AWS Solutions Architect - Associate (SAA-03) Study Guide

EC2

- **Placement Groups**
  - *Cluster* - single AZ, low latency
  - *Spread* - maximum 7 instances per AZ
  - *Partition* - spread over racks, up to 7 partitions per AZ, used for large sequential workloads like Hadoop, Cassandra, Kafka
- **Types**
  - *General Purpose*
  - *Compute Optimized* - good for high performance required workloads (HPC, batch processing, media transcoding)
  - *Memory Optimized* - good for large datasets in memory (high performance databases)
  - *Storage Optimized* - good for workloads that require high sequence read/write access to large datasets on local storage (OLTP)
- **Security Groups**
  - Locked to one VPC or region
  - Can be attached to multiple instances
  - By default, all inbound traffic blocked, all outbound traffic allowed
- **Purchasing Options**
  - *On-demand Instance* - short and unpredictable workloads, pay per second
  - *Reserved Instance* - long workloads, one or three years, can be convertible
  - *Savings Plan* - long workloads, commitment to pre-defined usage, one or three years
  - *Spot Instance* - short workloads, cheap, unreliable
    - *Spot Fleet* - set of spot and on-demand instances that tries to meet target capacity and price constraints
    - *Spot Block* - try to reserve spot instances for a set time, no guarantees
  - *Dedicated Host* - entire physical server, look for “server bound software licenses”
  - *Dedicated Instance* - use up an entire instance hardware, no sharing

    - *Capacity Reservation* - reserve on-demand instances in an AZ for any duration
- **Tenancy**
    - Default, host, and dedicated; you can only change tenancy from host to dedicated and vice versa once the instance has started
    - Dedicated tenancy will always take precedence regardless of VPC or launch configuration defaults
- **ENI (Elastic Network Interfaces)**
    - Represents a virtual network card, can attach them on the fly on EC2 instances, bound to a single AZ
- Instances cannot be changed when running, they must be stopped first (ex: enabling hibernation mode)
- Instance metadata can be found at http://169.254.169.254/latest/meta-data
- Instance profiles let EC2 instances assume IAM roles, allowing them to do things like make API calls
- Recovered instances are exactly the same as they were before they became unhealthy
- Use an **EFA (Elastic Fabric Adapter)** for HPC

EBS

- Data at rest, between volumes and instances, and snapshots are ALL encrypted
- EBS Volume can only be mounted to one instance at a time
- EBS Volume is locked to one AZ
- By default is deleted when root instance is terminated
- EBS supports live configuration changes
- **EC2 Instance Store**
    - High performance hardware disk
    - Good I/O performance
    - Loses storage when stopped
    - Look for “dataset replicated across instances”
- **AMI**
    - Customization of an EC2 instance
- **Types**
    - *gp2/gp3* (SSD) - general purpose, 3 IOPS per GiB of storage
    - *io1/io2* (SSD) - low latency, mission-critical, high-throughput workloads, look for “more than 16,000 IOPS”
    - *st1* (HHD) - low cost, high-throughput, frequent access
    - *sc1* (HDD) - infrequent access
    - Only the first two can be used as boot volumes

- **Data Lifecycle Manager** - automatically takes snapshots of EBS volumes on a regular basis
- **RAID** - RAID 0 stripes volumes together for better I/O performance, RAID 1 mirrors volumes for better redundancy

EFS

- Can be multi-AZ
- Use cases: content management, web serving, data sharing
- Scales automatically
- Maximum life cycle rule is 90 days
- **Types**
    - Performance
        - General purpose
        - Max I/O: higher latency, used for things like big data
    - Throughput
        - Bursting
        - Provisioned: set your own throughput
- **EFS Infrequent Access** - cost to retrieve files, but lower cost to store
- Look for "POSIX compliant" and "Linux AMI"

ELB

- Health checks done on a specific port and route
- ELBs can only run within one region (can have multiple AZs)
- **CLB (Classic Load Balancer)**
    - Supports TCP and HTTP/S
- **ALB (Application Load Balancer)**
    - Supports HTTP
    - Good fit for microservices and container based applications
    - Target groups must have private IP

