AWS Certified solutions associate

- Udemy practice exams
- Whizlab practice exams
- ExamTopics questions
- DataCumulus courses
- AWS FAQs
- TutorialsDojo AWS cheatsheets

---

1. Compute

1.1. EC2

> Instance

- Amazon Machine Image (AMI) provides the info required to launch an instance
- Instance metadata: instance ID, public keys, public IP address
- Multi-AZ uses synchronous replication ensuring almost no RPO (recovery point objective). Read replicas take longer
- To monitor custom metrics, you must install the CloudWatch agent on the instance
- Billing
    - On-demand: billed only when it's running or stopping, not when pending
    - Spot: not billed if instance is in stopping state
    - Reserved instance: billed until end of the term even if it's terminated
- There is a vCPU-based on-demand instance limit per region, if you want more you can submit the limit increase form to AWS and retry the failed requests once approved
- When restarting an instance, the Elastic IP address remains associated with the instance, and the ENI (elastic network interface) stays attached as well

Storage

- Instance store volume and RAM: ephemeral, data is lost when instance is stopped. RAID 0 configuration enables you to improve your storage volumes' performance by distributing the I/O across the volumes in a stripe
    - If data should be kept, enable hibernation and hibernate instance before shutdown. Snapshotting the instance won't help because RAM contents are reloaded
- EBS and EFS are persistent. If you start and stop an EC2 instance, the EBS volume associated with it is preserved but the data is erased

Instance types

- Standard reserved instance: more discount, can't exchange instances but can change AZ, scope, network platform, or instance size
- Convertible reserved instance: flexibility to change families, OS types and tenancies
- Scheduled reserved instance: purchase capacity reservations that recur on a daily/weekly/monthly basis with a specified start time and duration, for a one-year term

If you purchased a reserved instance but you want to stop it, terminate the RI asap to avoid getting billed at the on-demand price when it expires and sell it in the AWS RI marketplace

Networking

- If the instance should send/receive traffic over the Internet, it should have a public IP address associated with it
- To reduce data transfer costs between instances, deploy them in the same AZ
- To establish a SSH connection to the instance that's inside a VPC:
    - On the security group (access to instance), add inbound rule to allow SSH to instance
        - Security groups are stateful: if inbound traffic is granted, outgoing traffic is automatically granted as well
    - On the NACL (access to whole subnet), add inbound + outbound rule to allow SSH to instance
        - NACL are stateless
        - if you only enabled an Inbound rule in NACL, the traffic can only go in but the SSH response will not go out since there is no Outbound rule

- To accelerate HPC apps, add an Elastic Fabric Adapter (EFA), which is a network device but doesn't work for Windows instances. For Windows instances, use Elastic Network Adapter (ENA)

1.1.1. Autoscaling

A service that increases and decreases the number of instances based on certain metrics: CPU usage, memory usage etc. To create an autoscaling group, create a launch configuration and set an AMI. When downscaling, it deletes the ones with the oldest launch config, from the AZ with the most #instances, and the ones closest to the next billing hour.

Cooldown period: ensures that Auto scaling group doesn't launch/terminate any instances before the previous scaling activity takes effect. Default value is 300s. Prevents the instances to be scaling up and down very fast.

> High availability

For highly available instances, deploy in at least 2 AZ.

If at least 2 instances should be running all the time, we need 2 AZ and 2 instances in each. Worst case, one AZ fails but we still have 2 instances running. If we only had 1 instance in the unaffected AZ, Autoscaling would deploy the second instance but it would take some time, for a while there would be only 1 instance. Which we don't want.

If you need at least 6 instances, deploy 3x3x3 (3 in each AZ). If one AZ goes down, you still have 6 instances running. Also it's low-cost because there's 9 instances running in total. You could also have 6x6x0 but then you always have 12 instances running, more costly. If you need at least 4 instances, deploy 2x2x2.

- Min 2 instances: 1x1x1, or 2x2
- Min 4 instances: 2x2x2
- Min 6 instances: 3x3x3

> Scaling types

- Simple scaling: based on a single scaling adjustment
- Step scaling: choose scaling metrics and threshold values for the CloudWatch alarms that trigger the scaling. allows multiple targets
- Target tracking scaling: specify a target value for a specific metric. only one target
- Scheduled scaling: for predictable load changes, e.g. when expecting a load at 9AM when people come to work

1.2. ECS

Which services run the containers?

- EC2 instances (which have the ECS container agent running)
- Fargate launch type (serverless). Has less capacity than EC2

ECS can handle bursts in traffic but it takes minutes to set up new containers.

ECS tasks can be run on CloudWatch events, e.g. when a file is uploaded to a certain S3 bucket using a S3 PUT operation. You can also declare a reduced number of ECS tasks whenever a file is deleted on the S3 bucket

using the DELETE operation. First, create an Event **rule** for S3 that watches for object-level operations (PUT, DELETE). For object-level operations, it is required to create a CloudTrail trail first. On the Targets section, select the "ECS task" and input the needed values such as the cluster name, task definition and the task count. You need two rules – one for the scale-up and another for the scale-down of the ECS task count.

To insert sensitive data into containers, you can store it in Secrets Manager secrets or System Manager Parameter Store parameters and then you can reference them in the container definition.

- Use the `secrets` container definition parameter to inject sensitive data as environment variables
- Use the `secretOptions` container definition parameter to inject sensitive data in the log configuration of a container

1.3. Lambda

Event driven, serverless compute service. Thought for fast operations that take less than 15mins. They can be used to handle bursts of traffic in seconds.

You can use aliases when updating functions, to have the Lambdas versioned. This enables canary deployment (e.g. only sending 20% to the updated function)

For sensitive info in env variables, create a *new* KMS key and use it to enable encryption helpers that leverage on KMS to store and encrypt the sensitive info. Lambda encrypts the environment variables in the function by default, but the info is still visible to other users who have access to the Lambda console. Lambda uses a default KMS key to encrypt the variables, which is usually accessible by other users.

