EC2

Launch Instance from AMI

Placement Groups

Cluster - Low Latency group in Single AZ

- All instance in same rack, same AZ
- If Hardware fails all instances fail at the same time
- (Increased Risk)(Great Network Speed)

Spread - Spreads instances across underlying hardware (7 instances per group per AZ)
*Not Applicable to t2 Instances

- All Instances run on different hardware
- Reduced risk of simultaneous failure
- Good for hosted application to maximize high availability

If these exist you can place instances within whatever you want. Also tells you limit within launching.

Termination Protection

TIP: Instance where shutdown behavior = terminate and enable termination protection is ticked. (Instance will still be terminated)

Can change within the EC2 Console while instance is running. Can also change shutdown behavior.

Launch Troubleshooting

- InstanceLimitExceeded error: if you get this error you have reached your max limit of instances per region.

  You can go into Limits within the dashboard and request an increase or contact support for help.

- InsufficientInstanceCapacity error: if you get this error, it means AWS does not have enough On-Demand capacity in the particular AZ to which the instance is launched.

  Wait a few minutes and try again
  Launch one at a time

Change instance type later

- InstanceTerminatesImmediately error: (goes from pending to terminated)

  Reached EBS volume limit.
  EBS volume is corrupt
  The root EBS volume is corrupted and you do not have permissions to access key
  The instance store AMI that you used to launch the instance is missing a req part

SSH Trouble

- Make sure the pem file has 400 permissions otherwise you get Unprotected Private Key File error
- Make sure your username for the OS is given correctly when logging via SSH, or else you'll get "Host key not found"
- Connection Timeouts: SG not configured Properly/CPU load of the instance is high

Instance Launch Types

- On Demand: short workload, predictable pricing
  Billed per second after the first minute
  Has the highest cost but no upfront payment
  No long term commitment
  Short-term and un-interrupted (Good for Autoscaling)
- Reserved Instances: long workloads (Over a Year)
  Up to 75% discount compared to On-demand
  Pay upfront for what you use with long term commitment
  Reservation period of 1-3 years
  Reserve a specific instance type
  Recommended for steady state usage apps (think database)
- Convertible Reserved Instances: long workloads with flexible instance types (Up to 54% lower cost)
- Scheduled Reserved Instances: launch within time window you reserve
- Spot Instances: short workloads, for cheap, can lose instances
  Up to (90% lower vs On-Demand)
  You bid a price and get the instance as long as it's under the price you bid
  Price varies on offers and demand
  Spot instances are reclaimed with a 2 minute notification warning when the spot price goes above your bid
  Used for batch jobs, Big Data Analysis or Workloads resilient to failure
- Dedicated Instances: no other customers share your hardware
  You control the instances but not the hardware
  No control over instance placement (Can move to different hardware upon Start/Stop)
- Dedicated Hosts: book an entire server within a datacenter

Control Hardware and Sockets/Cores
Useful for software that uses (BYOL) model (Vendors that bill on cores)
Compliance regulations

Think of these instance types as hotels.
On demand - approach desk pay full price
Reserved - in advance and discounted
Spot - kicked out if higher bid
Dedication - book the entire hotel

Instance Type

- R:applications that need RAM
- C: applications that need CPU
- M: between Ran and CPU (Web App)
- I: applications that needs instance storage (database)
- G: applications that need GPU
- T2/T3: burstable instances good CPU for awhile until credit usage is damaged (high usage) (cannot purchase CPU credits if performance degrades)
- T2/T3 Unlimited: can be very $

  https://ec2instances.info/

AMIs

- Can create custom w/ preinstalled packaged
- Security concerns if left outdated
- You can sell and rent AMIs from other people
- All found on AWS Marketplace
- Don’t use an AMI you don’t trust and check for infections
- Store in S3 so they’re cheap to store
- By default your AMIs are private and looked by account/region
- Clean up your AMIs if you don’t use

Cross Account AMI Copy

- Sharing an AMI does not affect the ownership
- They can copy the AMI and they’ll be the owner
- You can’t copy an AMI with an associated billingProduct code that was shared with you from another account. This includes Windows AMIs and AMIs from the AWS Marketplace. To copy a shared AMI with a billingProduct code, launch an EC2 instance in your account using the shared AMI and then create an AMI from that instance

- Change permissions from AMI (Right-click and Modify Permissions Public/Private) Set account numbers within here if private
- If you choose create volume permissions they can still copy with or without

Elastic IP

- IP you own and doesn't change. If you start/stop an EC2 instance without one the public IP will change
- If you don't use the Elastic IP you're changed. If in use, you aren't

TIP: With an elastic IP, you can mask the failure of an instance or software by rapidly remapping the address to another instance in your account.

- You can only have 5 elastic IPs by default in your account. Can ask AWS to increase this limit
- Overall, try to avoid using elastic IPs

Cloud Watch EC2

- Collected at 5 minute intervals (Can change to 1 minute but it costs)
- Custom metrics (yours to push) basic: 1 minute | high: 1 second
- CPU Utilization + Credit Usage / Balance for Burst
- Network In / Out
- Status Check - Instance and System Status
- Disk I/O (EC2 Store Instances Only)
- No RAM Usage Statistics

EC2 Custom Metrics

- Can push RAM or Swap
- CloudWatch Monitoring Scripts (Old way, can use CW agent)

Cloud Watch EC2

- Not enabled by default but if you use instance this enables the options
- Check IAM to ensure you have the correct permission

EC2 at Scale (SSM & OPSWORKS)

Systems Manager Overview

- Helps you manage your EC2 and On-Prem systems at scale
- Operational insights about the state of infras
- Easily detect problems
- Patching automation for enhanced compliance
- Works for Windows and Linux
- Integrated with CloudWatch metric/dashboards
- Integrated with AWS Config
- Free Service

What if you lose your SSH key for EC2?

- If the instance is EBS: Run the AWSSupport-ResetAccess | Old method is to stop the instance > unmount volume > mount to another instance and modify ~/.ssh/authorized_keys file > remount and use new key
- Instance store backed EC2: You can't stop the instance (otherwise data is lost) - AWS Recommends Termination

AWS Parameter Store

- Securely store configuration and secrets
- Use seamless encryption using KMS
- Serverless, scable, durable, easy SDK, free
- Version tracking of configs/secrets
- Configuration management using path & IAM (restrict who can access what)
- Notification with CloudWatch Events
- Integration with CloudFormation

Can be accessed via EC2 >Parameter Store or CLI

Opsworks Overview

- Chef & Puppet help you perform server configuration automatically, or repetitive actions
- They work great with EC2 & On Premise VM
- AWS Opswork = Managed Chef & Puppet
- Alternative to AWS SSM

TIP: Anytime you seem Chef & Puppet answer AWS Opsworks

EC2 LB & AUTOSCALING

- Troubleshooting
- Advanced options and logging
- CloudWatch Integrations

Vertical Scalability

- Increasing the size of the instance (up/down)
- Very common for non distributed systems, such as a database
- RDS, ElastiCache are services that can scale vertically
- There's usually a limit to how much you can scale (hardware limit)

Horizontal Scalability

- Increasing the number of instances/systems for your application (scale out/in)
- Common for web/modern applications
- It's easy to scale using EC2 and cloning

High Availability

- Running in at least 2 data centers (multiple AZs)
- Both LB and Autoscaling have multi AZ options you can enable
- Goal is to survive data center loss
- Can be passive (RDS Multi AZ for example)

Why use a LB?