- **NLB (Network Load Balancer)**
  - Traffic routed using a private IP address
  - Supports TCP and UDP protocols,
  - Also used for extreme performance needs
  - One static IP per AZ, can assign elastic IPs
- *Cross Zone Load Balancing*
  - Always on for ALBs, paid for NLB, free for CLB
- *Deregistration delay*
  - Time that instances get to complete requests while they are de-registering or unhealthy
- **ASG Scaling Policies**
  - *Target tracking* - ex: I want CPU to stay around 40%, look for "over-provisioning"
  - *Simple/Step scaling* - ex: When a CloudWatch alarm is triggered, add-remove instance
  - *Scheduled actions* - ex: At 3:00 PM add 3 instances
  - Use *lifecycle hooks* to control what happens to instances as they are launched or terminated
  - ASG always creates new instances first when rebalancing, and terminates unhealthy instances before creating new ones
- Launch configurations (deprecated) for ASGs cannot be modified once created
- Launch templates for ASGs allow provisioning across multiple instance types
- When scaling in, the EC2 instance that was launched from the oldest launch configuration gets terminated first

Relational Databases (RDS, Aurora)

- RDS and Aurora both support IAM DB Authentication
- Automated backups have a maximum of 35 days (use AWS Backup if you need more time)
- **RDS (Relational Database Service)**
  - Uses SSL to encrypt data in transit
  - *Storage Auto Scaling* - increases storage dynamically
  - *Read Replicas* - asynchronous, increases read scalability, can promote a read replica to a primary database, use case: you want to run a reporting application on your production database to run analytics, so you create a read replica to do the work there instead
    - Can make these cross-region for disaster recovery

    - *Multi-AZ* - synchronous, increases availability, the replica is within the same region
    - *RDS Proxy* - must be accessed from VPC, allows apps to pool DB connections established within database, reduces stress on database resources, serverless, look for “too many connections”
    - *RDS Enhanced Monitoring* - use instead of CloudWatch to view things like CPU usage of your RDS instance
    - *RDS Custom* - allows you to access the underlying database and OS
- **Aurora**
    - Automatically spread across 3 AZs
    - High I/O and low latency => set up Aurora Replicas
    - Instantaneous failover
    - If using single DB cluster and it fails, it will try to recreate it in the same AZ
    - *Aurora Replicas* - increase availability and scales read operations, in case of failover, Aurora automatically fails over to the read replica with the highest tier and size, cannot promote Aurora read replicas to standalone
    - *Global Aurora* - cross region read replicas used to disaster recovery, global database decreases latency by a lot
    - *Aurora Serverless* - Used for intermittent or infrequent workloads, no need for capacity planning, in case of failures the DB is recreated in another AZ
    - *Aurora MultiMaster* - used to reduce write latency
    - *Custom Endpoints* - use these to direct users to other tasks that aren’t reading and writing
    - Can clone Aurora database from an existing one, very fast and cost-effective

ElastiCache

- **ElastiCache**
    - Helps make your application stateless
    - Involves heavy application code changes
    - Does not support IAM authentication
    - Used for read-heavy and compute-intensive workloads
    - Look for “in-memory”
- **Redis and Memcached**
    - *Redis* - uses Redis AUTH for security, multi-AZ, read replicas, backup and restore

    - *Memcached* - uses shards, no replication, non persistent, no backup and restore, only thing that is has going for it is that it supports multi-threading

