(Jai Shree [illegible])

AWS - SAA notes

In networking

7 Application.
6 presentation
5 Session
4 Transport layer (segment)
3 Network router → (packet)
2 data link switch → (frame)
1 physical

4 3 2 1

As we go down, data gets encapsulated before it is send into physical wires.

$2^n - 2 \geq$ $\quad$ $2^n \geq 4$

$\boxed{n = 2}$

Q) 10.16.0.0/16 → divide it into 4 networks.

$2^n \geq 4$ $\quad$ $\therefore n = 2$ = network will steal 2 bits from host

$\therefore$ earlier = 16, now = 16 + 2 = 18

10.16.00000000.0 $\quad$ $\therefore$ on intervals of 64

128 64 32 16 8 4 2 1

18

AWS resources are Not truly in the public internet Zone.

They are in the public AWS Zone. Any EC2 that we access we ~~are~~ in VPC, ~~they~~ are projected to AWS public Zones.

~~① EC2~~

EC2 summary.

Private by default → uses default VPC unless mentioned
↳ EC2 is launched in a SUBNET (AZ)

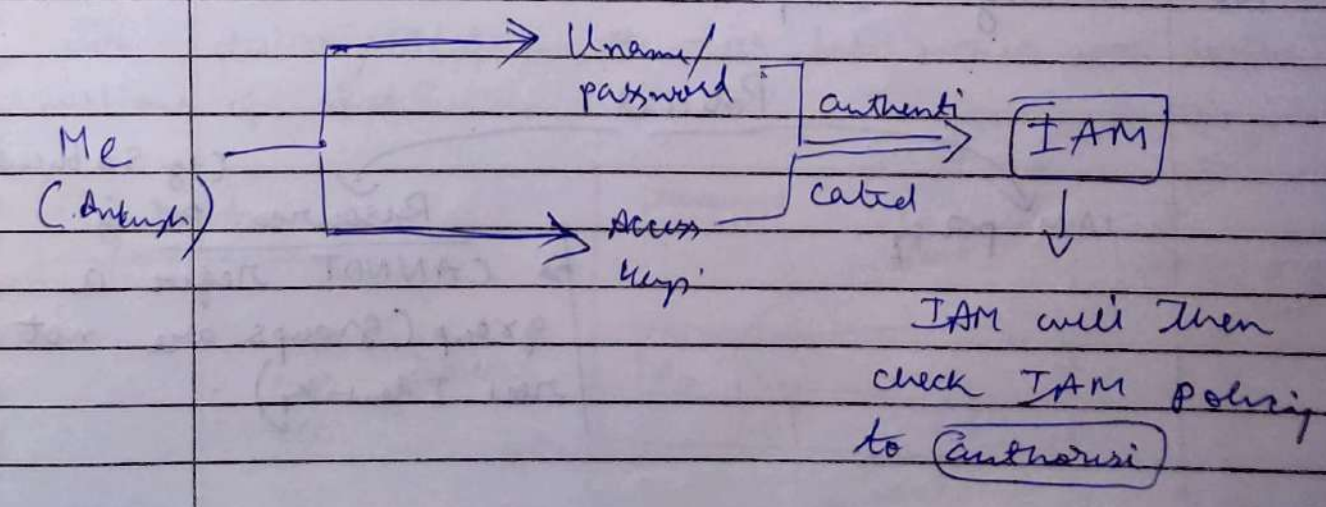
Billing → per second → min 60 secs.

States → running, stopped, terminated

IAM ①

Managed → individual policies can be attached to users
Inline → exceptions to managed policies

IAM Users → long term AWS access - a named thing. Named ~~thing~~ for long term access.



Access keys are for users [max 2 per user]

ARN

[arn : aws : service : region : acc-id : resource id
↳ for S3 → region & acc_id is not needed

FACTS

→ 5000 users per account (global)
→ Each user can be part of 10 groups

↳ If 75k users needed → we will use Federations / Roles.

Groups

Groups can have inline & managed policy.

2) How many users in a single group?
Ans) All users may (5000 users)

NO 1 GROUP ∀ | 300 groups per account
NO Nesting

Policies

IAM policy

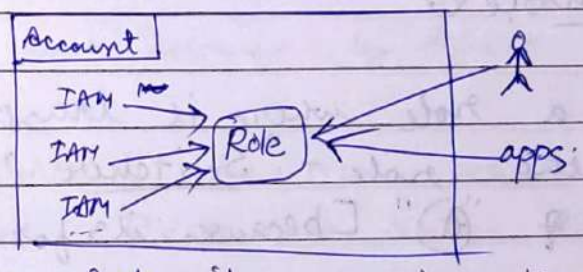
Resource policy (eg S3 bucket)
→ CANNOT refer a group (groups are not real Identity).

Meaning - We (can't) create bucket policy and add (allow) an entire group in principle section (groups can not be referenced).

CAN'T

Roles

Its an identity (real identity).



Will use role if we dont know how many principles gonna use it. OR if we need > 5000 users to use something.

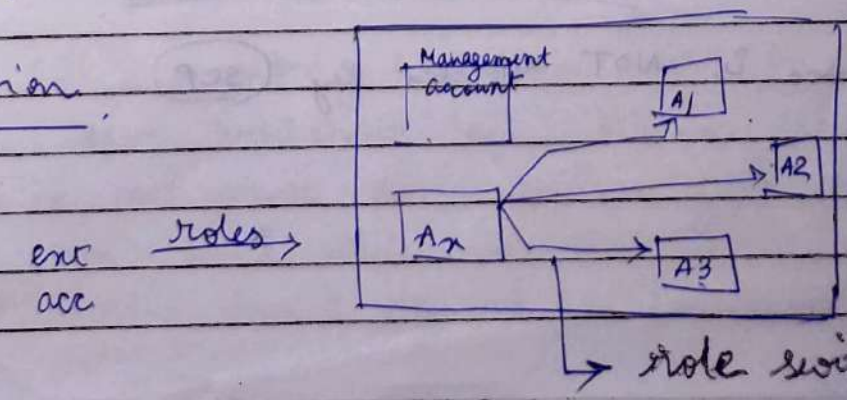
Temporary basis → something assumes the role for set time (SHORT TIME)

YOU BECOME THAT ROLE

External identities to assume this role = ID federation

For multi acc & cross account & for mobile sign in with twitter / FB / gmail etc. - i.e for mobile apps with millions of users

Organisation



We can federate to 1 account and then role switch.

Account A → management account
Account B → ext account

Now, (A) ~~wants~~ adds (B) into its organisation.
(A) wants to [ROLE- SWITCH] to (B)

(A) Steps in AWS Console (A)

(B) needs to create a role where it trusts the (A) account. Name of the role - "Switcher." (eg) for acc (A) using ID of "(A)" [because its for (A)]
Now in A
↳ switch role → account ID → (B).
Role name → Switcher
↳ By doing so, (A) assumes the role & A can switch to (B)

[Inside role policy should be admin access]

Name of the role - "OrganizationAccountAccessRole"

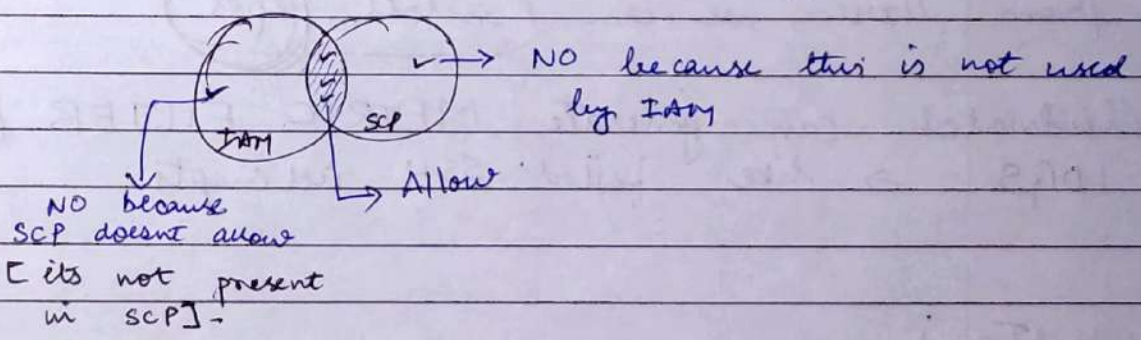
"Management" acc is NOT affected by (SCP)

~~Can~~ SCP can indirectly restrict acc root user & directly affect the account itself

SCP → important for larger & complex deployments
SCP → BOUNDARIES to an account
↳ CANNOT grant permission.
DEFAULT → Allow * for all

∴ It makes a deny list explicit architecture because every thing is ~~not~~ allowed by default

ONLY COMMON ALLOW services b/w IAM & SCP are allowed. Rest all are denied.



DEFAULT SCP → Full AWS Access → means NO restrictions

CLOUD WATCH

Metrics → CPU utilization, for on premises/AWS etc
↳ public service

Logs → collⁿ, monitoring & action based on log
eg - webserver logs, firewall logs etc

Events → if AWS service does something, → it will perform some other action.
↳ certain time of day → do particular action.

metric → CPU usage, network ~~IN~~ ↑/↓, IOPs

Alarms → Cloudwatch (CW) can take an action based on (Metric)

Alarm → OK.
→ ALARM → action [billing alarm].

- Certain stuffs are present in "amazon-linux-extras" repo.

Command - "amazon-linux-extras" → (it will list all packages available)

from there we can install epel

Cloudwatch can generate METRIC FILTER from LOGS → like failed SSH attempts.

CloudTrail

API calls / acts - logger (roles, users or anyone)
→ Default 90 days history (free).

Events

Management
creating, terminating VPC / EC2

Data
S3 objects access, λ function invoked

By default CloudTrail logs management EVENTS.

