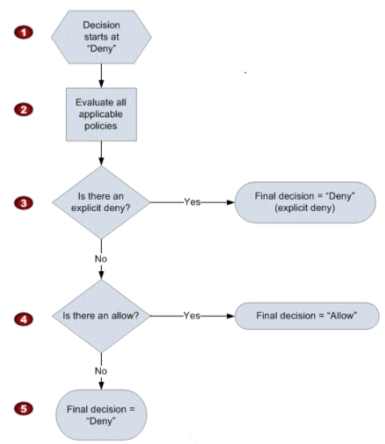
**AWS Certified Solutions Architect Exam Study Notes**

*Checkout my study notes for the CDA Exam. Can be useful for CSAA too;*[*AWS Certified Developer Associate Study Notes [WIP]*](https://renatoargh.wordpress.com/2018/03/01/aws-certified-developer-associate-study-notes/)

**IAM**

* Centralised control to your AWS account.
* Shared access to your AWS account.
* Granular permissions.
* Identity federations (connect to Active Directory or other identity providers).
* Set up multi-factor authentication.
* Provide temporary access for users or devices where necessary.
* Allows you to set up your own password rotation policy.
* Integrates with many different AWS services
* Critical terms:
  + User – End users (people).
  + Group – A collection of users under the same set of permissions.
  + Role – A set of permissions that you can assign to AWS resources.
  + Policies – A JSON document that defines one or more permissions. You can attach it to users, groups or roles.
* IAM is global. You don’t have to choose a region.
* Evaluation of IAM policies:
  + When an AWS service receives a request, the request is first authenticated using information about the access key ID and signature. (A few services, like Amazon S3, allow requests from anonymous users.) If the request passes authentication, AWS then determines whether the requester is authorized to perform the action represented by the request.

Requests that are made by the AWS account root user are allowed for resources in that account. However, if the request is made using the credentials of an IAM user, or if the request is signed using temporary credentials that are granted by AWS STS, AWS uses the permissions defined in one or more IAM policies to determine whether the user’s request is authorized.

* + Amazon S3 supports Access Control Lists (ACLs) and resource-level policies for buckets and objects. The permissions established using ACLs and bucket-level policies can affect what actions the root user is allowed to perform on a bucket.
  + Evaluation logic of IAM policies:

**Storage**

**S3**

* Virtual unlimited storage, files can be from 0 B to 5 TB in size.
  + With a single PUT operation you can upload objects up to 5GB in size.
  + Using the multipart upload API, you can upload large objects, up to 5 TB. The multipart upload API is designed to improve the upload experience for larger objects. You can upload objects in parts. These object parts can be uploaded independently, in any order, and in parallel. You can use a multipart upload for objects from 5 MB to 5 TB in size.
  + There is a limit of 1000 parts for multipart uploads.
* Immediate consistency (read after write) after PUT new file, eventual consistency for overwrite PUT and DELETE (can take some time to be consistent).
* Consists of: Key, Value, VersionId, Metadata and Sub-resources.
* Sub-resources are
  + Access Control Lists – For individual files or at the bucket level
  + Torrent – Supports bittorrent protocol
* Adding a random guid at the start of the key improves performance, since S3 stores files sorted lexicographically. Adding the guid make it distributed evenly across S3 servers.
* Tiered storage options
  + Standard – Eleven 9s of durability
  + Infrequent Access – Cheaper to store but implies a retrieval fee.
  + Reduced Redundancy – Less durability but a lot cheaper
* Glacier
  + Own independent service. Designed for archival, very cheap but takes from 3 to 5 hours to retrieve the file. Also charges retrieval fee.
  + Accessible via AWS CLI over S3 commands.-
  + You can’t export from Glacier using AWS Import/Export service.
  + Natively encrypts data at rest.
  + Archives are **immutable**.
  + Archives can range from 1 byte to 40 Terabytes.
  + There is multi-part upload feature for glacier
    - Maximum parts: 10000 (Differently form S3, which is 1000)
    - Maximum part size: 4GB (Differently form S3, which is 5GB)
    - The part size must be a megabyte (1024 KB) multiplied by a power of 2. For example, 1048576 (1 MB), 2097152 (2 MB), 4194304 (4 MB), 8388608 (8 MB).
* Life-cycle management
* You can secure data by using bucket policies
* Files are stored across multiple devices, across multiple facilities. Designed to sustain the loss of 2 facilities concurrently.
* Charged by
  + Storage
  + Requests
  + Storage management – Charged by tags associated with the file
  + Data transfer – Data into S3 is free, data out is charged
  + Transfer Acceleration
* Transfer Acceleration
  + Not guaranteed, charged only when enabled and acceleration succeeded. Allows users to upload their files via closest edge location.
  + Specific URL for uploading in the form of bucketname.s3-accelerate.amazonaws.com.
  + Has the greatest impact on people who are in far away locations from your region.
* Make sure to Read S3 FAQ before taking the exam
* By default all buckets are private and all objects are also private
* Securing your buckets
  + Bucket Policies
  + ACL (Access Control Lists)
  + Enable bucket to create access logs
  + Encryption
    - In transit – SSL/TLS is enabled by default
    - At rest – 4 methods of encryption
* Support for encryption
  + Client side – You encrypt your data on client side and send it to S3
  + Server side
    - S3 managed keys (AES-256) – SSE-S3
    - KMS (key management service) – SSE-KMS
    - Customer provided key – SSE-C
* Versioning
  + Once enabled it can’t be disabled (but suspended). Every version of the file is kept and charged as a regular stored object. Should be cautious with increasing cost (large files frequently changing).
  + Prevents from deletion. Only marks the object as deleted but keeps all versions of it, therefore keeps charging you for the all the objects stored. You must delete the delete marker in order to get fully rid of the file. You can delete the delete marker in order to restore the object.
  + You can enable MFA delete feature, so it will prevent people from accidentally deleting objects.
  + Versioning integrates with life-cycle rules.
* Cross-Region Replication
  + You need versioning enabled on both source and target buckets
  + Can be used as backup
  + Only new objects will be replicated after enabling cross-region replication. Existing objects will be kept only in the source bucket. You can use AWS CLI to copy existing contents.
  + Object permissions are not replicated, so one object made public on the source bucket is still private on destination bucket.
  + Deleting markers are replicated, but permanent deletes (deleting delete markers) are not. This behavior is interesting when using replication as backup mechanism.
  + Deleting a specific version of the object is not replicated.
  + You cannot replicate to multiple buckets
* Life-cycle Rules
  + Used to move objects to infrequent access storage class (at least 3 days after uploading the object), archive to glacier (at least 30 days after moving object to infrequent access) **and/or** expire (permanently delete) an object.
  + Can be applied to current versions, previous versions **and/or** incomplete multi-part uploads.
  + In order to move to IA object must have been stored for at least 30 days and have at least 128 KB. Moving to Glacier requires at least 30 days after moving to IA.
* Static web site hosting
  + Bucket name must be the same of your domain name.
  + Can be also used to only redirect requests.
* You CAN use IPv6 with Amazon S3
* You can enable CORS on a bucket so its contents may be accessed from a different domain. CORS is configured via an XML file under permissions tab on your bucket.
* If you want to make all objects in a bucket publicly available you will need to need to set a **bucket policy**, not a bucket ACL.
  + A you can see, bucket ACL does not have an option for public read:
  + A policy, which is a json document describing permission can allow public reads, like the following example

**Storage Gateway**

* Connects on-premise software appliance with the cloud based storage
* Natively encrypts data at rest.
* Four types
  + File Gateway (NTF) – Stored on S3
  + Volume Gateways (iSCSI)
    - Stored Volumes – Entire data set on site and backed up asynchronously on S3.
    - Cached Volumes – Entire data set is stored in the cloud, most recently accessed data in stored on premises.
  + Tape Gateway (VTL) – Used for back up.

**Snowball**

* Allows for sending huge amounts of data quickly, skipping the internet.
* Data is transferred physically, via mail.
* Imports and exports data to and from S3.
* There 3 types
  + Snowball – Petabyte scale
  + Snowball Edge – Storage of 100 TB and compute capabilities. Can run lambda functions.
  + Snowmobile – Petabyte (100 PB, but you can order 10 of these, reaching the exabyte scale). It is actually a truck.
* Created after failure of AWS Import/Export which depended non standardized media, sent by customers.

**CloudFront**

* Edge Location – the location where content is gonna be cached.
* You can also PUT to edge locations (not read only).
* Origin – origin of the files that the CDN will distribute, it can be an S3 bucket, an EC2 instance, an Elastic Load Balancer or Route53. You can have multiple origins for a distribution.
* Objects are cached for a TTL (time to live). You can clear the cache before TTL but you will be charged.
* TTL is in seconds and per-object. Default value is 24 hours.
* Distribution
  + Consists of collection of edge locations
  + Web distribution – For websites
  + RTMP – Used for media streaming (flash files)

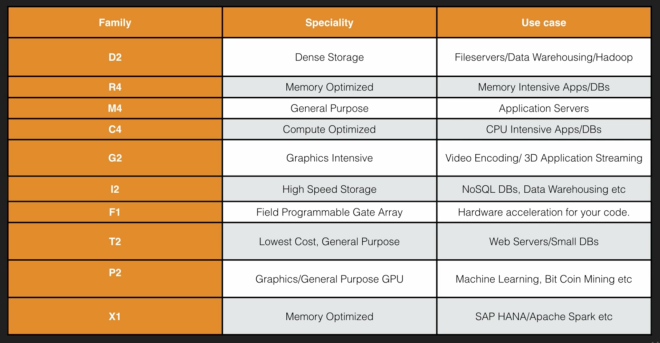
**AWS Data Pipeline**

* AWS Data Pipeline is a web service that helps you reliably process and move data between different AWS compute and storage services, as well as on-premise data sources, at specified intervals.
* You can regularly access your data where it is stored, transform and process it at scale, and efficiently transfer the results to AWS services such as S3, DynamoDB and EMR.