- Spread load across multiple downstream instances
- Expose a single point of access (DNS) to your app
- Seamlessly handle failures of downstream instances
- Do regular health checks of instances
- Provide SSL termination (HTTPS) for your websites (Client to ELB Encryption)
- Enforce stickiness cookies (same user session/same instance)
- Separate private from public traffic

ELB (EC2)

- AWS guarantees that it will work
- AWS takes care of upgrades/main/high avail
- AWS provides only a few configuration options

- Cost effective
- Integrated with mang AWS services

Types of Load Balancers

Classic - (v1 - old gen) 2009

Application - (v2 - new gen) 2016
Layer 7
Allow load balancing to multiple HTTP applications across machines
Allow multiple applications on the same machine
Load balance using route or hostname
(Work well with docker and ECS)
Has port mapping feature to redirect to a dynamic port
Classic is very inefficent
ALB supports HTTP/HTTPS & Websocket protocols
The application servers don't see the IP of the client directly
Client IP using: X-Forwarded-For
Client Port: X-Forwarded-Port
Client Protocol: - X-Forwarded-Proto

Network - (v2 - new gen) 2017
Layer 4 TCP Traffic (vs HTTP)
Handle millions of reqs per second
Support for static and elastic IP
Used for extreme performance
Overall the same process

Recommended to use v2 gen because of features

Health checks are done over port/route (/health) 200 = OK

TIPS -
CLB are depreciated
CLB and ALB support SSL certs and provide SSL termination
ALB can route based on Path
All have static hostnames. Do not resolve using underlying IP.
LBs can scale but not instantaneously - contact AWS for a “warm-up”
NLB directly see the client IP
4xx are client errors
5xx are application induced errors
If LB can't connect to application (Check Sec Groups)

Load Balancer Stickiness

- Same client always redirects to same instance behind LB works with CLB and ALB types
- Uses cookie with expiration date you control
- Use case: make sure the user doesn't lose his session data/progress
- Enabling stickiness may bring imbalance to the load over the backend EC2 instances

Load Balancer for SysOps

ALB:

- Layer 7 (HTTP, HTTPS, Websocket)
- URL Based Routing (Hostname or path)
- Does not support static IP but has fixed DNS
- Provide SSL termination

NLB:

- Layer 4 (TCP)
- No pre-warming needed
- 1 static IP per subnet
- No SSL Termination

TIP: Chain NLB and a ALB to "give" ALB a fixed IP address

If you expect high traffic (shopping season) warm up ELB with support:

Duration of traffic
Expected requests per second
Size of request

400 errors are passed on by ELB
500 errors throw healthchecks

TIP: Support legacy browsers by enabling a weaker cipher (DES-CBC3-SHA for TLS 1.0)
*Security Policy is ELBSecurityPolicy-TLS-1-0-2015-04 ***CHANGE ALB SEC POLICY****

Troubleshooting:
Check Sec Groups
Check Health Checks
Stick Sessions
For Multi AZ enable cross zone balancing
If you don't need public use internal LB

Enable deletion protection to protect ELB

Load Balancer Monitoring

- All LB metrics are pushed to CloudWatch metrics
- BackendConnectionErrors
- Healthy/UnhealthyHostCount
- HTTPCode_Backend_2XX: Successful
- HTTPCode_Backend_3XX: Redirected
- HTTPCode_ELB_4XX: Client error codes
- HTTPCode_ELB_5XX: Server error codes generated by LB
- Latency
- RequestCount
- SurgeQueueLength (If overloaded you see this) 1024 is the max value
- SpillOverCount: The total number of requests that were rejected because the surge queue was full

Auto Scaling Groups

- The goal of ASG is to scale out and in servers depending on load for EC2
- Ensure you have the max and min number running at all times (set limits)
- Automatically register new instances to a load balancer

Attributes:

- AMI + Instance Type
- EC2 User Data
- EBS Volumes
- Sec Groups
- SSH Key Pair
- Min/Max size / Initial Capacity
- Network + Subnet Information
- LB Info
- Scaling Policies

Auto Scaling Alarms

- Possible to scale ASG based on CloudWatch alarms
- An alarm monitors as a metric (such as avg CPU)
- Metrics are computed overall for ASG instances
- Based on the alarm can create scale out and in policies

New Rules:

- Number of requests on the ELB per instance
- Average Network In

These new rules are easier to set up and can make more sense

Can use Custom Metric

ex: number of total connected users

1. Send custom metric from application on EC2 to CloudWatch (PutMetric API)
2. Create CloudWatch alarm
3. Alarms scale policy for ASG

- IAM roles attached to an ASG will get assigned to EC2 instances
- ASG is free. You only pay for underlying resources being launched/used
- ASG can terminate instances marker as unhealthy by an LB (and replace them)
- ASG doesn't reboot unhealthy instances for you

CLI:

- Set-instance-health
- Terminate-instance-in-auto-scaling-group

Troubleshooting:

- <number of instances> instances are already running. Launch EC2 instance failed
  The auto scaling group has reached the limit set by the DesiredCapacity parameter.
  Update your Auto Scaling group by providing a new value for the desired capacity.

- Launching EC2 Instances is failing
  The sec group does not exist
  The key pair does not exist

- If the ASG fails to launch an instance for over 24 hours, it will automatically suspend the processes

CloudWatch ASG

Opt-in checks:

- GroupMinSize
- GroupMaxSize
- GroupDesiredCapacity

- GroupInServiceInstances
- GroupPendingInstances
- GroupStandbyInstance
- GroupTerminatingInstances
- GroupTotalInstances

Metrics above collected every 1 minute

Monitoring Underlying EC2 basic 5 mins, 1 min is paid

ELASTIC BEANSTALK

Developer centric view of deploying an application on AWS
Free managed service
Config/OS is handles by beanstalk
Deployment strategy is configurable but performed by ELB
Just the app code is the responsibility of the dev

Three arch models:

- Single instance deployment: good for dev
- LB+ASG: great for prod or pre-prod web apps
- ASG only: great for non-web apps in production (workers, etc.)

Has three total components:

- Application
- App Version and each deploy gets assigned a version
- Environment name (dev, test, prod): free naming

-Deploy app versions to environments and can promote version to the next environment
-Rollback feature to previous app version
-Full control over life cycle of environments

Supports the following platforms:

- GO
- Java SE
- Java with Tomcat
- .NET on Windows with IIS
- Node,js

- PHP
- Python
- Ruby
- Packer Builder
- Single/Multi/Preconf Docker
- Can write your own custom platform

Elastic Beanstalk Deployment Modes

Single instance: (DEV)
High Avail with LB (PROD)

Deployment Options for Updates:

- All at once (deploy all in one go) - fastest but instances aren't available to serve traffic (bit of downtime)
- Rolling - Updates a few instances at a time (bucket), and then move onto the next bucket once the first bucket is healthy
- Rolling with additional batches - Like rolling, but spins up new instances to move the batch (so old is still available)
- Immutable - spins up new instances into a new ASG, deploys version to these instances, then swaps all the instances when everything is healthy $$$$

Beanstalk for SysOps

- Beanstalk can put app logs into CloudWatch Logs
- You manage the application, AWS will manage the underlying infras
- Know the diff deployment mods for your app
- Custom domain: Route 53 or CNAME on top of Beanstalk URL
- You are not responsible for patching the runtimes (Nodejs, PHP, etc.)

How Beanstalk deploys apps ex: ROLLING

1. EC2 has a base AMI (Can Config)
2. EC2 gets the new code for the app
3. EC2 resolves the app dependencies **(can take awhile)**
4. Apps get swapped on the EC2 instance

How can you shorten this time with a lot of dependencies?