Trail will log only on region which it is created in

Trail → 1 region
→ all region trail

Global services ~~m~~ like IAM, STS etc log ~~their~~ Their events to us-east-1 (N. Virginia)

Cloudtrail → CW logs → for using metric filter
→ S3 for indefinite storage

Can also enable "organisational trail"

Exam POV

1. CT enabled by default → 90 days NO → S3 ×
2. Management Events → default

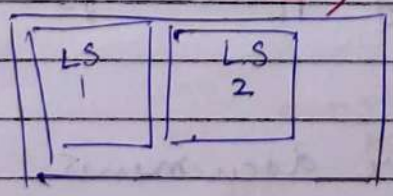
CT NOT REALTIME → min 15 mins from activity

3. Default → All regions enabled

CW

Log stream → log events from same source
eg from / var / log / messages
→ from 1 instance (specific source)

Log groups →



__/__/__

We define configuration on log groups - retention / permission & metric filters

Each AWS service will create their log group.

CloudTrail → we can either send that to S3
OR
can send that to CW for metric filter

TO check → Event History in CT → by default is 90 days

Even if we create a trail or not

S3

private by default.

Bucket policy → resource policy
↓
(*) Can allow / deny ~~from~~ other account as well (*)

For anonymous access - ONLY bucket/resource policy applies

(*) Block Public Access (*)

Final failsafe → only applicable for anonymous users

S3 Static Website
→ index & error documents
→ Endpoint is created

For static hosting :-

We need to
1. Turn off block all public access
2. Enable S3 static hosting
3. Implement bucket policy for all objects inside the bucket

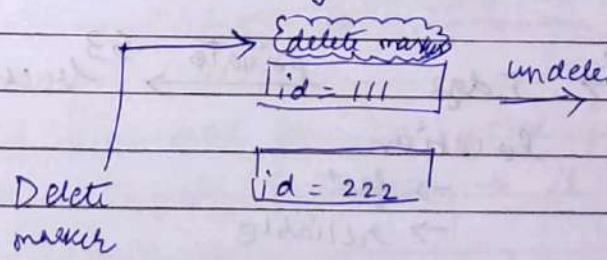
To use it with our own domain ⇒ Bucket name = domain name
[Simple routing].

Versioning

Disabled → Enabled → Suspend

once Enabled, can NEVER be Disabled

Deleting means using a delete marker



undelete with delete the delete marker. ~~thus ... de~~ ~~leting objects~~

MFA Delete (in versioning).

Enabled in versioning configuration.

- changing state i.e from Versioning enabled to suspend
- delete versions

MFA needed to do these things

__/__/__

MFA delete

- → Versioning must be on
- → Only be enabled with CLI, SDK, Rest API
- →

Needed for

~~prevent~~ for perma delete bucket objects

changing ~~ba~~ versioning status

S3 performance optimization:

→ Multipart Upload.

min data → 100 MB.

max no. of chunks → 10,000 [parts can fail but can be restarted].

→ Transfer Acc" (Bucket name ⇒ NO period).

My data $\xrightarrow{\text{public}}$ Edge location $\xrightarrow{\text{private}}$ S3 bucket

- → fast
- → reliable

ENCRYPTION

Asymetric encryption is expensive so parties will ~~gen~~ ~~send~~ exchange symmetric keys (asymmetrically) & then will do encryption symmetrically after successfully exchanging [symmetric key]

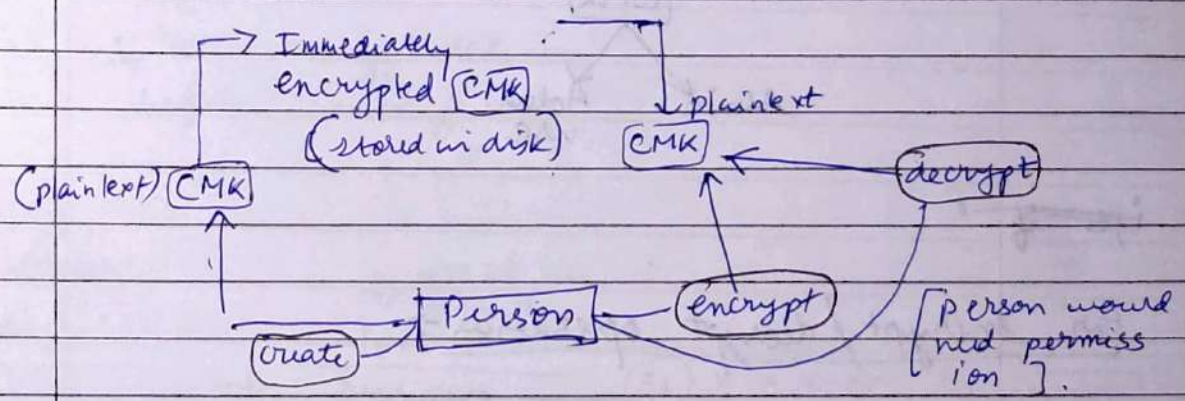
KMS

→ regional & public [AWS public zone]
→ Crypto operations eg - encryption, decryption,
FIPS - 140 - L2

KEYS never leave KMS.

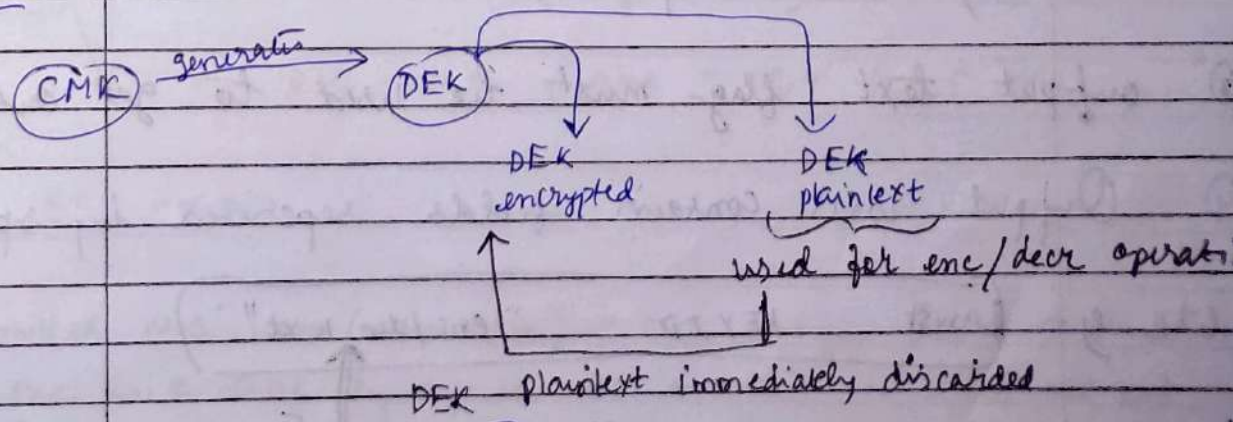
CMK [Customer Master key] - used by KMS for crypt action.
↳ LOGICAL → backing by PHYSICAL key material.

Encry/Decr → 4KB max



CMK itself is stored in encrypted format in disk.
During any crypto operation → it is ~~e~~ decrypted and then used

DEK



CMK can enc/dec DEK

__/__/__

CMK never leave the region & never leave KMS

CMK → AWS managed → ~~at~~ 3 yr rotation
CMK → Customer managed → opt. rotation (1 yr)

CMK portal. [for customer managed keys]
↳ We need to ~~allow~~ attach a resource policy i.e a KMS policy where we need to select

(Users & roles for Admin access to the key
Users & role for encrypt/decrypt with key.)

CMK
Admin — Actual use

~~i.e if~~

For encrypt/decrypt operation →

① output is always (base64 encoded) so need to decode it
② For encryption →
↳ key ID / alias is needed but for decryption, ~~manual~~ KEY is not needed (automatically taken by AWS).

③ output text flag must be used to get output.

④ Output will contain fields seperated by spaces

like, eg - (AWS KEY ID "enc/dec text")

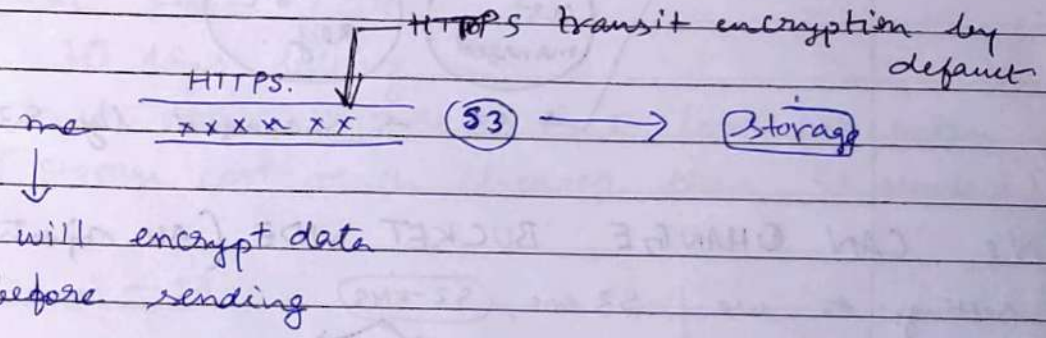
We need to query only this part [we dont need entire output].

We cannot immediately delete a key → DISABLE & Schedule key deletion min (7 days)

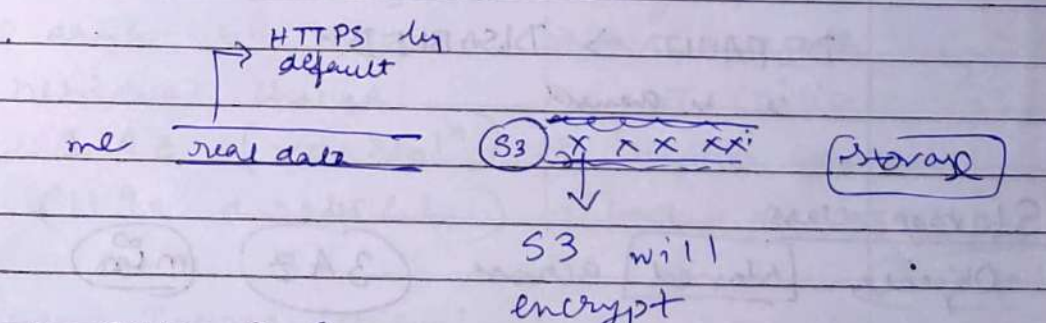
S3 encryption. (Object level).