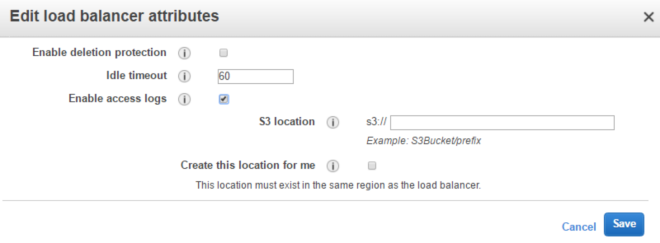
**Compute**

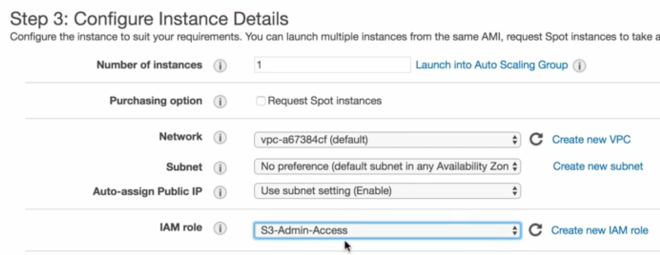
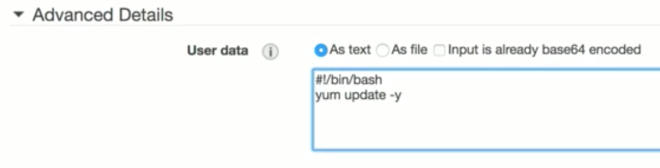
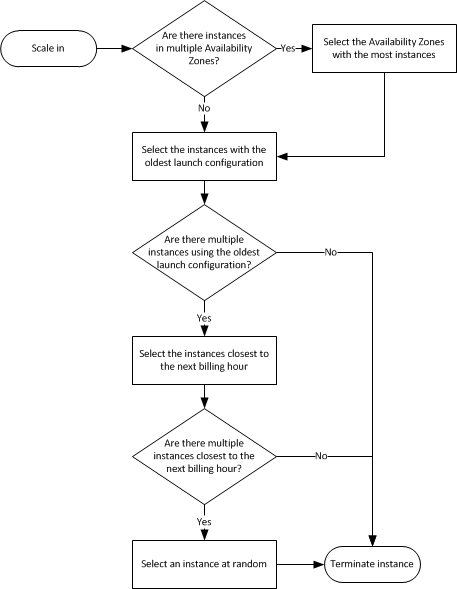
**EC2**

* Virtual machines on the cloud
* On demand instance – Pay a fixed rate by the hour or by the second (linux instances only).
* Reserved instances – Capacity reservations. Sign a one to three year contract (pay a bit upfront).
  + Standard – Up to 75% price reduction.
  + Convertible – Up to 54%, allows for the changing of instance attributes (when resulting instance is of equal or greater pricing).
  + Scheduled – Available to launch within the time window you reserve. Available for fractions of day, week or month.
  + Reserved instances are available for multi-AZ deployments.
  + You CAN’T move a reserved instance between regions.
  + You CAN transfer a reserved instance from one availability zone to another.
  + You can sell your unused reserved instances on the Reserved Instance Marketplace (see <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ri-market-general.html>)
* Spot instances – Enables you to bid for the price you want. Provides great savings but requires flexible start and end times for you applications.
  + If you stop the instance before a complete hour they will charge you this full hour, otherwise, if Amazon terminates your instance you get the free hour.
* Dedicated hosts – Physical EC2 servers dedicated to your use. Allows you to use your existing server-bound licenses. Good for clients that don’t want a multi-tenant scenario. Useful for regulatory requirements. Can be purchased on demand or as discounted reserved instances.
* Assuming all instances are in the same AWS organization, the reserved instance pricing for the unused QA instances will be applied to on demand instances.
* There are currently 10 different families of instance types. See picture below. [Instance Types Cheat Sheet](https://renatoargh.files.wordpress.com/2018/01/instance-types-cheat-sheet.pdf).



* EBS – Elastic Block Storage
  + Storage volumes that you can attach to EC2 instances. Block based storage, so you can install databases, SOs, etc. Automatically replicated within the availability zone it was created.
  + Must be at the same availability zone as the instance.
  + Types of EBS
    - General Purpose SSD (GP2) – General purpose, balance of price and performance.
      * Maximum size is 16TiB (Tebibyte, not Terabyte), as opposed to instance store volumes which are limited to 10GiB.
      * Minimum size us 1GiB (Gibibyte, not Gigabyte)
    - Provisioned IOPS SSD (IO1) – I/O intensive applications, like relational databases or NoSQL databases.
      * There is a maximum ratio of 50:1 when provisioning IOPS, so for example; a 8GB Volume would allow 400IOPS at most (8 x 50 = 400).
      * You can provision 100 IOPS at minimum
      * You can provision 20000 IOPS at maximum
    - Throughput Optimised HDD (ST1) – Big data, data warehouse, log processing. Cannot be a boot volume.
    - Cold HDD (SC1) – File server
    - Magnetic (Standard) – Lowest cost per GB that is bootable. Ideal when data has infrequent access.
  + When terminating EC2 instances, root volume will be terminated by default. Additional volumes will be kept and you will have to delete them manually.
  + You cannot mount one EBS volume to multiples EC2 instances, instead you should use EFS (elastic file system). On the other hand you can have an instance with multiple EBS volumes attached.
    - All EBS volumes attached to an instance must be at the same availability zone.
  + You can upgrade an EBS volume on the fly (except for the magnetic types). You can for example change its storage capacity without downtime.
  + If you want to change the type or the availability zone of a volume you must first take an snapshot of it and then create a new volume based on that snapshot, at this point you are able to change its type, from magnetic to SSD, for example.
    - If you wanna change the volume region you will need to use the copy function from the snapshots listing.
  + Snapshots – Point in time copy of volumes. Commonly used for backups. They are not deleted once you terminate an EC2 instance.
    - Stored on S3 (but you can’t directly view them)
    - Snapshots are incremental. Only blocks that have changed are moved to S3.
    - Snapshots of encrypted volumes are going to be encrypted automatically.
    - You can share snapshots with other people, only if they are not encrypted (because your encryption keys are tied to your AWS account).
  + What can happen to an EBS volume when an EC2 instance is terminated:
    - By default It will be deleted.
    - You can set it up to persist upon instance termination.
  + A root volume is the volume used to boot the instance with an AMI. Default AMI’s root volume cannot be encrypted even tough you can create an AMI with root volume encrypted. Additional volumes can be encrypted.
  + RAID = Redundant Array of Independent Disks. You can use RAID in AWS when you are not getting the IOPS that you require.
    - RAID 0, 1 and 10 – Can be used on AWS
    - RAID 5 – Discouraged
    - To make a snapshot of a RAID you must either:
      * Stop the EC2 instance to prevent inconsistencies among the discs, take the snapshot and then turn it on again.
      * Freeze the file system.
      * Unmount the RAID array.
  + EBS Encryption Key – Amazon EBS encryption uses AWS Key Management Service (AWS KMS) customer master keys (CMKs) when creating encrypted volumes and any snapshots created from them.
    - CMK – Customer Master Key
    - You cannot change a CMK that is associated with an existing snapshot or encrypted volume
    - You can associate a different CMK during a snapshot copy operation so that the resulting copied snapshot uses the new CMK.
* Termination protection prevents from accidentally terminating your instance and it is turned off by default.
* Network Interfaces
  + Can be attached while:
    - Instance is running – hot attach
    - instance is stopped – warm attach
    - instance is being launched – cold attach
* EC2 Network
  + Into which VPC the instance is going to be started.
  + Subnet – One subnet equals one availability zone.
* To connect via ssh to your instance you must remember to change privileges on the .pem key file with the command chmod 400 key.pem.
* Security Groups – Virtual firewall to control traffic. One instance can have multiple security groups.
  + Any inbound rule change to security group applies immediately.
  + All inbound traffic is blocked by default. All outbound traffic is allowed by default.
  + Security groups are stateful. If you create an inbound rule allowing traffic in, that traffic is automatically allowed back out again.
  + Everything is blocked by default, so you can’t just deny traffic to any specific port or from one specific IP address (instead use Network Access Control Lists).
* Can I delete a snapshot of an EBS volume that is used as the root device of a registered AMI? You must deregister the AMI before being able to delete the root device.
* When moving an AMI between regions: AWS does not copy launch permissions, user-defined tags, or Amazon S3 bucket permissions from the source AMI to the new AMI.
* EBS vs Instance Store
  + EBS Backed Volumes – Persistent.
  + Instance Store – Ephemeral storage, if the underlying host fails you will loose your data. You can’t stop it, only terminate or reboot. Less durability of data.
  + You can’t detach instance stored volumes.
  + You can reboot both without loosing your data.
* Elastic Load Balancer
  + Spread the load of your traffic across different instances.
  + You don’t get a public IP address for the load balancer, only a public DNS name.
  + Instances monitored by ELB are listed as InService or OutOfService, depending on health checks.
  + Read the ELB FAQ before the exam. Focus on classic load balancers.
  + You are charged for the use of ELB.
  + Each Classic Load Balancer has an associated IPv4, IPv6, and dualstack (both IPv4 and IPv6) DNS name. However, IPv6 for Classic Load Balancers is not yet supported in VPC.
  + If an instance belonging to an Elastic Load Balancer fails its health check the ELB will de-register the instance and stop sending traffic to it.
  + Three types
    - Application load balancers
      * Relatively new. Operates at the application layer (layer 7)
      * Support for path based routing. Example: Sending all requests to example.com/videos/\* to an specific target group.
      * Does not support TCP protocol (it acts at request level). In this case choose either a classic load balancer or a network load balancer.
    - Classic load balancers – Operates on transport layer (layer 4).
    - Network Load Balancers – Operates at TCP level.
      * Good use case for WebSockets protocol
  + There are questions specific to application load balancers on the CSA exam. You will need to read the FAQ: https://aws.amazon.com/elasticloadbalancing/faqs/
  + SSL certs on load balancers.
    - Application load balancer – May have multiple certs to accomodate multiple sites.
    - Classic load balancers – Can have only one SSL cert.

You can enable access logs on the load balancer (can be useful for analysing traffic patterns). It will be stored on a S3 bucket.

