CloudFront (CF)	Content Delivery Network (CDN). Improves read performance by caching content at edge locations. DDoS protection, integration w Shield, AWS Web App Firewall Can expose external HTTPS and can talk to internal HTTPS backend Client make request to edge location, CloudFront forward request to origin including query string & request headers, return response. Next time client make same request, data in cache already	
CloudFront - Origins	S3 bucket: distribute files and cache them at edge, enhanced security w CloudFront OAI (Origin Access Identity, i.e. only allow communication from CloudFront to S3), CloudFront can be used as ingress, i.e. upload file to S3 from CloudFront (S3 transfer acceleration) use transfer acceleration if data > 1GB Custom Origin (HTTP): ALB, EC2, S3 static website, any HTTP backend For ALB or EC2, security group must be public to allow access from public IP of edge locations	
CloudFront - Geo Restriction	Whitelist: only allow users from approved list of countries to access content Blacklist: deny users access if they are from a blacklist of countries "Country: is determined using a 3rd party Geo-IP database. Use cases: Copyright laws	
CloudFront vs S3 CRR	CloudFront: - global edge network, - files are cached for a TTL, - great for static content that must be available everywhere	S3 CRR: - must be setup for every region u want replication in, - files are updated in near real time, - read only, - great for dynamic content that needs to be available at low-latency in few regions
CloudFront - Signed URL / Cookie	Attach a policy w: URL expiration, IP ranges to access data from, Trusted signers (AWS accts that can create signed URLs). Signed URLs = access to individual files (1 signed URL per file) Signed Cookies = access to multiple files (1 signed cooker for many files)	
	CloudFront Signed URL: - allow access to path, no matter origin, - acct wide key-pair, only root acct can manage it, - can filter by IP, path, date, expiration, - can leverage cache features	S3 Signed URL: - issue a request as the person who presigned the URL, - uses the IAM key of the signing IAM principal, - limited lifetime
Pricing	Cost of data out per edge location varies. The more data is transferred, the lower the cost becomes Can reduce num of edge locations to reduce cost. Price Class All: all regions – best performance. Price Class 200: exclude most expensive regions. Price Class 100: only least expensive regions	
Extra	Multiple Origin: Route to diff origins based on content type. Based on path pattern: /images/*, /api/*, /* Origin Groups: increase HA and do failover. Origin grp contains 1 primary & 1 secondary origin	
	Field Level Encryption: protect user sensitive info through application stack. Adds additional layer of security w HTTPS. Uses asymmetric encryption Sensitive info encrypted at edge close to user. Server behind origin will decrypt w private key Specify set of fields to be encrypted in POST requests (≤ 10 fields) & specify public key for encryption	
Global Accelerator (GA)	Unicast IP: 1 server holds 1 IP address Anycast IP: all servers hold the same IP and client is routed to nearest one	
	GA leverage AWS internal network to route to ur app. 2 Anycast IP are created for ur app (e.g. ALB) Anycast IP send traffic directly to edge locations, which then send traffic to ur app. Works w Elastic IP, EC2 instances, ALB, NLB (public or private). Has Health Checks Consistent performance: - intelligent routing to lowest latency and fast regional failover (< 1min) and DR, - no issue w client cache (since IP doesn't change) Blue/Green deployment Security: - only 2 external IP need to be whitelisted, - auto DDoS protection due to AWS Shield	
CF vs GA	Both use AWS global network and edge locations. Both integrate w Shield for DDoS protection	
	CF: - improve performance for both cacheable content (images & videos) & dynamic content (API acceleration & dynamic site delivery), - content is served at the edge	GA: - improve performance for wide range of app over TCP/UDP, - proxying package at edge to app running is 1 or more AWS region, - good for non HTTP use cases (gaming–UDP, IoT–MQTT, Voice over IP), - good for HTTP use cases that require static IP/fast regional failover

Snow Family	Highly secure, portable devices to collect and process data at the edge, & migrate data in and out of AWS Data migration: Snowcone, Snowball Edge, Snowmobile. Edge computing: Snowcone, Snowball Edge Could have problems w limited bandwidth, connection stability, high network cost, then use Snow Family Rule of thumb: If it takes more than a week to transfer over the network, use Snow Family
Snowball Edge	Physical data transfer soln: move TBs or PBs of data in and out of AWS. Pay per data transfer job. Provide block storage and S3-compatible object storage. Have compute capabilities Snowball Edge Storage Optimized: 80 TB of HDD capacity for block vol and S3 compatible obj storage

	Snowball Edge Compute Optimized: 42 TB of HDD capacity for block vol and S3 compatible obj storage Use cases: large data cloud migration, decommision data center, DR Cannot import directly to Glacier. Must import to S3 1st, then create lifecycle policy for transition	
Snowcone	Small, portable computing, anywhere, rugged & secure, withstand harsh environments Light (2.1kg), used for edge computing, storage & data transfer. 8 TB of usable storage Use snowcone where snowball does not fit (space-constrained environment) Must provide own battery & cables. Can sent to AWS offline or connect to internet and use AWS DataSync to send data	
Snowmobile	Transfer exabytes of data (1 EB = 1000 PB = 1000000TB). Better than Snowball if transfer > 10 PB Ea Snowmobile has 100 PB of capacity (use multiple in parallel) High security: temperature controlled, GPS, 24/7 surveillance	
Usage Procedure	1. Request Snowball devices from AWS console for delivery 2. Install snowball client / AWS OpsHub on ur server 3. Connect snowball to servers and copy files over	4. Ship back device when done 5. Data will be loaded into S3 bucket 6. Snowball is completely wiped
Edge Computing	Process data while it's being created at an edge location (limited/no internet, limited/no easy access to computing power). Then use Snowball Edge/Snowcone device to do edge computing Use cases: preprocess data, ML at edge, transcoding media streams Eventually, can ship back data to AWS to transfer to AWS	
	All devices can run EC2 instances & AWS Lambda Functions (using AWS IoT Greengrass) Long term deployment options (1 & 3 years discounted pricing) Snowcone: - 2 CPUs, 4GB of memory, wired or wireless access - USB-C powered using a cord or optional battery	Snowball Edge - Compute Optimized: - 52 vCPUs, 208 GiB RAM, 42 TB usable storage - optional GPU Snowball Edge - Storage Optimized: - up to 40 vCPUs, 80 GiB of RAM - Object storage clustering available
AWS OpsHub	Used to need CLI to mange Snow Family Device. Now can use OpsHub (download on comp) to manage Can unlock & configure single or clustered devices, transfer files, launch and manage instances on Snow Family Devices, monitor device metrics, launch compatible AWS services on devices (EC2, AWS DataSync, Network File System (NFS))	
FSx	Launch 3rd party high performance file sys on AWS (eg Lustre, Windows File Server, NetApp ONTAP). Is a fully managed service. Compared to EFS which is a shared POSIX sys for Linux users.	
FSx for Windows (File Server)	Fully managed Windows file sys shared drive. Supports SMB protocol & Windows NTFS Has Microsoft Active Directory integration, ACLs, user quotas. Can be mounted on EC2 Linux instance Scale up to 10s of GB/s, millions of IOPS, 100s PB of data Support DFSR Storage – SDD: for latency sensitive workloads (DB, media processing, data analytics,...) Storage – HDD: for broad specturm of workloads (home directory, CMS,...) Can be used from on-premises infrastructure (VPN or Direct Connect). Can have Multi-AZ (HA). Data backed up daily to S3	
FSx for Lustre	Lustre is a parallel distributed file sys, for large-scale computing. Lustre derived from Linux & cluster For ML, High Performance Computing (HPC) (video processing, financial modelling, electronic design automation). Scales up to 100s GB/s, million of IOPS, sub-ms latencies SSD – low latency, IOPS intensive workloads, small & random file ops HDD – throughput intensive workloads, large & sequential file ops Seamless integration w S3: can "read S3" as a file sys, can write output of computations back to S3 Can be used from on-premises infrastructure (VPN or Direct Connect).	
File sys Deployment for Lustre	Scratch file sys: - temp storage, - data is not replicated (data lost if file sys fails), - high bursts (6x faster, 200MBps per TiB) - Used for short-term processing/to save cost	Persistent file sys: - long term storage, - data is replicated within same AZ, - replace failed files within mins, - Used for long-term processing/sensitive data
Storage Gateway	Bridge btw on-premises data and cloud data in S3. Use cases: DR, backup & restore, tiered storage Types of storage gateway: file gateway, volume gateway, tape gateway. Gateway into EBS, S3, Glacial	
File Gateway	Configured S3 buckets are accessible through NFS and SMB protocol Support S3 standard, S3 IA, S3 One-Zone IA, Glacial. Bucket access using IAM roles for ea file gateway Most recently used data is cached in file gateway. Can be mounted on many servers on premise Integreated with Active Directory (AD) for user authentication	