1. Client 2. Server [both at Rest].

1. Client



2. Server. (SSE).



SSE
- → SSE - C (customer provided keys).
- → SSE - S3 (keys managed by S3).
- → SSE - KMS. (CMK managed by KMS).

• In SSE - KMS

Permitted KMS user (with CMK), CMK will gen DEK. DEK will be enc with CMK by default. As soon as it sees object, CMK will decrypt DEK, then DEK will encrypt object & then

~~DEK~~ plaintext DEK is "immediately discarded".
So if we want to decrypt, we need that CMK access

ROLE separation

SSE-S3 → AES-256

In S3 SSE → S3-KMS
- CMK cust managed
- AWS managed key
- S3 → managed by S3

We CAN CHANGE BUCKET WIDE (all objects) encryption settings to use | S3 enc, S3-KMS
- AWS managed
- cust managed

DEFAULT → DISABLED.
can be changed

Storage class
Objects stored across 3AZ min

① Standard 11 9s durability i.e 99.....9 %

Fees
- /GB monthly
- data transfer out from S3 /GB
- /1000 requests

objects are ~~re~~ served within milisecond of request ed → ∴ FIRST BYTE LATENCY = millisecond

② S3 - IA

→ min 3 AZ repl^n, same durability (11 9s)
same MD5 & CRC checks.
→ RETRIEVAL FEE / GB
→ min 30 days billing
→ Long lived data, imp data but less frequently accessed [storage cost much cheaper than S3 standard].

③ S3 - 1 Zone - IA

→ ~~min~~ less than S3 & S3 - IA
→ min 30 days billing.
→ data retrieval charge
→ ONLY 1 AZ (no repl^n)
→ Same (11 9s durability)

→ Long lived data
→ less accessed
→ NOT critical
→ replica copies.
→ backup of backup
(IF AZ fails)

④ S3 glacier - Instant retrieval.
→ cheaper storage
→ more expensive retrieval cost
→ min 90 days billing.
→ accessed once / quarter
→ long lived data with millisecond access.

⑤ S3 glacier - Flexible
→ COLD objects. Not ready to use
→ retrieval process. pay

Retrieval jobs

Expedite	Standard	Bulk
1-5 min (exp)	3-5 hr	5-12 hr (cheapest)

First byte latency → min - hrs

CANNOT make objects public.

min 90 days → eg archival yearly access.

(6) S3 Glacier Deep Archive.

→ Frozen state
→ min 180 days

Retrieval

Stand	Bulk
12 hr	48 hrs

1st byte latency = hrs to days

use → legal data, regulation, archival data, data that needs hrs to days.

(7) S3 - Intelligent Tiering (5 classes).

No retrieval fee but management fee. Known pattern changes.

1. FA
2. IA
3. Archive instant access
4. Archive
5. Deep archive

∴ IA, 1Zone IA → 30 days min bill (30×1)
Glacier instant,
Glacier flexible → 90 days " " (30×3)
Glacier deep archive → 180 days " " (30×6)

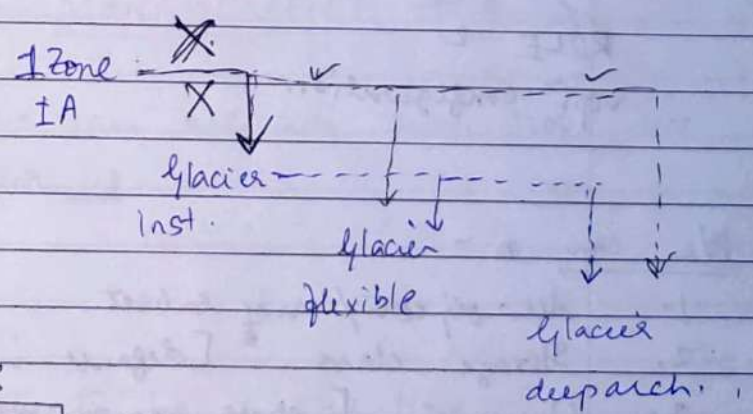
Lifecycle config.

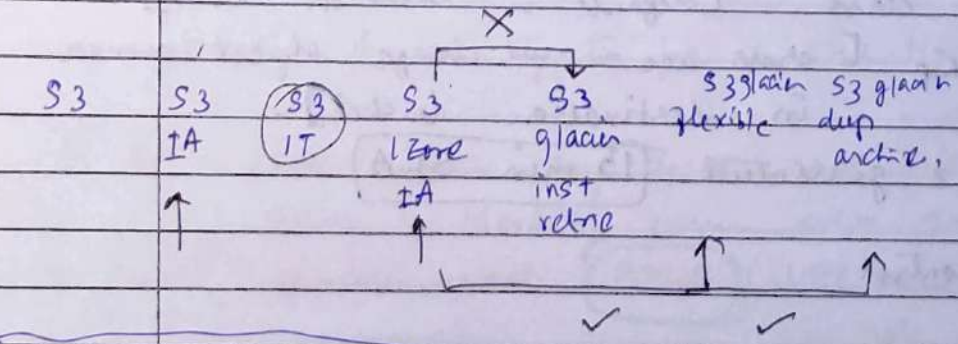
→ Transition action & expiration action.
[not based on how frequently data is accessed].

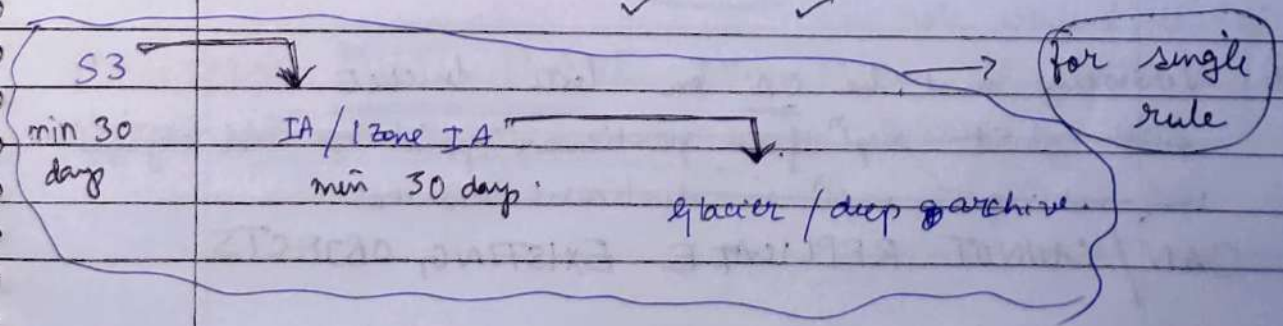
S3 std

IA

intelligent tier







__/__/__

S3 replication.

CRR (cross reg) — Soure bucket → destination bucket (diff region)

SRR (same region)

~~SSB~~

Same acc.

source —encrypted→ destination.

ROLE & repl^n configuration.

Diff acc

Source —encrypted→ dest^n

ROLE & repl^n config

bucket policy to allow ~~source~~ account

We can :-

1. All objects / ~~subset~~ subset
2. Storage class [default is same in both]
3. Ownership [cross acc - we change object owner in destination bucket].
4. RTC → guaranteed $\boxed{\text{15 min SLA}}$

repl^n time control

(*) Versioning must be ON on both buckets

(*) ~~will start repl^n from particular point → old objects will not get repl^n in destination bucket~~

CAN / CANNOT REPLICATE EXISTING OBJECTS.

→ Glacier & Glacier deep archive → cannot repl^n [cold & frozen storage]

DELETES ✗ not repl^n [by default but CAN be ENABLED]

SRR → Logs, Region resilience & strict data regulation.
CRR → Global resilience & latency reduction.

(*) In addition to Block Public access for bucket,

There is Account wide "Block Public Access for this ACCOUNT"
↓
Needs to be turned OFF

Practical:
Repl^n will be under ~~man~~ "MANAGEMENT TAB".

It basically a rule. We can use our own role or let S3 create one for us

PRESIGNED URLs.

Access private bucket objects for limited time.

(*) Can generate for any object (w or w/o access)
(*) URL perm = Identity perm who generated it
(*) DONT generate with ROLE (URL will stop as soon as role credentials expire)

IDENTITY MUST HAVE permission to access that object AT the current time of accessing the link

S3 Select & Glacier Select

On S3 side → SQL like statements used in S3 - SELECT.

happens in S3. Used for filtering.

↳ 400% faster

↳ much cheaper

Note Retrieval = retrieve = make available from cold storage

Transfer = actual download

Request = GET request made before download.

S3 Event Notifs

→ Notifs when (event) occur in (bucket)

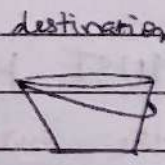
↳ notif can deliver to SNS, SQS & λ

Notifs for create (put, post, copy, multipart upload), delete, restore, repli^n.

So create, repl^n/restore, delete

C R D. (like crud).

ACCESS LOGS.



bucket & object are accessed on source → that accessing is logged to destination bucket

that log delivery will be done by "Log Delivery Group"
so it will need access to destination bucket by ACL

Networking

$2^n - 2 \geq 64$, $n = 7$

VPC Size

min = /28 max = /16

Sample architecture

	US-east-1 VPC	DB	APP	WEB
AZ-A	Reserved 10.16.0.0/20	10.16.16.0/20	10.16.32.0/20	10.16.~~64~~ 48.0/20
AZ-B	Reserved 10.16.64.0/20	10.16.80.0/20	96.0/20	112.0/20
AZ-C.	Reserved 128.0/20	144.0/20	160.0/20	176.0/20
AZ-x [future]	reserved.	-	-	-

↳ VPC = 10.16.0.0/16

/16 & we need 16 subnets, $2^4 = 16$ $16 + 4 = 20$

∴ subnet = 16 subnets of /20

∴ add 16 to 3rd octet

10.16.16.0/20
10.16.31.255/20
-1 10.16.32.0/20

Critical Setup:

→ Tenancy → shared (default) vs dedicated (expensive)

DNS → VPC + 2 i.e here 10.16.0.2

enable DNS Hostname → public IP instance get public DNS hostname

enable DNS Support → instance will be able to DNS ip i.e DNS resolution will be enabled

AWS reserved IPs

/20

10.16.16.0 → network ID

10.16.16.1 → ~~VPC router~~ VPC router [default gateway for subnet]

10.16.16.2 → DNS [every subnet].

