19 Dec 2023 (1)

AWS - CP {CLF-C01}

✓ = revn (notes)
✓✓ = with quiz
~~skip~~ =

Revision Checklist :

- ✓✓ Basics
- ✓✓ IAM
- ✓✓ EC2
- ✓✓ EC2 Storage
- ✓✓ ELB + ASG
- ✓✓ S3
- ✓✓ Databases + Analytics
- ✓✓ Other Compute Services
- ✓✓ Deployment + Management at Scale
- ✓✓ Global Infrastructure
- ✓✓ Cloud Integrations
- ✓✓ Cloud Monitoring
- ✓✓ VPC + networking
- ✓✓ Security
- ✓✓ ML smol
- ✓✓ Acc mgmt + AWS support + Billings
- ✓✓ Adv Identity smol
- ✓✓ Ecosystem + Architecture
- ✓✓ other services smol

- ✓ Make a list of all global services, AZ tied services, hybrid services.
- A list of all services (flowchart).

AWS | Cloud Basics

Service Models ⇒ IaaS, PaaS, SaaS.

Deployment Models ⇒ Private
Public
Hybrid

Shared Res Model:

* Customer := security (IN) the cloud
* AWS = security (of) the cloud

(Data, IAM, appⁿ, configuration of OS + n/w + firewalls
encryption (client-side + server-side)

↳ customer's res.

(software ⇒ compute, storage, database, n/w
hardware ⇒ region, AZs, edge locⁿ

↳ AWS res.

* Cloud Shell = alternative to CLI. on console.

AWS can be accessed with:

→ console (manual) (protected by MFA)
→ CLI (protected by access keys)
↳ can be generated through the console
→ SDK → API.

IAM - Identity Access Management (3)

- IAM users, groups.
  - ↳ no sub-groups, only individual users
- IAM Policies — JSON docs → allow / deny
  - ↳ structure: version
    ID optional
    statement {
      sid optional
      Effect ⇒ allow / deny
      Principal ⇒ which AC/group
      Action ⇒ what is allowed / denied
      Resource
      condition optional
    }

IAM MFA (Multi-factor auth)
⇒ password + security device = login

to protect users/groups
↓ ↓
IAM pwd policy

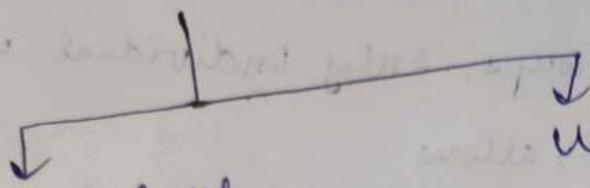
① Virtual → AWS Authy
② U2F Key → Yubikey (3rd party)
③ More hardware keys and devices

IAM Roles — for services
— EC2, lambda funcⁿ, CF, etc.
— topi → logo

IAM policies
↳ users, groups

IAM roles
↳ services

- IAM Security Tools



Account-level

⇓

Credentials report

↳ report of all user credential's status within the account.

user-level

⇓

Access Advisor

↳ shows permissions given and last access data for a user.

- IAM Best Practices

→ never use root account

→ MFA + strong pwd policy

→ Roles

→ use access keys for CLI/SDK. → rotation of keys is recommended.

→ use cred report + access advisor to revise policies.

→ least privilege principle.

EC2 - Elastic Cloud Compute (4)

* Virtual servers / VMs / instances
* EC2 - user data script → runs on initialization / booting
  → runs with root user privilledge.
* SSH-ing into a server ⇒ Key-pair
  Secure shell
  - public key on instance
  - pvt key on local machine used as auth to SSH into EC2 instance.

* Types of instances
  → general purpose = balance = ex: t2.micro
  → compute optimized
  → memory optimized = analyse / process large dataset
  → storage optimized = high r/w needed
  → Accelerated computing
  → HPC optimized

* Security Groups → firewall rules for EC2 instances.
  control:
  - access to ports
  - allowed IP ranges
  - inbound + outbound traffic.

  ⇓
  "allow" rules for IPv4, IPv6, another security group.

* Relationship ⇒ SGs (M) ←→ (M) EC2
  - Default → all inbound = blocked
    → all outbound = allowed

* Common Ports:

22 = SSH → linux instance , 3389 = RDP → window instan
21 = FTP , 22 = secure FTP
80 = HTTP , 443 = HTTPS .

. EC2 instance connect (console) is an easir alt. to SSH

* Instance purchasing options ($$):

1.) On-demand
2.) Reserved → normal
→ convertible (change config)
3.) savings plan → commit to amt of usage
4.) Spot instances → cheapest, unreliable
5.) Dedicated hosts, instances ⇒ book entire
⇒ no sharing
6.) capacity reservation in an AZ.

* shared Res Model :

customer → SG rules, OS patches + updates, anything installed on the EC2 machine, IAM roles assigned to EC2., data on instance.

EC2 Instance storage

✓ EBS
  ↳ snapshots
  ↳ also NFS
✓ AMI
✓ Image Builder
✓ EC2 Instance Store (hardware)
✓ EFS + FSx (NFS)

★ EBS = elastic block storage
= NFS = a drive on n/w you can attach to an EC2 instance. = persists data after termination.
= one instance ↗ many EBS volumes
one EBS volm ↗ one instance
= pined to an AZ. To move around — snapshots.
= to create an EBS volm ⇒ you need to provision certain capacity → GBs / IOPS.
= billed regardless of usage.