Use golden AMI: standardized company specific AMI w/
Package OS
Package App dependencies

Package company-wide software

Troubleshooting Beanstalk

- Review environment events
- Pull logs to view recent log file entries
- Roll back to a previous working version of the app
- When accessing external resources make sure the sec groups are correctly configured
- In case of command timeouts you can increase the value for processing

CLOUDFORMATION

- Infras as code
- No resources are manually created, which is excellent for control
- The code can be version controlled using git
- Changes to the infras are reviewed through code
- Can easily track costs by using resource identifiers so you know how much a stack costs you
- You can estimate cost using the CloudFormation template
- Savings strategy: In dev, you could automate deletion of templates at 5PM and recreate at 8 AM, safely
- Productivity - create and destroy easily
- Automated generation of diagram for your templates!

Separation of concern: create many stacks for many apps and many layers-

Example:

- VPC stacks
- Network stacks
- App stacks

Don’t reinvent the wheel-

- Leverage existing web templates
- Documentation is extensive

How it works

- Templates have to uploaded to S3 and referenced in CF

- To update a template, we can’t edit a previous one. You have to reupload a new version of the template each time
- Stacks are ID’d by a name
- Deleting a stack deletes every single artifact that was created by CF

Deploy Templates

Manual Way-

- Editing templates in the CF designer
- USing the console to input params, etc

Automated Way-

- Editing templates in a YAML file
- Using the AWS CLI to deploy the templates
- Recommended way when you want to automate your flow

Building Blocks

1. Resources: your AWS resources declared in the template
2. Parameters: the dynamic inputs for your template
3. Mapping: the static variables for your template
4. Outputs: references to what has been created
5. Conditionals: List of conditions to perform resource creation
6. Metadata

YAML CRASH COURSE

You can use YAML and JSON for CF but JSON is unreadable. YAML is the preferred way.

- Key value Pairs
- Nested objects
- Support Arrays
- Multi line strings
- Can include comments!

CF Parameters

- Allow you to reuse your templates across the company
- Some inputs can’t be determined ahead of time (Don’t have all info from DEV)

- Use when: Is this CF resource config likely to change in the future? If so, use a parameter.
- How to reference a parameter: The Fn:Ref function. Parameters can be used anywhere in a template. The shorthand for YAML is !Ref. These are specified at the top of your CF template. Used for parameters or resources and runs from top to bottom.

CF Resources

- Resources are the core of CF templates
- They represent different AWS components that will be created and configured
- Resources are declared and can ref each other
- AWS figures out creation, updates and deletes resources for you
- There are over 244 types of resources
- Resource types identifiers are of the form:

AWS::aws-productname::data-type-name

CF Mappings

- Mappings are fixed variables within CF templates
- They're very handy to differ between diff types of environments (dev vs. prod), regions, AMI types, etc
- All values are hardcoded within the template

When to use Mappings vs. Parameters

- Mapping are great when you know in advance the values that can be taken and that they can be deduced from variables such as:
- Region
- AZ
- AWS Account
- Environment
- Etc

***CF Outputs* IMPORTANT**

- Outputs section declares optional outputs values that we can import into other stacks (if you export first)
- You can also view the outputs in AWS Console or the AWS CLI
- Great to Output VPC ID and Subnet ID from your CF template
- It's the best way to perform collab cross stack, as you let experts handle their part of the stack
- You can't delete a CF Stack if its outputs are being referenced in another CF stack

- *EXPORT: NAME: SSHSECGROUP* is what's used to create the output of the info you request
- Fn::ImportValue grabs the output value and inputs it in: *!ImportValue SSHSECGROUP*

CF Conditions

- Used to control the creation of resources or outputs based on a condition
- Conditions can be whatever you want they to be common ones are: Environment, AWS Region, Parameter Values
- Each condition can ref another condition, parameter value or mapping
- Functions as:

Fn::And
Fn::Equals
Fn::If
Fn::Not
Fn::Or

- Conditions can be applied to resources/outputs/etc

Functions to Know for Exam

- Fn::Ref - parameter - returns value of the para | resource - returns ID of the underlying resource | Shorthand for this in YAML !Ref
- Fn::GetAtt - attributes are attached to every resource you create | AZ, priv DNS name, public DNS, private DNS, pub IP, private IP
- Fn::FindInMap - returns a named value from a specific key | YAML !FindInMap
- Fn::ImportValue - imports from exported values in other templates | YAML !Import
- Fn::Join - Join values with a delimiter | combines comma lists | YAML !Join [":", [a, b, c]] | Creates: "a:b:c"
- Fn::Sub - YAML !Sub - Substitute Values
- Condition Functions (Above)

EC2 Storage

EBS -

- An EBS volume is a network drive you can attach to you instance while they run
- It allows your instances to persist data
- Can be removed and reattached easily
- Cannot move across AZs
- Has a provisioned capacity (based on total data not used)

- Increase the drive size (optimisation phase) usable but not ideal
- Snapshots stored in S3
- Can copy across region
- Can make AMI from snapshot

Volume Types

- GP2 (SSD) - Recommended for most workloads, system boot volumes, virtual desktops, low latency
  I/O Burst - Can burst similar to t2/t3 CPU
  Get burst credit over time - good perf when needed. Bigger the volume the more credits acquired. Monitor with CloudWatch.
  No credits for over 1TB volumes.
- IO1 (SSD) Highest Perm SSD (database) - Critical biz apps that require sustained IOPS performance. MongoDB, Cassandra, MS SQL, MySQL, PostgreSQL, Oracle SIZE: 4 GB -16TB
- STI (HDD) Low cost HDD volume for frequent access - Streaming workloads, Big Data and Log processing, Cannot be boot volume, Apache Kafka SIZE: 500 GB - 16TB
- SCI (HDD) Lowest cost HDD volume for less frequently accessed data - When low cost is important, Cannot be boot volume SIZE: 500 GB - 16 TB, Max throughput of 250 MBPs

EBS Migrations

- Snapshot and move to different AZ
- EBS backups use IO and you shouldn't use when application is handling traffic
- Create new volume from Snapshot after moving to preferred AZ

EBS Encryption

- Data at rest encrypted inside the volume
- Snapshots and volumes created from are encrypted
- AES-256 using KMS
- Snapshot Unencrypted EBS volume create new volume from Snap and attach to original instance

EBS vs Instance Store

- Instance store is physically attached to the machine (EBS is a network drive)
- Pros: Better IO Perf and Good buffer for Instance Store
- Cons: On stop or term, instance store is lost. Can't resize the instance store. Backups must be operated by the user.

For SysOps

- If you plan to use root vol after termination term delete on term to NO
- If you use EBS for high performance use EBS-optimized instance types
- If EBS volume is unused you still pay for it
- For cost saving over a period, it can be cheaper to snapshot a volume and restore it later

Troubleshooting

- High wait time slow - increase IOPS
- EC2 won't start with EBS volume as root. Make sure volume names are properly mapped (/dev/xvdb instead of /dev/xvda)
- After increasing volume size, you still need to repartition the incremental storage

RAID Options

- Possible if OS supports it
- RAID 0, RAID 1, RAID 5 & 6 aren't recommended
- RAID 0 = Performance - If one disk fails all data is failed (Good for High IOPS)
- RAID 1 = Fault Tolerance - Mirrors one volume to another (Uses 2x network) - Application that needs increased fault tolerance | Application where you need to service disks

CloudWatch for EBS

- VolumeIdleTime: # of seconds when no read/write submitted
- VolumeQueueLength: # of ops waiting to be executed (app issues or IOPS limit)
- BurstBalance: if 0, we need a volume with higher IOPS
- GP2: 5 min interval IO1: 1 min interval

EFS (Elastic File System)

- Managed NFS that can be mounted on many EC2
- EFS works with EC2 instances in Multi-AZ
- Highly avail, scalable, expensive (3x gp2), pay per use
- Use case: CMS, Web Serving, Data Sharing, WordPress
- Sec Groups control access
- Works with Linux but not Windows
- Encryption using KMS