Volume Gateway	Block storage using iSCSI protocol backed by S3 using EBS snapshots which can help restore on-premise vols	Cache vol: low latency access to most recent data. Stored vol: entire dataset is on premise, scheduled backups to S3
Tape Gateway	Some companies have backup process using physical tapes W tape gateway, companies use the same process but in cloud Virtual Tape Library (VTL) backed by S3 and Glacial Back up data using existing tape-based processes (and iSCSI protocol) Works w leading backup software vendors	
FSx File Gateway	Native access to Amazon FSx for Windows File Server. Has local cacher for frequently accessed data Has Windows native compatibility (SMB, NTFS, Active Directory,...) Useful for group file shares and home directories	
Hardware appliance	Previous 4 gateways require virtual servers to be run on-premise. If don't have, can use Storage Gateway Hardware Applicance. Hardware works w previous 4 gateways Has the required CPUs, memory, network and SSD cache resources. Helpful for daily NFS backups in small data centers	
Transfer Family	Fully managed file sys for file transfers in and out of S3 of Amazon EFS using the FTP protocol Supported protocols: - AWS Transfer for FTP (file transfer protocol), AWS Transfer for FTPS (FTP over SSL), - AWS TRanfer fro SFTP (Secure FTP) Fully managed infrastructure, scalable, reliable, HA (multi-AZ) Pay per provisioned endpoint per hours + data transfer in GB Can store and managed user credentials within service. Can integrate w existing authentication sys (Microsoft Active Directory, LDAP, Okta, Amazon Cognito, custom source) Usage: sharing files, public datasets, CRM, ERP ... using FTP	

SQS	Fully managed service, used to decouple apps. Unlimited throughputs, unlimited num of messages in queue Defauly retention of message = 4 days (max 14 days). Low latency. Limit of 256 kb per message Can have duplicate messages (at least once delivery). Can have out of order messages (best effort ordering) Sent to SQS w API SendMessage. Message persisted in queue until consumer deletes it (DeleteMessage API) Consumers (EC2, lambda ...) poll SQS for messages (receive up to 10 at a time). Delete message using DeleteMessage API CloudWatch metric – Queue length (ApproximateNumberOfMessages) -> CloudWatch alarm -> ASG	
Message Visibility Timeout	After a message is polled by a consumer, it becomes invisible to other consumers. Default 30s During processing by consumer, can call ChangeMessageVisibility API to get more time Shorter visibility -> duplicate processing. Higher -> if consumer crash, more time before re-processing	
Dead Letter Queue (DLQ)	Set threshold of how many times a message can go back to queue when consumer fail to process After MaximumReceives threshold exceeded, the message goes into the dead letter queue (DLQ) Need to process message in DLQ before they expire Redrive to Source: redrive message from DLQ to source queue in batches after figuring out whats wrong	
Delay Queue	Delay a message (consumers don't see it immediately) up to 15 mins. Default 0s (available immediately) Can set delay at queue level. Use DelaySeconds parameter to override default delay at message level	
Long Polling	Consumer wait for message to arrive if queue is empty currently. Wait time 1 - 20s Decrease latency & decr num of API calls to SQS Can be enabled at queue level or at consumer level (use WaitTimeSeconds API)	
Request-Response System	SQS Temporary Queue Client to implement request-response sys – leverage virtual queues instead of creating/deleting SQS queues	
FIFO queue	Ordering of message in queue. Limited throughput: 300 msg/s w/o batching; 3000 msg/s w batching Exactly-once send capability (remove duplicates). Can use Group ID (similar to partition key) to have multiple consumers	
SQS w ASG	Use CloudWatch sutom metric – queue length / num of instances -> CloudWatch alarm -> ASG	
SNS	Send 1 message to multiple receivers. Publish/Subscribe (Pub/Sub) sys. Event producer only send 1 message to SNS topic. All subscribers to SNS topic will get message (new feature to filter messages). Can send to email, SMS & Mobile notifications, HTTP(S) endpoints, SQS, Lambda, Kinesis Data Firehose	
SNS Publish	Topic Publish (Using SDK): - create topic, - create subscription, - publish to topic	Direct Publish (for mobile apps SDK): - create platform app, - create platform endpoint, - publish to platform endpoint

SNS Transport protocols:
HTTP/HTTPS, Email/Email-JSON, SQS, SMS

SQS & SNS security	In-flight encryption using HTTPS API. At-rest encryption using KMS or Client-side encryption Access Controls: IAM policies to regulate access to SQS API. SQS / SNS Access Policies (similar to S3 bucket policies): useful for cross-acct access to SQS queues / SNS topics, or allowing other services (SNS, S3...) to write to SQS / SNS topic
SNS + SQS: Fan out Pattern	Multiple SQS queues are subscribers to SNS topic. Fully decoupled, no data loss Ability to add more SQS subscribers over time (instead of sending same message directly to multiple SQS) Make sure SQS access policies allow for SNS to write Can use for S3 event notification (since only can have 1 S3 event rule)
SNS FIFO	Only SQS FIFO can subscribe. Limited throughput: same as SQS FIFO Can have ordering by message group ID (message in same group are ordered) Deduplication using a deduplication ID / Content Based Deduplication
SNS - Message Filtering	JSON policy used to filter messages sent to SNS topic subscribers. No filter -> all messages are received

Kinesis	Ingest real-time streaming data (app logs, metrics, website clickstreams, IoT telemetry data...) Kinesis Data Streams: capture, process and store data streams Kinesis Data Firehose: load data streams into AWS data stores Kinesis Data Analytics: analyze data streams w SQL or Apache Flink Kinesis Video Streams: capture, process & store video streams	
Kinesis Data Streams (KDS)	Ea stream made of multiple shards. Producer send record (contains partition key & data blob (up to 1 MB)) to stream. Consumer get record (now have partition key, seq num & data blob) from stream	
	Retention btw 1 - 365 days. Can reprocess (replay) data. Once data inserted, cannot be deleted Data that share the same partition goes to the same shard (ordering). Real-time (≈ 200 ms latency) Producers: AWS SDK, Kinesis Producer Library (KPL), Kinesis Agent Consumers: Write your own (Kinesis Consumer Library (KCL), AWS SDK), AWS: Lambda, KDF, KDA	
	Provisioned mode: - choose num of shards provisioned, scale manually or use API, (shard splitting/merging) - ea shard get 1 MB/s in (or 1000 records/s) - ea shard get 2 MB/s out (classic or enhanced fan-out) - pay per shard provisioned per hr	On-demand mode: - dafault capacity provisioned (4 MB/s in or 4000 records/s), - Scale automatically based on observed throughput peak during the last 30 days - pay per stream per hour & data in/out per GB
Kinesis Data Firehose (KDF)	Fully managed service, auto scaling, serverless. Pay for data gg through Firehose. Producers (SDK, KPL, Kinesis Agent, KDS, CloudWatch (Logs & Events), AWS IoT) Producer send record (up to 1 MB) to KDF. KDF does batch writes to destination Can send to AWS managed (Redshift {COPY through S3}, S3, ElastiSearch), 3rd party or custom (any HTTP endpoint) Near real time: - min 60s latency for non-full batches or min 32 MB of data at a time Supports many data formats, conversions, transformations (lambda) & compressions Can send failed or all data a backup S3 bucket	
Kinesis Data Analytics (KDA)	Producers & Consumers (KDS, KDF). Perform real time analytics on streams using SQL Can send result of query to KDS -> lambda/ec2 -> anywhere OR KDF -> S3, Redshift,... Fully managed, serverless, auto-scaling. Pay for actual consumption rates Use cases: Time-series analysis, real-time dashboards, real-time metrics	
MQ	Traditional app running on-premise using open protocols like MQTT, AMQP, STOMP, Openwire, WSS, ... When migrating these app to cloud, no need to re-engineer to use SQS, SNS. Can just use Amazon MQ Amazon MQ = managed Apache ActiveMQ. Doesn't scale as much as SQS, SNS. Runs on a dedicated machine, can run in HA w failover. Has both queue and topic features For HA: have MQ Broker in diff AZs, use EFS as storage	

Docker	Use cases: microservice architecture, lift-and-ship apps from on premise to cloud Amazon Elastic Container Service (ECS): AWS container platform. Amazon Elastic Kubernetes Service (EKS): AWS managed Kubernetes (open-source). AWS Fargate: AWS serverless platform (works w ECS & EKS). Amazon Elastic Container Registry (ECR): store container images	
ECS	Launch Docker container on AWS = launch ECS task on ECS Clusters	
	If using EC2 launch type: must provision and maintain infrastructure (EC2 instances)	If using Fargate launch type: no need to provision infrastructure (serverless)