EBS Snapshots = backup for EBS volm
= recommended to detach an EBS volm block from EC2 instance before taking a snapshot
= use snapshots to copy data from EBS in ~~sugi~~ AZ A to AZ B.
= snapshots are stored in S3
= features: snapshot archive, recycle bin

★ EC2 AMI = amazon machine image
= snapshot / blueprint of an OS / VM launch template
= AMI are copy-able across regions.
= AMI template also includes the snapshot of EBS attached
= creating AMI = speeds up boot process.

↓ launch
public AMIs
custom AMIs
marketplace AMIs.

* EC2 Image Builder = automation of AMI — creation, maintenance, validation, testing
  = does so by creating a Buidir EC2 instance
  = regional but lets you distribute AMIs globally.
  = free services
  = can be run on a schedule

* EC2 instance store
  = hardware soln.
  = physically connectable to the instance.
  = solve the issue of latency. higher IOPS.
  = lose storage if stopped
  = data loss possible, ensure backup.

* EFS = elastic file system
  = a managed NFS
  = can be attached to thousands of instances at once.
    ↳ adv over EBS. and also, EFS = regional and EBS = AZ.
  = on-demand, no pre capacity planning
  = only works with Linux EC2.

* EFS-IA : 92% cost saved.

* FSx (3rd party)
  = native microsoft file sys.
  = managed NFS for AWS.
  = FSx → lustre = HPC
    → windows file server = windows
    → NetApp

ELB + ASG

terms ✓ load balancing ✓ load balancing types
- ALB - NLB - GLB - classic

✓ ASGs ✓ ASG strategies

Scalability - vertical ↑ - horizontal →
- ability to accomodate increase in load

elasticity - ability of a system to scale up and down
- cloud feature

agility - unrelated - rapid IT infra. access.

★ ELB = servers that control and distribute incoming traffic.
= manged service
= users are exposed to ELB instead of servers directly.
= advantages : single point of access (DNS)
failure recovery
regular health checks of instances
provides SSL termination for your app^n (HTTPS) - secure connection + encrypt^n.

★ Types

1.) App^n load balancer - Layer 7 - HTTP/HTTPS - Static DNS

2.) N/w load balancer - Layer 4 - ultra high performance - (HTTPS) TLS / TCP/UDP - static IP through elastic IP.

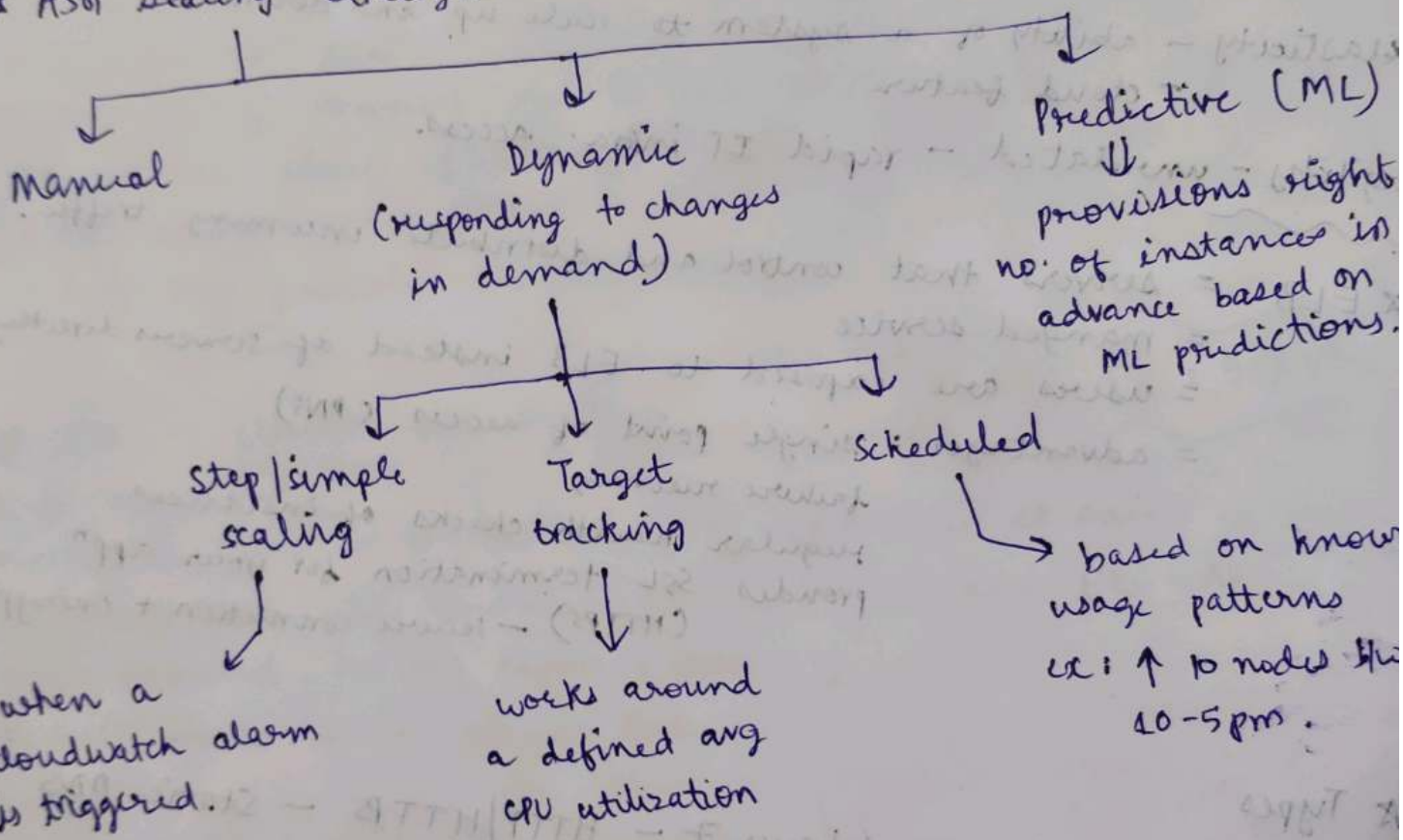
3.) Gateway load balancer - layer 3 - routes traffic to firewalls attached to your instances.
- GENEVE protocol traffic.