10.16.16.3 → RESERVED.

10.16.31.255 → Broadcast

CANNOT use them

VPC setting	Subnet Setting
DHCP option set → Controls DNS, NTP servers etc.	IP allocation
we can attach but not edit	1. Auto assign public IPv4
	2. Auto assign public IPv6

~~sub~~ VPC has to be enabled with ipv6

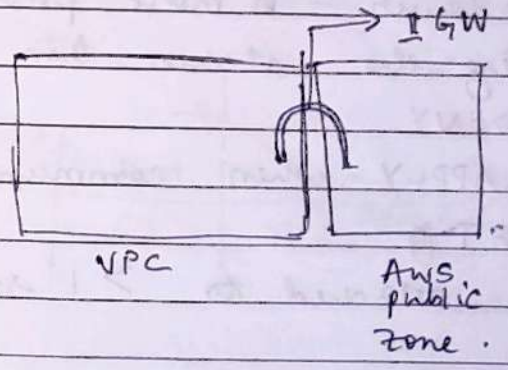
VPC routes

router = network IP + 1 address.

↳ routes traffic b/w different subnet

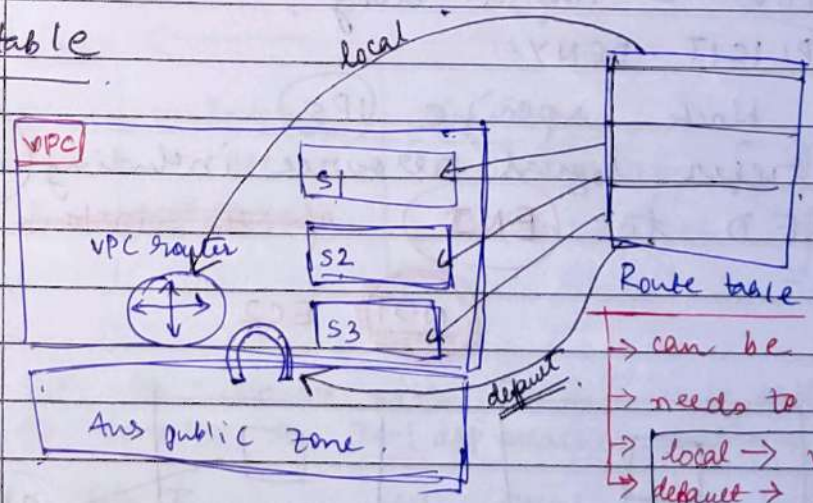
OS doesnt know about public IPV4, only pvt ipV4
OS knows about ipV6 because always public.

IGW



So IGW needs to be attached to VPC

Route table



Route table
- → can be many in VPC
- → needs to attached to subnet
- → local → VPC route
- → default → IGW

1 subnet = 1 route table

Finally we need to assign → ASSIGN PUBLIC IPV4 for public subnet.

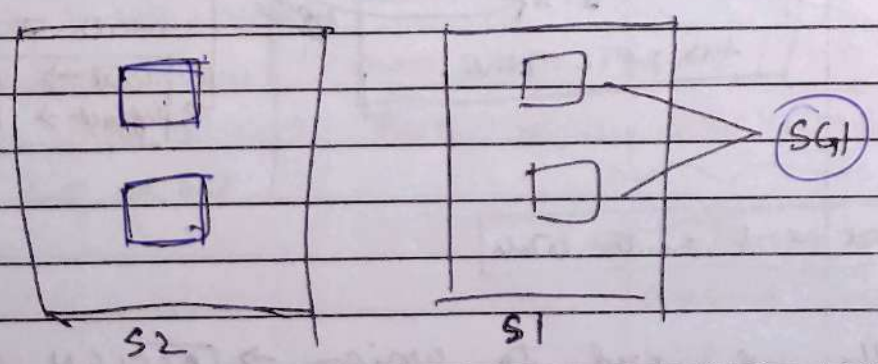
__/__/__

NACLs (stateless)

→ Inbound & outbound rules.
→ can allow / deny both inbound / outbound
→ Rule no = lower = more priority
→ * = if nothing else matches then *
= IMPLICIT DENY
→ RULES DONT APPLY when communicating INSIDE SUBNET
→ 1 NACL can be attached to > 1 subnet

Security group (SG)

→ Only ALLOW & implicit deny.
→ NO EXPLICIT DENY.
→ CANNOT block specific IPs
→ can refer logical resource including itself
→ ATTACHED to ENI ~~of the instance~~
NOT EC2



If resource on (S1) use Security group (SG1),
any other resource that refrences from (SG1),
the (S1) resource will be able to connect
if NACL ALLOWS

SG1

Protocol	Port	Source
TCP	80	0.0.0.0/0

SG2

Protocol	Port	Source
	69	SG1

Resource with SG1 will be able to communicate with resource with SG2 where SG2 is referencing SG1

i,e parent can connect to child

BUT on port 69 only

SG-1/app

Type	Protocol	Source
All	All	SG-1/app
TCP	TCP/1337	SG-2/web

these are not IPs

These are SGs

All Instance with SG-2 will be able to connect to SG-1 app attached instance

All Instance with SG-1 will be able to connect to each other &

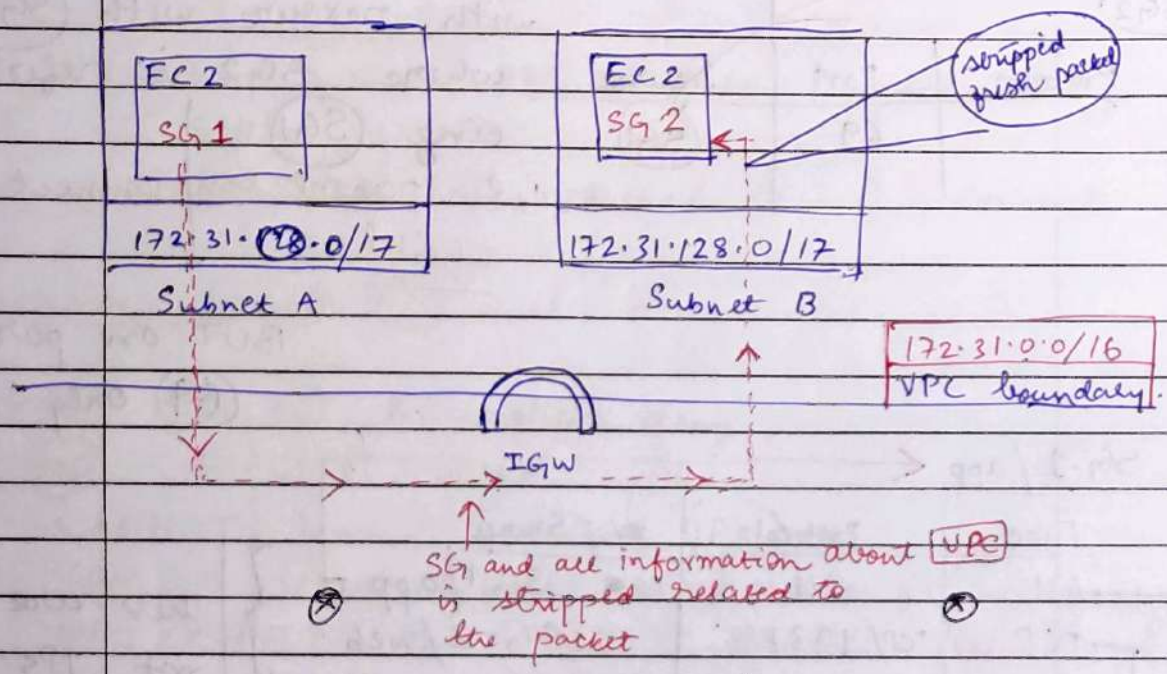
Flexible & Independent of IPs

Scalable & every web can connect to app if app refers the SG-2/web SG

SG concept

→ ~~SG has no knowledge of public IPs~~

→ SG only has knowledge of stuff ONLY inside the VPC



∴ packet originated from ENI containing SG1, is not known by ENI containing SG2 if packet crosses VPC.

∴ PUBLIC = IGW = Stripping of all VPC related info

IGW will give public IP to EC2. IGW is also responsible for Natting in public instances.