Route 53

- Fully managed DNS that is able to perform health checks
- Record types:
    - *A* - hostname to IPv4
    - *AAAA* - hostname to IPv6
    - *CNAME* - hostname to hostname
    - *NS* - name servers for hosted zone
- CNAME vs. alias
    - CNAME only for non-root domain, while alias works for both non-root and root domains (works for www.example.com but not example.com)
- Alias records
    - Maps hostname to an AWS resource
    - Cannot set TTL (time to live) and cannot set an alias record for an EC2 DNS name
- Routing policies
    - *Simple* - route traffic to a single resource, no health checks
    - *Weighted* - control the percent of requests that go to each resource
    - *Latency-based* - direct to the resource that has the least latency
    - *Geolocation* - based on user location
    - *Geoproximity* - shift traffic to resources based on defined bias values
- Health checks
    - Can use logical operators to combine results of multiple health checks into one
    - Health checks are for public resources, so if you want to perform a health check on a private endpoint, use CloudWatch
    - Failovers can be “active-active” (all resources available in case of failure) or “active-passive” (primary resources with backup resources in case of failure)
- Pre-reqs to route traffic to S3
    - S3 bucket with same name as domain or subdomain
    - Registered domain name
    - Route 53 as the DNS service for the domain

S3 (Simple Storage Service)

- Object storage service
- Minimum storage duration for S3 is 30 days
- Object values can be up to 5TB, must use S3 Multi-Part Upload if uploading more than 5GB
- *CORS* - allows requests to other origins, must enable correct CORS headers in S3 bucket
- *S3 Access Logs* - any requests made to S3 will be logged into another bucket
- *S3 Replication* - must enable versioning in both buckets, asynchronous, must have proper IAM permissions
    - *Cross-region Replication* - compliance, lower latency, replication across accounts
    - *Same-region Replication* - log aggregation, live replication
- *S3 Requester Pays* - have the requester instead of the bucket owner pay for the cost of request, helpful for when you want to share large datasets with other accounts
- *S3 Transfer Acceleration* - uses CloudFront to quickly transfer files (size of GBs) <u>from client</u> to an S3 bucket, do not need to pay if transferring from the Internet or if the transfer was not successfully accelerated
- To transfer files from bucket to bucket, use the *sync* command or batch replication
- **Storage Classes**
    - *Standard, General Purpose* - no storage minimum, frequently accessed data, low latency, high throughput
    - *Standard, IA* - minimum storage duration of 30 days, use case includes disaster recovery and backups
    - *Intelligent Tiering* - no storage minimum, moves objects between tiers based on usage
    - *One Zone, IA* - minimum storage duration of 30 days, high durability in single AZ, but lost if AZ is destroyed, use case includes storing secondary backups of on-site data
    - *Glacier Instant Retrieval* - minimum storage duration of 90 days, instant retrieval
    - *Glacier Flexible Retrieval* - minimum storage duration of 90 days, retrieval can be expedited, standard, or bulk
    - *Glacier Deep Archive* - minimum storage duration of 180 days, retrieval can be standard, or bulk
- **Lifecycle Rules**

    - *Transition actions* - when objects are moved to another storage class
    - *Expiration actions* - when objects get deleted
- *S3 Event Notifications* - can send to many AWS services, like SNS, SQS, Lambda, EventBridge
- *S3 Glacier Vault Lock* - objects cannot be deleted from Glacier storage when enabled
- **S3 Object Lock**
    - Versioning must be enabled
    - *Retention mode: Compliance* - objects can't be overwritten or deleted by <u>any</u> user
    - *Retention mode: Governance* - most users can't write or delete, mode can be changed or deleted
    - *Legal Hold* - protect object indefinitely, independent of retention, can be freely placed and removed
- Accidental deletion => MFA delete and bucket versioning
- Look for "static website" or "static content"
- "Data archiving" => S3 Glacier

CloudFront

- *Content Delivery Network (CDN)*
- Improves read performance, static content is cached at the edge
- Best for uploads/downloads under 1GB
- Origins include S3 buckets or custom origins (HTTP) like ALBs, EC2 instances, S3 websites
- Use *OAI (Origin Access Identity)* for security when an S3 bucket is the source of the CloudFront distribution
- Use *Signed URLs or Signed Cookies* to allow certain users to access private content, cookies for individual files, URLs for multiple files
- *CloudFront GeoRestriction* - black or whitelist users from certain countries, <u>cannot</u> be used with a VPC
- **CloudFront vs. S3 CRR**
    - CloudFront is good for static content that must be available everywhere, while S3 CRR is good for dynamic content that needs to be available at low-latency in a few areas
- **Global Accelerator**
    - Improves availability and performance of applications by providing static IP addresses
    - Look for "reduced latency," "global users"
    - Look for "UDP"
- **CloudFront vs. Global Accelerator**