- Provisioned concurrency: keep instances provisioned, more expensive
- Reserved concurrency: dedicated reservations of parallel execution for your function. This number will be subtracted from your default account soft limit of 1000 parallel executions
    - Guarantees that this concurrency level is always possible for your function
    - But concurency can't be exceeded

Lambda@Edge: allows you to execute code at different times when the CloudFront distribution is called, processing done closer to the edge.

Step functions: serverless orchestration, coordinates AWS services into serverless workflows.

Other compute services:

- Fargate: useful for microservices and launching containers in a serverless way, also helps with container cluster management. Removes the need to provision and manage servers, let you specify and pay for resources per application, and improve security through application isolation by design. For ECS & Fargate, specify CPU and memory at the task definition.
- Beanstalk: PaaS service for webapps. Supports deployment from Docker containers. Capacity provisioning, load balancing, scaling and health monitoring are automatically handled.

2. Storage

> Hybrid storage, AWS Storage Gateway

Integrate the on-premises network to AWS, used for hybrid cloud storage, on-premises systems are kept. Enables Active Directory users to deploy storage on their workstations as a drive.

- File Gateway: supports a file interface into S3, and you get a virtual software appliance on-prem. Supports protocols NFS and SMB. Don't use S3 API to access data moved to S3
- Volume Gateway: cloud-backed storage volumes that you can mount as iSCSI devices from on-prem servers, uses EBS volumes
    - Cached volumes: store data in S3, copy of frequently accessed subsets on-prem. used if you need access to the frequently accessed data subsets locally
    - Stored volumes: store data on-prem, asynchronously backup snapshots to S3. used if you need low-latency access to your entire dataset
- Tape Gateway: archive backup to Glacier, supports iSCSI

> Data migration

- AWS DataSync: upload all data to AWS, 100% cloud architecture. nothing stored on-prem. Used for replication of data to and from AWS storage services
- AWS Snowball edge: type of Snowball device with on-board storage and compute power. Each Snowball Edge device can transport data at speeds faster than the internet. Can't directly integrate backups to S3 Glacier, only to S3.
- AWS Snowmobile: exabyte-scale data transfer service. Up to 100PB per Snowmobile

2.1. S3

Object type storage service. Supports up to 3500 PUT and 5500 GET requests per second.

> Storage types

- S3 standard: no minimum storage duration charge
- S3 Standard IA and One Zone-IA: for infrequent access, but rapid access. Not for backup. the lifecycle policy can only transition data to this storage type *after 30 days of creation*. min data storage duration: 30 days
- S3 Glacier *for archiving*. Lifefycle policy can transition data to this storage type at any point
    - Expedited retrieval: allows you to quickly access data if you have an urgent request. Provisioned capacity ensures that retrieval capacity for expedited retrievals is available when you need it.
    - Glacier supports vault lock policy, which helps enforce regulatory and compliance requirements
    - Glacier: min data duration: 90 days
- Glacier deep archive: min data storage duration: 180 days

With lifefycle policy, you can specify that the data is moved to another storage class (like for archiving).

> Access

- All objects public to the Internet: configure the bucket policy to set all objects to pubic read. Grant public access to the object when uploading it using the console
- Access only to specific objects: pre-signed URLs
- Grant access to trusted buckets: set and endpoint policy. Bucket policy also works but it takes a lot of time.

> Extract data

- Amazon Athena: analyze data directly in S3 using standard SQL. Can work on many objects
- S3 select: retrieve only a subset of the object by using simple SQL expressions
- Amazon Macie: ML-powered service that monitors and detects usage patterns on S3 data, detect anomalies, risk of unauthorized access or inadvertent data leaks. It can recognize PII (personally identifiable info)

Other info

- Asynchronously replicates data in all AZs in a region
- CORS (cross-origin resource sharing) allows webapps loaded in one domain to interact with resources in a different domain. For instance, to add JavaScript to the webapp
- Cross-region replication needs to have versioning enabled first
- S3 object lock allows you to store objects using a write-once-read-many (WORM) model. Changes to objects are allowed but their previous versions should be preserved and remain retrievable. If you enabled S3 Object Lock, you won't be able to upload new versions of an object. This feature is only helpful when you want to prevent objects from being deleted or overwritten for a fixed amount of time or indefinitely
- Server access logs for S3 buckets provide detailed records for the requests that are made to an S3 bucket, e.g. requester, bucket name, request time, request action, referrer etc.

Retention modes

- Governance - overwrites/deletes are only possible with specific rights
- Compliance - no deletes or overwrites possible for the duration of the retention period

Securely serve private content via CloudFront

- Require that users access the private content by using special CloudFront signed URLs or signed cookies
- Requre that users access S3 content via CloudFront urls, not s3 urls. set up an origin access identity (OIA) for the bucket and give it permission to read files in the bucket

S3 notification feature

It can send notifications when events happen in a bucket. S3 event notifications are designed to be delivered at least once and to one destination only, e.g. SNS, SQS or Lambda but only one. You cannot attach two or more SNS topics or SQS queues for S3 event notification. If you need 2 SQS queues, set up a SNS queue and fan out to several SQSs.

S3:ObjectRemoved:DeleteMarkerCreated is triggered when a delete marker is created for a versioned object and not when an object is deleted or a versioned object is permanently deleted.

S3 Transfer acceleration

Transfer Acceleration enables fast, easy, and secure transfer of files over long distances between your client and your Amazon S3 bucket. Transfer Acceleration leverages Amazon CloudFront's globally distributed AWS Edge Locations. As data arrives at an AWS Edge Location, data is routed to your Amazon S3 bucket over an optimized network path.