NAT

→ outgoing only internet access

→ Runs from public subnet [They need public ipV4 address from IGW]

↳ public ipV address is = ELASTIC IP ~~for~~ for (NAT)

→ AZ resilient NOT region resilient

↳ they are placed in public subnet (AZ)

So private subnet ⟶ route table default path ~~of~~ 0.0.0.0/0

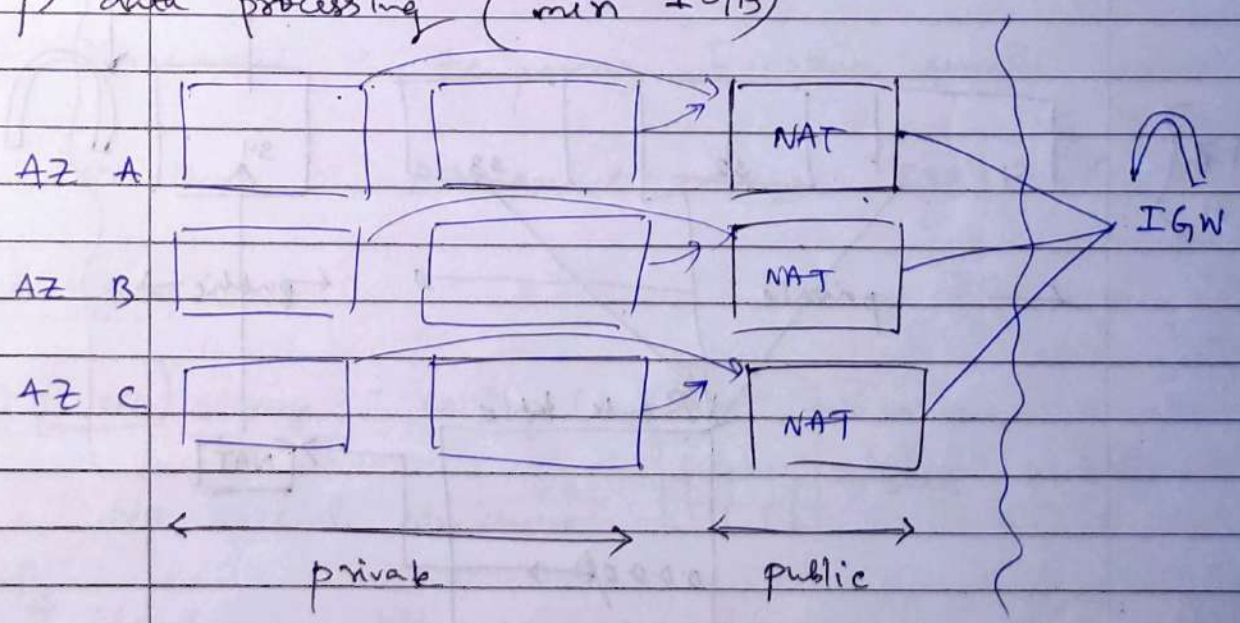
↓

POINT TO (NAT gateway.)

Cost

→ hourly (min 1 hr)

→ data processing (min 1GB)



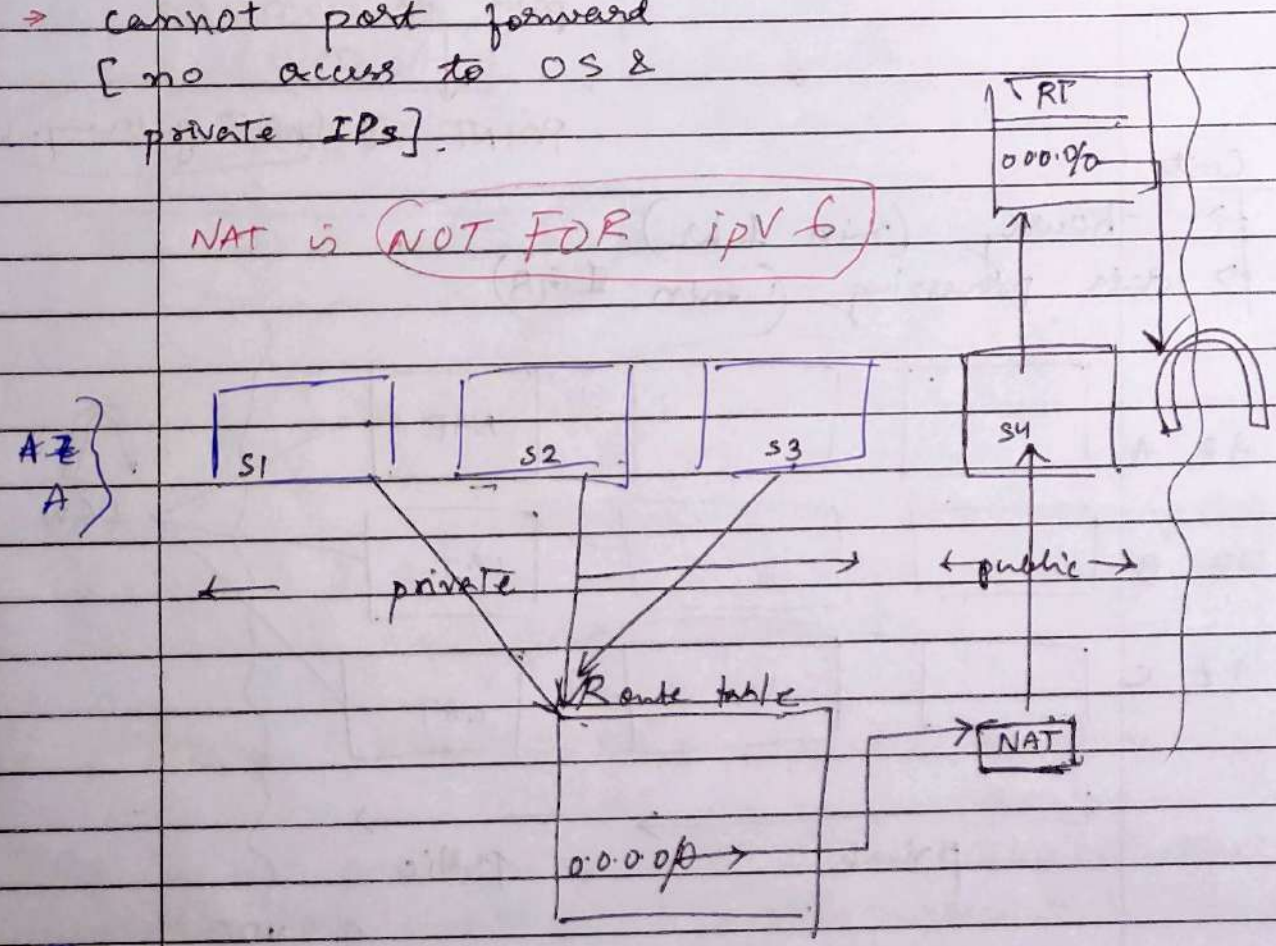
So every AZ needs to have 1 NAT in public subnet for other private subnets in same AZ to connect to ~~NAT~~ NAT and then to (IGW)

↳ this is imp for region resilience because NAT Gateway is NOT REGION RESILIENT

__/__/__

NAT Gateway	NAT instance (just EC2 instances)
High bandwidth (45 GBPs)	→ EC2 need to disable source-destination checks.
Highly available	
Highly scalable	
→ Costly (hourly + data transferred)	→ cheaper (depends on EC2 instance)
→ only be controlled with NACL	
→ cannot port forward [no access to OS & private IPs].	

NAT is NOT FOR ipV6.



EC2

1. General purpose - [Default] → diverse workload → equal resource ratio (CPU, RAM, HDD/SSD).

2. Compute - [more CPU] HPC (high perf comp), ML, gaming.

3. Memory optim → [more RAM] in memory DB, DB workload

4. Acc^n compt → GAMING, GPU, FPGA

5. Storage → [high IOPS]. sequential R/W - I/O, data warehousing, Elastic Search, Analytics [very high Input / Output].

Storage types

1. Direct (local) = instance store vol attached directly to EC2

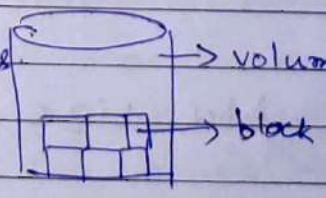
2. Network = over network (EBS)

3. Ephemeral = temporary, [instance store].

4. Persistent = permanent storage [EBS].

Category

1. Block storage - coll^n of blocks



No special structure

↳ used by filesystem & so they are bootable & mountable

by path / → root
C: → c drive.

2. File storage. as a share. MOUNT ✓✓✓
~~BOOT~~ ~~XXX~~

__/__/__

∴ file storage already has a structure [filesystem].

Object storage → flat structure (no structure).

Mountable ✗ ✗ [S3].
Bootable

Very scalable
& simultaneous access.

Performance

IO (block) size. × IOPs = Throughput

□ □
16kb 64kb.

eg 16kB × 100 IOPs = 1600 KBps

Increasing block size = IOPs will reduce

• EBS (Block ID - block storage).
unique.

EBS ~~Dec~~ → AZ only. 1 AZ → AZ resilience

EBS ⇒ PERSISTENT. IT follows the EC2 instance & can be attached & detached. Not linked to EC2

Can take EBS snapshots & then migrate it from 1AZ to another AZ in same Region.

EBS snapshots will be stored in S3 thus EBS snapshot will become Region resilient

Bill - GB/month

Copy snapshot into another REGION

EBS type

GP2 (General Purpose SSD)

Baseline = 3 times the vol (3 times X 1 GB)
refiling rate

∴ for 100 GB, the refiling rate = 100 X 3
= 300 IO
credit / sec.

However

(1) so for size upto 1TB, BURST = 1000 X 3 = 3000
IO credit
per second
∴ 3000 IOPs per sec for ≤ 1TB

(2) For size ≥ 1TB ≤ 16TB, baseline is = burst
i.e
min 3000 IOPs at all
time ~~E credit bucket refilling speed const~~

But max IOPs ≥ 1TB & ≤ 16TB = 16000 IOPs

GP2 → boost vol, low-latency, dev & test env

GP3 (16GB – 16TB) [more economical]

No credit bucket arch.

Normal (standard IOPs) = 3000 IOPs.

Upto 16,000 IOPs [for extra money]
& 1000 MBPs → still more economical.

Provisioned IOPs (IO1/IO2)
↳ consistent low latency & jitter (independent of volume).

max 64000 IOPs (IO1/2)	1000 MBPs
max 256,000 IOPs (Block Express)	4000 MBPs (Block Express).

here - size doesn't matter & IO/1/2 are used

if we want - consistent high performance, low latency, NO SQL DB etc. (all in less size)

HDD based (NOT FOR IOPS intensive) ~~BOOT~~

Throughput opt (ST1) [500MBPS]	Cold HDD (SC1) · [250MBPS]
→ Throughput intensive, frequent access & see SEQUENTIAL ACCESS & big data, data warehouse, logs streaming	→ CHEAPEST storage. INFREQUENTLY access, few scans/day.