    - CloudFront improves performance for cacheable content and dynamic content
    - Global Accelerator is good for non-HTTP use cases, HTTP use cases that require a static IP address, and HTTP use cases that require deterministic fast regional failover
    - CloudFront uses the edge to cache content while Global Accelerator uses the edge to find the optimal path to the nearest regional endpoint

Storage and Migration

- **Snow Family**
    - *Snowcone* - small, light, portable, can use AWS DataSync, 8TB
    - *Snowball Edge* - Storage optimized has 80TB, edge-computed optimized has 42TB
    - *Snowmobile* - a literal truck that has 100PB of capacity, used to transfer entire data center to AWS, not suitable for hybrid model
    - Data is general loaded into S3 buckets, and then rolled into S3 Glacier using lifecycle policies if necessary (cannot do it directly)
    - *Edge Computing* - processing data while it's being created on an edge location, uses cases: preprocessing data, machine learning at the edge
    - Snowball cannot import to Glacier directly
- **FSx**
    - *FSx for Lustre* - high performance computing, seamless integration with S3, can be used on-site, look for "parallel hot-storage" and "HPC"
    - *FSx for Windows* - can be mounted on Linux EC2 instances, backed-up daily to S3, can be multi-AZ, look for "Service Message Block (SMB)"
- **Storage Gateway**
    - Bridge between on-premises data and cloud, use cases: disaster recovery, backup and storage, tiered storage
    - Look for "hybrid model"
    - *S3 File Gateway* - most recently used data is cached in the file gateway, does not support direct transfer into S3 Glacier
    - *FSx File Gateway* - local cache for frequently accessed data
    - *Volume Gateway* - block storage backed by S3 and EBS snapshots which can help restore on-premises volumes
        - *Cached Volumes* - low latency access to most recently access data stored locally, everything else in AWS

        - *Stored Volumes* - entire dataset on premises
    - *Tape Gateway* - Virtual Tape Library (VTL) backed by Amazon S3 and Glacier
    - Can backup to S3 Glacier directly
- **Transfer Family**
    - Uses FTP to transfer files in and out of S3 and EFS
    - Managed infrastructure, multi-AZ
    - Look for "FTP, FTPS, SFTP"
- **DataSync**
    - Used to move large amounts of data to and from on-premises to AWS, and AWS to AWS
    - Can send directly to S3 Glacier

Messaging (SQS, SNS, Kinesis)

- **SQS (Simple Queue Service)**
    - Fully managed, used to decouple apps
    - Low latency, unlimited throughput, messages retained for 4 days by default, can be set from 1 minute to 14 days
    - Set *Long Polling* to have consumers wait for messages to arrive in the queue, decreases the number of API calls, increases efficiency and reducing latency
    - "Polls" messages, which allows messages to sit in the queue until they are consumed
    - *FIFO Queue* - guarantee that messages are sent in order, supports 300 messages per second but you can "batch" messages (up to 10 per batch) to do 3000 messages per second
    - *Temporary Queue* - used for request-response messaging patterns
    - *Dead Letter Queue* - used to debug SQS queues by isolating problematic messages
- **SNS (Simple Notification Service)**
    - Send topic notifications to multiple receivers
    - Fan-out model: Use SNS to send a message to multiple SQS queues
    - Can set filters to publish specific messages to specific destinations
    - "Pushes" messages, which assumes that applications are ready to consume messages
    - Look for "subscribe to" <u>NOT</u> "poll from"
- **Kinesis**
    - Used to collect, process, and analyze streaming data in real time
    - **Kinesis Data Streams**
        - Retention between 1 day to 1 year