	Ea EC2 instance must run the ECS Agent (similar to Docker Daemon) to register in the ECS cluster. AWS takes care of starting/stopping the containers in the EC2 instances	Just create task definitions. AWS run ECS task based on CPU/RAM needed To scale, just incr num of num of tasks
IAM roles for ECS	EC2 instance profile (EC2 launch types only): Used by the ECS agent to make API calls to ECS service, send container logs to CloudWatch Logs, pull Docker image from ECR, reference sensitive data in Secrets Manager/SSM Parameter Store	ECS Task Role: - allows ea task to have a specific role, - use diff roles for the diff ECS task services u run - defined in the task definition
LB integration	ALB supported and work for most cases. CLB supported but not recommended (no advanced features – Fargate) NLB recommended only for high throughput/performance use cases, or to pair w AWS PrivateLink	
Data volumes	Mount EFS file sys onto ECS tasks. Works for both EC2 and Fargate launch types Tasks running in same AZ will share the same data in the EFS file sys. Fargate + EFS = serverless FSx for Lustre not supported. S3 cannot be mounted as file sys	
ECS Auto Scaling	Auto incr/decr num of ECS tasks. Uses AWS Application Auto Scaling Metrics: 1. ECS Service Avg CPU utilization, 2. ECS Service Avg Memory Utilization (RAM), 3. ALB Request Count Per Target (metric coming from ALB) From these metrics, can do Target tracking (CloudWatch metric), Step Scaling (CloudWatch Alarm) or Scheduled scaling. ECS Service Auto Scaling (task level) ≠ EC2 Auto Scaling (EC2 instances level)	
EC2 launch type Auto Scaling	If need to scale num of EC2 instance instead of num of tasks. 1. ASG Scaling: based on CPU utilization -> add EC2 instance over time 2. ECS Cluster Capacity Provider: - auto provision and scale infrastructure for ECS tasks, - paired w an ASG, - add EC2 instance when missing capacity (CPU, RAM,...)	
ECS Rolling Update	When updating ECS from v1 to v2, can control how many can be started and stopped, and in which order Set min healthy percent (how many v1 task must be running) & max healthy percent (how many new v2 task can create) before all become v2	
ECS extras	EventBridge: S3 --event--> Amazon EventBridge --rule: run ECS task--> ECS task EventBridge Schedule (CRON job): EventBridge --every hr, use rule: run ECS task--> create new ECS task	
ECR	Store images on AWS. Can have private & public repository. Access is controlled through IAM Support image vulnerability scanning, versioning, image tags, image lifecycles,...	
EKS	Launch managed Kubernetes clusters on AWS. Kubernetes is cloud-agnostic (can be used on any cloud) Kubernetes is an open-source sys for auto deployment, scaling & management of containerized apps Support EC2 (worker nodes) or Fargate (serverless containers) Use cases: already using Kubernetes on-premise or on another cloud and want migrate to AWS, or want to use Kubernetes API ECS tasks = EKS Pods. EC2 instance = Worker nodes	

Serverless	Lambda,DynamoDB, Cognito, API Gateway, S3,SNS,SQS,KDF, Aurora serverless, Step Functions, Fargate
Lambda	Virtual fns (no servers to manage). Limited by time – short execution. Run on-demand. Auto-scaling Pay per request and compute time. Easy monitoring through CloudWatch. Increasing RAM will incr CPU and network Language support: Node.js (Javascript), Python, Java, C# (.NET Core), Golang, C#/Powershell, Ruby, Custom Runtime API (community supported, e.g. Rust), Lambda Container Image (image must implement Lambda Runtime API) Memory: 128MB - 10GB (1 MB increments). Max execution time: 15 mins. Environment vars: 4 KB Dick capacity in container (in /tmp): 512 MB. Concurrency executions : 1000 (can be increased) Deployment size (compressed .zip: 50 MB), (uncompressed code + dependencies: 250 MB)
Lambda @ Edge	Deploy lambda fns alongside CloudFront CDN: for more responsive apps Can use lambda to change CloudFront requests and responses: use this if need CUP for CloudFront 1. Viewer requests: After CloudFront receives a request from a viewer 2. Origin requests: Before CloudFront forwards a request to a origin 3. Origin response: After CloudFront receives a response from origin 4. Viewer response: Before CloudFront forwards a response to viewer Can generate response to viewer w/o sending the request to origin (viewer request + response only)

	Use cases: Website security & privacy, Dynamic web app at edge, Search engine optimization (SEO), Intelligently route across origins & data centers, Bot mitigation at edge, Real-time image transformation, A/B testing, User authentication & authorization, User prioritization, User Tracking & Analytics
DynamoDB	Fully managed, HA w replication across multiple AZs. NoSQL DB – not relational database Millions of request/sec, trillions of rows, 100s of TB of storage. Low cost & auto-scaling capabilities Fast & consistent performance (low latency). Integrated w IAM for security, authorization & admin Enables event driven programming w DynamoDB Streams. Standard & IA Table class
	Made of tables (database managed by AWS). Ea table has a primary key (defined at creation) Can have infinite items (rows). Ea item has attributes (columns; can be added over time) Max size of item is 400 KB. Data types supported: 1. Scalars (string, num, binary, boolean, null), 2. Document Types (List, Map), 3. Set Types (string set, num set, binary set) Primary key can be made of 1/2 col. Partition key & Sort key
	Provisioned mode (default): - pay for provisioned read capacity units (RCU) & write capcity units (WCU) - you specify num of read/writes per sec, - need to plan capacity beforehand, - can add auto-scaling mode for RCU & WCU On-demand mode: - read/write auto scale based on workload - no need to plan capacity - more expensive - great for unpredictable workload
DynamoDB Accelerator (DAX)	Fully managed, HA, seamless in-memory cache for DynamoDB. 5 mins TTL for cache (default) Help solve read congestion by caching. Microsecond latency for cached data Doesn't require app logic modification (compatible w xisting DynamoDB API)
DynamoDB Streams	Ordered stream of item-level modifications (create/update/delete) in table Can be sent to KDS, lambda, KCL apps. Data retention for 24 hrs Use cases: react to changes in real-time (send welcome email to users), analytics, insert into derivative tables, insert into ElasticSearch, implement cross-region replication
DynamoDB Global Table	Make DynamoDB table accessible w low latency in multiple regions Active-Active replication (apps can read/write to table in any regions) Must have DynamoDB streams as pre-requisite
TTL	Auto delete items after an expiry timestamp Use cases: reduce stored data by storing only current items, adhere to regulatory obligations
Indexes	Global Secondary Index (GSI) & Local Secondary Index (LSI) Allow queries on attributes other than primary key
Transactions	1 write writes to either multiple tables or none at all (eg deposit money writes to Transactions Logs & Account Balance Table)
API Gateway	Client --REST API--> API Gateway --Proxy Request--> Lambda Support for WebSocket protocol. Handle API versioning. Handle diff environments (dev, test, prod) Handle security (authentication & authorization). Create API keys, handle request throttling Can use Swagger/Open API import to quickly define API. Cache API response Transform and validate request & response. Generate SDK & API specifications
	Integrates w 1. Lambda: easy way to expose REST API backed by Lambda 2. HTTP: HTTP API on premise, ALB,... (add rate limit, caching, user authentications, API keys,...) 3. AWS Service: start an Step Fn Workflow, post message to SQS (add authentication, deploy publicly, rate control...)
	Endpoint Types: 1. Edge-optimized (default): - for global clients, - request routed through CloudFront edge locations (improves latency) - API Gateway still in same region (where we created) 2. Regional: - for clients within same region, - can manually combine w CloudFront (control over caching strategies & distribution) 3. Private: - accessed from ur VPC using an interface VPC endpoint (ENI), - use resource policy to define access
Security	IAM Permissions: Create an IAM policy authorization and attach to user/role API Gateway verifies IAM permissions by the calling app. Easy access within own infrastructure Uses Sig v4 capability where IAM credential are in headers
	Lambda Authorizers (formerly Custom authorizers): Use lambda to validate token passed in header Option to cache result of authentication. Helps to use OAuth/SAML/3rd party authentication Lambda will then return an IAM policy for user
	Cognito User Pools: Cognito fully managed user lifecycles

	API Gateway verifies identity automatically from AWS Cognito. No custom implementation required Cognito only helps with authentication, not authorization
AWS Cognito	Want to give user an identity for interaction w our apps Cognito User Pools: sign in functionality for app users. Integrate w API Gateway, ALB Cognito Identity Pools (Federated Identity): provide AWS credentials to users. Integrate w User Pools as an identity provider Cognito Sync: synchronize data from device to Cognito. May be deprecated and replaced by AppSync
Cognito User Pools (CUP)	Create serverless DB of users for ur mobile app. Simple login: username/email + pw. Possible to add email/phone num and do MFA Can enable Federated Identities (Facebook, Google, SAML,...). Sends back a JSON Web Token (JWT)
Cognito Federated Identity Pools	Provide direct access to AWS resources from client side Log in to federated identity provider (or remain anonymous) -> get temp AWS credentials from Federated Identity Pool -> credentials comes w a pre-defined IAM policy stating their permissions
Process	App --login to Identity Provider (CUP, Facebook, SAML...)--> send token back to app --authenticate to FIP--> Federated Identity --verify token w Identity Provider--> get credentials from STS --sends temp AWS credentials to app--> App ---> Identity Provider ---> App ---> FIP ---> Identity Provider ---> FIP ---> STS ---> FIP ---> App ---> AWS
Cognito Sync	Deprecated, should use AWS AppSync. Cross device synchronization (iOS, Android...) Store user preferences, config, state of app. Offline capability (sync when back online) Requires Federated Identity Pools (not User Pools). Store data in datasets (up to 1 MB) Up to 20 datasets to sync
Serverless Application Model (SAM)	Framework for developing and deploying serverless apps. All configuration (lambda, DynamoDB, API Gateway, Cognito User Pools) in YAML code Can run lambda, DynamoDB, API Gateway locally. Can use CodeDeploy to deploy lambda fns