4.) Classic Load balancer - Layer 4 and 7 - old & retired.

★ Target group - a group of instances you want your ELB to direct the user traffic towards.

- ASGs = auto scaling groups
  - = automatic scale up or down in case of traffic fluctu
  - = horizontal scaling (add / remove nodes).
  - = max cost optimization, truly on-demand.
  - = need to specify a min, max no. of instances.
  - = unhealthy instances are replaced by new ones through ASGs.

* ASG Scaling Strategies:



S3 - Simplified Storage Service

(7)

basics ✓ security ✓ versioning ✓ replication (CRR, SRR)
storage classes • IAM access analyser • encryption

S3 = store objects in buckets = each object has a key for unique identification
= regional = bucket name must be globally unique.

Security

1) IAM Policies (for users specific to S3 - allow or deny)
2) Object encryption
3) Resource Based
   - Bucket Policies = bucket wide rules = .json
   - Bucket ACL = who can and cannot access the bucket
   - Object ACL = " for objects.

   * Cross-account access is allowed or denied using bucket policy..

4) Bucket setting for "Block Public access : on".
   overrules even if allowed in bucket policy.

* Versioning = can be enabled at the bucket-level
= roll back to any prev versions

* Replication = asynchronus replication from source bucket to target bucket. = enable versioning for both buckets
= two buckets can belong to diff accounts.

* S3 storage classes = data can be moved b/w diff storage classes using Lifecycle rules. = cost-opt.
= all classes: dur = 99.9999%. (11 9's)
avail = 99.99% exept one-zone IA (99.5%)

Classes:

- S3 Standard
- S3 IA — standard, onezone (→ one AZ) — less frequently used BUT rapid access req
- S3 Glacier — backup, archive — cost = storage + retrieval
  - S3 glacier instant retrieval
  - " " flexible retrieval → expited (5min), Std (5h), Bulk (12h)
  - " " deep archive
    - 12h – 48h
    - (std) (bulk)
- Intelligent Tiering = auto-tiering rules
  = no retrieval fee

★ S3 encryption

- server-side (enabled by default by aws)
- client-side (client's choice)

★ IAM Access Analyser for S3:

→ find which resources are publically shared.

→ ensures only intended people have access

S3 – Snow family

highly secure, portable set of physical devices
for – data migration
– edge computing

- advised to used snowball for data migration if it takes more than a week to transfer data through the n/w.

* Data Migration → snowball client = AWS OpsHub

1.) Snowball edge – move TBs or PBs of data in or out of AWS.
   - → storage optimized = 80 TB of HDD
   - → Compute optimized = 42 TB of HDD

2.) Snowcone – small, portable computing. rugged + secure even in harsh env.
   - lightweight, used where snowball cannot fit.
   - → Snowcone – 8TB of HDD
   - → Snowcone SSD – 14TB of SSD
   - you need to provide battery / cables for it.
   - used with DataSync to set data using internet to AWS.

3.) Snowmobile (TRUCK) – to transfer EB of data. 1EB = 1000 PB = 1000 000 TB
   - capacity of 100 PB
   - secure with GPS, 24/7 surveillence
   - if data to transfer > 10 PB – this is better than snowball.

* Edge computing

1.) Snowcone (HDD) and Snowcone SSD
2.) Snowball edge – compute opt and storage opt.

* These come with long term 1 or 3 year commitments.

★ AWS OpsHub

→ use OpsHub to manage your Snow family devices can be installed.

★ Storage Gateway

→ used to expose on-premises data to AWS S3.

→ used esp for hybrid cloud.