* + Load balancer has some CloudWatch metrics enabled by default, they are reported at a 60s rate when requests are coming through the LB instance. (<https://docs.aws.amazon.com/elasticloadbalancing/latest/classic/elb-cloudwatch-metrics.html>)
  + You can create internal load balancers. To balance internal load across multiple database servers.
  + Cross zone load balancing
    - With cross-zone load balancing enabled, your load balancer nodes distribute incoming requests evenly across the Availability Zones enabled for your load balancer. Otherwise, each load balancer node distributes requests only to instances in its Availability Zone. For example, if you have 10 instances in Availability Zone us-west-2a and 2 instances in us-west-2b, the requests are distributed evenly across all 12 instances if cross-zone load balancing is enabled. Otherwise, the 2 instances in us-west-2b serve the same number of requests as the 10 instances in us-west-2a.
    - Cross-zone load balancing reduces the need to maintain equivalent numbers of instances in each enabled Availability Zone, and improves your application’s ability to handle the loss of one or more instances. However, we still recommend that you maintain approximately equivalent numbers of instances in each enabled Availability Zone for higher fault tolerance.
  + Connection draining
    - To ensure that a Classic Load Balancer stops sending requests to instances that are de-registering or unhealthy, while keeping the existing connections open, use connection draining. This enables the load balancer to complete in-flight requests made to instances that are de-registering or unhealthy.
    - When you enable connection draining, you can specify a maximum time for the load balancer to keep connections alive before reporting the instance as de-registered. The maximum timeout value can be set between 1 and 3,600 seconds (the default is 300 seconds). When the maximum time limit is reached, the load balancer forcibly closes connections to the de-registering instance.
    - If your instances are part of an Auto Scaling group and connection draining is enabled for your load balancer, Auto Scaling waits for the in-flight requests to complete, or for the maximum timeout to expire, before terminating instances due to a scaling event or health check replacement.
    - You can disable connection draining if you want your load balancer to immediately close connections to the instances that are de-registering or have become unhealthy. When connection draining is disabled, any in-flight requests made to instances that are de-registering or unhealthy are not completed.
  + Terminology
    - Register an instance – Registering an EC2 instance adds it to your load balancer. The load balancer continuously monitors the health of registered instances in its enabled Availability Zones, and routes requests to the instances that are healthy.
    - Deregister an instance – Deregistering an EC2 instance removes it from your load balancer. The load balancer stops routing requests to an instance as soon as it is deregistered. If demand decreases, or you need to service your instances, you can deregister instances from the load balancer. An instance that is deregistered remains running, but no longer receives traffic from the load balancer, and you can register it with the load balancer again when you are ready.When you deregister an instance, Elastic Load Balancing waits until in-flight requests have completed if connection draining is enabled.
* Cloud Watch
  + Allows for monitoring instance metrics.
  + Can sample your metrics each 5 minutes for basic monitoring or 1 minute for detailed monitoring. You are charged for detailed monitoring.
  + RAM memory usage is not available by default though you can implemented it as a custom metric.
  + Allows from the creation of neat dashboards for monitoring all these metrics.
  + Allows for the creation of alarms that can trigger different actions active
    - Send email notifications
    - Trigger auto-scaling actions
  + CloudWatch Events. AWS resources trigger events that you can use to take actions using lambda functions.
  + CloudWatch Logs. Helps you to aggregate, monitor and store logs. You can use to track host logs by installing an agent that will send logs to CloudWatch. You can even send application logs to CloudWatch.
  + CloudWatch vs CloudTrail
    - CloudWatch – For monitoring performance metrics
    - CloudTrail – Auditing whats happening on your account, to the level of API calls.
  + CloudWatch stores metrics for terminated Amazon EC2 instances or deleted Elastic Load Balancers for 15 months.
* Command Line Interface with EC2
  + You first need to run aws configure to configure appropriate credentials, it will then create a file at ~/.aws/credentials to store your credentials information.
  + There is a more secure way to provide CLI credentials, instead of plain text as in the previous item you can define a role and attach it to your instances. You can attach roles to instances during its creation or after when it is already running (new feature, wasn’t possible before).
  + By attaching a role to an instance you won’t need to configure anything. If you issue a command like aws s3 ls it will work right away.
  + Remember to use the flag --region when using s3 command line interface to prevent problems when accessing buckets in a different region than your instance.
* Bootstrap Scripts
  + Runs when EC2 first starts up.
* Instance Metadata – Data about your running instance.
  + You can list available metadata via command line by issuing the following command curl 169.254.169.254/latest/meta-data.
  + You can access the specific attribute value you want by concatenating one of the attributes listed in the previous item with the metadata data URL, for example: curl 169.254.169.254/latest/meta-data/public-ipv4 gives you the public IP address of your instance.
* Auto Scaling
  + You will have to set a up a launch configuration which is very similar to how you provision a regular instance except that it won’t start any instance right away but will use this configuration to spin up new instances as needed.
  + You are required to set how many instances you want to start and at which availability zones they will be launched (select all regions so your app is more resilient to failures).
  + You can configure scaling policies where you will define what metric triggers an instance to be added to the group and what metrics triggers the removal from the group. These policies are actually CloudWatch alarms. Alternatively you can configure your auto scaling group to remain at a fixed size so whenever one instance fails it will trigger a new one to replace the crashed instance.
  + If an instance returns an unhealthy status then auto scaling will terminate the instance and launch a new instance.
  + Auto scaling can be configured to send an SNS notification when it needs to terminate an instance.
  + The following diagram shows default auto scaling policy for choosing which instance it will terminate first (scale in)(<https://docs.aws.amazon.com/elasticloadbalancing/latest/classic/elb-cloudwatch-metrics.html>): 
  + As it can be seen from the previous chart auto scaling **does not** take into account the number of active sessions per instance in order to choose the one that will be terminated.
  + You can protect instances from scale in policy. Auto scaling will not select protected instances for termination during scale in.Screenshot 2018-01-31 22.06.56.png
  + You can provide custom termination policies
    - OldestInstance
    - NewestInstance
    - OldestLaunchConfiguration
    - ClosestToNextInstanceHour
    - Default
  + You can suspend and then resume one or more of the scaling processes for your Auto Scaling group. This can be useful when you want to investigate a configuration problem or other issue with your web application and then make changes to your application, without triggering the scaling processes.
  + Auto Scaling processes are as follow:
    - Launch – Adds a new EC2 instance to the group, increasing its capacity.
    - Terminate – Removes an EC2 instance from the group, decreasing its capacity.
    - HealthCheck – Checks the health of the instances. Amazon EC2 Auto Scaling marks an instance as unhealthy if Amazon EC2 or Elastic Load Balancing tells Amazon EC2 Auto Scaling that the instance is unhealthy. This process can override the health status of an instance that you set manually.
    - ReplaceUnhealthy – Terminates instances that are marked as unhealthy and later creates new instances to replace them. This process works with the HealthCheck process, and uses both the Terminate and Launchprocesses.
    - AZRebalance – Balances the number of EC2 instances in the group across the Availability Zones in the region. If you remove an Availability Zone from your Auto Scaling group or an Availability Zone otherwise becomes unhealthy or unavailable, the scaling process launches new instances in an unaffected Availability Zone before terminating the unhealthy or unavailable instances. When the unhealthy Availability Zone returns to a healthy state, the scaling process automatically redistributes the instances evenly across the Availability Zones for the group.
    - AlarmNotification – Accepts notifications from CloudWatch alarms that are associated with the group. If you suspend AlarmNotification, Amazon EC2 Auto Scaling does not automatically execute policies that would be triggered by an alarm.
    - ScheduledActions – Performs scheduled actions that you create.
    - AddToLoadBalancer – Adds instances to the attached load balancer or target group when they are launched. If you suspend this processes you will have to manually add instances to the ELB after you this process is resumed.
  + Cool-down period – The number of seconds after a scaling activity completes before another can begin. Default cool-down value is 300 seconds.
  + Lifecycle Hooks – Provides the ability of running scripts during launch and termination of your instances.
    - *Lifecycle hooks enable you to perform custom actions by pausing instances as an Auto Scaling group launches or terminates them. For example, while your newly launched instance is paused, you could install or configure software on it.*
    - If you provide such scripts then instances will transition to states Pending:Wait and Terminating:Wait before entering service and indeed terminating.
  + Instance lifecycle when in auto scaling:
  + Pricing – There is no additional charge for AWS Auto Scaling. You pay only for the AWS resources needed to run your applications and Amazon CloudWatch monitoring fees.
* Placement Groups – Logical grouping of instances within a single availability zone. Using placement groups enables applications to participate in a low latency, 10Gbps network.
  + A placement group can’t span multiple availability zones,
  + Name of a placement group must be unique within your AWS account.
  + Only certain types of instances can be put in a placement group (compute optimised, GPU, memory optimised, storage optimised). AWS recommend homogenous instances (same size and same family) within a placement group.
  + You can’t merge placement groups.
  + You can’t move an existing instance to a placement group. You can create an AMI from your existing instance, and then launch a new instance form the AMI into a placement group.
* EFS – Elastic File System
  + File storage service for EC2.
  + Supports protocol NFSv4 (network file system).
  + Like S3, there is no pre provision. Can scale up to petabytes.
  + Can be shared among multiple EC2 instances.
  + Data is stored in multiple availability zones within a region.
  + Read after write consistency.
  + Instances using EFS must be in the same security group as the EFS.
  + To mount EFS on your instance just follow the instruction given on EFS console web page.
  + The underlying Hypervisor for EC2 is Xen.
* AMI – You can customise an EC2 instance and then save its configuration by creating an Amazon Machine Image (AMI).
  + A golden image is simply an image that you have customised to your liking with all necessary software/data/configuration information ready to go and then saved as a personal AMI from which you can launch instances.

**ECS and ECR**

* Amazon managed service for Docker.
* Elastic Container Service is a highly scalable, fast, container management service that makes it easy to run, stop, and manage Docker containers on a cluster of EC2 instances. Amazon ECS lets you launch and stop container-based applications with simple API calls, allows you to get the state of your cluster from a centralised service, and gives you access to many familiar EC2 features.
* ECR is a managed Docker registry that is secure, scalable and reliable. Amazon ECR supports private Docker repositories with resource-based permissions using AWS IAM so that specific users on or EC2 instances can access repositories and images. Developers can use Docker CLI to push, pull and manage images.
* A task definition is required to run Docker containers in Amazon ECS. Task definition are text files in JSON format that describe one or more containers that form your application.
  + Think of a task definition as a cloud formation template but for Docker. Configure things such as the amount of CPU, RAM, etc.
* Ideal way to diagnose errors in docker container is using Docker Diagnostic Tools.