DB	RDBMS (SQL, OLTP (online transaction processing)): RDS, Aurora – great for joins NoSQL DB: DynamoDB (~JSON), ElastiCache (key/value pairs), Neptune (graphs) – no joins, no SQL Object Store: S3 (for big objs) / Glacial. Graphs: Neptune – display relationships btw data Data Warehouse (SQL Analytics/BI) : Redshift (OLAP; online analytical processing), Athena Search: ElasticSearch (JSON) – free text, unstructured search
Redshift	Based on PostgreSQL but for OLAP. Columnar storage of data (instead of rows) 10x better performance than other data warehouse, scale to PBs of data Has Massively Parallel Query Execution engine (MPP). Pay based on ec2 instance provisioned Has SQL interface for performing queries. BI tools like Quicksight or Tableau integrate w it Data is loaded in from S3 (COPY command), DynamoDB, DMS, other DBs... From 1 - 128 Nodes, up to 128 TBs of space per node Leader node: query planning, result aggregation. Worker node: perform queries, send result to leader Backup & Restore, Security VPC / IAM/ KMS, monitoring Reshift Enhanced VPC Routing: COPY/UNLOAD goes through VPC instead of internet
	Has no Multi-AZ feature (all in 1 AZ). Can create snapshots of cluster (stored in S3) Snapshots are incremental (only what has changed is saved). Can restore a snapshot into a new cluster Automated: set schedule/storage usage as trigger. Set retention Manual: snapshot retained until u delete it Can configure redshift to auto copy snapshots of a cluster into another region ideal for running long complex queries. can cache results
	KDF -> S3 --S3 copy command--> Redshift OR S3 --S3 copy command--> Redshift Copy command can enable Enhanced VPC Routing to traverse AWS private connection EC2 instance w JDBC driver -> Redshift (better to write data in batches)
Redshift Spectrum	Perform queries on S3 (w/o loading data into redshift). Must create Redshift Cluster first Query is submitted to thousands of Redshift Spectrum nodes
Glue	Managed Extract, Transform, and Load (ETL) service. Fully serverless service Useful to prepare and transform data for analytics. Glue Data Catalog: catalog of datasets (metadata)
Neptune	Fully managed graph DB. For high relationship data, social networking, knowledge graphs (Wikipedia) HA acorss 3 AZ, up to 15 read replicas. Point-in-time recovery, cts backup to S3. KMS + HTTPS
OpenSearch Service	Successor to ElasticSearch. W OpenSearch, can search any field (even partial matches) Common to use as complement to other DB. Has usage for Big Data apps

	Can provision a cluster of instances. Integrate w KDF, AWS IoT, CloudWatch Logs for data ingestion Security through Cognito, IAM, KMS, SSL & VPC. Comes w OpenSearch Dashboard (visualization)

CloudWatch Metric	Metric is a var to monitor (CPUUtilization, NetworkIn...). Metrics belong to namespaces Dimension is an attibute of metric (instance id, environment,...). Up to 10 dimensions per metric Metrics have timestamp. Can create CloudWatch dashboard of metrics
	EC2 instances have metric updated every 5 mins. Detailed monitoring: every 1 min (pay more) EC2 memory usage is not pushed by default (need push as custom metric)
Custom Metric	Eg: Memory (RAM) usage, disk space, num of logged in users.... Use PutMetricData API Can use dimensions to segment metrics: instance.id, environment.name Metric resolution (StorageResolution API param): 1. Standard (60s). 2. High Resolution (1/5/10/30s) w higher cost Can push metric 2 weeks in the past or 2 hrs in the future (need ensure ec2 instance time set correctly)
CloudWatch Dashboards	Quick access to metrics and alarms. Dashboards are global. Can change timezone & time range Can include graph from diff AWS accts and regions. Can setup auto refresh (10s, 1min, 2m, 5m, 15m) Can share w ppl w/o AWS acct (public, email, 3rd party SSO w Cognito)
CloudWatch Logs	Log groups: arbitrary name, usually representing an app Log stream: instances within app / log files / containers Can define log expiration policy (never, 30 days...) Can send logs to S3, KDS, KDF, Lambda, ElasticSearch Can send logs from SDK, CloudWatch Logs Agent, CloudWatch Unified Agent (deprecated), Elastic Beanstalk (collection of logs from app), ECS (collection from containers), Lambda (from fn logs), VPC Flow Logs, API Gateway, CloudTrail based on filter, Route53 (log DNS queries)
	Can use filter expressions (eg find specific IP inside log, count occurrence of 'ERROR' in logs) Metric filter can then be used to trigger CloudWatch alarm CloudWatch Logs Insights can be used to query logs and add queries to CloudWatch Dashboard
	CloudWatch Logs -> S3. Use CreateExportTask API. Log data takes up to 12 hrs to be exported Not near-real time or real-time. If want stream logs, use Logs Subscriptions (filter) instead
	Logs Aggregation Multi-Account, Multi-Region: Multiple accts and regions logs have subscription filters on ea -> KDS -> KDF -> S3 (eg)
CloudWatch Agent & Logs Agent	By default, no logs from EC2 instance will go to CloudWatch Logs Need to run CloudWatch Agent on EC2 to push log files. Instance need IAM role to push to CloudWatch Logs. CloudWatch Logs Agent can be run on-premise as well CloudWatch Logs Agent: old version. Can only send to CloudWatch Logs CloudWatch Unified Agent: new version. Collect additional system-level metrics like RAM, processes... Can still send logs to CloudWatch Logs. Has centralized config w SSM Param Store
CloudWatch Alarm	Used to trigger notifications for any metric. Various options: sampling, %, min, max, ... Alarm States: OK, INSUFFICIENT_DATA, ALARM. Period: length of time to evaluate metric. High res custom metric (10s, 30s, multiple of 60s) Alarm Targets: 1. Stop, Terminate, Reboot, Recover EC2 instance. 2. Trigger Auto-Scaling action 3. Send notification to SNS (from which you can do anything) EC2 instance recovery: will keep the same Private, Public, Elastic IP, metadata & placement group
CloudWatch Events	Event Pattern: Intercept events from AWS services (EC2 instance start, CodeBuild failure, S3 Trusted Advisor, API call w CloudTrail integration) Schedule or CRON: e.g. Create events on every 4 hrs JSON payload created from event and passed to target (compute: lambda, batch, ECS task; integration: SQS, SNS, KDS, KDF; orchestrations: step functions, CodePipeline, CodeBuild, maintenance: SSM, EC2 actions)
EventBridge	Newer version of CloudWatch Events. Event bus can be accessed by other AWS accts Has Default Event Bus: generated by AWS services (CloudWatch Events) Partner Event Bus: events from SaaS services or apps (Zendesk, DataDog, Segment, Auth0...) Custom Event Bus: for your own apps Can archive events (all/filter) sent to an event bus (indefinitely or set period) Can replay archived events. Rules: defined how to process events (Event Pattern, Schedule or CRON)

Schema Registry	EventBridge can analyze events in bus and infer schema. Schema can be versioned Schema Registry allows you to generate code for app, that will know in advance how data is structured in event bus	
Resouce-based policy	Manage permissions for a specific Event Bus (same as S3 bucket policy) Allow/deny access from another AWS accts/ regions Use Case: aggregate all events from ur AWS Organisation in a single AWS acct or region	
CloudTrail	Provide governance, compliance and audit for your AWS acct. Enabled by default Get history of events/API calls made within ur acct by SDK, Console, CLI, AWS services Can put logs from CloudTrail into CloudWatch Logs or S3 Trail can be applied to all regions (default) or a single region. Events stored for 90 days by default To keep events beyond 90 days, log to S3 and query w Athena	
CloudTrail Events	Management Events: Ops performed on resources in AWS acct e.g. configure security (IAM AttachRolePolicy), config rules for routing data (EC2 CreateSubnet), set up logging (CloudTrail CreateTrail) By default, trails are configured to log management events Can separate Read Events (don't modify resources) from Write Events (may modify resources)	Data Events: - by default not logged - S3 obj level activity (GetObject, DeleteObject, PutObject): can separate read and write events - lambda function execution activity (Invoke API) CloudTrail Insights Events
CloudTrail Insights	Enable it to detect unusual activity in ur acct: inaccurate resource provisioning, hitting service limits, bursts of AWS IAM actions, gaps in periodic maintenance activity Does this by 1st analyzing normal management events to create a baseline -> continuously analyze write events to detect unusual pattern -> anomalies appear in CloudTrail console + event sent to S3 + EventBridge Event created (for automation needs)	
AWS Config	Helps w auditing and recording compliance of your AWS resources. Help record configurations and changes over time. Can receive alerts (SNS) for any changes E.g. is there unrestricted SSH access to SG? do my bucket have public access? how has my ALB config change over time? Per-region service. Can aggregate across regions and accts. Can store config data into S3	
Config Rules	Can use AWS managed Config rules or make custom Config rules (defined in Lambda) Eg. evaluate if ea instance type is t2.micro Rules can be evaluated / triggered for ea config change or at regular time intervals Does not prevent actions from happening (no deny) Can automate remediation of non-compliant resources using SSM Automation Documents (use AWS Managed Automation Documents or create custom Automation Documents) Can set remidation retries if resource is still non-compliant	
	Use EventBridge to trigger notifications when resource non-compliant. Can send config changes and compliance state notifications to SNS (all events – use SNS filtering or filter at client-side)	

Security Token Service (STS)	Grant limited and temp access to AWS resources. Token valid for 1 hr (must be refreshed) AssumeRole API: - within own acct for enhanced security, - Cross Acct Access (assume role in target acct and perform actions there) AssumeRoleWithSAML API: return credentials for users logged in w SAML AssumeRoleWithWebIdentity API: - return creds for users logged in w IdP (Facebook, Google login, OIDC compatible...), - not recommended, use Cognito instead GetSessionToken API: for MFA, from user or AWS acct root user
Identity Federation	Federation lets users outside AWS assume temp roles for accessing AWS resources Types of federation: SAML 2.0, Custom Identity Broker, Web Identity Federation w or w/o Cognito, Single Sign On (SSO), Non-SAML w AWS Microsoft Active Directory (AD) Using federation, don't need to create IAM users
SAML 2.0 Federation	Integrate AD/ ADFS w AWS (or any SAML 2.0). Provide access to AWS Console or CLI Need to setup trust btw AWS IAM and SAML (both ways). Enable web-based, cross domain SSO Use AssumeRoleWithSAML API. Amazon SSO is newer way of SAML 2.0 Federation
Custom Identity Broker	Use only if IdP not compatible w SAML 2.0. Identity broker must determine the appropriate IAM policy. Use AssumeRole or GetFederationToken API