3 types → file gateway — AWS EBS
→ volm gateway — AWS S3
→ tape gateway — S3 glacier

Databases + Analytics

Basics DBs ✓RDS ✓Aurora ✓RDS deployment options
✓Elasticache ✓DynamoDB + DAX + Global tables ✓Redshift
✓EMR ✓Athena ✓QuickSight ✓DocumentDB ✓Neptune
✓QLDB ✓Glue (ETL) ✓Managed Blockchain ✓DMS

Databases → relational = SQL → RDS, Aurora
→ non-relational = noSQL ex: JSON → DynamoDB
→ in-memory = Redis and Memcached → Elasticache

Relational DB → RDS, Aurora

★ RDS = relational DB service = SQL
= managed service = continuous backups by EBS
= you cannot SSH into your DB.
= free-tier = OLTP (transactions)

★ Aurora = aws' DB technology = not open source
= MySQL, PostgreSQL
= performance opt. = cloud optimized DB.
(5x more than RDS)
= serverless DB. = OLTP (transactions)

★ RDS deployment options

(1) Read Replicas → upto 15 read replicas – load balancing

(2) Multi-AZ → failover DB in one more AZ for disaster recovery.
passive until main DB is working fine.

(3) Multi-region → read replicas in multiple regions. You can read from any but write only to the main RDS.

★ Elasticache → get managed in-memory DB services
(Redis, Memcached)
→ Reduces load off of main DB (like RDS) by caching freq accessed data in-memory DB. → faster access.

★ Managed Blockchain = join public blockchain n/w or create decentralized blockchains.

* DynamoDB = managed, nosQL non-rel^n DB w/ replication in
= serverless DB = prousses millions of req per sec.
= Key-value DB. = scalable, secure, cheap, serverless.
= standard and IA table types → cost opt.

* DAX – DynamoDB Accelerator
= managed, in-memory cache only for DynamoDB

* Dynamo DB Global Tables
= allows 2-way replication (R/w both allowed) of tables accross multiple regions.

* RedShift = data warehousing + analysis = PostgreSQL = OLAP (online analytical processing)
= allows MPP ⇒ massively parallel query exec^n.
= serverless = can be integrated with BI tools.

* EMR (Elastic MapReduce) = a tool that helps in creating hadoop clusters
||
helps you analyse and process large amounts of data using Apache Hadoop (ex.)
cluster of nodes to perform parallel computations like big data analytics.

* Athena = serverless query engine = queries over data stored in S3 buckets. = uses SQL

* QuickSight = dashboard (for analysed data) on databases
= ML-powered = visual = easier to conclude = it is run on top of databases, so, no server provisioning

* Document DB = cloud native nosQL DB for mongoDB → used to store and query JSON.

* Neptune = graph DBs = ex: social n/w

* QLDB = Quantum Ledger DB = record financial transactions
= centralized = SQL queried

* Glue = ETL service = serverless
extract, load, transform

* DMS = data migration service
bas = from one DB to another
= self healing = homo/hetro

Other Compute Services (10)

Basics ✓ ECS → ECR, fargate ✓ Lambda ✓ API Gateway ✓ Batch ✓ Lightsail

Docker = containerize app^n
VMs vs containers → each VM has its own OS but all containers on top of a host machine share the host's OS.

ECS = Elastic Container Service = used to run containerized app^{n}s.
= to run these ~~serverless~~ containerized apps
→ fargate = serverless = each container is given own CPU + RAM
→ ec2 servers = ECS lets you run multiple containerized app^{n}s on the same instance

- ECR = Elastic container ~~Repocit~~ Registry
= aws' private docker container repo lets you store docker images

Lambda = FaaS = Functions as a Service = serverless
= reactive type of service (event-driven then fnc is called).
= pricing ⇒ pay per req and duration of func^n running.
- Lanabda allows to run docker containers if container img has implemented Lambda Runtime API.

API Gateway = serverless = for devs to create, maintain, monitor APIs.
= API gateway is for end-users to access the API.
= supports RESTful APIs + WebSocket APIs.

Batch = not serverless but relies on EC2. = runs 10000s of computing batch jobs.
Batch job = a job with a start and end point.
= schedule tons of batch jobs with AWS Batch.

Lightsail = server, storage, DB, n/w in simpler (light) way.
= no-autoscaling. = simpler alt. = for beginners.

Deploy + Manage at scale

CloudFormation ✓ Elastic Beanstalk ✓ Dev tools
- codedeploy
- code commit
- code build
- code pipeline
- artifact - codestar

Cloud9 ✓ SSM ✓ Opsworks.

★ CloudFormation

= IaC = YAML or JSON support on console
= use CDK (cloud dev kit) for more lang support.
= declerative programming = destroy / create CF templates on the go
= stack designer in cf = view architecture created by your code
= "view in designer" : hands-on.

★ Elastic Beanstalk

= PaaS = developer centric service. = all services in one-view.
= developer can focus on app^n code and data, else is managed by AWS.
= free

- Deployment Models in beanstalk
  - ↳ single instance deployments
  - ↳ load balanced autoscaling
  - ↳ only auto scaling
- Health monitoring — beanstalk has its own.

Developer Tools

- CodeDeploy = deployment sol^ = fully managed = servers must be provisioned ahead of time with CodeDeploy Agent. = hybrid
- CodeCommit = git-based AWS' internal repository
- CodeBuild = compiles, run tests, produce packages from source code so its ready to deploy.
- CodePipeline = orchestrate the steps involved in pushing code to production.
  = CI/CD = fully managed.
- CodeArtifact = store and retrieve dependencies involved w/ a software package. = npm, yarn, etc. ↳ by codecommit.
- codestar = quick one shot way to start your dev project. Provisions and sets up everything you need. star fr.