2.2. EBS

- **They can only be attached to instances in the same AZ**, therefore don't support multi-AZ resiliency. Don't allow concurrent connections from multiple instances
- EBS volumes spport live config changes in production, you can change volume type, size and IOPS capacity without service interruptions
- EBS volumes are off-instance, they can persist independently from the life of an instance. To prevent EBS from being deleted when an instance terminates, set the value of DeleteOnTermination to False

> Types

Features	**SSD**	**HDD**
Best for	small, random I/O operations	large, sequential I/O operations
Can be used as a bootable volume?	Yes	No
Suitable use cases	Transactional workloads, sustained IOPS performance, large db workloads	large streaming workloads, big data, data warehouses, throughput-oriented storage for large volumes of data that's infrequently accessed
Cost	Moderate/high	Low

- General purpose: SSD-backed, used as boot volume
- Provisioned IOPS: I/O intensive, low-latency. best for NoSQL or large RDBS. Requires lots of storage
- Throughput optimized: large operations (large data), also magnetic
- For infrequently accessed data, always cold HDD. Magnetic volume has lowest cost per GB

> Snapshots

- To back up all EBS volumes, use Amazon Data lifecycle manager (DLM) to automate the creation of snapshots
- The EBS can still be used while the snapshot is being created
- Snapshots are automatically encrypted, and all data moving between the volume and the instance are encrypted
- To ensure that restored volumes from unencrypted snapshots are automatically encrypted, enable EBS Encryption by default feature for the *Region*

2.3. File storage

- Amazon FSx For Lustre: high-performing *parallel* file system for fast processing of workloads, useful for HPC. It has POSIX interface
- Amazon FSx For Windows File Server: fully managed Microsoft Windows filesystem with support for SMB protocol, Windows NFTS, Microsoft Active Directory integrations. Useful for app workloads that require shared file storage
- EFS: only supports Linux workloads, allow concurrent connections from multiple instances hosted on multiple AZs

2.3. RDS

Relational databaser service. Provides storage autoscaling to scale storage capacity with 0 downtime. Supports Multi-AZ RDS db, for higher availability in case of AZ failure. This has synchronous replication to the secondary db. If the primary db were to fail, the canonical name record (CNAME) switches from the primary to standby instance.

- To migrate databases to AWS, use Schema Conversion tool to convert the source schema and application code to match that of the target databse, and then use DB migration service to migrate data from the source db to target db
- ElastiCache caches database query results for faster retrieval, caches data storage for user sessions
- Cloudwatch has the following enhanced monitoring metrics for RDS: RDS child processes and OS processes
- To encrypt network traffic from and to the RDS db with SSL, enable the IAM DB authentication. IAM database authentication works with MySQL and PostgreSQL. With this authentication method, you don't need to use a password when you connect to a DB instance. Instead, you use an authentication token.

2.4. Aurora

Relational db, supports dynamic storage scaling and can conduct table joins. Automatically scales to accomodate data growth. Can have more throughput than MySQL and PostgreSQL.

An Aurora DB cluster consists of one or more DB instances and a cluster volume that manages the data for those DB instances. Each Aurora DB cluster can have up to 15 Aurora Replicas in addition to the primary DB instance.

If you have an Amazon Aurora Replica in the same or a different Availability Zone, when failing over, Amazon Aurora flips the canonical name record (CNAME) for your DB Instance to point at the healthy replica, which in turn is promoted to become the new primary. But if you only have one instance, Aurora will attempt to create a new DB Instance in the same Availability Zone as the original instance. This replacement of the original instance is done on a best-effort basis and may not succeed, for example, if there is an issue that is broadly affecting the Availability Zone

Aurora serverless DB cluster: DB cluster that automatically starts up, shuts down and scales up/down based on the app's needs. It's a simple, cost-effective option for infrequent, intermittent sporadic or unpredictable workloads. You can create a db endpoint without specifying the DB instance class size, you only set the min and max capacity. The endpoint connects to a proxy fleet that routes the workload to a fleet of resources that are automatically scaled.

A non-Serverless DB cluster for Aurora is called a provisioned DB cluster.

Aurora Global db is designed for globally distributed apps, it spans multiple regions. It supports storage-based replication (RPO) with a latency of <1s. If there's an unplanned outage, one of the secondary regions you assigned can be promoted to read and write capabilities (RTO) in <1min. Consists of one primary AWS Region where your data is mastered, and one read-only, secondary AWS Region.

- Reader endpoint: load-balances each connection request among the Aurora replicas, read-only connections
- Cluster/writer endpoint: connects to the primary db instance, used for write operations

2.5. DynamoDB

DynamoDB is a multi-AZ, NoSQL db (suitable for key-value stores) that can handle frequent schema changes and doesn't have downtime with schema changes. Automatically scales storage capacity. But write capacity can be increased.

- DynamoDB + AppSync: to keep shared data updated in real time where users from around the world submit data
- DynamoDB + ElastiCache: provide high performance storage of key-value pairs
- DynamoDB autoscaling: can be used directly to dynamically adjust provisioned throughput capacity in response to traffic patterns
- Global tables: synchronized tables in different regions
- CloudWatch by default captures: user read capacity units, user write capacity units, throttles
- By default, tables are encrypted with KMS. You can use a customer-managed key (CMK)
- DynamoDB Stream: invoke of other services if items are created/updated/deleted. A stream record contains info about data modifications. You can configure the stream so that the record captures additional info, e.g. the before/after images of modified items. You can set up triggers so that a specific change in a table triggers a Lambda function.