Web Identity Federation	AssumeRoleWithWebIdentity not recommended, use Cognito instead as allows for anonymous users, data sync and MFA
Microsoft AD	Found on any Windows Server w AD Domain Services. Contains DB of objs: user accts, printers, computers, file shares, security groups Centralized security management, create acct, assign permissions. Objs organised in trees AWS Managed Microsoft AD: - Create AD in AWS, manage users locally, support MFA. - establish trust connections w on-premise AD AD Connector: - Directory Gateway (proxy) to redirect to on-premise AD, support MFA. - users managed on on-premise AD Simple AD: - AD-compatible managed directory on AWS. - cannot be joined w on-premise AD
AWS Organizations	Global service. Manage multiple AWS accts. Main acct is master acct (cannot change) Other accts are member accts. Member acct can only be part of 1 organization Get consolidated billing across all accts - single payment method Get pricing benefits from aggregated usage (vol discount for EC2, S3...) API available to automate acct creation. Can enable CloudTrails to send logs to central S3 acct Can send CloudWatch Logs to central logging acct. Establish cross acct roles for admin purposes
AWS Orgs	Organizational Units (OU). E.g. sales + retail + finance, prod + dev + test, proj 1 + proj 2 + proj 3 Ea OU then have multiple member accts in it
Service Control Policy (SCP)	Whitelist or blacklist IAM actions. Applied at the OU or acct level. Does not apply to master acct SCP applied to all users and roles of acct, including root user. SCP does not apply to service-linked roles (which enable other AWS services to integrate w AWS Orgs) SCP must have explicit Allow (does not allow anything by default) Use cases: Restrict access to certain services, Enforce PCI compliance by explicitly denying services
Moving Acct	Migrate acct from 1 org to another org: 1. Remove acct from old org. 2. Send invite from new org. 3. Accept invite to new org from member acct Migrate master acct: 1. Remove all member acct from org using process above. 2. Delete old org. 3. Repeat process above to invite master acct to new org
IAM Conditions	aws:SourceIP : restrict client IP from which the API calls are made aws:RequestedRegion : restrict region API calls are made to Can restrict based on tags. Can force MFA. When assume a role (IAM role), u give up ur original permissions and take the permissions assigned to the role. Conversely, w resource based policy (S3 bucket policy), the principal don't have to give up permissions
IAM Permission Boundaries	Supported for user and roles (not for groups). Use managed policy to set the max permissions an IAM entity can get. Can be used w AWS Organization SCP and AWS IAM policy
Resource Access Manager	Share resources that you own w other AWS accts. Avoid resource replication. Can share: VPC Subnets: - allow to have resources launched in same subnet, - must be from same AWS Org, - cannot share SG and default VPC, - participants can manage own resources but cannot view, modify, delete resources belonging to other participants, - resources in same VPC can talk to one another using private IP, - SG from other accts can be referenced for max security Can share: AWS Transit Gateway, Route53 Resolver Rules, License Manager Configurations
AWS SSO	Centrally manage SSO to access multiple accts and 3rd party apps. Integrated w AWS Orgs Support SAML 2.0 markup. Integration w on-premise Active Directory. Centralized permission management. Centralized auditing w CloudTrail

Encryption	1. Encryption in flight: Data encrypted before sending and decrypted after receiving. SSL certs help w encryption (HTTPS). Ensure no MITM (man in the middle attack) can happen 2. Server side encryption at rest: data encrypted after being received by server. Data decrypted before being sent. Stored in encrypted form thanks to a data key Encryption/decryption keys must be managed somewhere and server has access to it 3. Client side encryption: data encrypted by client and never decrypted by server. Data decrypted by receiving client. Could leverage Envelope Encryption
Key Management System (KMS)	AWS manages keys for us. Fully integrated w IAM for authorization Integrated w: EBS vols, S3 server side encryption, Redshift, RDS, SSM Parameter Store... Customer Master Keys (CMK):

	1. Symmetric (AES-256 keys): - single key for encryption/decryption, - AWS services using KMS use symmetric CMK, - necessary for envelope encryption, - cannot get access to key unencrypted (must use KMS API) 2. Assymetric (RSA & ECC Key pairs): - Public (encrypt) and private key (decrypt) pairs, - public key downloadable but cannot access private key unencrypted
	Able to fully manage keys & policies: create, rotation policies, disable, enable keys Can audit keys using CloudTrail. Types of CMK: AWS Managed Service Default CMK, User Keys created in KMS, User Keys imported Can only encrypt up to 4KB of data per call. If data > 4KB -> use Envelope Encryption To give access to KMS: ensure key policy allow user + IAM policy allow API calls
	KMS bound to specific region. To copy snapshots across regions: 1. Create snapshots of encrypted vol. 2. Copy snapshot to new region but re-encrypt w new KMS key in new region. 3. Recreate encrypted snapshot in new region
	KMS Key policies similar to S3 bucket policies. W/o key policies -> no one can access keys Default KMS Key policy give complete access to root user. Give access to IAM policies to KMS keys Custom KMS Key Policy: define user, roles that can access KMS keys. Define who can administer the keys. Useful for cross acct access of KMS keys
	Copy snapshots across accts: 1. Create snapshot of encrypted vol. 2. Attach a KMS Key policy to authorize cross-acct access. 3. Share encrypted snapshot. 4. Create a copy of encrypted snapshot and re-encrypt w KMS key in new acct. 5. Recreate vol from snapshot
KMS Key Rotation	Auto Key rotation for Cusomter-managed KMS (not AWS-managed CMK) If enabled: auto rotation every 1 year. Previous key kept active so that can decrypt old data New key has same CMK ID (only backing key is changed) Manual Key rotation: to rotate every 90 days, 180 days,... New Key has diff CMK ID. Previous key kept active so that can decrypt old data Better to use alias (to hide change of key for app)
SSM Parameter Store	Secure store for configuration and secrets. Can further encrypt secrets w KMS in SSM param store Serverless, scalable, durable, easy SDK. Version tracking of secrets/configs. Config management using path & IAM Notifications w CloudWatch Events. Integration w CloudFormation Use GetParameters or GetParametersByPath API
	Parameter policies: assign TTL to parameter to force updating and deleting of old pw Can assign multiple policies at a time
Secrets Manager	Newer service, meant for storing secrets. Can force rotation of keys, after X num of days Can automate generation of secrets on rotation (using lambda). Integrate w AWS RDS Encrypted w KMS. Mostly meant for RDS integration
CloudHSM	KMS: AWS manages software for encryption. CloudHSM: AWS provision encryption hardware Dedicated Hardware (Hardware Security Module). You manage encryption keys entirely HSM device is tamper resistent, FIPS 140-2 Level 3 compliance Support both symmetric and asymmetric keys (SSL/TLS keys). Good option to use w SSE-C encryption Must use CloudHSM Client Software. Redshift support CloudHSM CloudHSM clusters are HA. Great for availability and durability
AWS Shield	Shield Standard: free tier activated for all customer, protect against SYN/UDP floods, reflection attacks, layer 3/4 attacks, DDoS attacks... Shield Advanced: optional DDoS mitigation service. Protect against more sosphisticated attack on EC2, ELB, CloudFront, Global Accelerator, Route 53 24/7 access to DDoS Response Team (DRP). Protect against higher fees due to DDoS attacks
Web Application Layer (WAF)	Protect web apps from common web exploits (layer 7 = HTTP) Can be deployed on API Gateway, ALB and CloudFront Define Web ACL: - rules can include IP addresses, HTTP headers, HTTP body, URI strings. - protect from common attacks like SQL injection or cross-site scripting (XSS). - can put size constraints, geo-match (block countries). - has rate-based rules (count occurrences of events) – for DDoS protection
	Firewall Manager: manages all rules of WAFs in all accts of AWS Org Define common set of rules that apply to WAF, Shield Advanced (ALB, CLB, Elastic IP, CloudFront), SG for ECS and ENI resources in VPC

GuardDuty	Intelligent Threat Discovery to protect AWS acct. Use ML algo to detect anomaly using 3rd party data No need to install software. Input data includes: 1. CloudTrail Event Logs (unusual API calls, unauthorized deployments): Management events & S3 data events. 2. VPC Flow Logs. 3. DNS Logs. 4. Kubernetes Audit Logs Can set up CloudWatch Event Rules to notify findings to lambda, SNS Can protect against CryptoCurrency attacks
Inspector	Automated Security assessments For EC2 instances: - must run AWS System Manager agent, - detect unintended network accessibility, - analyze running OS against known vulnerabilities For containers pushed to ECR: assessment of containers as they are pushed Reporting and integration w AWS Security Hub. Can send findings to AWS EventBridge
Macie	Fully managed data security and data privacy service that use ML and pattern matching to discover and protect sensitive data in AWS. Currently only support S3 Helps identify and alaert u to sensitive data such as Personally Identifiable Information (PII)

API Gateway
RESTful APIs enable stateless client-server communication
WebSocket APIs (WebSocket protocol): enables stateful, full-duplex communication btw client & server

With caching for a stage enabled, you can reduce the number of calls made to your endpoint and also improve the latency of requests to your API.