S3

- Allows the storage of objects (files)
- Buckets must have a globally unique name
- No uppercase or underscore, 3-63 char long, not an IP, must start w/ lowercase

Objects

- Stored as a key (entire full path) uses bucketname/folder/object.txt
- Object Values are the content of the body can be up to 5TB
- If uploading more than 5GB, must use “multi-part upload”
- Version ID (If versioning is enabled)

Versioning

- Enabled on bucket level
- If overwrite will increment the objects
- Best practice to protect against unintended deletes or revert back
- Any file that is not versioned prior to enabling with have *NULL*

Encryption

- SSE-S3 - Encrypts S3 objects using keys handled & managed by AWS | Encrypted Server Side using AES-256
- SSE-KMS - Leverage AWS KMS to manage | Same as SSE-S3 but give you an audit trail and user control
- SSE-C - manage your own keys | HTTPS only from the customer using
- Client Side Encryption - Client must encrypt before sending data to S3 | Decryption by Client as well | Client manages both functions
- ***Encryption in transit - AWS S3 exposes: HTTP endpoint: non encrypted, HTTPS is encrypted | Encryption in Flight = (SSL/TLS)***

Security

- User based = IAM Policies - call API to verify access or deny by user
- Resource based = Bucket Policies - bucket wide rule from S3 which allows cross account and it easier to manage | JSON Based | Set allow or deny and gives effect
- Networking - Supports VPC endpoints
- Logging and Audit - S3 access logs can be stored in another S3 bucket
- API calls can be logged in AWS CloudTrail
- User Security - MFA can be required in versioned buckets to delete objects | Signed URLs - valid for limited time (video service for logged in users)

Websites

- Can host static sites open to WWW | URL ex: <bucket-name>.s3-website-<AWS-region>.amazonaws.com
- 403 errors require the bucket to allow public reads

CORS

- Cross Origin Resource Sharing - allows you to limit the number of websites that can request your files in S3
- If you request data from another S3 bucket, you need to enable CORS

Consistency Model

- Read after write consistency for PUTS of new objects ex (PUT 200 -> GET 200)
- This is true, except if you do a GET before the object existed (cache)
- As soon as you write you can recieve it (Read after Write)

S3 FOR SYSOPS

S3 Versioning

- Versioning also happens when you encrypt a file this allows you to back out of ransomware
- Deleting a file in the S3 bucket just adds a delete marker
- To delete a bucket, you have to remove all file versions within it
- ls command won't list delete markers (versions) that still exist

MFA

- Forces users to use hardware key or phone when doing important ops on S3
- To use versioning must be enabled
- You will need MFA to: permanently delete an object version. Suspend versioning on the bucket. Won't need to: enable versioning, list del versions
- Only the bucket owner can enable/disable MFA-Delete
- MFA-Delete can only be enabled using the CLI

Default Encryption vs Bucket Policies

- Option you enable in console when creating a bucket

Access Logs

- Log all access to S3 buckets for audits
- Any request from any account, auth or denied will be logged into another S3 bucket
- Can be analyzed using data analysis tools or using Amazon Athena

Cross Region Replication

- Versioning must be enabled on source and destination
- Buckets must be in different AWS regions
- Can be in different accounts
- Must give proper IAM permissions to S3

Pre-Signed URLs

- Can generate using SDK or CLI (For Downloads can use CLI) (For Uploads can use SDK)

Inventory

- Helps manage storage
- Audit and report on the replication and encryption status of your objects
- You can query all the data using Athena, Redshift, Presto, Hive, Spark
- Data goes from a source bucket to a target bucket (setup policy)

Storage Tiers

- Standard - General Purpose | (11 9s) across multiple AZs | 99.99% avail
- Standard-Infrequent Access (IA) | (11 9s) across multiple AZs | Low Cost | Disaster Recovery Data
- One Zone-Infrequent Access | Lost when AZ is destroyed | 99.95% avail | Lower than IA by 20% | Storage secondary backup copies
- Reduced Redundancy Storage (Depreciated) (4 9s) | Non-critical Data | Don’t Use
- Intelligent Tiering (New) | Same low latency as Standard | Small monthly monitoring and auto-tiering for small fee
- Glacier

Lifecycle Rules

- Can setup to move data between diff tiers, to save cost
- Example: Standard -> IA -> Glacier (After X Days)
- Allows cleanup of multipart uploads if not finished (After X Days)

Analytics

- You can setup to help determine when to transition objects from Standard to IA
- Does not work for ONEZONE_IA or GLACIER
- Report updates daily
- Takes about 24h to 48h to generate on first use
- Helps put together lifecycle rules

Storage Gateway

- 3 Types
- Hybrid Cloud Storage
- Gives S3 access through a gateway
- Configured S3 buckets are accessible using the NFS and SMB protocol

File Gateway

- Bucket access using IAM roles
- Most recently used data is cached
- Can be mounted on many servers

Volume Gateway

- Block storage using iSCSI protocol backed by S3
- Backed by EBS snapshots which can help restore on-premise volumes
- Cached volumes: low latency access to the most recent data
- Stored volumes: entire dataset is on premise, scheduled backups to S3

Tape Gateway

- Some companies still use tape
- Virtual Tape Library backed by S3 and Glacier
- Back up data using existing tape-based process
- Works with leading backup software vendors

HINTS

- File access / NFS = File Gateway (Backed by S3)
- Volume / Block / iSCSI = Volume gateway (S3 with EBS snapshots)
- VTL Tape Solutions / Backup = Tape Gateway (S3 & Glacier)

GLACIER

- Low cost object storage for archiving/backup
- Data is retained for the longer term (10s of years)
- Alt to on-premise tape storage
- Cost is ($.004 / GB) + Retrieval Cost
- Each item in glacier is called “Archive” (Up to 40 TB)
- Stores in Vaults not Buckets
- Functions - Upload (MultiPart Support) | Download | Delete

Retrieval Options

- Expedited (1 to 5 mins) - .03 per GB and .01 per request (Must Purchase Capacity Units)
- Standard (3 to 5 hours) - .01 per GB and .05 per 1000 requests
- Bulk (5 to 12 hours) - .0025 per GB and .025 per 1000 requests

*Value Policies & Vault Lock

- Vault is a collection of archives
- Each Vault has one access policy and one lock policy
- Vault policies are written in JSON
- Lock policies are for regulatory and compliance requirements | These policies can never be changed once created | WORM (write once read many)

SNOWBALL

- Used to move large amounts of data to AWS Data Centers
- Pay per transfer job
- Secure/Tamper Resistant uses KMS 256
- Supports SNS Alerts | Test | AWS Console Alerts
- Data is loaded into S3 buckets

CLOUDFRONT

- Content Delivery Network (CDN)
- Improves read performance, content is cached at the edge
- 136 Points on Presence globally (edge locations)
- Popular with S3 but works with EC2, load balancing
- Can help prevent network attacks

- Can provide SSL (HTTPS) at the edge using ACM
- Supports RTMP Protocol (videos/media)
- Origin Access Identity (Access account to ensure only this user accesses and goes through CF) Created Distribution is the option used to create (Restrict Bucket Access needs to be enabled to ensure S3 URLs aren't exposed)

Access Logs

- CloudFront access logs: logs every request made to CF into a logging S3 bucket

Reports

- Cache Statistics
- Popular Objects
- Top Referrers
- Usage
- Viewers
- These reports are based on data from the access logs

Troubleshooting

- CloudFront caches HTTP 4xx and 5xx status codes returned by S3 (origin server)
- 4xx error indicates the user doesn't have access (403) or the object doesn't exist (404)
- 5xx error codes indicate Gateway issues