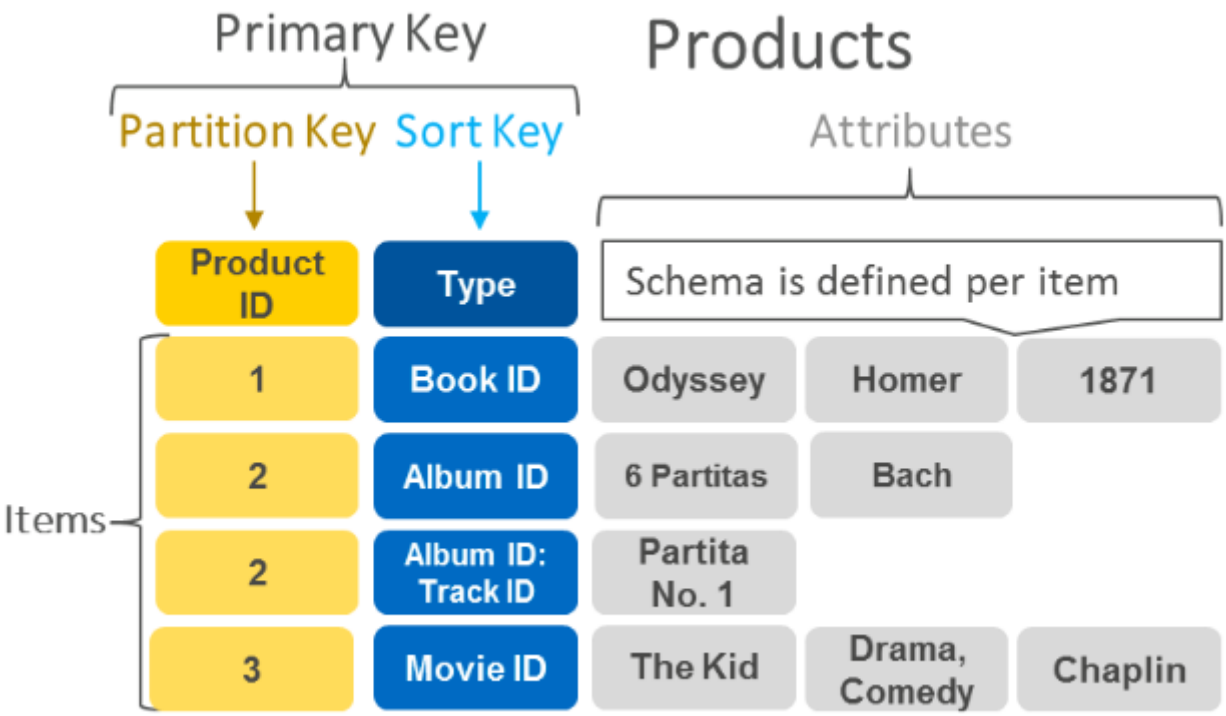
Capacity

- Provisioned capacity: specify the capacity units for the table and get billed for them. Useful for steady load or known patterns
- On-demand: paying per request (good for unpredictable traffic). Pricing is pay-per-request for read and write requests

Partitions

A document in DynamoDB doesn't have a fixed schema. Each table defines the primary key, the unique identifier for each table and it must be provided when inserting a new item:

- Simple (single field): also the partition key
- Composite: build-up via the partition and range key. Range key can be used with expressions



Internally, DynamoDB has different partitions where the items are stored. The partition key runs through a hash function whose result determines the partition. A good partition should be equally distributed. This is important because the read/write capacity units are distributed among partitions. If items aren't well distributed, your requests are more likely of being throttled because you'll have hot partitions /partitions receiving high load).

To distribute db workload evenly and using provisioned throughput efficiently, use partition keys with high-cardinality attributes, which have a large number of distinct values for each item.

Secondary indexes:

- Local (local secondary index - LSI): needs to have the same hash/partition key, but an alternative range key. max 5 per table
- Global (GSI): partition & range key can be different. max 20 per table

> Search

- Query: looks for items at a specific partition. You're billed only for the retrieved items. Query works on indexes (partition & range key, if any). Cheaper and faster than a scan
- Scan: runs through the table looking for items that match your expression. You're billed by the items that are scanned

> Backups:

- On-demand: regularly trigger on-demand backups. A lambda function that triggers backups via aws-sdk. EventBridge rule that invokes the function regularly
- Continuous backups via point-in-time-recovery: more costly, allows you to restore the table to any state within the last 35 days.

2.6. Redis

If users need to authenticate, use Redis AUTH by creating a new Redis Cluster with both the `--transit-encryption-enabled` and `--auth-token` parameters enabled. The second parameters asks users for a password.

3. Network and content delivery

3.1. CloudFront

Low-latency and high-transfer speed content delivery network. Can't route the traffic to the closest edge location via an Anycast static IP address, for that use Global accelerator

- Geo restriction: block access for certain countries
- Cache: CloudFront serves an object from an edge location until the cache duration passes, then next time the edge location gets a user request for the object, CloudFront forwards the request to the origin server to verify that cache == latest version. The Cache-Control max-age directive lets you specify how long (in seconds) you want an object to remain in the cache before CloudFront gets the object again from the origin server
- If users around the world have HTTP 504 errors, set up an origin failover by creating an origin group with 2 origins: specify one as the primary origin, the other as secondary origin which CloudFront automatically switches to when the primary origin returns specific HTTP status code failure updates. You can also deploy the app to multiple AWS regions and set up a Route53 record with latency routing policy to route incoming traffic to the region that provides the best latency, but this has more costs

---

- Lambda@Edge: run general-purpose code on regional edge locations, reduces delay of processing. Only for serverless architectures. Executed in one of AWS' 13 regional edge caches. Supports JS/Python, 5s (viewer), 30s (origin triggers), max memory is 128MB (viewer) & 10GB (origin), has network access. Used in scenarios:
  - Viewer request/response: invoked at the start/end of all requests
  - Origin request/response: only when cloudfront requests the origin/retrieves a response
- Cloudfront functions: lightweight version of Lambda@Edge, less capabilities but better latency and cheaper. Executed in one of 218 edge locations. Used for access control & authorization, HTTP redirects, cache manipulation.
  - Supports JS, max exec time is 1ms, max memory is 2MB, has no network access.

AWS Global accelerator: uses the AWS global network to optimize the path from your users to your applications, improving the performance of your TCP and UDP traffic. It can direct user traffic to the nearest application endpoint to the client.

3.2. API Gateway

Service to build RESTful APIs and WebSocket APIs optimized for serverless workloads. Billing is only for the API calls you receive and the amount of data transferred out.

API Gateway has three major parts:

- Request flow: authentication, authorization
- Integration (what the client wants to do), e.g. a Lambda function
- Response flow: what happens after the integration, e.g. transformation

*Authorizer*s protect routes of the API. Protects downstream services and allows forwarding a security context, e.g. the details of an authenticated user. More info on twitter thread

---

Amazon API Gateway provides throttling at multiple levels including global and by a service call. Throttling limits can be set for standard rates and bursts. For example, API owners can set a rate limit of 1,000 requests per second for a specific method in their REST APIs, and also configure Amazon API Gateway to handle a burst of 2,000 requests per second for a few seconds

API Gateway can scale using AWS Edge locations, but for bursts of API, you need to configure throttling limits. Any request over the limit will receive a 429 HTTP response.