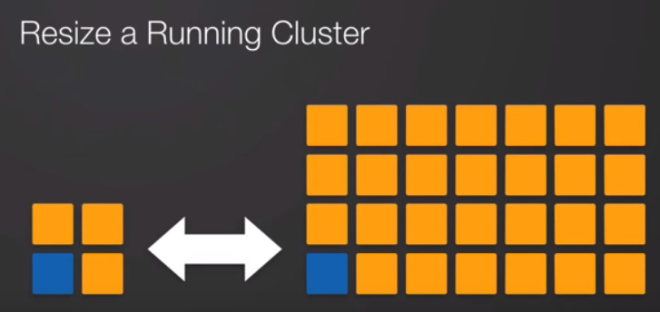
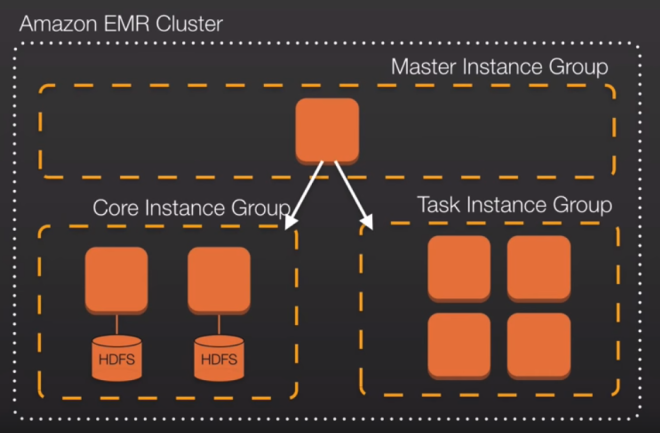
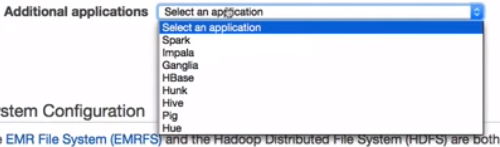
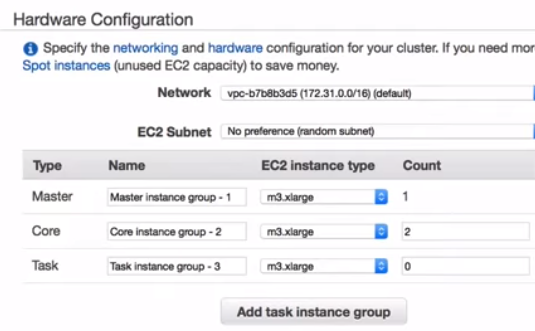
**Lambda**

* Server-less code execution.
* Scales out (not up) automatically.
* 5 minutes timeout limit for function execution.
* Architecture may become extremely complicated.
* Execute code in response to events
  + Service triggers
  + Scheduled executions
* What services can trigger lambda events (bear in mind that this triggers are region specific).
  + API Gateway
  + AWS IoT
  + Alexa Skills Kit
  + Alexa Smart Home
  + CloudFront
  + CloudWatch Events
  + CloudWatch Logs
  + CodeCommit
  + CognitoSync Trigger
  + DynamoDB
  + Kinesis
  + Lambda functions can trigger lambda functions
* Languages supported:
  + Node.js (JavaScript)
  + Python
  + Java
  + C#
  + Go (recently added – might not be considered on the exam)
* Pricing based on:
  + Number of requests (you get the first million requests free).
  + Request duration to the millisecond.
  + Amount of RAM memory allocated (price is $ 0.00001667 per GB-second used).
* Resource limits per invocation
  + Memory allocation: from 129MB to 1536MB (in 64MB increments)
  + Temporary space (/tmp): 512MB
  + Number of file descriptors: 1024
  + Number of processes and threads combined: 1024
  + Maximum execution duration: 300 seconds
  + Maximum request body size: 6MB

**API Gateway**

* Automatically integrated with CloudFront
* Stages allows for different environments and versioning of the API
* You can use API Gateway to generate an SSL certificate and use its public key in the backend to verify that HTTP requests to your backend system are from API Gateway. This allows your HTTP backend to control and accept only requests originating from Amazon API Gateway, even if the backend is publicly accessible.
  + API caching in Amazon API Gateway is not eligible for the AWS Free Tier.
  + Calling methods with the [authorization type](http://docs.aws.amazon.com/apigateway/api-reference/resource/method/#authorizationType) of AWS\_IAM, CUSTOM, and COGNITO\_USER\_POOLS are not charged for authorization and authentication failures.
  + Calling methods requiring API keys are not charged when API keys are missing or invalid.
  + API Gateway-throttled requests are not charged when the request rate or burst exceed the pre-configured limits.
  + Usage plan-throttled requests are not charged when rate limits or quota exceed the pre-configured limits.
* Paths /ping and /sping are reserved for health checks and can’t be used.
* Stage variables are name-value pairs that you can define as configuration attributes associated with a deployment stage of an API. They act like environment variables and can be used in your API setup and mapping templates.
* Apart from IAM roles and policies Amazon Cognito can be used to control access to your API gateway.

**EMR – Elatic Map Reduce**

* Provides a managed Hadoop framework
* Quickly and cost effectively process vast amount of data
* Run other popular frameworks such as Spark
* Parallel distributed execution of tasks
* Pricing – hourly rate for each instance that you use
* Support for spot and reserved instances
* Elastic – You can add and remove instances whenever needed
* You have full SSH access to your instances allowing for the customization of your cluster.
* Data store integrations with
  + Amazon S3 – Can be used as underlying “file system” for input/output data
  + Hadoop Distributed File System
  + Amazon DynamoDB
  + Amazon Redshift
  + Amazon Glacier
  + Amazon RDS
* Use cases examples:
  + Click-stream analysis
  + Genomics
  + Log processing
* Work is distributed via the master node
* You can select specific Hadoop distribution when configuring the cluster
* You can install adittional aplications when configuring the cluster
* You can specify different instance types for each type of node.
* Steps – A step is a unit of work you submit to the cluster. A step might contain one or more jobs, or contain instruction to install or configure an application. You can submit up to 256 steps to a cluster.

**Route53**

* Support for IPv4 and IPv6.
* VPCs are now IPv6 compatible.
* Global service, you don’t have to choose a region.
* Features
  + Domain registrar
  + Regular DNS server
  + Health checks
* Charged by
  + Hosted zone
  + DNS query
    - Regular queries (simple or weighted)
    - GEO based queries
    - Latency based queries
* TTL – “Time to Live” or the time in which resolving server (or client) should cache the result of a DNS query. The lower the TTL the faster DNS changes propagate throughout the internet.
* There are 50 domain names available by default, however it is a soft limit and can be raised by contacting AWS support.
* Records
  + SOA  – “Start of authority” record. The name of the server that suplied the data for the zone. The administrator of the zone
  + NS – Name Server records used by top level domains to direct traffic to the content DNS server which contains the authoritative DNS records.
  + A – Most fundamental record. A stands for Address. Points to a name to an IPv4 address.
  + AAAA – Same as A records but for IPv6.
  + CNAME – Canonical name, translates a domain to another domain.
  + ALIAS – Used to map resource record sets in your hosted zone to Elastic Load Balancers, CloudFront distributions or S3 buckets that are configured as websites.
    - Key difference to CNAME records: CNAME records can’t be used for naked domain names (zone apex record). You can’t have a CNAME for example.com, it must be either an A records or an ALIAS.
    - When you make a request to R53 for a CNAME record you will be charged, on the other hand you will not be charged if making a request to an ALIAS record.
    - If possible, always choose an ALIAS over a CNAME.
* Routing Policies
  + Simple – Default route policy. Common when you have a single resource for your domain. No intelligence, simple forwarding of the request.
  + Weighted – Define weights or percentages of requests that will be routed to a region or another (or different load balancers in the same region).
    - Can be used for A/B tests. Route a small percentage of requests to a load balancer running instances that serve a different version of the UI for example.
    - If you are testing you might have to wait for domain TTL until you are pointed to another region (or load balancer).
  + Latency – Allows you to route your traffic based on the lowest network latency for your end users (the region that will give them the fastest response time).
  + Failover – Routes users based on health checks. Can direct users to a different region when health check fails. Allows you to have a primary and secondary site in different regions. Interesting for disaster recovery scenarios.
  + Geolocation – Allows you to route your traffic based on the geographical location of your users. Send European users to London region or Amercian users to North Virginia.
* Route53 has a security feature that prevents internal DNS from being read by external sources. The work around is to create a EC2 hosted DNS instance that does zone transfers from the internal DNS, and allows itself to be queried by external servers.

**VM Import/Export**

* VM Import/Export enables you to easily import virtual machine images from your existing environment to Amazon EC2 instances and export them back to your on-premises environment.
* Pricing – VM Import/Export is available at no additional charge beyond standard usage charges for Amazon EC2 and Amazon S3.
* Use cases:
  + Migrate Your Existing Applications and Workloads to Amazon EC2
  + Copy Your VM Image Catalog to Amazon EC2
  + Create a Disaster Recovery Repository for your VM images

**Databases**

**RDS**

* OLTP – Online transaction processing
* Managed relational database service
* Databases available
  + SQL Server
  + Oracle
  + MySQL
  + PostgreSQL
  + Aurora
  + MariaDB
* You can configure DB instance security group to only allow connections from instances in your DMZ security group, making sure only IPs from inside your VPN can reach the database.
* Backups – There are two types of backups in RDS
  + Automated Backups
    - By default, Amazon RDS enables automated backups of your DB instance created via the RDS API or the AWS CLI with a 1-day retention period.
    - Stored in S3 (you get free storage equal to the size of the database).
    - Allow you to recover your database to any point in time within a retention period. Retention period can be between 1 to 35 days. Will take a full daily snapshot and will also store transaction logs throughout the day. When you do a recovery, AWS will first choose the most recent daily backup and then apply transaction logs relevant to that day. This allows for a point in time recovery down to a second within the retention period.
    - Backups are taken within a predefined window when you can experience elevated latency.
    - Deleted after you delete original RDS.
    - Changes to the backup window take effect immediately.
  + Database Snapshots
    - Manually initiated
    - Stored even after you delete your RDS instance.
    - You can use a database snapshot to start a new instance in a different region and even resize your underlying instance (scale up or down).
    - IO operations are suspended for the duration of the snapshot.
  + When restoring either Automated Backup or Database Snapshots the restored version of the database will be a new RDS instance with a new endpoint.
* Encryption – Encryption at rest is supported using AWS KMS. Encrypting an existing non-encrypted instance is not supported, but you can create a new encrypted instance and migrate your data.
* Multi-AZ
  + Allows you to have an exact copy of your database in a different availability zone so if you. RDS will automatically failover to the secondary instance without you having to worry about connection strings in your apps or administrative intervention.
  + Disaster recovery only. Not used to improve performance.
  + Synchronous replication of data
* Read Replicas
  + Read only replicas used to improve performance, not suitable for disaster recovery.
  + You can have up to 5 copies of your main database.
  + Good for intensive report generation tasks.
  + Not supported for MS SQL or Oracle.
  + Must have automatic backups turned on in order to use read replicas.
  + You can have read replicas of read replicas, but watch out for latency.
  + Each read replica will have its own DNS endpoint.
  + You cannot have read replicas that have Multi-AZ (they are single availability zone instances).
  + You can promote read replicas to be a main database but then you break replication.
  + For MySQL and MariaDb you can have a read replica in a secondary region (not available for PostgreSQL).
  + Can’t scale out in terms of writes.
  + Asynchronous replication of data
* You can always simulate a fail-over scenario by resetting the main instance of your database cluster (or when available clicking the button “Action > Fail-over”).
* If you are using Amazon RDS Provisioned IOPS storage with MySQL and Oracle database engines the maximum size RDS volume you can have by default is 6TB.
* By default, the maximum provisioned IOPS capacity on an Oracle and MySQL RDS instance (using provisioned IOPS) is 30,000 IOPS.