SQS
You can use message timers to set an initial invisibility period for a certain message added to a queue. The default (minimum) delay for a message is 0 seconds. The maximum is 15 minutes

By default, all DynamoDB tables are encrypted under an AWS owned customer master key (CMK), which do not write to CloudTrail logs

Only Standard SQS queue is allowed as an Amazon S3 event notification destination, whereas FIFO SQS queue is not allowed

Lambda auto tracks number of requests, the latency per request, and the number of requests resulting in an error
Can view on lambda console, CloudWatch console, …

ALB supports authentication from OIDC compliant identity providers such as Google, Facebook and Amazon. It is implemented through an authentication action on a listener rule that integrates with Amazon Cognito to create user pools.

RAID 0 can stripe multiple volumes together; for on-instance redundancy, RAID 1 can mirror two volumes together.
EBS volumes (irrespective of the RAID types) are local disks and cannot be shared across instances

You can specify instance store volumes for an instance only when you launch it.
You can't detach an instance store volume from one instance and attach it to a different instance. The data in an instance store persists only during the lifetime of its associated instance. If an instance reboots (intentionally or unintentionally), data in the instance store persists.
If you create an AMI from an instance, the data on its instance store volumes isn't preserved
When you stop, hibernate, or terminate an instance, every block of storage in the instance store is reset.
An instance store provides temporary block-level storage for your instance.

If you restrict access by using, for example, CloudFront signed URLs or signed cookies, you also won't want people to be able to view files by simply using the direct Amazon S3 URL for the file. Instead, you want them to only access the files by using the CloudFront URL, so your content remains protected.
To generate this URL we must code, and Lambda is the perfect tool for running that code on the fly.

CIDR	Classless Inter-Domain Routing – mtd for allocating IP addresses. Used in SG and AWS Networking CIDR IPv4 = base IP (XX.XX.XX.XX) + Subnet mask (define how many bits can change in base IP; /8 = 255.0.0.0, /16 = 255.255.0.0, /32 = 255.255.255.255.255) E.g. 11.22.0.0/32 -> 2^0 = 1 -> 11.22.0.0 | 11.22.0.0/31 -> 2^1 = 2 -> 11.22.0.0 - 11.22.0.1 11.22.0.0/16 -> 2^16 = 65,536 -> 11.22.0.0 - 11.22.255.255 | 0.0.0.0/0 -> 2^32 -> 255.255.255.255 IP contains 4 octets 1.2.3.4, /32: no octet can change, /24: last octet can change, /16: last 2 octet can change, /8: last 3 octet can change, /0: all octet can change
	Private IP can only allow certain values: 10.0.0.0/8 for big networks. 172.16.0.0/12 for AWS default VPC 192.168.0.0/16 for home networks. All the rest of IP on the internet are public
Virtual Private Cloud (VPC)	All new AWS acct have default VPC. New EC2 instance launched in default VPC if no subnets specified Default VPC have internet connectivity and all EC2 instances in it have public IPv4 addresses EC2 instance also have public and private IPv4 DNS name Can have multiple VPC in a AWS region (max 5 – soft limit). Max CIDR per VPC is 5 For ea CIDR: min size is /28 (16 IP addresses), max size is /16 (65536 IP) Since VPC is private, only private IPv4 ranges are allowed (10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16) VPC CIDR should not overlap w your other networks
Subnet	Sub-range of IPv4 addresses withing your VPC. AWS reserve 5 IP addresses (1st 4 and last) in ea subnet These 5 IP are not available for use and can't be assigned to an EC2 instance E.g. 10.0.0.0/24 -> reserved IPs: 10.0.0.0 (Network Address), 10.0.0.1 (reserved by AWS for VPC router), 10.0.0.2 (reserved by AWS for mapping to Amazon-provided DNS), 10.0.0.3 (reserved by AWS for future use), 10.0.0.255 (Network Broadcast Address. AWS don't support broadcast in VPC, so IP is just reserved)
Internet Gateway (IGW)	Allow resources in a VPC to connect to the Internet. Scales horizontally, HA and redundant Must be created separately froma VPC. 1 VPC can only be attached to 1 IGW and vice versa IGW on their own do not allow Internet access for subnets Can do NAT for instance w public IP Need to edit route table so that resources in public subnet can connect to the Router -> IGW -> Internet
Bastion Host	Users --SSH--> Bastion Host (EC2 instance in public subnet) --> EC2 instance in private subnet Edit SG of private EC2 instance to allow access from Bastion Host SG. Edit SG of Bastion Host to allow port 22 (SSH) access from users IP addresses Must use NLB w them
NAT Instance	NAT = Network Address Translation. Allow EC2 instance in private subnet to connect to the internet Must be launched in public subnet. Must disable EC2 setting: Source / destination Check Must have elastic IP attached to it. Route tables must be configured to route traffic from private subnets to NAT Instance
	Pre-configured Linux AMI available. (No more standard support) Not HA/resilent to setup out of the box (need create ASG in multi-AZ + resilent user data script) Internet traffic bandwidth depends on EC2 instance type Need edit SG to allow inbound HTTP/HTTPS traffic from private subnets & SSH from home network Need edit SG to allow outbound HTTP/HTTPS traffic to internet
NAT Gateway	AWS-managed NAT, highly resilient, HA, no administration. NATGW is created in a specific AZ, need elastic IP. Can't be used by EC2 instance in the same subnet (only from other subnets) Private subnet -> NATGW -> IGW -> Internet. Has 5 Gbps of bandwidth w autoscaling up to 45 Gbps No SG to manage. NATGW resilient within a single AZ Must create multiple NATGW in multiple AZ for fault tolerance. No cross-AZ failover since if AZ goes down, then won't need NATGW
DNS Resolution in VPC	DNS Resolution (enableDnsSupport): - decide if DNS resolution from route 53 resolver is supported for the VPC - True by default: queries the Amazon Provider DNS Server at 169.254.169.253 or the reserved IP address at the base of the VPC IPv4 network range plus two (.2) DNS Hostnames (enableDnsHostname): - default true for default VPC, default false for new VPC - won't do anything unless enableDnsSupport = True - if true -> will assign public hostname to EC2 instance if it has a public IPv4 - will have private hostname (DNS) by default If u use a custom DNS domain names in a private Hosted Zone in route 53, must set enableDnsSupport & enableDnsHostnames to True
SG & NACL	Network Access Control List (NACL): firewall to control traffic in and out of subnet 1 NACL per subnet, each subnet are assigned the Default NACL (accept everything in/out of subnet) NACL rules: - rules have a num (1-32766), lower num have higher precedence

<table>
<tr><td></td><td colspan="2">- 1st rule match will drive decision, - last rule is an *, and denies all request in case of no rule match
- AWS recommends adding rules by increment of 100, - Newly created NACL will deny everything
- NACL are a great way of blocking a specific IP address at the subnet level</td></tr>
<tr><td>Ephemeral Ports</td><td colspan="2">For any 2 endpoints to establish a connection, they must use a port
Clients connect to a defined port and expect a response on an ephemeral port
Diff OS use diff port ranges: IANA & MS Windows 10 (49152 - 65535), Linux Kernel (32768 - 60999)
So when using NACL, need to specify range of ports to allow inbound and outbound</td></tr>
<tr><td></td><td>SG: - instance level
- support allow rules only
- stateful (return traffic auto allowed)
- all rules are evaluated
- manually apply to ec2 instance</td><td>NACL: - subnet level
- support allow and deny rules
- stateless (return traffic must be explicitly allowed)
- rules evaluated in order (lowest to highest), first match wins
- auto apply to all ec2 instance in subnet</td></tr>
<tr><td>VPC Reachability Analyzer</td><td colspan="2">Network diagnostic tool that troubleshoots network connectivity btw 2 endpoints in VPC. Builds a model of the network config, then checks reachability based on these configs (doesn't send packets)
When destination is reachable – it produces hop-by-hop details of the virtual network path
When destination not reachable – it identifies the blocking component(s)</td></tr>
<tr><td>VPC Peering</td><td colspan="2">Privately connect 2 VPC using AWS network. Make them behave as if they were in the same network
Must not have overlapping CIDRs. VPC Peering Connection is not transitive (must be established for ea VPC that needs to connect to another)
Must update route tables in ea VPC subnets to ensure EC2 instances can connect to one another
Can create VPC Peering connection btw VPCs in diff AWS accts/regions
Can reference a SG in a peered VPC (works cross acct – same region)</td></tr>
<tr><td rowspan="2">VPC Endpoints (AWS - AWS)</td><td colspan="2">Every AWS Service is publicly exposed (public IP). Redundant and scale horizontally
VPC endpoints, powered by AWS PrivateLink allows u to connect to AWS services through private network instead of the public Internet
Remove need for IGW, NATGW, ... to access AWS Services
Troubleshooting: check DNS Setting resolution in VPC, check route tables</td></tr>
<tr><td>Interface Endpoint:
Provisions an ENI (private IP address) as an entry point (must attach SG)
Supports most AWS services</td><td>Gateway Endpoint:
Provisions a Gateway and must be used as a target in a Route Table
Supports S3 and DynamoDB</td></tr>
<tr><td>VPC Endpoints (AWS - Customer)</td><td colspan="2">PrivateLink: Most secure and scalable way to expose a service to 1000s of VPC (own or other accts), unlike VPC Peering which exposes all resources in the VPC
Requires a NLB (AWS VPC) and ENI (Customer VPC) or GWLB
If NLB in multiple AZ and ENI in multiple AZ, then solution is fault-tolerant
ALB -> NLB --PrivateLink--> ENI (customer VPC) OR NLB --PrivateLink--> VGW -> CGW</td></tr>
<tr><td>VPC Flow Logs</td><td colspan="2">Flow logs capture info abt IP traffic gg into ur interfaces: VPC Flow logs, Subnet Flow logs, ENI Flow logs
Helps to monitor and troubleshoot connectivity issues. Flow logs can be sent to S3 or CloudWatch Logs
Captures network info from AWS managed interfaces too: ELB, RDS, ElastiCache, Redshift, Workspaces, NATGW, Transit GW...
srcaddr & dstaddr – help identify problematic IP, srcport & dstport – help identify problematic ports
Action – accept/reject of request due to SG/NACL
Can query VPC Flow Logs w Athena on S3 or CloudWatch Logs Insights</td></tr>
<tr><td rowspan="3">Site-to-Site VPN (S2S VPN)</td><td colspan="2">Connect to AWS VPC from on-premise using VPN connection (public but encrypted). Needs VGW + CGW
Virtual Private Gateway (VGW): - VPN concentrator on AWS side of the VPN connection
- VGW is created and attached to the VPC from which u want to create the S2S VPN connection
- Possible to customize the ASN (Autonomous System Number)
Customer Gateway (CGW): - soft app/physical device on customer side of VPN connection</td></tr>
<tr><td colspan="2">If CGW is public: use public internet-routable IP address for CGW
If CGW is private: if behind a NAT device (NAT-T; NAT traversal enabled), use public IP of NAT device
Still must enable Route Propagation for the VGW in the route table of ur subnets
If need to ping EC2 instances from on-premise, need add the ICMP protocol on the inbound of SG</td></tr>
<tr><td colspan="2">AWS VPN CloudHub: provide communication btw multiple sites, if u have multiple VPN connection
Low cost Hub-and-Spoke model for primary or secondary network connectivity btw diff locations (VPN only). VGW + multiple CGW -> customer network can talk to one another as well
VPN connection -> traverse public internet but encrypted</td></tr>
</table>

