



**But….**

**The Model changes for different service types:**

* Infrastructure
* Container
* Abstracted

**Infrastructure Services: -**

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 an 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;**

* Amazon Machine Images (AMIs)
* 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 or 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 containers services include RDS, EMR & 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.

**Exam Tips: -**

You are responsible for things like EC2 OS Patching/Antivirus, Security Groups etc.

You are not responsible for things like RDS OS updates, RDS Database Updates, PHP updates with ElasticBeanstalk etc.

**AWS Artifacts: -**

AWS artifacts provides on-demand downloads of AWS security and compliance documents, such as AWS ISO certifications, Payment Card Industry (PCI), and Service Organization Control (SOC) reports. You can submit the security and compliance documents (also known as audit artifacts) to your auditors or regulators to demonstrate the security and compliance of the AWS infrastructure and services that you use.