Athena

- Serverless service to perform analytics directly against S3 files
- Uses SQL language to query the files
- Has a JDBC / ODBC driver
- Charged per query and amount of data scanned
- Supports: CSV, JSON
- Use Case: BI / Analytics / Reporting, Analyze & Query, VPC Flow logs, ELB Logs, CloudTrail, etc…

HINT

- Analyze data directly on S3 = Use Athena

DATABASES

Parameter Groups

- Can configure the DB engine using Parameter Groups
- Dynamic parameters are applied immediately
- Static parameter are applied after instance reboot
- You can modify parameter group associated with a DB (must reboot)

RDS

- It’s a managed DB service for DB use SQL as a query language
- Allows you to create DBs in the cloud that are managed by AWS and supports: Postgres, Oracle, MySQL, MariaDB, Oracle, MS SQL, Aurora (AWS)
- Advantages over deploying DB on EC2: OS Patches | Continuous backups and restore to specific timestamps | Monitoring Dashboards | Read replicas for improved read perf | Multi AZ for DR | Maintenance windows for upgrades | Scaling capability (vertically and horizontal | Can’t SSH to…

API for SysOps

- DescribeDBInstances - Helps get a list of all DB Instances you have deployed including Read Replicas | Helps get DB version
- CreateDBSnapshot - Makes snapshot of DB
- DescribeEvents - Helps return info about events related to your DB instance
- RebootDBInstance - Helps to initiate a ‘forced’ failover by rebooting

Read Replicas

- Up to 5 read replicas
- Within AZ, Cross AZ or Cross Region
- Replication is ASYNC so reads are eventually consistent
- Replicas can be promoted to their own DB
- Applications must update the connection string to leverage read replicas
- Helps scale read traffic (SPEED UP Queries)
- Can be promoted as a standalone DB (manually)
- Each replica has its own DNS endpoint
- Can have read replicas of read replicas
- Clones of the main DB
- Can be Multi-AZ
- Help with DR by using Cross Region
- Not supported by Oracle

- Can be used to run BI / Analytics Reports

Multi AZ (DR)

- SYNC Replication
- One DNS name automatic app failover to standby (name convention)
- Increases availability (Automatic recovery should RDS get stuck or fail)
- Failover in case of loss of AZ, loss of network, instance or storage failure
- No manual intervention in apps
- Multi AZ does not support the reads | Only fails over if primary fails, AZ outage, The DB instance server type is changed, the OS of the DB instance is undergoing patching, manual reboot of the DB instance was initiated using reboot with failover | No failover for DB operations: long running queries, deadlocks or db corruption errors | Endpoint is the same after failover (no URL change in app) | Backups are created from the standby
- Only within a single region, not cross region. Region outages impact availability.
- If RDS Backups impact of PROD DB performance Enable Multi AZ

Backups

- Automated Backups - Daily full snapshot of the database | Capture transaction logs in real time | Ability to restore to any point in time | 7 days of retention by default (can set anywhere from 0-35 days)
- DB Snapshots - Manually triggered by user | Retention of backup for as long as you want (Pre-Build or Long Term Backup) | If taken on Multi-AZ DB this doesn’t impact the master - just standby | After the first (FULL) all others are incremental (FASTER) | Can COPY and SHARE

Encryption

- Encryption at rest with KMS (Must be enabled)
- SSL Certs to encrypt data in flight
- *To ENFORCE SSL - ProsgreSQL: rds.force_ssl=1 in AWS RDS Console* | *MySQL: within DB - GRANT USAGE ON *.* TO ‘mysqluser’@’%’ IDENTIFIED BY ‘...’ REQUIRE SSL;*
- *To CONNECT using SSL - Provide the SSL trust cert* | *Provide SSL options when connecting to database*

Security for SysOps

- RDS databases are usually deployed within a private subnet, not public
- Works by leveraging sec groups
- IAM policies help control who can MANAGE AWS RDS

- Traditional Username and Password can be used to LOGIN to the DB
- IAM users can bo by used for (MySQL / Aurora)
- Encryption at rest can only be done when creating the DB instance
- Can unencrypted -> snapshot -> copy snapshot as encrypted - > create DB from snapshot (Same as EBS method)
- Check Ports / IP / Sec Group - inbound rules in DBs SG
- In-database user creation and perms
- Create a database with or without public access
- Ensure parameter groups or DB is configured to allow SSL connection
- Can't SSH, No Manual Patching of DB/OS | Can't audit the underlying instance

CloudWatch Metrics

- DatabaseConnections
- SwapUsage
- ReadIOPS / WriteIOPS
- ReadLatency / WriteLatency
- DiskQueueDepth
- FreeStorageSpace
- Enhance Monitoring (gathered from an agent on DB instance) - Useful when you need to see processes and thread use on CPU | Access to over 50 new CPU, mem, file sys and disk I/O metrics

Performance Insights

- Visualize your DB perf and analyze any issues that affect it
- Can visualize the DB load and filter: By Waits -> Find the resource bottleneck | By SQL statements -> find the SQL statement that is the problem | By Hosts - find the server that is using the most of the DB | By Users - find the user that is using the most of the DB
- DBLoad - The number of active sessions for the DB engine
- Can view the SQL queries that are putting load on your database
- Requires one of the following to run:

-Aurora with MySQL 1.17.3+
-RDS MySQL version 5.7.22+
-RDS MySQL version 5.6.41+
-Aurora with PostgreSQL compatibility
-RDS PostgreSQL version 10
-RDS Oracle (All Versions)

Aurora

- Aurora is an AWS tech (not open source)
- Postgres and MySQL are both supported as Aurora DB (Means your drives will work as if Aurora was a Postgres or MySQL DB)
- Aurora is Cloud Optimized and claims 5x performance improvement over MySQL on RDS and 3x performance of Postgres on RDS
- Aurora storage automatically grows in increments of 10GB, up to 64TB
- Aurora can have 15 read replicas while MySQL has 5, and the replication process is faster (sub 10 ms replica lag)
- Failover multi AZ by default
- Costs 20% more than RDS but more efficient

High Avail and Read Scaling

- 6 copies of your data across 3 AZs | 4 copies out 6 needed for writes (2 AZs) | 3 copies out of 6 needed for reads | Self healing with peer-to-peer replication | Storage is striped across 100s of volumes
- Only one Aurora instance takes writes (master)
- Automated failover for master in under 30 seconds
- Aurora can have 15 read replicas from 1 master
- Support for Cross Region
- Replication | Self Healing | Auto Expanding

Features

- Automatic failover
- Backup and Recovery
- Isolation and Security
- Industry Compliance
- Push-Button scaling
- Automated Patching with ZERO Downtime
- Advanced Monitoring
- Routine Maint
- Backtrack: restore data to any point in time without using backups

Security

- Encrypted as rest w/ KMS
- Automated backups, snapshots and replicas are also encrypted
- Encryption in flight using SSL

- Auth using IAM
- You are responsible for protecting the instance with sec groups
- You can't SSH

Aurora Serverless

- No need to choose instance size
- Only supports MySQL 5.6 (Jan 2019) & Postgres (BETA)
- Helpful when you can't predict the workload
- DB cluster starts, shutdown and scales automatically based on CPU / connections
- Measured by ACU (Aurora Capacity Units) - Billed in 5 minute increments

ElastiCache

- RDS for caches
- Managed Redis or Memcached
- Caches are in-memory DBs with really high perf, low latency
- Helps reduce load off of DBs for read intensive workloads
- Helps make your application stateless
- Write scaling using sharding
- Read scaling using read replicas
- Multi AZ with Failover Capability
- AWS does: OS maint/patching/optimizations/setup/config/monitoring/failure recovery and backups
- Application queries ElastiCache. If not avail, gets from RDS and stores in ElastiCache.
- If user hits another application instance it still using the cached data from the previous instance. User doesn't have to login again if passed between instances.
- *Relieves load of the DB and share state (user sessions store) all apps can be stateless*