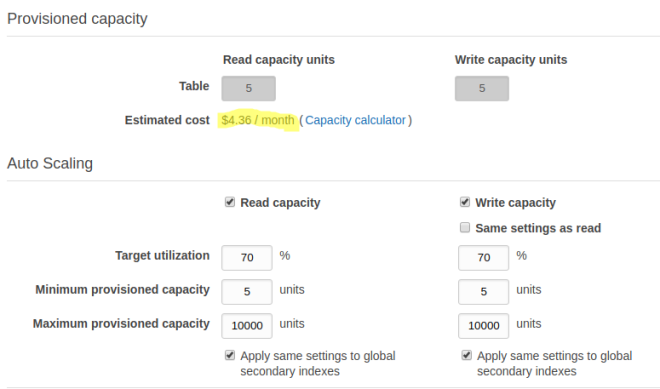
**MySQL**

* The MyISAM storage engine does not support reliable crash recovery and might prevent a point-in-time restore or snapshot restore from working as intended.

**Aurora**

* Relational database engine created by Amazon, only ran on the cloud.
* MySQL compatible tough five times better performance.
* Storage auto scales in increments of 10 GB up to 64 TB.
* Compute resources can scale as well, to 32vCPUs and 244 GB of memory.
* Maintains 2 copies of your database per availability zone, with minimum of 3 availability zones. Means 6 copies of your data.
* Aurora storage is self-healing. Automatically scanned for errors and repaired automatically.
* 2 types of replicas available
  + Aurora replicas – up to 15 (auto failover)
  + MySQL replicas – up to 5
* You should always use the cluster endpoint in your app for automatically fail-over.
* You can always simulate a fail-over scenario by resetting the main instance of your database cluster (or when available clicking the button “Action > Fail-over”).
* Amazon RDS does not currently support increasing storage on a SQL Server database instance.

**DynamoDB**

* Amazon proprietary fully managed NoSQL database.
* Offers “push button” scaling. A lot easier than RDS.
* Supports both document and key-value store models.
* SSD storage
* Redundant across 3 different data centers
* Consistency models
  + Eventual consistency reads – Replication takes around one second (best read performance)
  + Strong read consistency – Replication is under one second.
* Pricing
  + Write and read throughput (1 capacity unit = 1 write/read per second).
    - Write capacity – billed in blocks of 10
    - Read capacity – charged in block of 50
    - You can change read/write capacity on the fly.
  + Storage costs per GB.
* Good use case is when your database is being stressed due to lots of OLAP transactions on it.
* DynamoDB allows for the storage of large text and binary objects, but there is a limit of 400 KB.
* With DynamoDB you provision an amount of transactions per second, and you can either manually increase/decrease this amount or configure auto scaling (see below).
* You can create triggers for DynamoDB with Lambda functions
* DynamoDB allows for cross region replication (for disaster recovery).

**ElastiCache**

* ElastiCache is a web service that makes it easy to run in-memory cache in the cloud. Can serve a layer that helps enhancing performance compared to disk-based databases.
* Good use case is when your database is a read-heavy and not prone to  to frequent change.
* Supports two open-source in-memory caching engines
  + Redis – Multi-AZ capabilities, protocol compatible with memcached.
  + Memcached – No multi-AZ capabilities

**Redshift**

* OLAP – Online analytics processing
* Data warehouse service
* Columnar data storage
  + When storing data in columnar format Redshift uses a 1024Kb blocksize.
* Advanced compression due to columnar architecture.
* Doesn’t require indexes or materialized views.
* Massive parallel processing
* Single Node (up to 160GB)
* Multi-node
  + Leader node
  + Compute nodes (up to 128 compute nodes)
* Pricing
  + Compute node hours (not charged for leader node hours)
  + Backup
  + Data transfer (only within a VPC, not outside it).
* Ecnryption
  + In transit with SSL
  + At rest with AES-256
  + By default redshift takes care of key management, but you can use HSM and KMS.
* Only available in one available zone (not Multi-AZ)
* Can restore snapshots to a new AZ in case of outages.

**DMS**

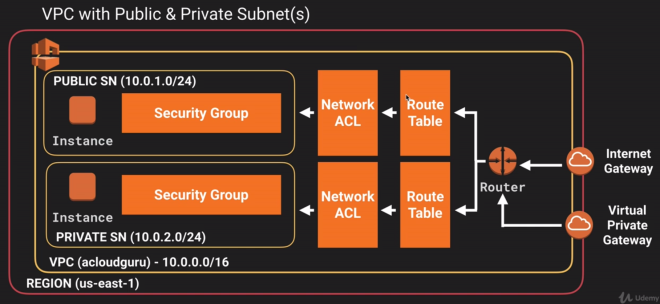
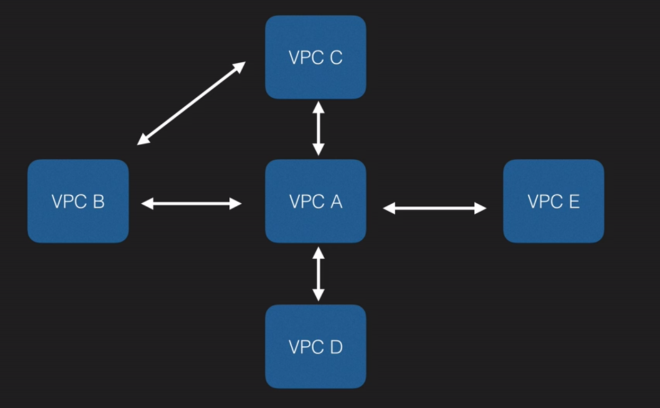
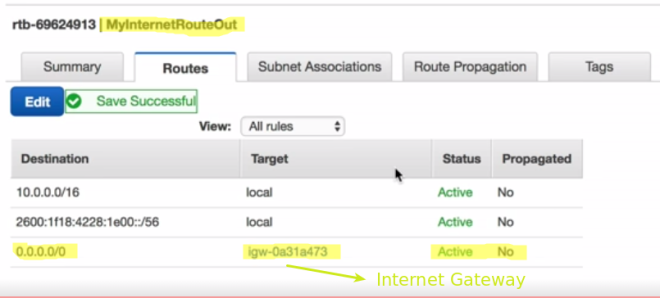
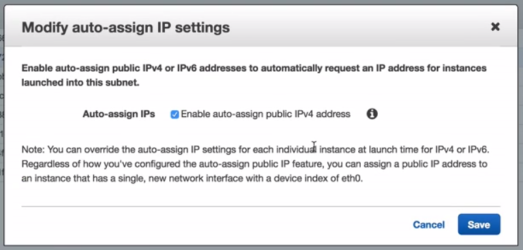
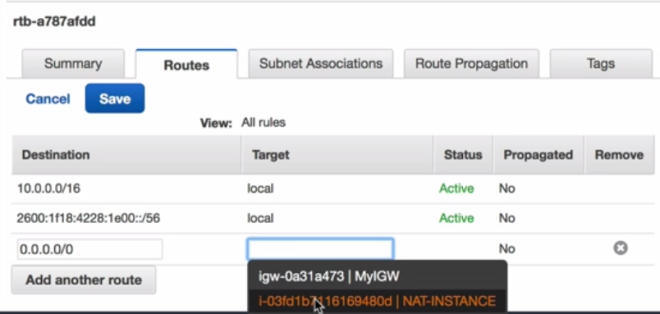
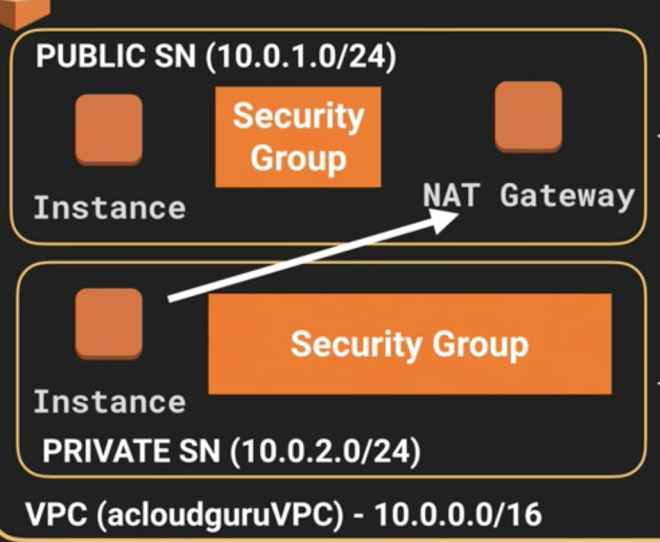
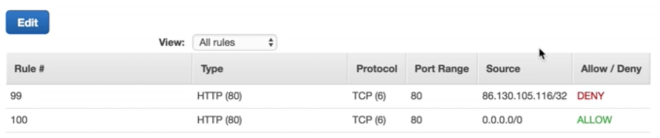
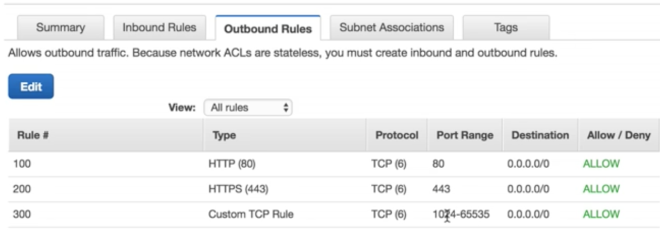
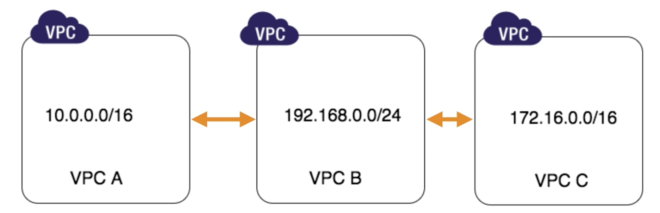
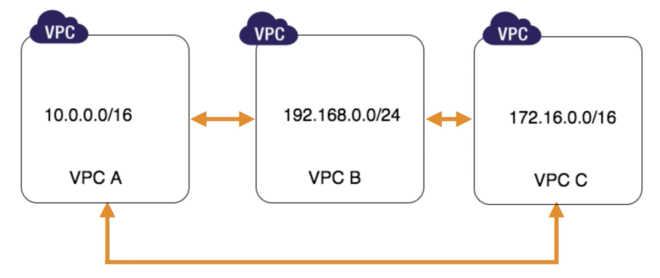
* Database migration service
* Helps migrating data one database to another. The process is fully managed and handles aspects like data transformations, compression and parallel transfer. You can even migrate data from one database vendor to another.
* AWS schema conversion tool helps automatically converts the source database schema and a majority of the custom code, including views, stored procedures and functions to a format compatible with the target database. Converting Oracle legacy databases to MySQL for example.