X-Ray: traces and analyzes **user requests** as they travel through the API Gateway and other microservices, not EC2 instances.

3.3. Load balancers

Load balancers distribute traffic within their respective region. For cross-region, choose Route 53.

- Must be deployed on a public subnet
- Load balancers can have access logs enabled to see info about the HTTP requests going through it
- Load balancers can route requests based on the host field, path url, HTTP header, HTTP method etc. Path-based routing allows you to route a request based on the URL path of the HTTP header
- Cross-zone load balancing allows load balancing across multiple AZs

Load balancer types:

- Network load balancer: 4th layer of the OSI model
- Application load balancer: supports path-based routing, host-based routing, bi-directional communication with WebSockets. Also supports weighted target groups routing
    - An ALB sends requests to healthy instances only. It performs periodic health checks on targets in a target group, if an instance fails the health check a configurable amount of times it'll be marked as unhealthy and won't receive traffic until it passes another health check

3.4. Route 53

DNS web service, it redirects traffic via domain names to your apps. DNS resolves domains into their IP addresses

> DNS routing

- Simple routing: used for single resources that are performing given functions in your domain. Can't create multiple records with the same name for this type
- Weighted routing: define multiple records for the same (sub-)domain name and choose how much traffic is routed to each one of them. Useful for load balancing
- Geolocation routing: route traffic based on the grographic location of *users*
- Geoproximity routing: routes traffic based on the geographic location of the *users+resources*. By specifying *bias*, you can choose how much of the traffic should be routed
- Latency routing: serves user requests from the AWS region with lowest latency. Users from the same location might get sent to different regions

> Health checking

- Active-active failover: when you want all of your resources to be available the majority of the time. When a resource becomes unavailable, Route 53 can detect that it's unhealthy and stop including it when responding to queries
- Active-passive failover: when you want a primary resource or group of resources to be available the majority of the time and you want a secondary resource or group of resources to be on standby in case all the primary resources become unavailable

Each record includes the name of the (sub)domain, a record type (A, M2..) and other info.

- To set up DNS failover to a static website, use Route 53 with the failover option to a static S3 website bucket or CloudFront distribution
- To route traffic to a website hosted on a S3 bucket: the bucket should be configured to host a static website, the bucket name = domain/subdomain name. You need a registered domain name (you can use Route 53 for that), and Route 53 must be the DNS service for the domain

To check all healthy instances, use multivalue answer routing policy to help distribute DNS responses across multiple resources. For example, use multivalue answer routing when you want to associate your routing records with a Route 53 health check.

To route traffic to an ELB load balancer, use Route 53, create an alias record that points to the LB. It's similar to the CNAME record but o can create the alias record for the root domain + subdomains (CNAME can only be used for subdomains). To enable IPv6 resolution, create a second resource record

- Alias with type "MX" record set: for mail servers
- Alias with type "CNAME" record set: can't be created for zone apex
- Non-Alias with type "A" record set: for IP addresses
- Alias type with "A" record set: for domains
- Alias type with "AAAA" record set: for subdomains

3.5. VPC

Virtual private cloud.

- A VPC spans all the AZs in the region. After creating a VPC, you can add one or more subnets in each AZ
- A VPC allows you to specify an IP address range for the VPC, add subnets, associate security groups, and configure route tables
- A subnet is a range of IP addresses in your VPC. You can launch AWS resources into a specified subnet
- Use a public subnet for resources that must be connected to the internet, and a private subnet for resources that won’t be connected to the internet
- To protect the AWS resources in each subnet, use security groups and network access control lists (ACL)
- ENI (elastic network interface): logical networking component in a VPC that represents a virtual network card. It includes a primary private IPv4 address, 1+ secondary private IPv4 addresses, 1 Elastic IPv4 per private IPv4 address, 1 public IPv4, 1+ IPv6

> Subnets

Each subnet maps to a single AZ, and each subnet is automatically associated with the main route table for the VPC.

- To create an IPv6 subnet, you need to create IPv4 subnet first
- For 2 EC2 instances inside a VPC to communicate (each instance in its own subnet), the Network ACL should allow communication between the 2 subnets and the security groups allow the app host to communicate to the other instance on the right port and protocol

Security groups and NACL

- Security group: firewall for *EC2 instances*
  - They can allow inbound/outbound access from other security groups
  - They're stateful, everything is blocked by default. Supports allow rules only
  - To allow only clients connecting from the IP address XXX should have access to the host, set the security group inbound rule, protocol tcp, range-22, source XXX/32. /32 is to specify one IP address, /0 refers to the entire network
- NACL (network access control list): firewall for associated *subnets*, for the whole subnet
  - Supports allow + deny rules
  - Inbound rules for NACL subnets are evaluated starting the lowest numbered rule: if rule #100 says allow and rule #* says deny, #100 is evaluated first -> allow. if source is allowed on rule #100, it won't further evaluate rule #101 etc

Connect from VPC to the Internet

- Internet Gateway: allows instances with public IPs to access the internet
  - A subnet is deemed to be a public subnet if it has a route table that directs traffic to the internet gateway
  - Egress-only Internet Gateway: used for VPCs with IPv6
  - Entry in route table: 0.0.0.0/0 -> my_internet_gateway
- NAT Gateway: allows subnets access to the Internet, managed by AWS, scales based on demand
  - allows instances with no public IPs to access the internet. Internet traffic can't access the instances
  - Can't be associated with security groups
  - Used only for IPv4
  - Must be deployed in a public subnet
- NAT Instance: managed by the user with no auto scaling

Connection between 2 VPCs

- Peering connection: allows connection just within the VPCs, not with the connections that the other VPC has
  - If B peers to A and C, A and C aren't connected
- CIDR-ranges should not overlap
- To connect instances between 2 different VPCs, set up peering connection + re-configure route table's target and destination of the instances' subnet