        - Can replay data
        - Immutable data
        - Uses shards to organize data (more shards means better performance)
        - Real time, used when you only need to analyze parts of the data and don't need to store it all somewhere
        - Look for "concurrent consumption of data"
    - **Kinesis Firehose**
        - Fully managed
        - Near real time, automatically scaled, no replay
        - Used when you need to store all the data somewhere
        - Usually dumped into S3 and Redshift and does not offer data storage
        - Easily integrated with Lambdas
    - **Kinesis Data Analysis**
        - Can process data in real time
- **MQ**
    - Used when migrating non-cloud applications to AWS

Serverless (ECS, EKS, Lambdas, DynamoDB)

- **ECS (Elastic Container Service)**
    - EC2 launch type: must provision and maintain the EC2 instance, ECS agent uses a EC2 instance profile
    - ECS task role allows each task to have a specific role, defined in task definition
    - Integrated with ALBs and NLBs
    - Uses Application Auto Scaling
- **EKS (Elastic Kubernetes Service)**
    - Look for "cloud-agnostic"
    - Can support EC2 or Fargate
- **ECR (Elastic Container Registry)**
    - Store and manage Docker images, access controlled through IAM
- **Fargate**
    - AWS's serverless container platform, all managed by AWS
    - Fargate with EFS is entirely serverless
    - Does not incur costs when application is idle
- **Lambda**
    - Serverless, scaling is automated, run on-demand, short executions (under 15 minutes by default)
    - Concurrency set to 1000 executions by default
    - Look for "scaling in seconds"

    - By default, is launched in an AWS-owned VPC
    - *Lambda@Edge* - Used to customize content delivered by CloudFront
- **DynamoDB**
    - Non-relational database, best suited for key-value and document model store
    - Incredibly fast, millions of requests per seconds, 100s of TBs of storage
    - Made up of tables, each table has a primary key, each item in the table has attributes
    - *DAX (DynamoDB Accelerator)* - Uses caching to solve read congestion, microseconds latency
    - *DynamoDB Streams* - shows modifications made to table, 24 hour retention
    - Use *DynamoDB Global Tables* to make it accessible in multiple regions with low latency (must have Streams active)
    - To improve performance, use high-cardinality keys
    - Not optimal for very large datasets
    - Use point in time recovery for backing up lost data
    - Look for “key-value pairs”
    - Look for “resilient to schema changes”
- **API Gateway**
    - Serverless, can invoke Lambdas, expose HTTP endpoints, expose an AWS API
    - Can use throttling limits to protect the back end from high traffic
    - Can use Gateway Caching to improve performance
    - RESTful APIs are stateless, while WebSocket APIs are stateful

Data Analytics (Athena, Redshift, EMR, FSx, Glue)

- **Athena**
    - Serverless query service to analyze data in S3
    - Uses standard SQL
    - Does not analyze data in real time!
- **Redshift**
    - Look for “business analytics”
    - Look for “OLAP” and NOT “OLTP” (that’s for RDS/Aurora)
    - **Redshift Spectrum** - efficient querying and retrieval of data from S3 without loading it
- **OpenSearch**

    - Successor to ElasticSearch
    - Can search any field, even partial matches
- **EMR (Elastic Map Reduce)**
    - Distributes vast amounts of data and processing across multiple EC2 instances called Hadoop clusters
    - Look for “machine learning and data processing”
    - Look for “Spark and Hadoop”
- **QuickSight**
    - Serverless machine learning business analysis service that creates interactive dashboards
- **Glue**
    - Extract, transform, and load service to prepare data for analytics
- **Lake Formation**
    - Sets up data lakes, which are central places to have all your data for analytics
- **Kinesis Data Analytics**
    - Real-time analytics on Kinesis Data Streams and Firehose using SQL

Machine Learning

- *Rekognition* - used to find people, texts, objects in images and videos
- *Transcribe* - speech to text
- *Polly* - text to speech
- *Translate* - language translation
- *Lex and Connect* - same as Alexa, virtual contact center
- *Comprehend and Comprehend Medical* - NLP and sentiment analysis
- *SageMaker* - fully managed service to build ML models
- *Forecast* - uses ML to make accurate forecasts
- *Kendra* - document search service
- *Personalize* - build apps with real time personalized recommendations
- *Textract* - extracts text from any scanned documents