Redis

- Is an in-mem key-value store
- Super low latency
- Cache survive reboots by default (persistence)
- Great to host - User sessions | Leaderboard (Gaming) | Distributed States
- Relieve DB pressure (Such as RDS)
- Multi AZ with Automated Failover for DR
- Support for Read Replicas

Memcached

- In-mem object store
- Cache doesn’t survive reboots
- Uses: Quick objects from memory | Cache often accessed objects
- Overall Redis is better

Monitoring, Auditing and Performance

- CloudWatch provides metrics for every service in AWS
- Metric is a variable to monitor (CPUUtilization, NetworkIn, etc
- Metrics belong to namespaces
- Dimension is an attribute of a metric (instance id , evn, etc.)
- Up to 10 dimensions per metric
- Metrics have timestamps
- Can create CloudWatch dashboards of metrics

CloudWatch EC2 Detailed Monitoring

- Instance metrics update every 5 mins
- With detailed (cost) can get every 1 min
- Use detailed if you want to more prompt scale your ASG
- Free tier allows you to have 10 detailed monitoring metrics
- Memory usage for EC2 is by default not pushed (must be pushed from inside the instance as a custom metric)

CloudWatch Custom Metrics

- Possible to define and send your own metrics to CW
- Ability to use dimensions (attributes) to segment: Instance.id Enviornment.name
- Metric Resolution: 1 min, Up to 1 second
- Use API call PutMetricData
- Use exponential back off in case of throttle errors (increase time for metric check)

CloudWatch Dashboard

- Setup for quick access to key metrics
- Dashboards are global
- Dashboards can include graphs from diff regions
- You can change time zone and time range of the dashboards
- You can setup auto refresh (10s, 1m, 2m, 5m, 15m)
- Pricing: 3 Dashboards (up to 50 metrics) for free | $3/dashboard/month afterwards

CloudWatch Logs

- Applications can send logs to CloudWatch using the SDK
- CloudWatch can collect logs from:
- Elastic Beanstalk: collection of logs from application
- ECS: collection from containers
- AWS Lambda: collection from function logs
- VPC Flow Logs: VPC specific logs
- API Gateway
- CloudTrail based on filter
- CloudWatch log agents: for example on EC2 machines
- Route53: Log DNS queries

- CloudWatch logs can go to:
- Batch exporter to S3 for archiving
- Stream to ElasticSearch cluster for further analytics

- Log storage architecture
- Log groups: arbitrary name, usually reps an application
- Log stream: instance within app/log fails/containers

- Can define log expiration policy (never exp, 30 days, etc.)
- Using the AWS CLI you can tail CloudWatch logs
- To send logs to CloudWatch make sure IAM is properly configured
- Can encrypt logs using KMS at group level

CloudWatch Alarms

- Alarms trigger notifications for any metrics
- Alarms can go to Auto Scaling, EC2 Actions, SNS Notifications
- Various Options (sampling, %, max, min, etc.)
- Alarm States: OK, INSUFFICIENT_DATA, ALARM
- Period: Length of time in seconds to evaluate | Custom Metrics: can only choose 10 sec or 30 sec
- EC2: Stop, Terminate, Reboot, or Recover | Trigger Auto Scale Option | Send notification to SNS
- Can be created based on CloudWatch Logs Metrics Filters
- CloudWatch doesn’t validate of test the action that is assigned
- To test alarms or notifications, set the alarm state to ALARM using the CLI:
  aws cloudwatch set-alarm-state --alarm-name “myalarm” --state-value ALARM
  --state-reason “testing”

CloudWatch Events

- Source + Rule => Target
- Schedule: Cron jobs
- Event Pattern: event rule to react to service doing something
- Trigger Lambda, SQS/SNS/Kinesis Messages
- They create a small JSON doc to give info about the change

CloudTrail

- Used to provide governance, compliance and audit AWS
- Tracks every API call from everywhere
- Is Enabled by Default
- Can put logs from CloudTrail into CloudWatch Logs

TIP: If a resource is deleted, check CloudTrail first

- Shows only the past 90 days of activity
- The default UI only shows: Create, Modify and Delete
- CloudTrail Trail allows you to get any events you choose | Ability to store in S3 | Can be region or global | Stored using SSE-S3 encryption by default | control access using S3 IAM or Bucket Policy

Config

- Helps with auditing and compliance of your AWS resources
- Helps record configs and changes over time
- Helps record compliance over time
- Possibile to store in AWS S3 (can query w/ Athena)
- Questions solved with Config: Is there unrestricted SSH access to my sec grps? | Do my buckets have any public access? | How has my ALB config changed over time?
- Can setup SNS notifications for any changes
- Per-region service
- Can be used across regions and accounts
- AWS managed config rules (over 75)
- Can make custom config rules (defined in AWS Lambda)
- Triggers for rules: each time config changes | and/or at interval time
- Pricing: No free tier | $2 USD per rule per region per month (less after 10 rules)
- In console shows compliance of a resource over time | can view CloudTrail API calls to track down who did what

CloudWatch vs CloudTrail vs Config

- CloudWatch is for perf metrics and dashboards | events & alerts | log aggregation & analysis
- CloudTrail records api calls made within your account by everyone | can define trails for specific resources | global service
- Config records config changes | evaluate against compliance rules | get timeline of changes and compliance through UI

AWS Account Management

Health Dashboard

- Shoes all regions and all service health
- Shows historical info for each day
- Has an RSS feed you can subscribe to

Personal Health Dashboard

- Global service
- Shows how AWS outages directly impact you
- Shows impact on your resources
- Lists issues and actions you can do to remediate

AWS Organizations

- Global
- Allows you to manage multiple AWS accounts
- The main account is master account
- Other accounts are member accounts
- Member accounts can only be apart of one organization
- Consolidated billing across all accounts - one payment method
- Pricing benefits
- API is avail to automate account creation
- SCPs - OUs and Service Control Policies | can nest OU within OU | Apply service control policies to OU | SCPs are a filter to IAM | Helpful for sandbox account creation and separating DEV from PROD

AWS Service Catalog

- Users that are new to AWS have too many options and may create stacks that are not compliant with the rest of the organization
- Some users just want a quick self-service portal to launch a set or authorized products by admin
- Create and manage catalogs of IT services that are approved by admins of the account
- The “products” are CloudFormation templates, this helps with consistency and standardization
- They are assigned to portfolios (teams)
- Teams are presented with a portal where they can launch products
- Easy for users to use without deep AWS knowledge

AWS Billing Alarms

- Stored in US-EAST-1 ONLY
- Billing data are for overall worldwide AWS costs

AWS Cost Explorer

- Graphic tool to view and analyze cost/usage
- Review charges and cost associated with your AWS account or org
- Forecast spending for the next 3 months
- Get recommendations for which EC2 reserved instance to purchase
- Access to default reports

AWS Budgets

- Create budget and send alarms when cost exceeds the budget
- 3 types: Usage, Cost, Reservation
- For RESERVED INSTANCES: Track utilization on ec2, elasticache, rds, redshift
- Filters: service, linked account, tag, purchase option, instance type, region, az, api operation, etc.
- 2 free, then $.02 day/budget

AWS Cost Allocation Tags

- With Cost Allocation Tags we can enable detailed cost reports
- Just like TAGS, but they show up as columns in reports
- Auto gen to resources | aws:createdBy | Not applied to resources created before the activation of service
- Users tags can be defined by users | user:
- Just appear in billing console but can take up to 24 hours to populate