★ Cloud 9 = cloud IDE for coding. = accessible through a browser.
= allows real-time pair-programming.

★ SSM (Systems Manager)
= manage EC2 instances and on-premises servers.
= gives insights about state of your infrastructure.
= automatic patching of servers, run commands across all servers in one go.
= install SSM agent onto systems we control.

• SSM session manger
= secure shell into servers without SSH-ing into them.

• SSM Parameter store
= store parameters (API Key, passwords), secrets securely.
= versioning ✓ = encryption ✓.

★ AWS OpsWork = managed chef & Puppet in the cloud.
= alt. to SSM.

Global Infrastructure

sic ✓ Route 53 ✓ CDN CloudFront ✓ Global Accelerator
S3 acceleration transfer.

Regions → AZs (3 or more) → edge locations / PoPs.

* Route 53 → DNS → route users to closest deployment (least latency) = global anycast routing
  domain name system.
  → domain registration, manage DNS records, load balance, health checks, routing policy
    → simple
    → weighted
    → latency min
    → failover

  A → IPv4
  AAAA → IPv6
  CNAM → alias domain

* CloudFront = content delivery n/w or servic
= caches the content (data) to several edge locations (216 edge locn).
= DDos protection
= delivers content responding to any HTTP request.

* S3 Transfer Acceleration = file transfer by i) forward to edge location ii) then use private AWS n/w to reach target S3 bucket.

* Global Accelerator = improves global appn availability by leveraging the AWS internal n/w. (60% speed up).
Through this, appn from around the world is sent to edge locn nearest to you and the to you using public IP. www n/w.

• Outposts • Wavelength • Local zones.

* AWS outposts = hybrid = extends aws infrastructure and tools to on-premises data centres.
= setup physical outpost server racks on-premises. comes preloaded with aws.

* Wavelength = 5G

* Local zones = mini extensions of regions.
= located closer to cities.
(small data centre outposts)
= reduced latency.

Cloud Integrations

SQS ✓ SNS ✓ Kinesis ✓ MQ

decouple 2 app^n by inserting : SQS (queue)
SNS (pub/sub)
kinesis (streaming)

★ Amazon SQS = allows to store messages too.
= serverless = rapidly scalable = no limit to no. of messages in the queue.

= produces send mess to quee, consumers poll the quee for mess, once read, mess are deleted from the queue.
= offers FIFO strategy. = consumers share messages.

★ Kinesis = real-time big data streaming
= managed service to collect, process, analyze real-time data.
= analysed data sent to S3 / redshift. ~~throw~~

★ SNS = simple Notif service = send one message to many.
Pub/sub model = publisher = sender
= subscriber = receiver

SNS topics are created.

★ Amazon MQ = message broker service for RabbitMQ, ActiveMQ
= easier to switch to MQ from on-premises than to convert to SQS/SNS.
= servers. = not super scalable.
= has both queue and topic feature.

★ Kinesis Data Firehose ⇒ ETL services for kinesis?

Cloud Monitoring

CloudWatch – metrics, alarms, logs, events
CloudTrail ✓ X-Ray ✓ CodeGuru ✓ Health Dashboard

✱ CloudWatch

1.) Metrics → parameters used to quantify the performance of your resources. → has timestamps → create dashboard for visualizing usage.

2.) Alarms → trigger notifs for any metric. → lead to an alarm action.

3.) Logs → collect, store, analyse log files from app^n & resources.
→ valuable for troubleshooting, monitoring.
→ CloudWatch Log Agent fetches log files from servers for feeding into Logs.

4.) EventBridge → React to events happening on AWS. Create rules that match events with trigger action responses.

✱ CloudTrail → history of events / API calls made within your AWS acc. → governance, compliance, audit. → through: SDK, CLI, console, services.

→ enabled by default.
→ tracks every happening.
→ this data is sent to → S3
→ CloudWatch Logs.

✱ X-Ray → just like real life x-ray. → traces all requests made and gives a visual representation. → identify exactly where the issue / bottleneck is. → pinpoints issues.

✱ CodeGuru → ML-based → automated code reviews and performance recs. → Java and Python.

→ codeguru reviewer ⇒ code-review for static code (dev)
→ codeguru profiler ⇒ recs about app^n during runtime (production)

- Health Dashboard
→ personalized info about status of ALL your services.
→ use dashboard to analyse ongoing and past issues.

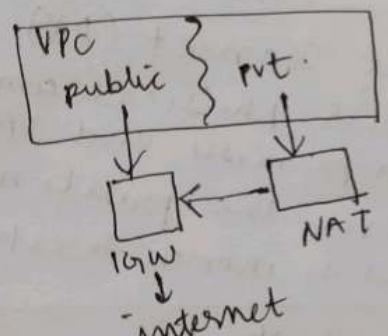
- Account health dashboard
↳ events that impact your infrastructure.

VPC & Networking

✓ Elastic IP ✓ VPC + Subnets + IGW + NAT ✓ NACL + SGs.
✓ VPC Flow logs ✓ Peering ✓ endpoints ✓ Private Link
✓ Site-to-site VPN + DX ✓ Client VPN ✓ ~~Open VPN~~ Transit GW.