Instance store vs EBS:

Instance store when > 260,000 IOPS needed.

Snapshots

~~then~~ → Restored Lazily.

Sol^n = FSR = Fast Snapshot Restore (instant restore).
(50 max FSR per region).

We can force read of every block manually to get the performance

Charge = GB / month

Snapshot → incremental & DATA USED, NOT (total allocated)
↳ 1st is always full.

EBS Encryption:

→ Vol uses 1 DEK & snapshot of that vol will also use that same DEK.

→ CANNOT CHANGE FROM ENCRYPTED to NOT Encrypted.

→ OS NOT AWARE (no performance loss)
(AES 256 - KMS)

__/__/__

SNAPSHOT will be stored in SAME AZ. When creating a vol → we can choose different (AZ)

Can set ACCOUNT WIDE EBS encryption for current (REGION)

Networking

ENI

EC2

→ 1 primary private IP (from subnet).
private DNS for 10.10.0.0 = ip-10-10-0-0.ec2.internal.

→ ≥0 secondary pvt IP

point to private IP inside (VPC)

→ 1 Public IPv4, public DNS ec2-3-89-165-200.....
other place = public IP

→ 1 EIP per private IP
(*) removes public IP & EIP becomes public IPV4

→ Source/Destⁿ check

→ Sec group (≥1)

ENI
Secondary
↓
Same AZ as EC2

EIP can be associated with Private (IP), and private IP can be associated to ENI so ENI will have :

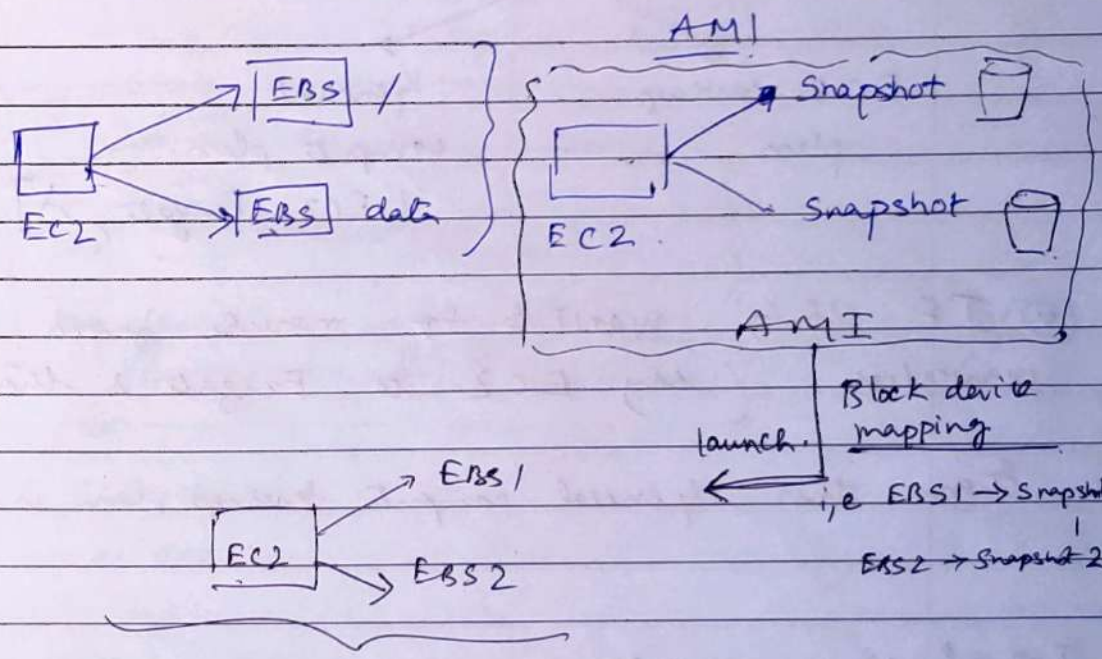
ENI → EC2

EIP	–	primary private IP
EIP	–	sec pvt IP
EIP	–	... pvt IP

→ (totally optional)

AMI

Regional & Unique ID for each region



EBS is newly created from snapshot

AMI itself doesnt contain those EBS vols. It just refers to their snapshots

Purchase Options:

Standard reserved

scheduled reserved [duration & time windows, sales analysis]

Capacity reservation
- → Region [say 40 EC2 in IAD].
- → Zone (1 AZ) [say 10 EC2 in US-east-1a]
- → On demand [say 5 EC2 in any AZ]. cap reserve

__/__/__

Savings plan.

fixed hourly commitment (1 or 3 year)

- EC2 savings plan
- General compute plan. [EC2, Fargate, λ].

(*) IF ORG. WANTS to moves from EC2 to serverless. (say EC2 to Fargate & then to λ]

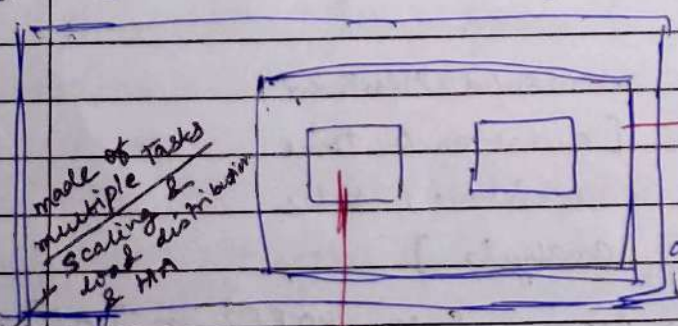
then this General compute savings plan is useful.

Termination protection.

USERS need to have permission to ~~use~~ use

* "DISABLEterminationAPI"

ECS (Container service) (EC2 mode & Fargate mode)



→ Task definition. (multiple ~~a~~ container def^n. App can be made of multiple container. This contain overall app info)

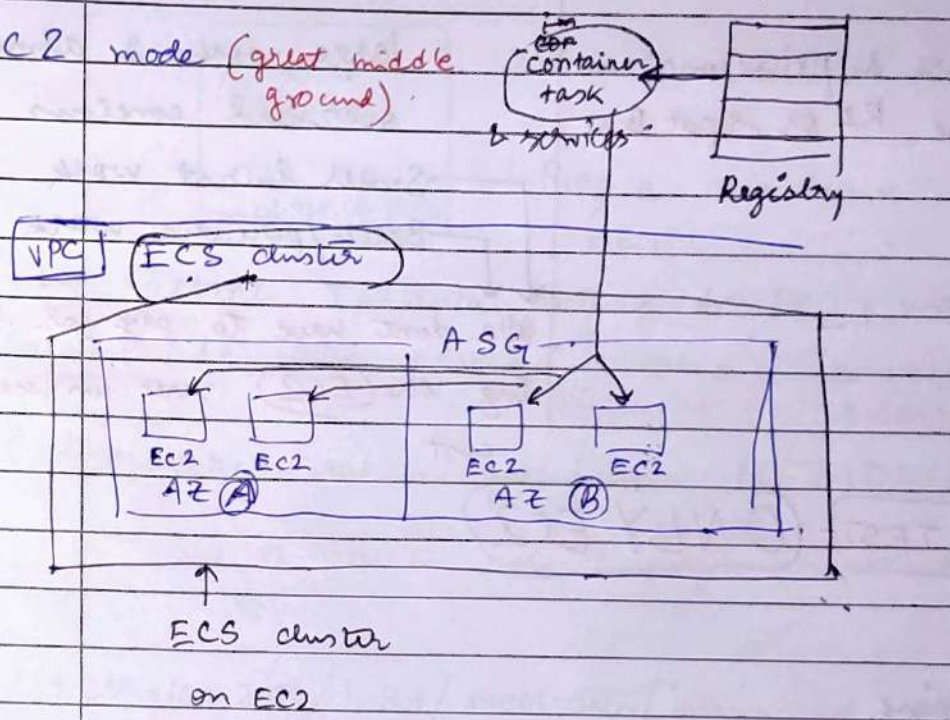
Service def^n
how we want task to scale

Container definition
where my image is, port (its a link to my image)

Contain CPU, Networking Roles that containers will use

Cluster type

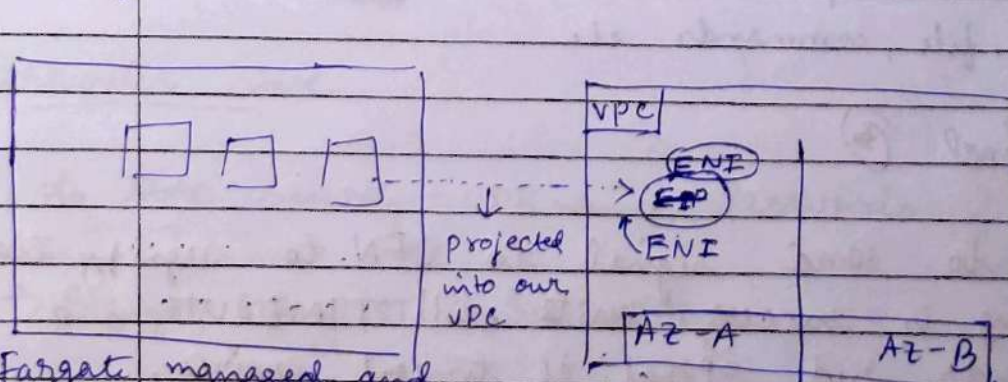
1. EC2 mode (great middle ground)



ECS will deploy the container from registry into EC2 & those will be termed as instances running container

Usr — manage EC2, pay for all EC2 (regardless of EC2 is running container or not). Can use reserved instances & spot instances.