	Connect multiple S2S VPN connection btw same VGW and multiple CGW, setup dynamic routing and configure route tables
Direct Connect (DX)	Provide a dedicated private connection from a remote network to ur AWS VPC Dedicated connection must be set up btw ur DC and AWS Direct Connect locations (physical location) Need to setup VGW on VPC. Support both IPv4 and IPv6. VIF = virtual network interface Can access public resource (S3 on public VIF) and private (EC2 on private VIF) on same connection Use cases: Increased bandwidth throughput (large datasets lower cost), more consistent network experience (app using real-time data feeds), hybrid environments (cloud + on-premise)
	Dedicated Connection: 1 Gbps & 10 Gbps capacity, - physical ethernet port dedicated to customer - request made to AWS first, then completed by AWS Direct Connect Partners Hosted Connections: 50,500 Mbps,... to 10 Gbps, - connection request made to Direct Connect Partners - Capacity can be added/remove on-demand, - 1,2,5,10 Gbps available at select Direct Connect Partners Often longer than 1 month to set up connection
	Data in transit not encrypted but private If need encryption: can set up DX + VPN to provide an IPsec-encrypted private connection
	High resiliency for critical workloads -> set up DX in multiple locations Maximum resiliency -> set up DX in multiple locations + ea location has multiple connections
Direct Connect Gateway	To setup Direct Connect to > 1 VPC in diff regions (same acct), must use Direct Connect GW Customer network --DX connection--> DX Location --private VIF--> DX GW --private VIF--> VGWs
Transit Gateway	For transitive peering btw thousands of VPC and on-premise, hub-and-spoke (star) connection Can connect w Direct Connect Gateway or connect w VPN connection Regional resource, can work cross-region. Share cross-acct using Resource Access Manager (RAM) Can peer transit gateways across regions. Route table: limit which VPC can talk to which VPC Only AWS service that support IP Multicast. Can use to share DX btw multiple accts
	S2S VPN ECMP (equal-cost multi-path routing). Routing strategy to forward a packet over multiple best path. Use case: create multiple S2S connections to increase bandwidth of connection to AWS
VPC - Traffic Monitoring	Capture and inspect network traffic in ur VPC. Route the traffic to security app that u manage Capture the traffic from (source) – ENI and to (target) – ENI or NLB Can have filter to capture all packet or only packets of interest Source and Target can be in same VPC or diff VPCs (VPC Peering) Use cases: content inspection, threat monitoring, troubleshooting...
IPv6	Every IPv6 address is public and Internet routable (no private range) X.X.X.X.X.X.X.X (X is hexadecimal from 0000 to ffff) IPv4 can never be disabled for VPC and subnets. Can enable IPv6 to operate in dual-stack mode EC2 instance will get at least a private IPv4 and public IPv6
Egress-only IGW	Used for IPv6 only (similar to NAT Gateway but for IPv6 only). Must update route table Allow instances in private VPC outbound connection over IPv6 while preventing internet to initiate an IPv6 connection to instances

DR	RPO (recovery point objective) and RTO (recovery time objective). Ideal low RPO and low RTO RPO = time before disaster that backup is made. RTO = time after disaster that backup is restored. DR strategies: Backup and Restore, Pilot Light, Warm Standby, Hot Site / Multi Site approach
	Backup and Restore: high RPO, high RTO, low cost Pilot Light: critical core always running in the cloud, lower RPO and RTO than backup & restore Warm Standby: full sys is running, but at min size. After disaster -> can scale to production load. Multi Site/Hot Site: lowest RPO and RTO, most ex. Full production scale running on AWS and on-premise
DB Migration Service (DMS)	Quickly and securely migrate DB to AWS, resilient, self-healing. Source DB remain available during migration. Support homogeneous migration (same type of DB) and heterogeneous migration (diff type of DB) Support cts data replication using CDC (change data capture). Create EC2 instance to do the replication
	Sources: most DB, S3. Targets: most DB, Redshift, S3, Elasticsearch, KDS Use Schema Conversion Tool (SCT) only if doing heterogeneous migration
On-Premise Strategy	Can download Amazon Linux 2 AMI as VM (.iso format) VM Import/Export: migrate existing app to EC2, create a DR repository strategy for on-premise VM, can export back VM from EC2 to on-premise

	AWS Application Discovery Service: gather info abt on-premise servers to plan a migration, server utilization and dependency mappings, track w AWS Migration Hub AWS Server Migration Service: incremental replication of on-premise live servers to AWS
AWS DataSync	Move large amt of data from on-premise to AWS. Move data from NAS or file sys via NFS or SMB Can synchronize to S3 (any storage class), EFS, FSx (Windows, Lustre...) Replication task can be scheduled hourly, daily, weekly Need to install DataSync agent to connect to sys. Can setup bandwidth limit Can also use to replicate data across regions
AWS Backup	Fully managed service. Centrally managed and automate backups across AWS services Supported: EC2, EBS, S3, RDS, Aurora, DynamoDB, DocumentDB, Neptune, EFS, FSx (Windows, Lustre), Storage Gateway (Volume Gateway) Support cross-region/cross-acct backups. Support point in time recovery (PITR) for supported services Support on-demand and scheduled backup, tag-based backup policies Create backup policies known as Backup Plans: define backup frequency (schedule/CRON), backup window, transition to cold storage (never, days...years), retention period (always, days...years)
	Vault Lock: Enforce a WORM (write once, read many) state for all backups that you store in AWS Backup Additional layer of defense against: inadvertent or malicious delete ops, updates that shorten or alter retention period Even root user cannot delete backups when enabled

ML	Rekognition: face detecting, labeling, celebrity recognition Transcribe: audio to text Polly: text to audio Translate: translations Sagemaker: ML for developer and data scientist Personalize: real-time personalized recommendations	Lex: build chatbots Connect: cloud contact center Comprehend: NLP Forecast: highly accurate forecast Kendra: ML-powered search engine Textract: detect text and data in document

HPC	Cluster placement. EC2 enhanced networking (SR-IOV): - higher bandwidth, higher packer per sec (PPS), lower latency, - Option 1) Elastic Network Adapter (ENA) up to 100 Gbps, - 2) Intel 82599 VF up to 10 Gbps Elastic Fabric Adapter (EFA): - improved ENA for HPC, only for Linux, - great for inter-node communications, tightly coupled workloads, - uses message passing interface (MPI) standard, - bypasses underlying Linux OS to provide low-latency, reliable transport	
	AWS Batch: - support multi-node parallel jobs, enabling u to run a single job that span multiple EC2 instances - easily schedule jobs and launch EC2 instances accordingly	AWS ParallelCluster: - open-source cluster management tool to deploy HPC on AWS, - configure w text files, - automate creation of VPC, subnets, cluster type and instance types, - can enable EFA on cluster