**Monitoring**

**CloudTrail**

* Logs activity to the API level
* AWS CloudTrail is a web service that records activity made on your account and delivers log files to your Amazon S3 bucket.
* CloudTrail event history provides a viewable, searchable, and downloadable record of the past 90 days of CloudTrail events.
* A trail that applies to all regions has the following advantages:
  + The configuration settings for the trail apply consistently across all regions.
  + You receive CloudTrail events from all regions in a single S3 bucket and optionally in a CloudWatch Logs log group.
  + You manage trail configuration for all regions from one location.
  + You immediately receive events from a new region. When a new region launches, CloudTrail automatically creates a trail for you in the new region with the same settings as your original trail.
  + You can create trails in regions that you don’t use often to monitor for unusual activity.

**VPC**

* Virtual Private Cloud – Think of it as a virtual data center on the cloud.
* Amazon provides you with a default VPC so you don’t have to first configure it.
* You can create a Virtual Private Network and leverage AWS cloud as an extension of your own corporate data center.
* VPC lives inside a region.
* One security group CAN span multiple availability zones.
* One subnet CAN’T span multiple availability zones.
* By default all subnets will be able to communicate with each other using the main route table.
* Subnet can use the follwing IP sets
  + 10.0.0.0 – 10.255.255.255 (10/8 prefix) – BIGGER ADDRESS RANGE
  + 172.16.0.0 – 172.31.255.255 (172.16/12 prefix) – INTERMEDIATE ADDRESS RANGE
  + 192.168.0.0 – 192.168.255.255 (192.168/16 prefix) – SMALLER ADDRESS RANGE
  + Visit [http://cidr.xyz](http://cidr.xyz/) for more info
* Soft limit of 5 VPCs.
* You can configure route tables between subnets to specify which subnet can talk to another subnet.
* You can create an internet gateway and attach to your VPC so you can access it from the internet. You can have only one internet gateway. They are already distributed across all availability zones.
* All subnets in the default VPC has a route out to the internet.
* With VPCs you can isolate one set of resources from another.
* You can peer VPCs directly (no transitive peering). VPC peering only routes traffic between source and destination VPCs. VPC peering does not support edge-to-edge routing (only one hop allowed).
  + You can peer VPCs from different regions
  + When establishing a peering connection, you can specify the entire IP block as a destination or any subset. For example, when peering with subnet 10.0.0.0/16 these are valid destinations:
    - 10.0.0.0/16 – Whole ip range
    - 10.0.0.0/28 – Subset
    - 10.0.0.7/32 – Specific IP
  + VPC peering configuration is done at route table level
* You can have a dedicated VPC (quite expensive). Once a VPC is set to Dedicated hosting, it is not possible to change the VPC or the instances to Default hosting. You must re-create the VPC.
* You mus attach an internet gateway to your VPC in order to have internet connectivity. You can attach only one internet gateway per VPC.
* When you set up a subnet on a VPC there are 5 reserved IP addresses
  + 10.0.0.0 – Network address.
  + 10.0.0.1 – Reserved by AWS for the VPC router.
  + 10.0.0.2 – Reserved by AWS. The IP address of the DNS server is always the base of the VPC network range plus two; however, AWS also reserve the base of each subnet range plus two. For VPCs with multiple CIDR blocks, the IP address of the DNS server is located in the primary CIDR. For more information, see [Amazon DNS Server](https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_DHCP_Options.html#AmazonDNS).
  + 10.0.0.3 – Reserved by AWS for future use.
  + 10.0.0.255 – Network broadcast address. AWS does not support broadcast in a VPC, therefore they reserve this address.
* Route tables
  + Created by default after creating a VPC.
  + Each time you create a subnet it will be associated by default to your main route table.
  + To allow internet connectivity you will have to create another route table and add a rule out to the internet. This rule must point to the internet gateway.
* Subnets you create does not assign a public IP address by default. You will want to enable it for public subnets.
* Security groups don’t span VPCs
* In order to allow subnets to communicate you must create a security group that allows access from the CIDR range of the subnet you want to allow access from.
* Allowing internet access from private subnets
  + NAT Instances
    - Start a NAT instance (get one from AMI images) on the public subnet.
    - You will have to disable source/destination checks for this instance
    - You should now go back to the private subnet route table and create an outbound rule to the nat instance (not to the internet gateway).
    - The amount of traffic a NAT instance can handle depends on the instance size. If you are having a bottleneck then increase the instance size.
    - Having a NAT instance does not scale. You would have to put it behind auto scaling groups and distribute across multiple availability zones, etc. It is best to use a NAT ateway in this scneario.
  + NAT Gateways (under VPC service)
    - Operates on IPv4 only (Egress Only Internet Gateways are for IPv6)
    - Select the public subnet when creating a NAT gateway.
    - Charged by the hour and data processed.
    - After creating a NAT gateway you should edit the route table for the private subnet and add an outbound rule targeting the NAT gateway.
    - Scale automatically to 10 Gbps.
    - No need to patch.
    - You’ll want them in multiple availability zones.
    - No need to disable source/destination checks.
    - Elastic IP – You must specify an Elastic IP address to associate with the NAT gateway when you create it.
    - You cannot route traffic to a NAT gateway through a VPC peering connection, a VPN connection, or AWS Direct Connect.
    - There may have been an error when your NAT gateway was being created, and it failed. A NAT gateway with a status of failed is visible in the Amazon VPC console for a short while (usually an hour), after which it’s automatically deleted.
    - Far more secure than a NAT instance.
    - 
  + Network Access Control Lists (Network ACL)
    - By default it allows all traffic in and out.
    - You can associate one subnet to only one Network ACL, not to multiple.
    - Controls inbound and outbound traffic.
    - When you create a subnet it is attached to your default Network ACL.
    - Network ACLs cannot span multiple VPCs
    - Newly create network ACLs deny all inbound and outbound traffic by default (different form the default Network ACL created with a subnet, which allows all in and out).
    - It is configured similar to a Security Group, but differently from a Security Group, Network ACL are stateless, meaning that you will have to configure outbound rules explicitly.
    - Can be used to deny inbound/outbound traffic to specific IPS. In this case IP addresses must came first than 0.0.0.0/0. Network ACLs contain a numbered list of rules that is evaluated in ascending order.
    - You must remember to allow for ephemeral ports (rule number 300 above).
    - ELBs and Custom VPCs – You must put your elastic load balancer in at least two internet enabled subnets in two different availability zones.
    - Rules are evaluated starting with the lowest numbered rule. As soon as a rule matches traffic, it’s applied regardless of of any higher-numbered rule that may contradict it. As an example, the following configuration would ALLOW traffic from 54.12.34.34/32:
  + Bastion Host – A way to connect to an instance in a private subnet
    - A NAT is used to provide internet traffic to EC2 instances in private subnets.
    - A bastion is used to securely administer (with SSH or RDP) EC2 instances in private subnets (also called jump boxes).
  + VPC Endpoints – Is a way to allow traffic to internal AWS services without having to go through the internet (and thus leaving the datacenter). You add them to your route table.
  + Network Flow Logs
    - Enable you to capture IP traffic information for the network interfaces in your resources.
    - You cannot tag a flow log.
    - You cannot enable flow logs to peered VPCs unless the peer VPC is in your account.
    - After you have create a flow log you cannot change its configuration, for example, you can’t associate a different IAM role with the flow log.
    - Not all traffic is monitored
      * Traffic generated by instances when they contact Amazon DNS server.
      * Traffic generated by Windows instance for Amazon Windows license activation.
      * Traffic to and from 169.254.169.254 for instance metadata.
      * DHCP traffic
      * Traffic to the reserved IP address for the default VPC router.
  + VPC Clean-up
    - Delete EC2 instances
    - Delete NAT Gateways
    - Delete Internet Gateways (detach it first)
    - Delete Endpoints
    - Finally delete the VPC (won’t delete any key pairs)
* Internal DNS is set at the VPC level, not the subnet level.
* For instances launched in a VPC, a private IPv4 address remains associated with the network interface when the instance is stopped and restarted, and is released when the instance is terminated.
* VPC Peering
  + VPC peering is simply a connection between two VPCs that enables your to route traffic between them using private IP addresses.
  + You can create a VPC peering connection between your own VPCs, or with a VPC in another account within a single region.
  + In order to work all VPCs must use different CIDR blocks (VPCs peered cannot have a matching or overlapping CIDR block).
  + Transitive peering is not support. For example, in the picture above VPC A can’t communicate with VPC C. VPCs must be at a single hop of distance for peering to work. You can’t peer via another VPC. You would have to peer VPC A to VPC C if you wanted them to communicate.

**Management Tools**

**OpsWorks**

* Orchestration service that uses chef.
* Chef consists of recipes to maintain a consistent state.
* Cookbook is a collection of recipes.