Monitoring (CloudWatch, CloudTrail, Config)

CloudWatch

- Used to monitor applications and AWS cloud resources
- CloudWatch Metrics are specific values to monitor
- CloudWatch Metric Streams send metrics to a destination of your choice
- CloudWatch Logs can be sent from and be received by many sources
- CloudWatch Alarms send emails via SNS whenever EC2 instances breach a certain threshold

    - CloudWatch Container Insights collect logs from ECS, EKS, Fargate
    - Default metrics are CPU utilization, disk read activity, and network packet info
    - Use CloudWatch Metric Streams to stream continuously to Kinesis Firehose
    - Must be installed onto EC2 instances if you want to monitor them
    - **EventBridge**
        - Create rules to schedule jobs, react to events, or trigger lambdas
        - Needs permissions on the target
        - Can send to SNS, SQS, and Lambda (but not FIFO SQS)
        - Look for “Software as a Service (SaaS)”

CloudTrail

- Used to monitor event history of AWS account activity, like API calls
- Provides governance, compliance, and audit for the AWS account
- Can be multi-AZ or single region
- *Management Events* - actions performed on AWS resources, logged by default
- *Data Events* - actions performed on data, must be enabled
- CloudTrail Insights detect unusual activity in your account
- Look for “API calls”

Config

- Used to record compliance of AWS resources
- Does not prevent actions from happening

IAM and Organizations

- **Organizations**
    - Global service that allows you to manage multiple AWS accounts
    - Main account is the management account, others are member accounts
    - Shared reserved instances and savings plans across accounts
    - Uses SCPs (Service Control Policies)
        - IAM policies applied to OU (Organizational Units) or Accounts to restrict Users and Roles
        - Must have explicit allows
        - SCPs will overrule IAM rules
        - SCPs do not affect service-linked roles
- **IAM**
    - *IAM Users* - are people within the organization, can be assigned an *IAM Role* or *IAM Policy*
    - *IAM Groups* - groupings of users, can be assigned an *IAM Policy*

    - *IAM Permission Boundaries* - used to restrict single users from escalating their own permissions by specifying exactly what maximum permissions they get
    - *Trust Policy* - the only resource based policy that IAM supports, the "resource" is the role itself, controls who can take on that role
    - IAM policies can never restrict the root user, and cannot be attached to OUs
    - When you assume a role, you give up your original permissions and take the permissions assigned to the role
    - When using a resource based policy, the principal doesn't have to give up permissions
    - An explicit deny overrides everything
    - IAM DB authentication allows you to access databases with an authorization token
    - To access AWS, you have to create access keys using the AWS Console
    - Never store API credentials anywhere, create IAM roles instead
    - "Windows" => Active directory
- **Cognito**
    - Gives users an identity to interact with an application
    - *User pools* - sign in functionality for app users, serverless database of users, integrates with ALBs and API Gateway
    - *Identity pools* - provides AWS credentials for direct access to AWS resources
    - Look for "many users," "mobile users," "authentication with SAML"

KMS/Security

- **KMS (Key Management Service)**
    - Look for "encryption for AWS"
    - Integrated with IAM
    - You cannot share the same key across regions
    - *Key Policies* - control access to KMS keys, similar to S3 bucket policies
    - *Multi-Region Keys*
        - Same key ID, key material, automatic rotation
        - Encrypt in one region and decrypt in others
        - They are NOT global
        - Each key is managed independently
- **S3 Encryption**