Connect from VPC to AWS resources

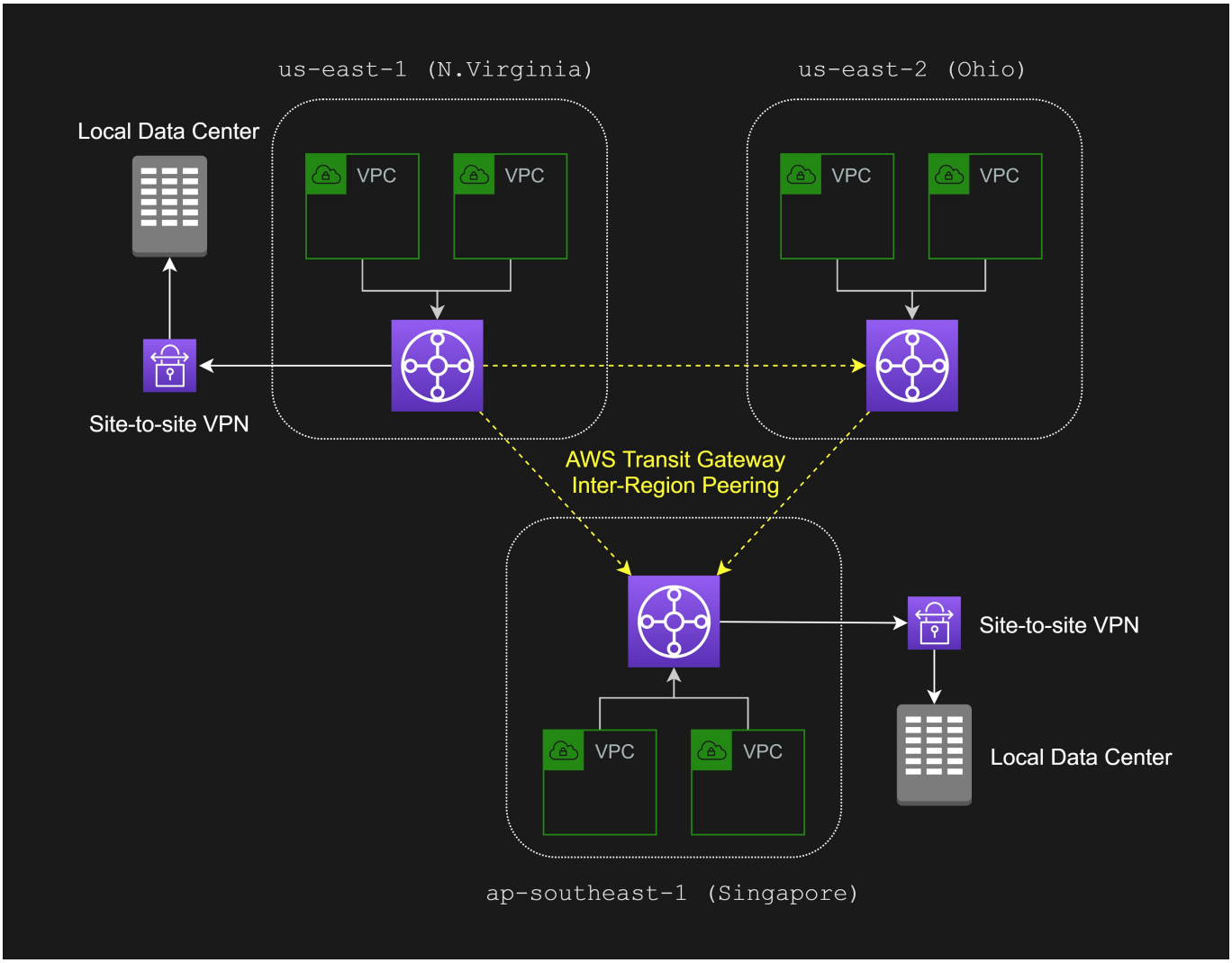
- VPC endpoint: connect the VPC to AWS resources (that aren't part of VPC) by PrivateLink without requiring an Internet gateway. Traffic between your VPC and the other services does not leave the

Amazon network. VPC endpoints for Amazon S3 provide secure connections to S3 buckets that do not require a gateway or NAT instances.

- Interface endpoint: used by most AWS resources
- Gateway endpoint: used by S3 and DynamoDB

Connect VPCs and on-prem networks

- Transit gateway: connect VPCs and on-prem networks, centrally manage point-to-point connectivity. You only create and manage 1 connection from the central gateway to each VPC/on-prem, and transit gateway acts like a hub that controls traffic
  - Without transit gateway, you'd need peering connection between all VPCs + attaching a VPN to each individual VPC