Security & Compliance

AWS Responsibility

- Protection of infras (hardware, software, facilities and networking)
- Managed services like S3, DynamoDB, RDS etc

Customer Responsibility

- You're responsible for EC2 sec groups and guess OS, IAM roles etc

RDS

AWS:

- Automated DB and OS Patching
- Manage the underlying EC2 instance, disable SSH access
- Audit the underlying instance and disk and guarantee it functions

Customer:

- Check ports/ip/sec group inbound rules for db
- In-database user creation and perms
- Creating a DB with or without public access
- Ensure parameter groups or DB is configured to only allow SSL connections

S3

AWS:

- Guarantee you get unlimited storage
- Guarantee you get encryption
- Ensure separation of data between diff customers
- Ensure AWS employees can't access data

Customer:

- Bucket config
- Bucket policy/public setting
- IAM users and roles
- Enabling encryption

Protection Products

- AWS Shield Standard: No additional cost DDoS protection by default
- Shield Advanced: 24/7 premium DDoS protection | Protection against more sophisticated attacks on CF, Route 53, LBs & Elastic IP/EC2

- AWS WAF: Filter specific requests based on rules | Protects from common web exploits | Allow and block web traffic | Protection from SQL injection and XSS | Protects against bots, base user agents, etc. | Size constraints | Geo blocking | Deploys on CF, Application LB and API gateway | Leverage existing marketplace of rules
- CloudFront and Route53: Protection using global edge network | This combined with AWS Shield provides attack mitigation at the edge
- Be ready to scale - leverage AWS Auto Scaling
- Separate static resources (S3/CloudFront) from dynamic ones (EC2/ALB)

Pentesting

- Permission is required for all pen tests
- Request permission using AWS root creds
- You must perform pen testing yourself - no 3rd party
- Applies to: EC2, ELB, RDS, Aurora, CF, API Gateway, Lambda, Lightsail
- Cannot test against nano/micro or small EC2 instances
- Takes 2 days to be approved

Inspector

- Only for EC2 instances
- Helps analyze known vulns
- Analyze against unintended network access
- Requires inspector agent to be installed on guest OS of EC2
- Define Templates which AWS manages
- No custom rule reporting

Security Logging

- Service Logs: CloudTrail trails - trace all API calls | Config rules - track config and compliance over time | CloudWatch logs - for full data retention | VPC Flow logs - IP traffic within your VPC | ELB Access Logs - metadata of requests made to LBs | CloudFront Logs - web distribution access logs | WAF logs - full logging of all requests analyzed by the service
- Logs can all be seen in Athena if stored in S3
- You should encrypt logs in S3, control access using IAM/Bucket Policies/MFA
- Move logs to Glacier for cost savings

GuardDuty

- Intel threat discovery to protect AWS Account
- Uses ML, anom detection and 3rd party data
- One click to enable (30 day trial), no need to install anything

- Input data includes: CloudTrail Logs - unusual API calls, unauthorized deployments | VPC Flow Logs: unusual internal traffic, unusual IP address | DNS Logs - compromised EC2 instances sending encoded data within DNS queries
- Notifies you in case of findings
- Integration with AWS Lambda

Trust Advisor

- Analyzes your AWS accounts and provides recommendations: Cost Optimization | Performance | Security | Fault Tolerance | Service Limits
- Core Checks and Recommendations - all customers
- Can enable weekly email notification from the console

Encryption

- SSL - encryption in flight (data encrypted before sending, server decrypts) HTTPS | Protects from MITM attacks
- Server side encryption at rest - data is encrypted after being received by the server | Decrypted and passed back over HTTPS
- Client side encryption - encrypted by client and never decrypted by the server | Data will be decrypted by a receiving client | The server should not be able to decrypt the data | Cloud leverage envelope encryption

KMS

- Key provider for encryption of service data
- Create/Rotate Policies/Disable/Enable
- Audit key usage using CloudTrail
- Fully integrated with IAM for authorization
- EBS, S3, RedShift, RDS, SSM, etc
- Can you CLI/SDK to encrypt on your side
- Anytime you need to share sensitive info… use KMS
- You never see the CMK (Customer Master Key) in KMS
- Encrypted secrets can be stores in the code or environment variables
- KMS can only encrypt up to 4KB of data per call
- If data is bigger than 4KB use envelope encryption
- User keys created in KMS - $1/mo.
- Only symmetric!

Require Migration for KMS Encryption (Snapshot/Backup)

- EBS Volumes
- RDS Databases

- ElastiCache
- EFS Network File System

In-place Encryption

- S3 on the fly!

CloudHSM

- AWS provisions encryption hardware
- Dedicated hardware
- You manage your own encryption keys entirely
- Hardware is tamper resistant
- FIPS 140-2 Level 3 Compliant
- Clusters are spread across multi AZ
- Supports both symmetric and asymmetric encryption
- No free tier
- Must use CloudHSM Client Software

IAM + MFA

- Added layer of security while accessing AWS account
- Accepts both virtual and hardware
- MFA for root user can be configured from IAM dashbaord
- MFA can also be configured using CLI
- Credential Report - CSV file on all the IAM users/creds | Shows who all have enabled MFA

IAM

- PassRole Option - In order to assign a role to an EC2 instance you need IAM:PassRole (applies to a bunch of services)

STS

- Security Token Service
- Allows to grant limited and temp access to AWS resources
- Valid for up to one hour (must be refreshed)
- Cross Account Access - Define IAM role for another account | Define which accounts can access the IAM role | Issue STS to impersonate the role (AssumeRole API)
- Federation with Cognito - Makes use of Facebook/Google/Amazon logins

Identity Federation with SAML & Cognito

- Federation lets users outside of AWS to assume temp roles for accessing resources
- These users assume an ID provided access role
- Federation Assumes a form of 3rd party auth: LDAP, Microsoft AD, Single Sign On, Open ID, Cognito
- Using federation, you don't need to create IAM users (user management is done outside of AWS)
- SAML Federation (For Enterprise) - to integrate AD or ADFS with AWS | Provides access to AWS Console or CLI (through temp creds) | No need to create IAM users for each employee
- Custom ID Broker App - Use only if ID provider is not compatible with SAML 2.0 | The ID broker must determine the appropriate IAM policy (You Apply Policies)
- AWS Cognito Federated ID Pools (For Public Apps) - Provide direct access to AWS resources from the Client Side | Example: provide temp access to write to an S3 bucket using Facebook Login

Compliance Frameworks

- Just because AWS is compliant with Standards doesn't mean your environments are
- SHARED RESPONSIBILITY

Artifact

- Portal that provides customers with on-demand access to AWS compliance docs and AWS agreements
- Allows tracking of agreements
- Can be used to support internal audit and compliance

Route 53

- Managed DNS (Collection of Rules and Records to URLs)
- Can purchase domains (Registrar)
- Can use public domain names you own or buy
- Can use private domain so instances can resolve within VPC
- Load balancing (through DNS)
- Health Checks (limited)
- Routing Policy: simple, failover, geolocation, latency, weighted, multi value
- You pay .50 per mo per hosted zone

Most common records:

- A: URL to IPv4
- AAAA: URL to IPv6
- CNAME: URL to URL
- Alias: URL to AWS resource

A

- DNS request from browser -> Route 53 (hands back IP to Browser)

CNAME vs Alias

- AWS Resources (LB, CloudFront, etc) expose an AWS URL: lb.us-east.elb.amazonaws.com
- CNAME: Points to a URL only for non root domains! (app.domain.com)
- ALIAS: Root or Nonroot (*.domain) | Free of charge | Native health check