    - *SSE-S3* - keys handled and managed by AWS, does not provide an audit trail for the customer
    - *SSE-KMS* - AWS KMS manages encryption keys, incurs a fee
    - *SSE-C* - client makes own encryption keys, HTTPS must be used
    - *Client Side Encryption* - fully managed by client
    - S3 does not encrypt metadata!
    - *Bucket Policies*
        - JSON based
        - Grants public access to bucket
        - Force objects to be encrypted
        - Grant access to another account
- **Parameter Store**
    - Storage for configuration and secrets
    - Encrypted using KMS
- **Secrets Manager**
    - Integrated with RDS, mostly used for this
    - Encrypted using KMS
    - Look for “rotating credentials”
- **ACM (AWS Certificate Manager)**
    - Provision, manage, and deploy TLS certificates
    - Can’t use with EC2, can deploy on ELBs, CloudFront
- **WAF (Web Application Firewall)**
    - Protects application on the HTTP layer (layer 7)
    - Can be deployed on ALBs, API Gateways, Cloudfront
    - Can use to block specific IPs, or specific geographic locations
    - “Rate-based” rules limit the maximum number of requests from a single IP
    - Look for “SQL injection” or “DDoS”
- **Shield**
    - Protect from DDoS attacks, standard and advanced tiers
    - Look for “DDoS”
- **Firewall Manager**
    - Manages rules in all accounts of an AWS Organization
    - Works with WAF, Shield, and VPC Security Groups
- **GuardDuty**
    - Uses machine learning to analyze event logs, integrated with CloudWatch Event rules to notify users in case of findings
    - Data sources are VPC flow logs, DNS logs, and CloudTrail event logs
- **Inspector**
    - Automated security assessments, sends reports to EventBridge or Security Hub

    - Only used for EC2 instances and containers
- **Macie**
    - Uses machine learning and pattern matching to protect sensitive data like PII

VPC (Virtual Private Cloud)

- *CIDR (Classless Inter-Domain Routing)* - allocates IP addresses
    - Base IP + subnet mask
    - Max 5 VPCs per region (NOT AZ), max 5 CIDRs per VPC
- **Default VPC**
    - All EC2 instances inside this have public IPv4 addresses, and both public and private IPv4 DNS names
    - You can only have one default VPC per region (but you can have many custom VPCs)
- **Subnets**
    - AWS reserved first 4 and last IP addresses in each subnet
- **Internet Gateway**
    - Allows resources in a VPC to connect to the internet, one to one attachment with VPC, must be used in conjunction with a route table to allow Internet access
    - These are always set up in public subnets
- **Bastion Hosts**
    - In the public subnet, then connected with other private subnets
    - Use with Network Load Balancers
    - Security group of bastion must allow inbound from the Internet on port 22
    - Security group of EC2 instances must allow private IP of bastion host
    - Used when connecting using SSH or RDP instead of the Internet
- **NAT Instances vs. NAT Gateways**
    - Used to allow instances in private subnets to initiate outbound traffic
    - These are always set up in public subnets
    - *Instances* - support port forwarding, can be associated with a security group, can be used as a bastion server, launched in public subnet, must be attached to an elastic IP
    - *Gateways* - can't do the above things, managed by Amazon, attached to a specific AZ, requires an Internet Gateway, no security groups
- **NACLs (Network Access Control Lists)**
    - Firewalls that control traffic to and from subnets
    - One NACL per subnet, new subnets assigned default NACL

        - Default accepts everything in and outbound with the subnets it's associated with
    - Used to block specific IP addresses at the subnet level
    - NACL rules are evaluated so that lower number rules take precedence over higher number rules
    - *Ephemeral ports* - clients connect to a defined port, expect a response on an ephemeral port
- **VPC Peering**
    - Privately connect two VPCs using AWS
    - Cannot have overlapping CIDRs!
    - Not transitive, must update route tables
    - Not used for sharing centrally-managed VPCs
    - Does not allow edge to edge routing (like VPN connections, Internet connections)
    - Can create VPC Peering connection between different accounts and regions
- **VPC Endpoints**
    - Allows you to connect to AWS services using a private network instead of the public Internet
    - Also known as **AWS PrivateLink**
    - **Interface Endpoints**
        - Provisions an ENI as an entry point
        - Must attach a Security Group
        - Supports most AWS services
    - **Gateway Endpoints**
        - Only supported by S3 and DynamoDB!
        - Free :)
        - Most likely going to be preferred on the exam
- **VPC Flow Logs**
    - Collect info about IP traffic going into interfaces
- **Site-to-Site VPN**
    - VPNs connect your on premises servers with your AWS servers over the Internet
    - *VGW (Virtual Private Gateway)* - on AWS side of VPN connection
    - *CGW (Customer Gateway)* - on customer side of VPN connection
    - *VPN CloudHub* - hub-and-spoke model for multiple VPN connections
- **DX (Direct Connect)**
    - Provides a dedicated private (as opposed to VPN which uses the Internet) connection from a remote network to a VPC
    - Data is not encrypted in transit by itself, pair with a VPN to do this
    - Must set up a VGW or a Transit Gateway on your VPC