2. Fargate



Fargate managed and shared infra

__/__/__

ECS - EC2	ECS - Fargate
1. large work & price concious ↳ can use RI or spot Inst	large work & admin overhead concious Small burst work Batch/periodic work We dont have to pay for running the EC2 ∴ no unnecessary cost

QUICKLY TEST - ONLY EC2

BOOT strapping

→ Instance Launch - run only 1ce.
→ Run as ROOT
→ No checks made by EC2 → can pass anything

user data stays in metadata

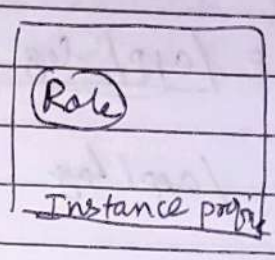
→ User data needs to be BASE64 encoded

- cfn-init (/var/log user data content).
  helper script & installed on EC2 : package, user, groups, file, commands etc.

cfn-signal

A way to send signal to CFN to signify that bootstrapping is success. Waits till TIMEOUT and then sends red signal if timeout expires.
Complete stack if signal is NOT RED

Instance profile

1. 

Wrapper around role

1 Profile can have only 1 role.

2. We attach Instance profile to EC2, not role.
3. Credentials are never invalid [always fresh from STS].
   ↳ they are taken ~~for~~ from METADATA.

CLI → need to creat role & profile separately

169.254.169.254/latest/meta-data/iam/security-credentials/ROLENAME

All credentials

hierarchy → env var ①
↓
· aws/creds ②
↓
· aws/config ③ → container cred ④ → instance profile. ⑤

Parameter store

→ to store securely - DB names, Passwords, full configs etc

→ Stringe, Stringlist & SecureString.

→ encrypted text with KMS

__/__/__

SSM parameter store

it accepts hierarchy structure = /app/ log

so it will show /app & /app/log

-> aws ssm get-parameters --names <parameter name>

-> can also get parameter by path

say /hi/hello
/hi/bye

so

/hi

path will list 2 other

/hello /bye.

Secure string get parameter

--with-decryption flag for SecureString

Dedicated host

Cannot run = Windows AMI
NO RDS & NO Placement groups.
Mainly for core & socket ~~licensi~~ licensing & compliance strict
can be shared

Enhanced Networking.

uses SRIOV. NIC is virtualization aware
Consistent low latency.

EBS optimised.
→ separate network for data & EBS storage
added ~~sto~~ dedicated capacity for storage networking

R53

Zone = DB for domain like google.com

NS $\xrightarrow{\text{hosts}}$ zone file.

Zone file - R53 ~~at~~ gives 4 Name servers.

In a VPC, VPC + 2 IP is the R53 resolver
→ used for querying.

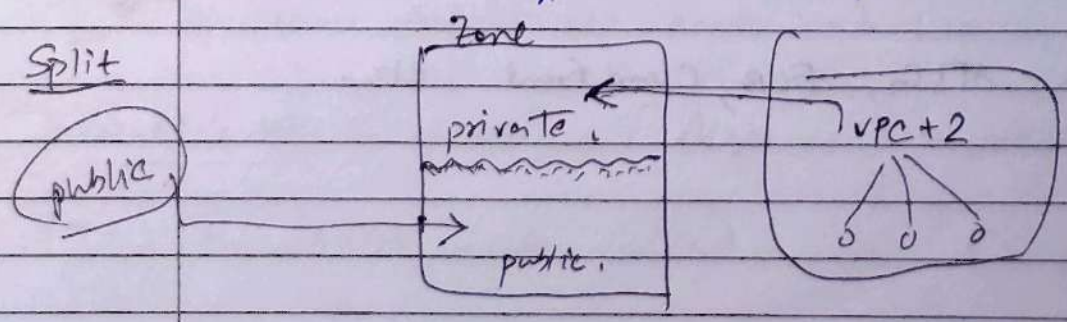
Private hosted zones

Not public. Only access in VPC

Split-view = public (internet) + private (in VPC) → with same zone.

we need to associate our VPC with private zone

Split



R53

CNAME - A name that points to raw domain

eg if raw domain is xyz.io,

CNAME = www.xyz.io → xyz.io

But xyz.io → abc.io ✗

raw domain cannot point to other things in CNAME

so, for ELB that gives only DNS name, we cannot point raw domain to ELB

so ~~xyz.io → ELB~~

SOLN = ALIAS → ONLY FOR R53 domain

NAM → AWS resource.

both naked domain & normal record

ALWAYS CHOOSE ALIAS, not CNAME (general case)

eg used in API G, ELB, Cloud Front, etc.

Routing (R53).

Simple routing

A → 1.2.3.4
A → 1.2.3.5
A → 1.2.3.6
. . . .

Single record can have multiple values.

Client gets one random value.
NO health checks (simple).

Health checks (separate from R53).

→ Located globally.

→ Checks every 30s or 10s (cost)

→ TCP, HTTP/HTTPS checks and string match.
↳ Response body must contain the ~~body~~ string else health check fails

3 types
→ endpoint
→ cloudwatch Alarm
→ calculated (checks of checks)

So health checkers are globally distributed & if 18% of them report the domain as healthy, then health check is healthy. (Also can create SNS topic if health check fails)

② Failover Routing.

www → EC2 [Primary] Zone

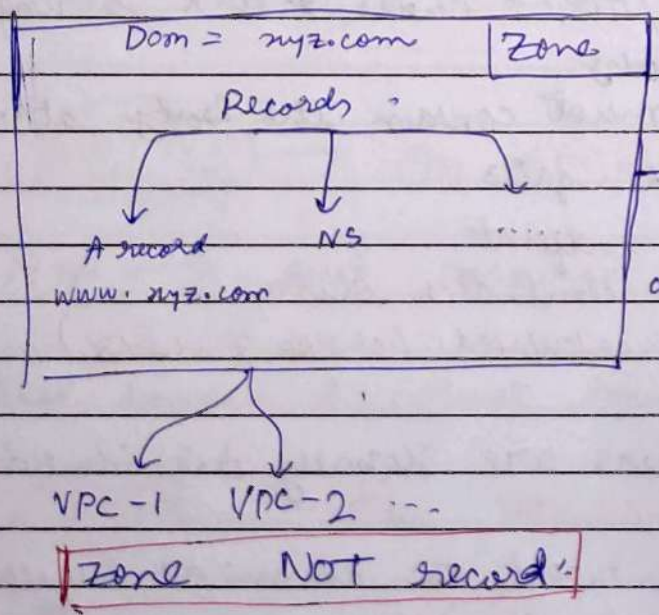
www → S3 (offloading) [Secondary]

multiple records.

→ if primary passes check then ✓

if primary doesnt, then routed to secondary.

We need to create health check for primary record only. Just need to mention (S3) bucket for secondary route



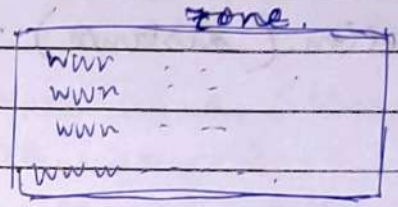
→ Zone can be private & associated with 1 or >1 VPC

so Zone NOT record.

When we create record we select a routing policy

Inside routing policy, we select a health check type

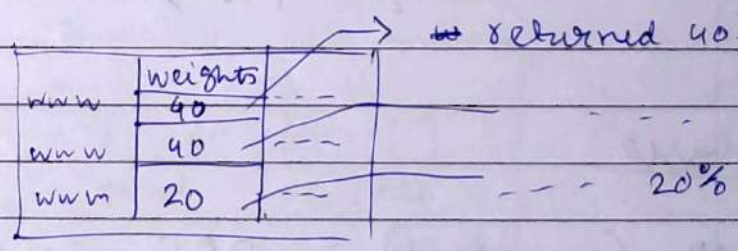
③ Multi Value routing



so many ~~healthy~~ health checkable (8) IPs are returned (multiple record health check) & client chooses at random

↳ CANNOT rules ELB

④ Weighted Rt



If record is unhealthy, it is skipped & new one is returned.

⑤ Latency Rt

Performance & user experience. Infra with lowest latency. (based on REGIONS)

⑥ Geolocation Rt.

NOT LOWEST LATENCY. It is based on my location.
Regional content (State, Country, Continent). or
Default record (for all rest).

__/__/__

(7) Geoproximity.

Routing based on location (distance) + bias for a location.

R53 interoperability.

domain hosting → hosting zone files on (NS)

R53 does both register + domain host
↓ (yearly fee) ↓ (monthly fee)

Databases.

ACID (consistant) → RDS (relational)
limited scaling

BASE (Availability) → maybe consistent
highly scalable (Dynamo DB).
→ no sql style

NO SQL + ACID = Dynamo DB.

Database on EC2 (bad idea).

→ OS level access
→ ROOT level access
→ version that AWS doesnt provide: (very specific version).

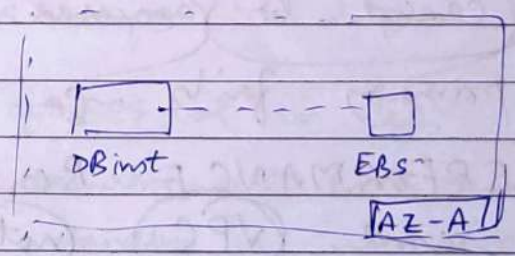
~~Bad ~~ imp.

RDS service

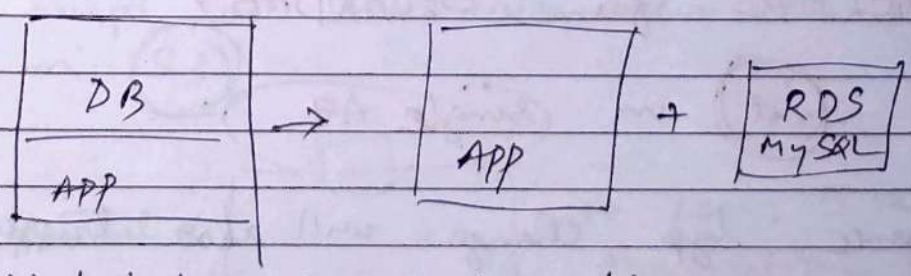
→ Database server as a Service (DbaaS).
→ Managed DB instance.
↳ MySQL, MariDB, Postsr..., Oracle, ... even Aurora.