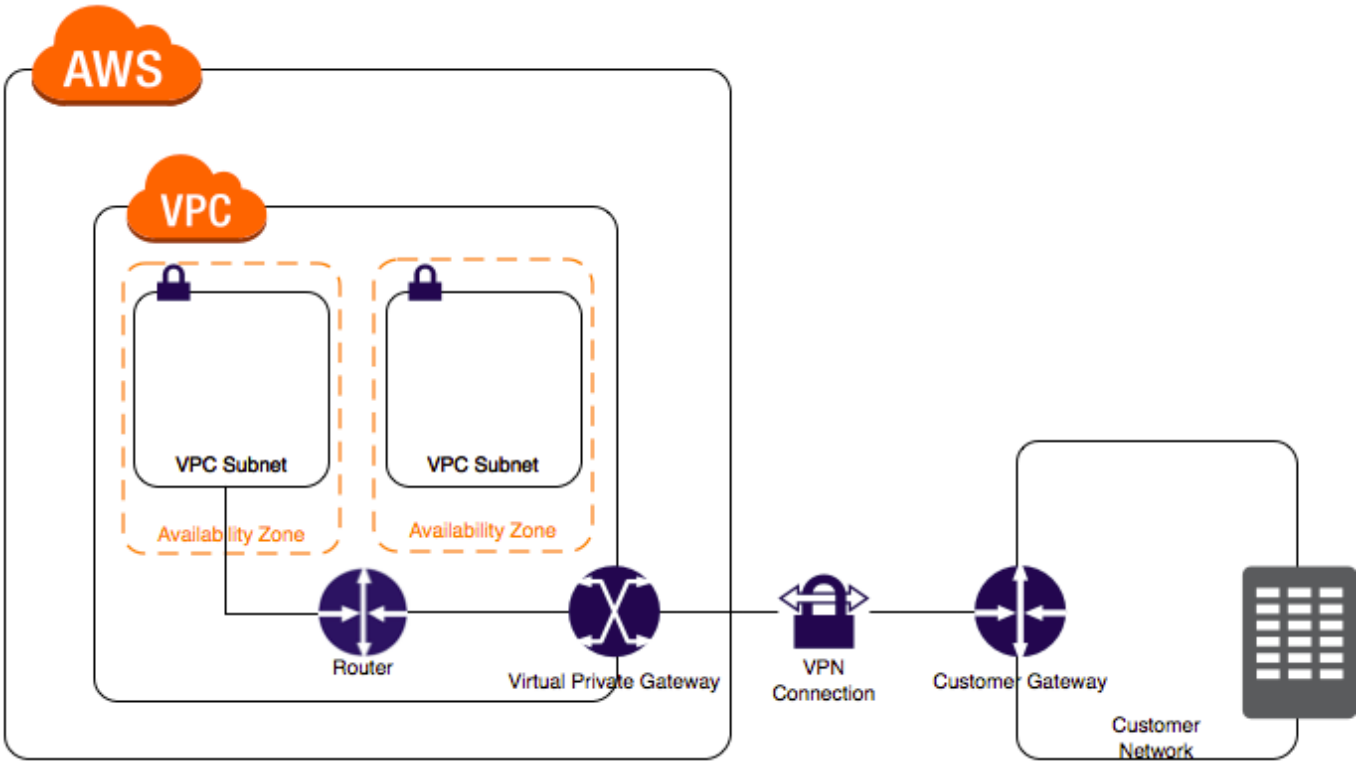
Securely connect on-prem and VPCs with VPN

A VPN allows you to connect your Amazon VPC to other remote networks securely using private sessions with IP security (IPSec) or transport layer security (TLS) tunnels. Maximum 2 tunnels and this can't be changed. But if you associate VPCs to an Equal Cost Multipath Routing (ECMR)-enabled transit gateway, you can attach additional VPN tunnels to it.

- AWS Site-to-Site VPN: to connect on-prem and AWS, cheap option with limited bandwidth and limited traffic.

- AWS Direct Connect: private network to connect on-prem and AWS, through Ethernet fiber-optic. Supports Transit Gateway. More bandwidth than Site-to-Site VPN

To connect from on-prem to VPCs with a VPN, the customer side needs a Customer Gateway, and an internet-routable IP address (static) of the customer gateway's external interface for the on-premises network.



> Windows Bastion

A bastion host is a special purpose computer on a network specifically designed and configured to withstand attacks, it's equivalent to an EC2 instance. It should be in a public subnet with either a public or Elastic IP address with sufficient RDP or SSH access defined in the security group. Users log on to the bastion host via SSH or RDP and then use that session to manage other hosts in the private subnets.

The best way to implement a bastion is to create a small EC2 instance in a public subnet which only has a security group only allowing access on port 22 from a particular IP address for maximum security. This blocks any SSH brute force attacks on the host. Use a private key (.pem) file to connect to the host. Recommended to create a small instance, since this only acts as a jump server to connect to other instances.

> Elastic IP address

Static IPv4 (can only connecto to NLB) address masks the failure of an instance by rapidly remapping the address to another instance. You can also specify the elastic IP in a DNS record of your domain, so that your domain points to your instance.

Can be used to use the trusted IPs as Elastic IP addresses (EIP) to the network load balancer. EIPs can't be associated to application load balancers.

4. Security

- GuardDuty: threat detection service that monitors for malicious activity and unauthorized behavior in your AWS accounts and workloads

- Inspector: automated security assessment service that helps improve the security and compliance of apps deployed on AWS
- Shield: detect and mitigate DDoS attacks
    - Shield standard: network and transport layer protection
    - Shield advanced: additional detection and mitigation. Near real-time visibility into attacks, integration with WAF
- WAF: blocks common attack patterns such as SQL injection or cross-site scripting
    - AWS Firewall manager: simplify WAF administration and maintenance tasks across multiple accounts and resources
    - To mitigate DDoS: create a cloudfront distribution, set an ALB as origin. Create a rate-based ACL rule using WAF and associate it to the cloudfront
    - WAF & ACL: can allow specific IP addresses, and block requests from a specific country using a geo match condition

> Encryption

Client-side encryption means encrypting data before sending it to AWS, useful for cases where master key and unencrypted data can't be sent to AWS.

> AWS Secrets manager

Secrets can be db credentials, passwords, API keys etc. It allows automatic rotation for all the credentials. Secrets Manager enables you to replace hardcoded credentials with an API call to the Secrets Manager.

4.1. IAM

To assign a bunch of permissions to a bunch of IAM users, create an IAM group and assign the policy to the group. IAM roles are for resources

Default policy is everything denied, which can be overruled by an explicit allow, which can overruled by an explicit deny. Each policy contains 1+ statements. Statement contains:

- Effect: allow/deny
- Action: list of actions
- Resource: list of resources for which the actions are granted

Policies can be:

- Identity based: attached to a user/group/role
- Resource based: attached to a resource
    - Needs a principal: for which account/user/role is getting the effect

> Import IAM from on-prem

If a company is using Active Directory in their on-premise system, use AWS Directory Service AD connector for easier integration. If the roles on-prem are already assigned using groups, in AWS use IAM roles or use Microsoft AD federation service.

Use IAM users only when creating new credentials, if the company already has then on-premises, they can be imported some other way.