    - Must use a *VIF (Virtual Interface)* to use Direct Connect
        - *Public*: to access AWS public services
        - *Private*: to access a VPC
        - *Transit*: to access a Transit Gateway
    - Takes months to set up
    - Globally available
    - Not redundant, so you have to set up another connection if you want to increase availability
    - Uses cases: increase bandwidth, more consistent network experience, hybrid environments
    - **Direct Connect Gateway** - used to set up a DX to one or more VPCs in different regions
    - Can set up a S2S VPN as a backup
- **Transit Gateway**
    - Hub-and-spoke connection between thousands of VPCs and on premises
    - Regional resource, can work cross-region
    - Use **Resource Access Manager** to share across accounts
    - Only AWS service that supports IP Multicast
    - Use ECMP (Equal-cost multi-path) routing to forward a packet over best paths
- **Traffic Mirroring**
    - Allows you to capture and inspect network traffic in your VPC
    - Used for troubleshooting, threat monitoring
- **Egress-only Internet Gateway**
    - Like a NAT gateway, but for IPv6 only!
- **Network Firewall**
    - Protect entire VPC from Layer 3 to 7

Other

- High costs, multiple accounts => Try consolidated billing
- **Elastic Beanstalk**
    - How to start up EC2 instances quickly: use a golden AMI to have static components setup already, use EC2 user data to customize dynamic installation
- **Storage**
    - *Neptune* - used for highly connected datasets such as social networks
    - *DocumentDB* - uses MongoDB to store JSON data
    - *Keyspaces* - serverless, used with Apache Cassandra
    - *QLDB (Quantum Ledger Database)* - records financial transactions
    - *Timestream* - time series database

- **Disaster Recovery**
  - *Backup and Restore* - RPO of hours, RTO of hours
  - *Pilot Light* - RPO of minutes, RTO between minutes and hours
  - *Warm Standby* - RTO of minutes
  - *Multi-site* - RTO nearly instantaneous
  - **DMS (Database Migration Service)**
    - Seamlessly migrates data from sources to databases in AWS Cloud, requires an EC2 instance to perform replication tasks
    - Uses change data capture (CDC) to do continuous data replication
    - Look for “ongoing updates” or “continuous replication”
  - **Backup** - fully managed service that automates backups across AWS services, you create backup policies called Backup Plans (backup frequency, backup window, retention periods)
  - **Application Discovery Service** - Gathers information about on-premises data centers to help plan migration projects
  - **Application Migration Service** - converts physical, virtual, and cloud-based servers to run natively on AWS
- **CloudFormation**
  - Declarative way of outlining AWS Infrastructure, for any resources
  - **Stacks and StackSets** - Stacks are AWS resources that are created when CloudFormation instantiates a template. StackSets allow you to use Stacks across different accounts and regions
- **SES (Simple Email Service)**
  - Fully managed service to send emails securely, globally and at scale
- **Ports**
  - FTP - 21
  - SFTP/SSH - 22
  - HTTP - 80
  - HTTPS - 443
  - Anything else is probably a RDS Database port
- **AWS Proton**
  - Like CloudFormation but for containers
- **AWS Control Tower**
  - Simplifies the creation of new accounts with preconfigured constraints
- **Exam Strategies**
  - “Least effort” => don’t choose anything that requires code (Lambda, Elasticache)
  - On demand => unpredictable workloads