**Application**

**SQS**

* First service released by Amazon Web Services.
* Simple Queue Service – Distributed queue service that allows you to store messages while they wait to be processed.
* Queue is a temporary repository for messages that are awaiting processing.
* Auto scale groups that sit behind SQS queues so can scale EC2 fleet based on queue size.
* Visibility Timeout
  + Is the amount of time that the message is invisible in the SQS queue after a reader picks up that message.
  + Provided the the job is processed before the visibility timeout expires, the message will then be deleted from the queue. If the job is not processed within that time, the message will become visible again and another reader will process it. This could result in the same message being delivered twice.
  + Maximum visibility timeout is 12 hours.
* Great way to decouple your infrastructure.
  + The queue resolve issues that arise if the producer is producing work faster than the consumer can process it, or if the producer or consumer are only intermittently connected to the network.
* Messages  can be up to 256KB of text in any format.
  + But you are billed for each chunk of 64KB. Each chunk is billed as one request.
  + Messages can be configured so the minimum MaximumMessageSize is 1Kb.
* SQS is pull based, not PUSH based.
* Messages can be kept in the queue from 1 minute to 14 days. Default is 4 days.
* Guarantees that the message will be processed at least once.
* Long Pooling
  + Is a way to retrieve messages from the queue. While the regular short pooling returns immediately, even if the message queue being pooled is empty, long pooling doesn’t return a response until a message arrives in the message queue, or the long pool times out.
  + Maximum timeout for long pooling is 20 seconds.
  + It is a way to save money when the queue is empty because you are charged by the number of requests.
* Two types of queues
  + Standard Queues – Default
    - Nearly-unlimited number of transactions per second
    - Guarantee that a message is delivered at least once, however, occasionally (because of distributed architecture) a more than one copy of message might be delivered our of order.
    - Provides best effort in delivering messages in the same order they arrived.
  + FIFO Queues – First In First Out
    - Guarantee the delivering order
    - Exactly once processing.
    - Duplicates are not introduced in the queue.
    - This type of queue is limited to 300 transaction per second but have all the capabilities of a standard queue.
* Default values when creating a queue are
  + Default visibility timeout: 30 seconds
  + Message retention period: 4 days
  + Maximum message size: 256KB
  + Delivery delay: 0 seconds
  + Receive message wait time: 20 seconds
* A single pull request can return at most 10 messages at a time, up to a maximum payload of 256KB.
* If a message must be delivered to multiple SQS queues you can create an SNS topic and subscribe all queues to this topic, all will receive the message.
* You can create any number of SQS queues, there is no limit as per the SQS FAQ.
* The message body should not be empty or null.
* SQS message may contain up to 10 optional attributes.
  + An attribute consists of name, type and value.
  + If an attribute is informed then name, type, and value must not be empty or null.
  + Attribute types can be; string, number, byte or a custom type (informed as a string).

**SWF**

* Simple Workflow Service – Provides a way to coordinate tasks. Tasks represent invocations of various processing steps in an application which can be performed by:
  + Executable code
  + Web service calls
  + Human actions
  + Scripts
* SQS vs SWF
  + SQS has a retention period of 14 days.
  + SWF up to 1 year for workflow executions.
  + SQS presents a message oriented API.
  + SWF presents a task oriented API.
  + SWF ensures that a message is assigned only once and is never duplicated.
  + SQS you need to handle duplicated messages
  + SWF keeps track of all tasks and events in an applicatio
  + SQS you need to implement your own application-level tracking, specially if your application uses multiple queues.
* Actors
  + Starters – Initiates a workflow.
  + Deciders – Control flow. If something has finished or failed in a workflow a Decider decides what to do next.
  + Activity Workers – Carry out the activity tasks.

**SNS**

* Simple Notification Service – Makes it ease to set up, operate and send notifications from the cloud to subscribers or other applications
* Notification types
  + Push notifications to Apple, Google, Fire OS, and Windows devices.
  + Android devices in China with Baidu Cloud Push.
  + SMS messages
  + Email
  + Post messages to SQS
  + Fire HTTP requests
  + Trigger lambda functions subscribed to it
  + Publish to other SNS topic
  + Send messages to other AWS services.
* One topic can deliver to multiple notification types.
* All messages published to SNS are stored redundantly across multiple availability zones.
* Auto-scaling integrates with SNS to notify about increasing/decreasing EC2 instances.
* Instantaneous push-based deliveries (no pooling).
* To prevent messages from being lost, all message published to SNS are stored redundantly across multiple availability zones.
* SNS vc SQS
  + Both are messaging services
  + SNS – Push
  + SQS – Pool
* Pricing
  + Per request
  + Per delivery transport

**Elastic Transcoder**

* Convert media files from their original source formats into different formats that will play on smartphones, tablets, PCs, etc.
* Pricing
  + Based on the minutes you transcode.
  + The resolution at you transcode.

**API Gateway**

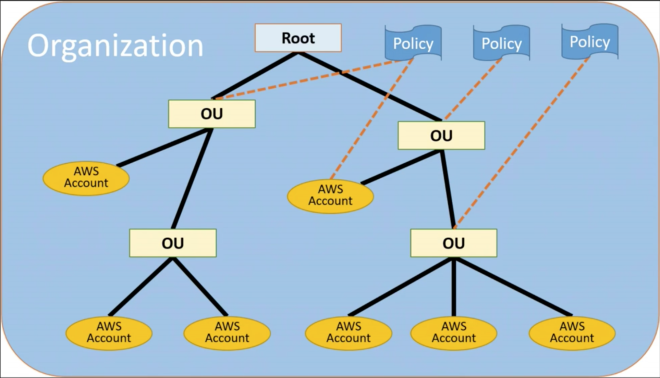
* Helps creating HTTP APIs that act as a front door for another compute services in AWS, such as EC2 or Lambda.
* You can enable cache for API responses (you will set a TTL in seconds).
* You can throttle requests to prevent attacks
* Connect to CloudWatch to log all requests
* You can enable CORS.
* Low cost and scales automatically
* API Gateway does not support unencrypted traffic, all endpoint created are HTTPS by default.

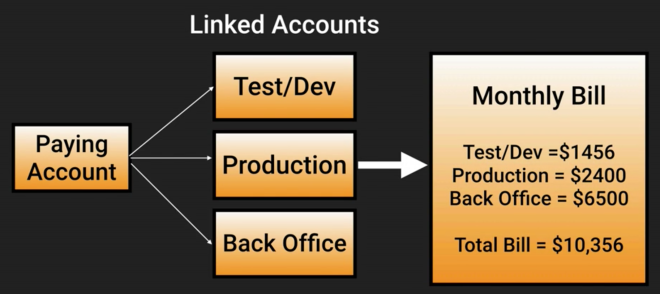
**Kinesis**

* Streaming data – Is data that is generated continuously by thousands of data sources, which typically send in the data records simultaneously, and in small sizes (KB).
* Kinesis is a platform to send your streaming data to. Makes it easy to load streaming data. A way to bring big data onto the cloud (hundreds of thousands of producers) and a way to consume it.
  + Example: Analyse thousands os tweets in real time to identify a trend about a subject. Wether people are thinking about Bitcoin pricing going up or down for example.
  + Other services for big data
    - Redshift – Business intelligence
    - Elastic map Reduce – Big data processing
* Three core Kinesis services
  + Kinesis Streams – Made up of shards. Retention period from 24 hours (default) to 7 days.
  + Kinesis Firehose – No shards. You can optionally use lambda to analyze data and then send to S3. Can send data to ElsticSearch or to Redshift through S3.
  + Kinesis Analytics – A way to analyze data stored inside Kinesis by running SQL queries.
* Retention Period – The time period from when a record is added to when it is no longer accessible is called the retention period. A Kinesis data stream stores records from 24 hours by default, up to 168 hours.

**Organisations & Consolidated Billing**

* AWS Organisations is an account management service that enables you to consolidate multiple AWS accounts into an organisation that you can create and centrally manage.
* Available in two feature sets:
  + Consolidated billing
  + All features

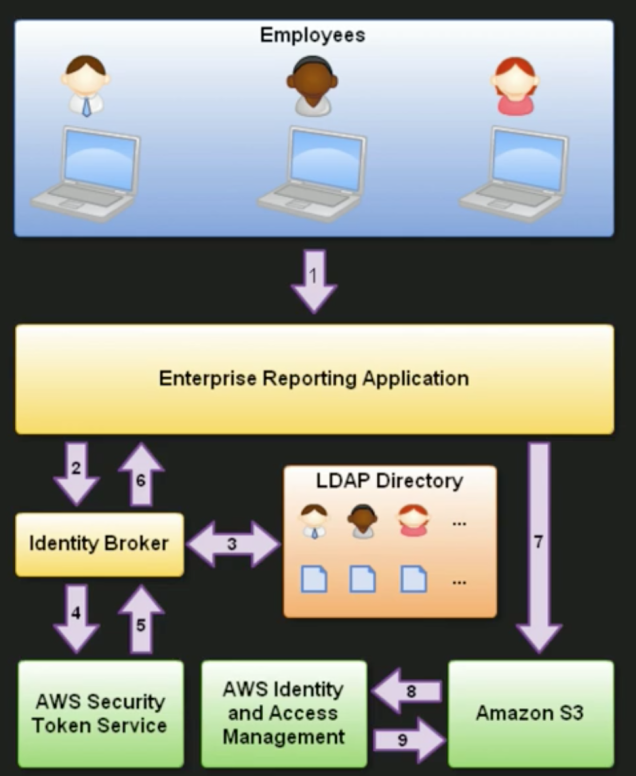




* Paying account is independent and cannot access resources of the other accounts.
* Currently there is a limit of 20 linked accounts for consolidated billing. You can add more but you have to open a support ticket.
* Advantages
  + One bill per AWS account
  + Very easy to track charges and allocate costs
  + Volume pricing discount
  + Unused reserved instances for EC2 are used across the group.
* Billing Alerts
  + When monitoring is enabled on the paying account the billing data for all linked accounts is included.
  + You can create individual alerts per individual account.
* CloudTrail – You can enable cloud trail to consolidate logs from individual accounts by enabling CloudTrail on all accounts and sending logs to the same bucket by using S3 cross account access. CloudTrail is on a per account and per region basis.
* Best practices for root account
  + Always enable MFA
  + Always use strong and complex password
  + Should be used for billing purposes only. You should do not deploy resources into root account.
* Cross account access – Allows for logging into multiple account and switch between them without having to reenter credentials.