Route Policies

- Simple: maps a domain to one URL | Use to redirect to single resource (no health check) | If multiple IP values are returned client figures out which to use
- Weighted: controls the % of reqs going to multiple endpoints (split reqs on load) | Health Checks | Helpful to split traffic between two regions
- Latency: Redirect to server that has the lowest ping | User Priority | Latency is evaluated in terms of use to designated AWS region | (Germany could go to US if lowest latency)
- Failover: Uses one primary and one secondary | Must use health check
- Geo: Based on user location (Country to IP) | Should create a default policy (in case there's no match)
- Multi Value: Route to multiple resources | Want to associate Route 53 health checks with records | Up to 8 healthy records are returned for each Multi Value query | Not a substitute for ELB | Cleans records if unhealthy

Health Checks

- If health check fails 3x it's marked unhealthy same in reverse
- Default time is 30s, can set to 10s at a cost
- About 15 health checkers will check the endpoint health -> req every 2 seconds on average
- Can have HTTP, TCP and HTTPS health checks (no SSL verification)
- Health checks can be linked to Route53 DNS queries

VPC

CIDR

- are used for sec group rules or AWS networking in General | Help define a range of IPs | /32 is one IP | Components: IP and Subnet Mask
- Max cider size is /16

Default VPC

- All new accounts have | New instances are launched into default VPC | Includes all public traffic to Private IP ranges allowed | Includes public and private DNS name | Internet Gateway Attached

Subnets

- Subnet: /32 = 2^0 (one) | Subnet /32 = 2^1 (two) | Subnet /30 2^2 (four) | /0 (all)
- Should not overlap other networks
- AWS reserves 5 IP addresses in each subnet created | Cannot be assigned to an instance
- TIP: If you need 29 addresses for EC2 instances, you can't choose a subnet size of /27 = (32 IPs) = 22 | Need at least /26 = (64 IPs) = 59

Virtual Private Cloud - (up to 5 per region - soft limit)
Max CIDR per VPC is 5 - Min size /28 = 16 IPs | Max size /16 = 65536

Internet Gateways (Allow access to public subnets) and Route Tables (Required for any effect)

Network ACLs vs Security Groups

- NACLs are like a firewall which control traffic to and from subnet
- Default NACL allows everything outbound and inbound
- One NACL per Subnet, new Subnets are assigned to the default NACL
- Number system for precedence between (1-32766) lower value # checked first
- Fresh NACLs deny everything by default
- NACLs are a great way of blocking a specific IP at the subnet level
- If Sec Groups allow outbound traffic they let back in (Stateful)

VPC Endpoints

- Meant for you to access AWS services within a private network - CloudWatch/S3
- They remove the need of IGW, NAT, etc for access

- Interface EndPoint - an ENI (Private IP) as an entry point (must attach sec group)
- Gateway EndPoint - provisions target and must be used in a route table - S3\Dynamodb
- Troubleshooting: Check DNS Setting Resolution in your VPC | Check route tables

VPC Peering

- Connect two VPC privately using AWS' network
- Make them behave as if they're in the same network
- Requires CIDR to not overlap
- Required to connect two VPCs at any point
- When setup the route tables in each VPCs subnet must be changed to connect

Troubleshooting Connection Issues-

- Incoming REJECT : NACL or SG | Inbound Accept, Outbound Reject: NACL
- Outbound REJECT: NACL or SG | Outbound Accept, Inbound Reject: NACL

Inbound-

- NACL inbound rules are first thing checked for incoming traffic
- Checks inbound rules. If approved, passes requests to Security Groups rules
- NACL then checks outbound rules
- NACL is stateless
- Sec Group allows in and back out of EC2

Outbound-

- From EC2 SG outbound rules are checked first. If approved, passes to NACL outbound rules
- Passes to the NACL outbound rules request comes back via NACL inbound as stateless. If approved, passes to EC2 (stateful)

Flow Logs

- Help monitor & troubleshoot connectivity issues
- Flow log data can go to S3/CloudWatch Logs
- Captures network information from AWS managed interfaces to: ELB, RDS, ElastiCache, Redshift, WorkSpaces
- Show success or failure of connections as well as IPs/Ports
- To enable create Log Group in CloudWatch Logs and assign to VPC
- IAM rule can be automatically created in the create flow log VPC function must be assigned (Can take up to 10 minutes)

- Can be analyzed with Athena must target location (S3), create table and add partition and source followed by SQL query to see

Types-

- VPC Flow Logs
- Subnet Flow Logs
- Elastic Network Interface Flow Logs

Bastion Host

- Used to SSH into private EC2 instances
- Bastion Host Security Group must be tightened (Should be only port 22 from the IP you need)
- Runs as an EC2 Instance

Site to Site VPN

- Used to connect onsite infrastructure to AWS
- Customer Gateway (Client side) - Software application or device for routing out | IP Address is often Static IP | If behind NAT, must use public IP address of NAT
- Virtual Private Gateway (AWS side) - Created and attached to the VPC from which you want to create site to site

Direct Connect

- Provides dedicated PRIVATE connection from a remote network to your VPC
- Must be setup between your DC and AWS Direct Connection locations
- You need to setup a Virtual Private Gateway on your VPC
- Access public resources (S3) and private (EC2) on the same connection
- Use Case: Moving Large Data Sets | Fast | Hybrid Environments
- Can use IPv4 or IPv6
- If you want to set up more than one region for your VPCs you must use a Direct Connect Gateway

Egress Only Internet Gateway

- Only supports IPv6
- Similar to a NAT, but NAT is only IPv4
- IPv6 are all public IPs
- Instance access TO internet but NOT directly reachable from internet
- Must edit route tables to implement

Misc Notes

AMI Copying

- Share from one account to another
- Does not change ownership of the AMI
- If AMI is copied to another region that account becomes owner
- Can't copy AMI with a billingProduct code | must spin up new instance from AMI and create AMI

ASG

- For High Avail you need at least 2 instances running across AZs in your ASG.
- ASG will launch a new instance after terminating an unhealthy one
- ASG will not reboot unhealthy hosts for you

Troubleshooting

- Number of instances are already running. Means ASG has reached the DesiredCapacity parameter limit
- When launching EC2 instances is failing - SG does not exist. Keypair might have been deleted
- If ASG fails to launch for over 24 hours it suspends the process

Load Balancer for SysOps

- Application LB - layer 7 http/https and websocket - URL based routing - provides SSL termination
- Network LB - layer 4 - tcp - no pre-warming - no SSL termination
- Chain NLB and ALB together to give ALB a fixed IP
- Pre-warm - ELB scales gradually to traffic, if 10x traffic doesn't happen automatically must contact support.
- Error Code - 200 means success | 4XX at client is unsuccessful | 5XX at server unsuccessful 503 service unavail |

Route 53

- Simple
- Weighted
- Latency
- Failover
- Geolocation

- Multi Value

Cloud Formation

- cfn-signal after cfn-init - tells CF service to keep going or fail | Need to define WaitCondition and CreationPolicy
- Wait Condition troubleshooting - AMI your using doesn't have AWS CF scripts installed | Verify cfn-init and cfn-signal ran successfully on the instance
- You can retrieve logs by logging into your instance | you need to disable rollback on failure to access
- Verify the instance has connection to the internet | CF targets must have internet to log
- OnFailure=DO_NOTHING allows you to troubleshoot instances | OnFailure=DELETE (removes everything)

CloudWatch vs CloudTrail vs Config

- CloudWatch is for performance monitoring and dashboards | events/alerting
- CloudTrail is to record API calls made within your account by everyone | Can define trails for spec resources | Global
- Config records config changes | evals resources against compliance rules | Get timeline of changes and compliance