CICD	Cts Integration: - developers push code to a code repository (Github, CodeCommit...) - testing/build server checks code as soon as it's pushed (Jenkins CI, CodeBuild...) Cts Delivery (ensure software can be released reliably, usually means auto deployment): - Jenkins CD, CodeDeploy,... Code (CodeCommit) -> Build + Test (CodeBuild) -> Deploy (CodeDeploy) -> Provision (User managed EC2 instances fleet; CloudFormation). Can do Deploy + Provision w Elastic Beanstalk (PaaS) AWS CodePipeline for orchestrating everything Code -> Build -> Test -> Deploy -> Provision
CloudFormation (CF)	Infrastructure as a code. Is declarative way of outlining AWS Infrastructure, for any resources (e.g. SG, 2 EC2 instances using this SG, 2 Elastic IP for the 2 EC2, S3 bucket...) CloudFormation create these resources, in the right order w the exact configuration u specify
CF	No resources manually created. Infrastructure code can be version controlled. Changes to infrastructure reviewed through code Ea resources within the stack is tagged w an identifier, so u can see how much a stack cost u Can estimate cost of resources using CloudFormation template
	Can destroy and recreate infrastructure on the fly. Automated generation of diagram for template Declarative programming (no need to figure out ordering and orchestrating) Can create many stacks for many app, and many layers (eg. VPC stacks, Network stacks, App stacks)
	Templates have to be uploaded in S3 and then referenced in CF. Cannot edit previous template. Must upload new version. Stack = actual collection of resources

	Stacks are identified by a name. Deleting a stack deletes all resources in it To deploy: 1) editing template in CF Designer + use console to input params, 2) edit template in YAML file + CLI to deploy templates can use change sets to see how infra would change
	StackSets: create, update, delete stacks across multiple accts and regions w a single operation Admin acct to create StackSets. Only Trusted Accts can create, update, delete stack instances from StackSets Updating StackSets will update all associated Stack instances throughout all accts and regions
Step Functions	Build serverless workflow to orchestrate lambda functions. Represent flow as JSON state machine Can use lambda fns in sequence, parallel, conditions, timeouts, error-handling... ^ & Glue Max execution time of 1 year. Can implement human approval feature
Simple Workflow Service (SWF)	Coordinate work amongst app. Code run on EC2 (not serverless). 1 year max run time Concept of 'activity step' and 'decision step'. Has built in 'human intervention step' Step Fns is newer than SWF, and hence is recommended except: - if require external signals to intervene in process - if need child processes that return value to parent processes
EMR	Elastic MapReduce help create Hadoop clusters (Big data) to analyze and process big amts of data Cluster can be made of 100s of EC2 instances. Supports Apache Spark, HBase, Presto, Flink... EMR takes care of all the provisioning and config. Has auto-scaling and integrated w spot instances
Opsworks	Chef & Puppet help perform server config automatically or repetitive actions Opsworks = managed Chef & Puppet. Alternative to SSM
AWS Workspaces	Managed, secure cloud desktop. Great to eliminate management of on-premise VDI (virtual desktop infrastructure). On demand, pay by usage. Integrated w Microsoft AD
AppSync	Store and sync data across mobile and web app in real time. Makes use of GraphQL
Cost Explorer	Visualize, understand and manage AWS costs and usage over time Can choose optimal savings plan. Forecast usage up to 12 mths based on previous usage

Well-Architected Framework & Tool	Pillars: Operational Excellence, Security, Reliability, Performance Efficacy, Cost optimization, Sustainability
Trusted Advisor	High level AWS acct assessment on cost optimization, performance, security, fault tolerance & service limits. These core checks available for all customers best practices Full Trusted Advisor (for business & enterprise support plants): - can set CloudWatch alarms when reaching limits, - programmatic access using AWS Support API

CloudFront content types that bypass the regional edge cache, and go directly to the origin.
1) Dynamic content, as determined at request time (cache-behavior configured to forward all headers)
2) Proxy methods PUT/POST/PATCH/OPTIONS/DELETE go directly to the origin

Providing shared access to services required by workloads in each of the VPCs
Consider an organization that has built a hub-and-spoke network with AWS Transit Gateway. VPCs have been provisioned into multiple AWS accounts, perhaps to facilitate network isolation or to enable delegated network administration. When deploying distributed architectures such as this, a popular approach is to build a "shared services VPC, which provides access to services required by workloads in each of the VPCs. This might include directory services or VPC endpoints. Sharing resources from a central location instead of building them in each VPC may reduce administrative overhead and cost.

Elastic Beanstalk automatically handles the deployment, from capacity provisioning, load balancing, auto-scaling to application health monitoring. At the same time, you retain full control over the AWS resources powering your application and can access the underlying resources at any time.

To resolve any DNS queries for resources in the AWS VPC from the on-premises network, you can create an inbound endpoint on Route 53 Resolver and then DNS resolvers on the on-premises network can forward DNS queries to Route 53 Resolver via this endpoint.
To resolve DNS queries for any resources in the on-premises network from the AWS VPC, you can create an outbound endpoint on Route 53 Resolver and then Route 53 Resolver can conditionally forward queries to resolvers on the on-premises network via this endpoint. To conditionally forward queries, you need to create Resolver rules that specify the domain names for the DNS queries that you want to forward (such as example.com) and the IP addresses of the DNS resolvers on the on-premises network that you want to forward the queries to.

X-ray
- provides an end-to-end view of requests as they travel through your application, and shows a map of your application's underlying components.
- trace data across AWS accounts and visualize it in a centralized account

Maximum resiliency: Opt for two separate Direct Connect connections terminating on separate devices in more than one Direct Connect location
High resiliency: Opt for one Direct Connect connection at each of the multiple Direct Connect locations
Non-critical production/development environment: Opt for at least two Direct Connect connections terminating on different devices at a single Direct Connect location

Dynamic port mapping with an Application Load Balancer makes it easier to run multiple tasks on the same Amazon ECS service on an Amazon ECS cluster.

RDS DB
OS updates standby first, promote standby to master, then update the old master/new standby
Upgrades to the database engine level require downtime. Even if your RDS DB instance uses a Multi-AZ deployment, both the primary and standby DB instances are upgraded at the same time.

As the Availability Zones got unbalanced, Amazon EC2 Auto Scaling will compensate by rebalancing the Availability Zones. When rebalancing, Amazon EC2 Auto Scaling launches new instances before terminating the old ones, so that rebalancing does not compromise the performance or availability of your application
Amazon EC2 Auto Scaling creates a new scaling activity for terminating the unhealthy instance and then terminates it. Later, another scaling activity launches a new instance to replace the terminated instance

A large multinational retail company has a presence in AWS in multiple regions. The company has established a new office and needs to implement a high-bandwidth, low-latency connection to multiple VPCs in multiple regions within the same account. The VPCs each have unique CIDR ranges.?
The company should implement an AWS Direct Connect connection to the closest region. A Direct Connect gateway can then be used to create private virtual interfaces (VIFs) to each AWS region.
Direct Connect gateway provides a grouping of Virtual Private Gateways (VGWs) and Private Virtual Interfaces (VIFs) that belong to the same AWS account and enables you to interface with VPCs in any AWS Region (except AWS China Region).
You can share a private virtual interface to interface with more than one Virtual Private Cloud (VPC) reducing the number of BGP sessions required.

Enhanced networking provides higher bandwidth, higher packet-per-second (PPS) performance, and consistently lower inter-instance latencies. If your packets-per-second rate appears to have reached its ceiling, you should consider moving to enhanced networking because you have likely reached the upper thresholds of the VIF driver. It is only available for certain instance types and only supported in VPC. You must also launch an HVM AMI with the appropriate drivers.
AWS currently supports enhanced networking capabilities using SR-IOV. SR-IOV provides direct access to network adapters, provides higher performance (packets-per-second) and lower latency.

AWS Serverless Application Model (AWS SAM) is an extension of AWS CloudFormation that is used to package, test, and deploy serverless applications.

With ALB and NLB IP addresses can be used to register:
- Instances in a peered VPC.
- AWS resources that are addressable by IP address and port.
- On-premises resources linked to AWS through Direct Connect or a VPN connection.

Amazon DynamoDB auto scaling uses the AWS Application Auto Scaling service to dynamically adjust provisioned throughput capacity on your behalf, in response to actual traffic patterns. This is the most efficient and cost-effective solution to optimizing for cost.

Run Command is designed to support a wide range of enterprise scenarios including installing software, running ad hoc scripts or Microsoft PowerShell commands, configuring Windows Update settings, and more on all target EC2 instances.
Run Command can be used to implement configuration changes across Windows instances on a consistent yet ad hoc basis and is accessible from the AWS Management Console, the AWS Command Line Interface (CLI), the AWS Tools for Windows PowerShell, and the AWS SDKs.

Default security groups have inbound allow rules (allowing traffic from within the group) whereas custom security groups do not have inbound allow rules (all inbound traffic is denied by default). All outbound traffic is allowed by default in custom and default security groups.

When you launch an instance into a default VPC, we provide the instance with public and private DNS hostnames that correspond to the public IPv4 and private IPv4 addresses for the instance.
When you launch an instance into a nondefault VPC, we provide the instance with a private DNS hostname and we might provide a public DNS hostname, depending on the DNS attributes you specify for the VPC and if your instance has a public IPv4 address.

You can control who can administer your file system using IAM. You can control access to files and directories with POSIX-compliant user and group-level permissions. POSIX permissions allows you to restrict access from hosts by user and group. EFS Security Groups act as a firewall, and the rules you add define the traffic flow.

You can associate an AWS Direct Connect gateway with either of the following gateways:
- A transit gateway when you have multiple VPCs in the same Region.
- A virtual private gateway.
In this case account Z owns the Direct Connect gateway so a VPG in accounts A and B must be associated with it to enable this configuration to work. After Account Z accepts the proposals, Account A and Account B can route traffic from their virtual private gateway to the Direct Connect gateway

A VPC automatically comes with a default network ACL which allows all inbound/outbound traffic. A custom NACL denies all traffic both inbound and outbound by default.