**Security Token Service**

* Grants users limited and temporary access to AWS resources. Users can come from 3 sources:
  + Federation – Active Directory, using SAML
  + Federation with mobile apps – Using Facebook/Google/OpenId providers to log in.
  + Cross account access – Lets users in one account access resources in another.
* Key terms
  + Federation – Combining or joining a list of users in one domain (such as AMI) 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 and join it (federate it) to point B. Calls GetFederationToken and passing an IAM policy and a duration, from 1 to 36 hours. Identity brokers are developed in house.
  + Identity Store – Services like Active Directory, Facebook, etc.
  + Identities – A user of a service like Facebook.
* Example Security Token Service flow:



* When federation with Active Directory
  + API call to AssumeRoleWithSAML​.
  + Sign-in endpoint is <https://signin.aws.amazon.com/saml>
* Web Identity Federation
  + Login with Facebook, Google, etc.
  + The Web Identity Federation Playground is available for testing under AMI.
  + API call to AssumeRoleWithWebIdentity
* Default session timeout for both AssumeRoleWithSAML and AssumeRoleWithWebIdentity is **1 hour**.
  + Session timeout can be configured from 15 minutes to a maximum of 1 hour.

**Workspaces**

* A Workspace is a cloud-based replacement for a traditional desktop.
* Windows 7 experience, provided by Windows Server 2008 R2.
* By default you will be given administrator access so you can install your applications.
* Workspaces are persistent.
* All data in the D:\ drive is persisted every 12 hours.
* You do not need an AWS account to log in to workspaces.

**Resource Groups & Tagging**

* Tags are key/value pairs attached to AWS resources
* 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.
* You can use the resource editor to find resources and tag them.
* *You are developing a web application, and you are maintaining separate sets of resources for your alpha, beta, and release environments. Each version runs on Amazon EC2 with an EBS volume. You use Elastic Load Balancing to manage traffic and Amazon Route 53 to manage your domain. What’s the best way to check the health and status of all three groups of services simultaneously?* With the Resource Groups tool, you use a single page to view and manage your resources.

**AWS Support**

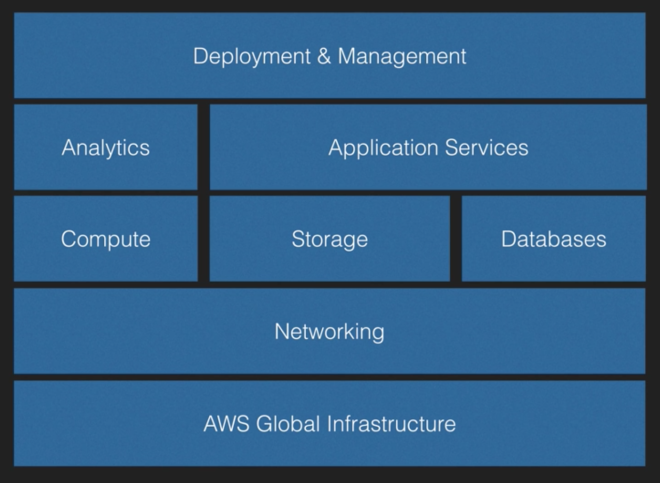
* Four levels of support available. All levels provide 24/7 access to customer service.
  + Basic – Access to 6 core Trusted Advisor checks
  + Developer – Access to 6 core Trusted Advisor checks
  + Business – Access to full set of Trusted Advisor checks
  + Enterprise – Access to full set of Trusted Advisor checks
* The maximum response time for a Business Level Premium Support Case is 1 hour

**Trusted Advisor**

* Trusted Advisor tool offers a one-view snapshot of your service and helps identify common security misconfigurations, suggestions for improving system performance, and underutilized resources.
* Security Aspects – Trusted Advisor checks for compliance with the following security recommendations:
  + Limited access to common administrative ports to only a small subsetof addresses. This includes ports 22 (SSH), 23 (Telnet) 3389 (RDP), and 5500 (VNC).
  + Limited access to common database ports. This includes ports 1433 (MSSQL Server), 1434 (MSSQL Monitor), 3306 (MySQL), Oracle (1521) and 5432 (PostgreSQL).
  + IAM is configured to help ensure secure access control of AWS resources.
  + Multi-factor authentication (MFA) token is enabled to provide two-factor authentication for the root AWS account.

**Whitepapers**

**AWS Whitepaper**

* Cloud computing is the on-demand delivery of IT resources and applications via the internet with pay-as-you-go pricing.
* Advantages
  + Trade capital expense for variable expense.
  + Benefit from massive economies of scale.
  + Stop guessing about capacity.
  + Increase speed and agility.
  + Stop spending money running and maintaining your data center.
  + Go global in minutes.
* Several regions across the world, comprising at least two availability zones.
* AWS adhere to several certifications.
* AWS comply to several industry-specific standards. 

**Security Whitepaper**

* Recommended reading: [click here to read it.](https://d0.awsstatic.com/whitepapers/Security/AWS_Security_Whitepaper.pdf)
* 20% of the exam is based on security. Very important to read the whole whitepaper.
* Shared security model
  + AWS is responsible for securing the underlying infrastructure that supports the cloud. AWS is also responsible for the security configuration of its products that are considered managed.
  + You are responsible for anything you put on the cloud or connect to the cloud.
* Storage decommissioning. AWs uses industry-standard practices to destroy storage devices and make sure no data can be accessed after it is disposed.
* Transmission protection – You can connect to AWS via HTTP or HTTPS. Additional layers of network security can be provided via VPC (virtual private cloud), which provides a private subnet within the AWS cloud, and the ability to use an IPsec Virtual Private Network (VPN) device to provide an encrypted tunnel between the Amazon VPC and your data centre.
* Amazon.com network is logically segregated from the AWS production network.
* Default protections
  + DDoS
  + Man in the middle
  + Ip spoofing – AWS won’t permit an instance to send traffic with a source IP or MAC address other than its own.
  + Port scanning – Port scanning by an EC2 instance is a violation of the AWS Acceptable Use Policy. You may request permission to conduct vulnerability scans as required to meet your compliance requirements.
  + Packet sniffing by other tenants
* AWS Trusted Advisor
  + Inspects your AWS environment and makes recommendations when opportunities may exist to save money, improve system performance, or close security gaps.
* Instance isolation
* Guest operating system
  + AWS does not have any access rights to you instances or the guest OS.
  + In order to be able to provide EBS encryption and to do this efficiently and with low-latency this feature is only available on more powerful instances like M3, C3, R3 and G2.
* Firewall – Amazon EC2 provides a complete firewall solution. Configured in a deny-all default mode. Customers must explicitly open the ports needed to allow inbound traffic.
* Direct Connect
  + Bypass internet service providers in your network path. Provides a more consistent network experience than internet-based connections.
  + Main benefits
    - Reduce costs when using large volumes of traffic
    - Increase reliability
    - Increase bandwidth
  + Direct Connect vs VPN
    - VPN connections can be configured in minutes and are a good solution if you have an immediate need, have low to modest bandwidth requirements, and can tolerate the inherent variability in internet-based connectivity.
    - AWS Direct Connect does not involve the internet; instead, it uses dedicated, private network connections between your intranet and Amazon VPC.
    - Having a VPN is considered a backup to a Direct Connect connection.
* AWS platform is PCI DSS 1.0 compliant, I can’t immediately deploy a website to it that can take and store credit card details. I do need to get delta accreditation from a QSA.

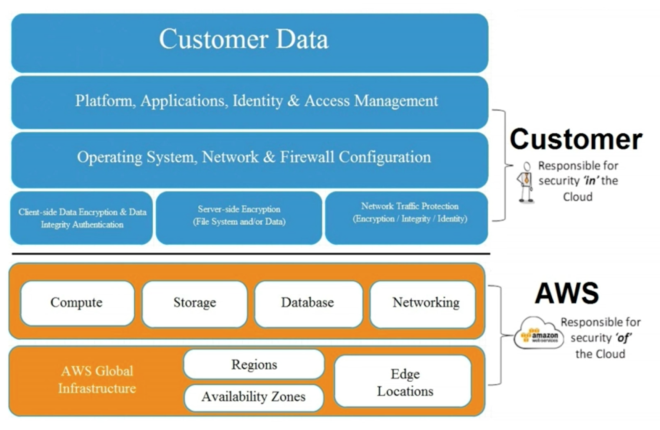
**Architecting for the cloud: best practices**

* Design for failure – Be a pessimist when designing architectures for the cloud. Assume things will fail. In other words, always design, implement and deploy for automated recovery from failure.
  + Example: Netflix Chaos Monkey – Software that randomly shuts down EC2 instances to check weather they are indeed resilient.
* Decouple your components – Think SQS.
* Implement elasticity.

**The Well-Architected Framework**

* A set of questions designed to evaluate how well your architecture is aligned with AWS best practices.
* General design principles:
  + Stop guessing you capacity needs.
  + Test systems at production scale.
  + Automate to make architectural experimentation easier.
  + Allow for evolutionary architectures.
  + Data-driven architectures.
  + Improve through game days (chaos monkeys)
* [Click here](https://renatoargh.wordpress.com/2018/01/20/the-well-architected-framework-checklist/) to see all questions in across all pillars

**Pillar 1 – Security**

* Design principles:
  + Apply security at all layers
  + Enable traceability
  + Automate events to security events
  + Focus on securing your system – data, application and operating system.
  + Automate security best practices.
* Shared responsibility model
* Security in the cloud consists of 4 areas:
  + Data protection
  + Privilege management
  + Infrastructure protection
  + Detective controls

**Pillar 2 – Reliability**

* Design principles:
  + Test recovery procedures.
  + Automatically recover from failure.
  + Scale horizontally to increase aggregate system availability.
  + Stop guessing capacity
* Reliability in the cloud consists of 3 areas:
  + Foundations
  + Change management
  + Failure management

**Pillar 3 – Performance Efficiency**

* Design principles
  + Democratize advanced technologies
  + Go global in minutes
  + Use server-less architectures
  + Experiment more often
* Performance efficiency in the cloud consists of 4 areas:
  + Compute
  + Storage
  + Database
  + Space-time trade-off

**Pillar 4 – Cost Optimisation**

* Design principles
  + Transparently attribute expenditure.
  + Use managed services to reduce the cost of ownership.
  + Trade capital expense for operational expense.
  + Benefit form economies of scale.
  + Stop spending money on data centre operations.
* Cost optimisation in the cloud consists of 4 areas:
  + Matched supply and demand
  + Cost-effective resources
  + Expenditure awareness
  + Optimising over time

**Pillar 5 – Operational Excellence**

* Design Principles
  + Perform operations with code.
  + Align operation processes to business objectives.
  + Make regular, small, incremental changes
  + Test for responses to unexpected events
  + Learn from operational events and failures
  + Keep operations procedures current
* Operational excellence in the cloud consists of 3 areas:
  + Preparation
  + Operation
  + Response