> Manage access centrally

To manage AWS resources centrally, use AWS organizations and AWS RAM (resource access manager) which enables you to share resources with any account or within organizations. You can share AWS Transit Gateways, Subnets, AWS License Manager configurations, and Amazon Route 53 Resolver rules resources with RAM

MySQL and PostgreSQL dbs instance can be authenticated with IAM DB authentication and then you only need an authentication token to access it.

awsu.me is a CLI tool to work with different roles in AWS.

- *Trusted advisor*: reviews permissions for unnecessary rights or best practice violations and checks that you've enabled AWS security features for services
- *Policy simulator*: build, validate and troubleshoot policies

---

- SSO: single sign-on, central management of access to AWS accounts and resources
- STS: security token service, create temporary credentials for AWS resources
    - For OpenID Connect: Web Identity Federation
    - Can be used for dealing with LDAP situations

5. Management

- AWS Config assesses, audits and evaluates resources. Can automate the evaluation of recorded configs against desired configs. By creating an AWS Config rule, you can enforce your ideal configuration in your AWS account
- OpsWorks: a configuration management service that provides managed instances of Chef and Puppet. Chef and Puppet are automation platforms that allow you to use code to automate the configurations of your servers

5.1. CloudWatch

CloudWatch by default monitors CPU, network and disk read activity on EC2 instances. To get memory utilization, need to have a custom metric. Install the CloudWatch agent in the EC2 instances that gathers all the metrics (memory usage for example). View the custom metrics in the CloudWatch console.

5.2. CloudTrail

Check who made changes/API calls to AWS resources, stores logs in S3, encrypted with SSE by default, you can also choose KMS key

- Management events: control management operations performed on resources in the AWS account
- Data events: resource operations performed on or within a resource, e.g. GetObject, DeleteObject

CloudTrail can track changes, can't enforce rules to comply with your policies.

6. Analytics

> Kinesis

Used Kinesis for streaming, **real-time** applications.

- Kinesis data streams: ordered sequence of data records meant to be written to and read from in real-time. Data records are temporarily stored in shards in the stream, default is 24h.
- Kinesis data firehose: loads streaming data into data stores and analytics tools e.g. S3, Redshift, Elasticsearch, Splunk. Only supports S3, Redshift, Elasticsearch and HTTP endpoint as destination. For DynamoDB, use streams.

To increase throughput, increase the number of shards by using the UpdateShardCount command. Throughput of streams and firehose is similar.

> EMR

Managed cluster platform for big data framework (Apache Hadoop, Spark). Processes and analyzes vast amounts of data. EMR can be used to transform and move large amounts of data into and out of other AWS datstores and dbs.

> Glue

AWS Glue is a serverless ETL service that crawls data, builds a data catalog, performs data preparation, transformation and ingestion. But doesn't allow the usage of big data frameworks.

> Redshift

Cloud data warehouse, it allows SQL and BI tools. You can run complex analytic queries against TB or PT of structured/semi-structured data.

Redshift automatically and continuously backs up your data to S3. It can asynchronously replicate your snapshots to S3 in another region for disaster recovery.

7. Messaging

7.1. SQS

Simple queue service, decouples downstream operations that don't need to be synchronous.

> Queue types

- Standard queue: guarantees that messages are delivered at least once, no guarantee for order
    - With Message groups and their identifiers, messages with the same ID are processed in order. Useful for processing messages of the same customer in order. Messages for one customer are delivered in FIFO, but messages for other customers are in parallel and FIFO is not guaranteed
- FIFO: limit 300 transactions/s, guarantees ordering, guarantees one-time processing of all messages, support for message groups
- DLQ (dead letter queues): if a message is considered unprocessable, it's sent to this queue. This helps unblock messaging systems without losing messages

> Features

- Visibility timeout: message is hidden from other consumers while it's being processed. If successfully processed, deleted. Else, message available again. Default timeout is 30s, max 12h
- Polling: retrieving messages from queue

    - It's not real time. If we receive empty messages when polling, enable long polling: set ReceiveMessageWaitTimeSeconds to higher than 0. In long polling, SQS waits until a message is available before sending a response to a ReceiveMessage request.
- Message priority: can't set a priority to individual messages
    - For users with different priority, create one SQS queue for each priority type. Consume messages from the high priority queue until it's empty, then the lower priority queue.
- Messages max size 256kb of text in json/xml format
- Message retention period from 1 minute to 14 days, default is 4 days. After that, messages are deleted
- Max in-flight msgs 120k, for FIFO 20k

Messages for Lambda triggers can be aggregated together into batches, so one function invocation processes several messages at a time.

> Amazon SWF

Task coordinator in the cloud, ensures a task is never duplicated and is assigned only once. A specific task is given to only one worker. These facilities enable you to coordinate your workflow without worrying about duplicate, lost, or conflicting tasks. Useful for creating decoupled architectures.

7.2. SNS

Simple notification service, publish-subscribe format. Notifications are delivered to clients using a "push" mechanism rather than to periodically check or "poll" for new information and updates.

- It can send notifications to SMS, email, HTTP
- Lambda is a valid subscriber, but EventBridge is not

Fanout scenario: when a message published to an SNS topic is replicated and pushed to multiple endpoints: SQS, HTTP(s), Lambda. This allows for parallel asynchronous processing:

- Create a topic and use two Amazon SQS queues to subscribe to the topic. If Amazon SNS receives an event notification, it will publish the message to both subscribers
- For example, you can develop an application that publishes a message to an SNS topic whenever an order is placed for a product. Then, SQS queues that are subscribed to the SNS topic receive identical notifications for the new order. An EC2 instance attached to one of the SQS queues can handle the processing or fulfillment of the order. And you can attach another Amazon EC2 instance to a data warehouse for analysis of all orders received

7.3. MQ

Service used for migrating messaging services to the cloud quickly and easily.

Managed message broker service for Apache ActiveMQ that makes it easy to set up and operate message brokers in the cloud and hybrid architecture. The user case is when migrating to a managed message broker to automate software administration and maintenance, without having to re-write existing applications.