★ Elastic IP → assigns permanent public IP to an instance
→ fee incurred when not in use.

★ VPC and Subnets → Public → Internet gateway (IGW)
↓ Private → NAT gateway → access internet while remaining private.
virtual private cloud n/w to deploy app^ns.
→ CIDR range is decided to allow certain IPs.



• VPC is region scoped

★ Network ACL → firewall that controls traffic to and from the subnet.
↳ allow/deny IPs.

★ Security Groups → firewall that controls traffic to and from EC2. → "allow" rules for IPs and other SGs.

★ VPC Flow Logs ⇒ log of all traffic flowing in and out of the VPC. ⇒ these help w/ monitoring and troubleshooting.
⇒ these logs can be fed into S3, CloudWatch Logs, Kinesis Data Firehose, etc.

★ VPC Peering ⇒ connect 2 VPCs privately on AWS' network.
⇒ VPCs must not have overlapping CIDR ranges. ⇒ non-transitive.

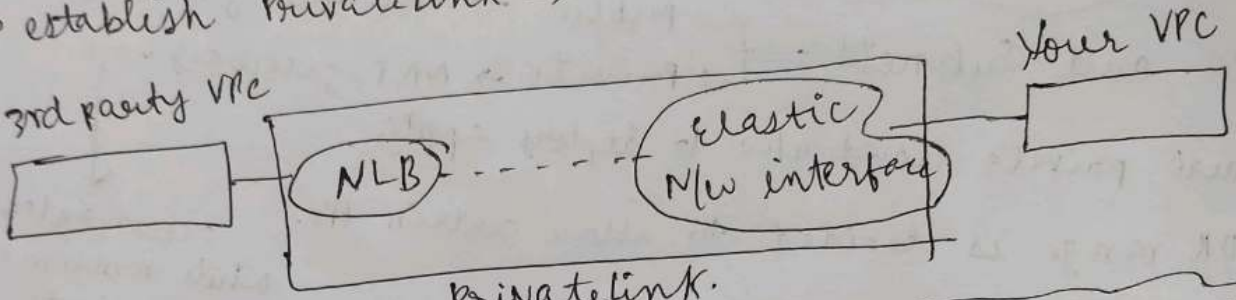
★ VPC Endpoints ⇒ connect to AWS services using private AWS n/w instead of public www. ⇒ security ↑ latency ↓

- Endpoints → Endpoint Gateway = S3, DynamoDB
  → Endpoint Interface = all others

★ PrivateLink (VPC endpoint services)

→ to expose a VPC to 1000s of services within another VPC. → way more scalable than peering.

→ to establish PrivateLink ⇒



★ Site-to-site VPN

- connect VPC to on-premises using the public www n/w.
- connection is encrypted.

Direct Connect (DX)

- establish physical connⁿ b/w on-premises and VPC
- goes over the private n/w
- needs a month to establish

→ both are ways to connect on-premises to VPC.

★ Client VPN (OpenVPN)

→ To access an EC2 instance deployed within the private subnet of a VPC. → connection is established publically.

★ Transit Gateway: peering connection b/w 1000s of VPCs in a star hub-and-spoke manner → simplified configuration.

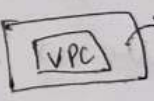
→ connect 1000s of VPCs and on-premises networks all together.

Security & Compliance

DDOS Attacks + protection ✓ Network Firewall ✓ Pen testing
✓ encryption - KMS + CloudHSM ✓ ACM ✓ secrets manager
✓ Guard Duty ✓ Inspector ✓ Config ✓ Macie ✓ Security Hub
✓ Detective ✓ AWS Abuse ✓ Root user privileges ✓ IAM access analyser

★ Shared Res Model ⇒ shared controls - patch mgmt
- configuration mgmt - awareness + train

★ DDOS Protection
1) Shield standard - protects from most common attacks
- enabled by default for free
2) Shield Advanced - 24/7 premium DDOS protection.
- subscription-based. - large scale DDOS.
3) WAF - Web app^n Firewall - not exclusively for DDOS - still offers DDOS protection by creating block/allow traffic and defining WACL.

★ Network Firewall → protect entire VPC. 

★ Penetration testing → simulation of cyberattacks.
→ DDOS-ing own infra, port flooding, req flooding, etc. = prohibited.

Encryption → KMS (key mgmt service)
→ customer managed
→ aws managed
→ aws owned
→ CloudHSM keys → keys generated from the CloudHSM (hardware security model) device.

★ ACM = AWS Cert. Manager = easy provision, maintenance and automatic renewal of SSL/TLS certificates.

* Secrets Manager → protect sensitive info. store and manage centrally - secure retrieval possible.
→ forced rotation of secrets, on rotation - automatically generates secrets with Lambda.
→ encrypted with KMS.

* Artifact → not a service → compliance docs → checks if you are compliant. → reports, agreements

* Guard Duty → Intelligent threat discovery using ML.
→ no software needed. → continous monitoring to find sus.
→ integrable with EventBridge rules and can target SNS or Lambda func^n.
→ CRYPTOCURRENCY attacks ✓.

* Inspector → finds s/w vulnerabilities in EC2, ECR, Lambda.
* Config → track config changes + compliance rules.
* Macie → Find sensetive PII data in S3 buckets.
* Security Hub → gather security findings from multiple AWS accounts.
* Detective → find root cause of security issues or sus activities.

* Abuse → report to Aws.
* Root user privileges
→ change acc settings
→ close your Aws acc
→ change / cancel AWS support plan
→ register as seller in Reserved instance marketplace.

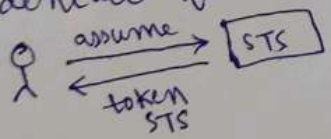
* IAM access analyser
→ defines a zone of trust, looks for risks outside that zone for findings.
↳ identify.

Machine learning

1.) Rekognition → face, object detection from img, videos, etc.
2.) Transcribe → speech to text → Automatic Speech Recognition (ASR)
3.) Polly → text to audio
4.) Translate → translations
5.) Lex → Alexa based on this → build chatbots using ASR
→ intent recognition.
6.) Connect → cloud call centre
7.) Comprehend → NLP based text analyser → group articles by topics within text
8.) SageMaker → easy ML at one place for devs and data scientist.
9.) Forecast → super accurate future predictions.
10.) Kendra → ML-powered search engine for inside documents.
11.) Personalize → real-time personalized recommendations.
→ ex: amazon.com → customized UX.
12.) Textract → extract text and data in documents.

Advanced Identity

1) AWS STS → security token service → temporary security credentials for accessing resources → define expiration period
→ considered safer for short-term usage like cross account access.



2.) Cognito → a service to manage DB of users, add authentication for them, secure access control to your apps. → provides the login with google | fb | github → this feature → social identity provider.

3.) Microsoft AD ⇒ centralize repo for managing metadata (info) about n/w resources. → It is a DB of users, devices on network, security groups, files, etc. ⇒ windows, central mgmt, active directory, domain ⇒ M-AD.

4.) AWS Directory Services

→ AWS = no AD of its own = you can extend it using dir services.
→ AWS Managed Microsoft AD (hybrid)
→ AD connector = redirects req to on-premises server which has actual AD.
→ Simple AD = a simple standalone AD in AWS
= cannot be connected to on-premises.

5.) AWS IAM Identity Centre (Single Sign-On)

→ provides one login / single sign-on for : all AWS accounts
: business cloud apps
: EC2 instances

→ could be built in AWS or 3rd party.

Well Architected framework – guiding principles
stop guessing capacity. test sys at production scale, automation of architectural experimentation (IaC, PaaS), evolving architecture. data-driven architecture, improve through game days.

★ Design Principle - Best Practices
Scalability – Disposable Resources – automatic – loose coupling your app^ns. – leverage max^m services not just servers.

★ Well architected framework – 6 pillars
1.) Operational excellence
2.) Security
3.) Reliability
4.) Performance efficiency
5.) Cost optimization
6.) Sustainability (nature)

① Operational Excellence
→ run and operate and monitor and improve smoothly.
→ IaC, annotate documents, make freq small reversible changes – easy to roll back, refine operations freq, anticipate and learn from all failure. → cloudFormation (IaC) ★.

② Security
→ protect info, systems and assets while delivering value.
→ Stong identity foundation. (IAM), enable traceability (logs), apply security at all layers, automate best security practices, protect data at rest and motion, keep everyone away from data, prepare for security events, shared res model.

③ Reliability
→ ability to recover from infra or service disruptions.
→ test recovery procedures, automatic failure recovery, scale horiz; stop guessing capacity, manage change through automation.

④ Performance efficiency
→ adapting and offering best use of resources while getting job done.
→ democratize adv technologies (adapt), go global in minutes, serverless, experiment more often, mechanical sympathy
be aware of all aws services.

⑤ Cost Optimization
→ deliver business value at lowest price point.
→ adopt consumption role (pay for what you use), measure overall efficiency, stop spending on data centres, analyse and attribute expenses and hence determine ROI, use tags, used "managed" services to reduce TCO.

⑥ Sustainability
→ minimize env impacts of running cloud env.
→ understand impact, establish goals, max utilization, adapt to newer more eff services, used managed services, reduce downstream impact.

★ To review your architecture against the 6 pillars, use the AWS well architected tool. → QnA type.

AWS CAF (Cloud Adoption Framework) = e-book guide for digital traformation by leveraging AWS.

→ Org capabilities
↓
→ grouped into 6 perspectives

→ Business capabilities → Business
→ People
→ Governance

→ Technical capabilities → Platform
→ Security
→ operations

• Cloud Transformation value chain
— 4 domains of transformation ⇒ tech, process, organization, product
• CAF – 4 transformation phases ⇒ envision, align, launch, scale.

★ AWS Right Sizing = process = before migration
= continually after
↳ action plan wrt the 6 CAF perspectives.

★ Help and guidance → AWS Professional Service
APN partners are granted w/ APN competency ← AWS Partner Network (APN)
↳ tech ↳ consulting ↳ training
• Navigate prog → improves APNs

ecosystem continued...

AWS IQ → help with experts (3rd party freelancers for AWS work within AWS — IQ).

* AWS re:post → Q&A forum (like stackoverflow). → free tier

* AWS Managed Service → not a usual "managed" service
  ↳ team of AWS experts → a typical team of people for support.
  → infrastructure and application support → implements best practices for your infrastructure → available 24/365.

* AWS Knowledge center ⇒ most common FAQs.

* AWS private training for orgs
  AWS academy for universities
  AWS training + certification → US Gov
  → enterprise

Other Services

1.) Workspaces → Desktop as a service → virtual-desktop (windows/linux)
→ multi-region deployments closest to end users for latency ↓.

2.) Appstream 2.0 ⇒ steaming service in the browser.

3.) IoT Core ⇒ n/w of interconnected devices for data trasfer
⇒ connect IoT devices to cloud ⇒ Build IoT app"s that gather, process, analyze data.

4.) Elastic Transcoder ⇒ convert media files in S3 into user-playback-device friendly formats (phone, ipad, etc.)

5.) Appsync → build backend, store data using GraphQL

6.) Amplify → set of tools to help you develop and deploy full stack web, mobile app"s. → all within Amplify studio.

7.) Device Farm → Test app"s on various devices at a large farm scale. Real devices not emulators.

8.) Backup → fully, centrally managed automatic backup for all AWS services, cross region + cross acc backup. on-demand, scheduled backups.
→ Supports point-in-time Recovery.

9.) Disaster Recovery strategies
Backup and restore.
Pilot light
Warm standby
Multi-site / Hot-site

10.) Elastic Disaster Recovery (DRS) -
→ continous block-level replication of servers into staging and production env.
→ quick and easy recovery

11) Data Sync
→ Move large Data from on-premiss to AWS.

12) App^n discovery service
→ Plan migration by gathering info about on-premiss data centers.
agent based ← → agentless

→ Simplest way to move ⇒ App^n migration service (MGN) (~~AMA~~)
⇒ same architecture as DRS.

13) Migration Evaluator
→ analyses migration plan to generate business case for cloud.

14) Migration Hub central location to disover, access, plan, track, modify migrations.

15) Fault Injection Simulator → chaos engineering.
→ uses experiment template.

16) Step functions
→ flowcharts / graph → posiblity of human approval at any step.

17) Satellite data = ground station.

18) Pinpoint ⇒ 2-way communications
⇒ better than SNS and SQS.
⇒ group / segmentize customers for personalization.

Account Mgmt, Billing, etc.

* AWS Organizations → consolidate multiple AWS accounts into an organisation → consolidated billing → discount from aggregated usage → pooling of reserved EC2 instances.
→ automate AWS account creation.
→ Restrict using SCP (Service control Policies) → JSON
  → dont apply to master account (of org).
  → blacklist or whitelist
  → OU level or individual level
  → doesnt apply to service-linked roles.
  → enforces PCI compliance at OU level.
  * → rule applied at OU level is => account level.

* AWS control tower → easy to setup, govern a secure and compliant multi-account environment → best practices ✓
→ automate environment setup → automate policy mgmt using guardrails (controls/rules) → monitors + detects policy violations.
→ runs on top of organizations.

* AWS RAM → resource access manager → share resources owned by you with other AWS accounts. → avoids resource duplication → always specify permissions and access levels.

* Service Catalog → for new AWS users, admins create portfolios which consist of products (Cloudformation templates) and gives users IAM policy defining which product to use.

* Pricing Models in AWS:
→ Pay as you go → Save when you reserve
→ Pay less by using more → Pay less as AWS grows.

* Free-tier services:
IAM, VPC, consolidated billing, Beanstalk, CF, ASG
pay for resources created.

• Service Pricing Models

① EC2 → instance type, no. of instance, data processed by ELB, detailed monitoring (per min), type of instance.

② Lambda → per func^n call, duration of func^n runtime

③ ECS → EC2 launch type
Fargate → CPU and memory allocated to containers in app^n..

④ S3 → free transfer IN, paid trasfer out, storage class, vol^m

⑤ EBS → provisioned in advanced, not on-demand
→ monthly per GB payments. → added snapshots cost

⑥ RDS → per hour, no. of I/O req per month

⑦ CloudFront → diff across regions.

⑧ N/w costs ⇒ inter AZ communication → pvt n/w ⇒ 0.01 $
→ public n/w ⇒ 0.02 $

⇒ inter region ⇒ 0.02 $

• Compute savings plan → most flexible → commit to usage.

• Compute optimizer → ML based resource recommender and analysis of utilization, efficiency, cost ↓ performance ↑ uses cloudwatch metrics..

★ Billing + costing tools

i) Estimation ⇒ Pricing calculator → includes free tier
ii) Tracking ⇒ Billing Dashboard ~~(only in us-east-1)~~ → monthly dashboard
Cost Allocation tags → group resources together
Cost and usage reports (most comprehensive)
Cost explorer → visual + next 12 months usage forecast.

iii) Monitoring | alarm | alerts ⇒ Billing alarms → us-east-1
⇒ Budgets
• alarms for exceeded budget and for forcasted to exceed budgets.
• allows upto 5 SNS notifs per budget.

★ Cost Anamoly Detection
↳ analyse + alert if smth "weird" happens. → ML
→ no need to define thresholds
→ sends report with root-cause.

★ Service Quotas – default limits on All AWS resources.

★ Trusted Advisor → recommendation on already existing resources.
↳ basic, developer = 7 core checks
↳ busines, enterprise = full checks.