DB instance.
We have to use (CNAME) to access DB instance

Type = db.m5, db.r5, db.t3
↓ general ↓ memory ↓ burst

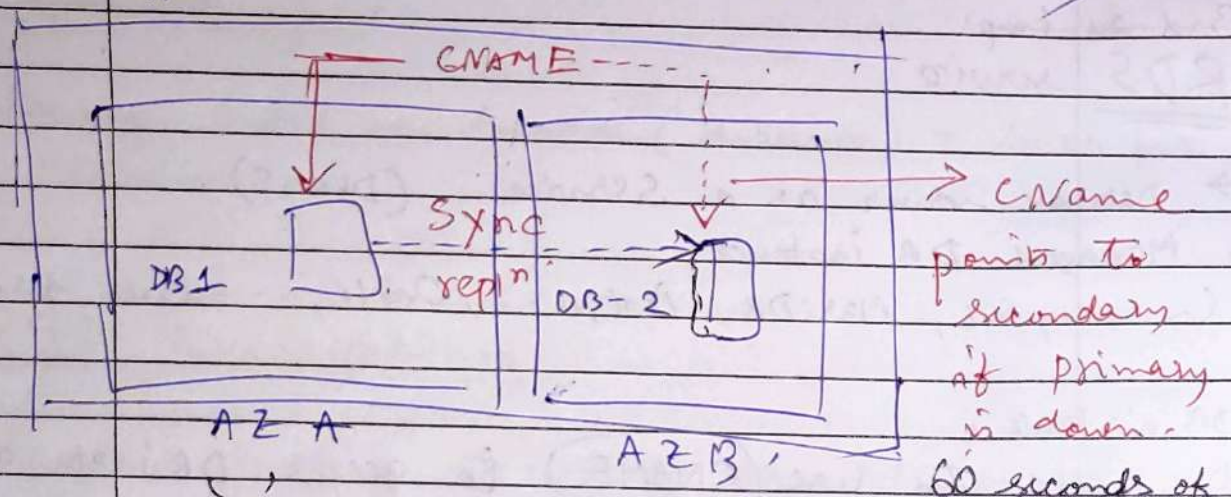


~~① create DB subnet group~~



1. e in original instance → create sqldump.
2. Use that in mysql command to load it on RDS with cname.
3. change wp-config.php file ~~to~~ to point ~~to~~ DBHOST to CNAME

MULTI AZ (for HA NOT fault tolerancy)



Almost '0' lag

It provides high Availability.

* MULTI AZ = NOT FREE (always constantly).
Secondary cannot be accessed directly
Only if primary fails. So
NO PERFORMANCE Benefit

* Inside same VPC not cross VPC

* BACKUPS from SECONDARY
but not in single AZ

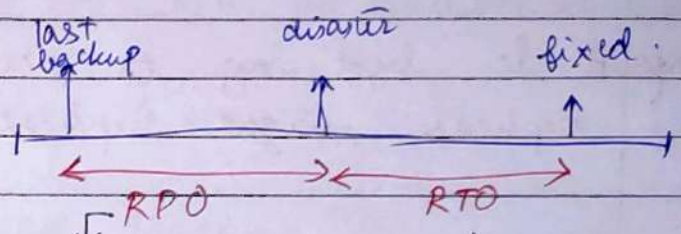
* Instance type change will also trigger failover

SYNC repl^n in same REGION

NO R or W oper^n benefit

RDS backups & restores.

RTO & RPO
↓ ↓
Time point



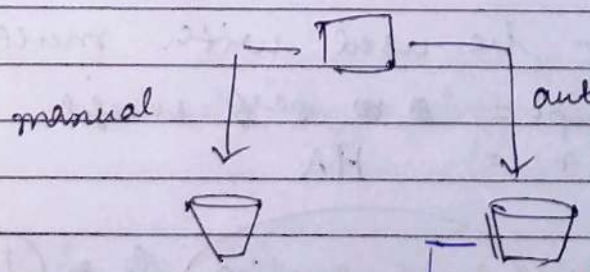
RPO ↓ amt. of max data loss. lower = ↑ exper

RTO ↓ time B/W DR & full recovery. lower = ↑ expensive

Automated backups
Manual ~~backups~~ snaps.
] AWS managed S3 bucket region resilient

snaps = snaps of EBS vol in DI.
single AZ = primary use kara hai (perf loss)
multi AZ = standby is used (no perf loss)

Manual snaps will live even after RDS DI is deleted, in (S3)



so just (5 min RPO)

to make these perma
→ final snapshot

→ 0 day retention to 35 day retention.
→ on DI delete, they will still expire on retention period (0-35 day)

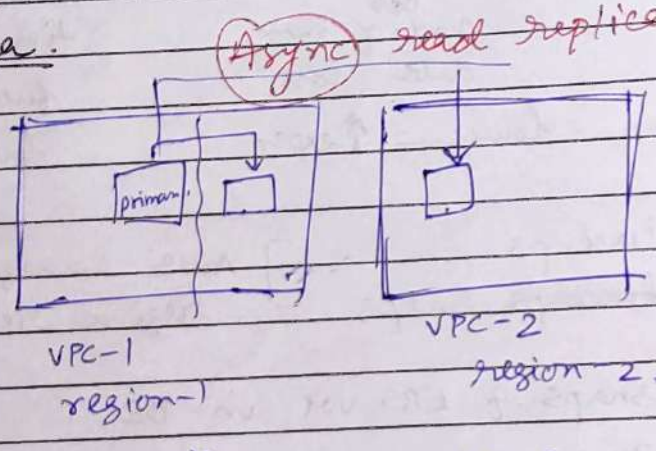
Restore

→ new DI with new CNAME

snapshot is best way to prevent corruption because even replicas might replicate same corrupt DB.

Read Replica.

ASYNC



cross region rep^n async read

read perf benefit

for 1 DI → 5 read repl

read replica can be used with multi AZ

so read repl = perf benefit
multi AZ = HA

Read replica of read replica but LAG

only read operations NO write.

ALMOST RPO (& read replica can be promoted to R/W in case of failure) with very less RTO so quick recovery.

Still need backups & snapshots because with RRs, data corruption can persist

RR can be promoted to R/W on failure.
(Read only)
RR until promoted. (irreversible).

Scale out read scaling.

Restoring → new instance
point in time recovery.

Encryption:

In transit = SSL/TLS in RDS.
Also EBS vol encryption - KMS.
storage, logs, snapshots & replicas are encrypt

Cant be removed once enabled.

RDS Ms SQL & RDS Oracle → TDE
transparent data encⁿ

Enc - handled by DB engine itself, not host.

ORACLE → Cloud HSM [very strong enc when customer dont want to trust AWS].

IAM Authent^n

IAM policy

RDS local DB user ⟷ IAM role/user

user can authenticate by IAM role/user

If policy allows, they can login to ~~RD~~ RDS DB user without using password.

(*) IAM usr/role policy is AUTHEN, NOT (AUTHORIZATION)
↓
(controlled by DB engine)

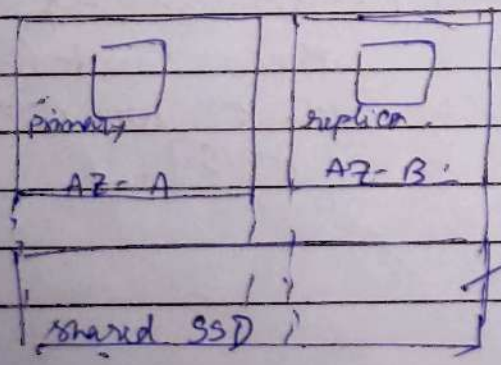
AURORA

→ Uses (cluster)

→ primary instance + more replica

→ (*) replica can be used for both Read and HA (*)

→



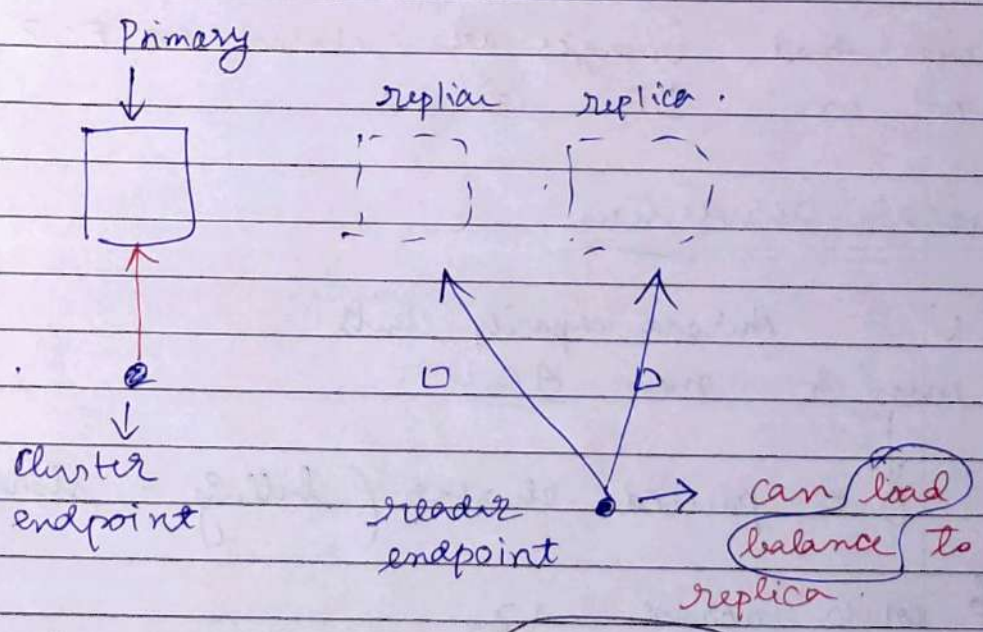
synchronous replication of 6 replica across AZs

Storage resilience is much better in Aurora.

Aurora can have 15 different replicas so quick recovery

high IOPS & ↓ latency (SSD).
storage → what's used → billing.

billing is based on highest storage watermark so freeing up space will no affect

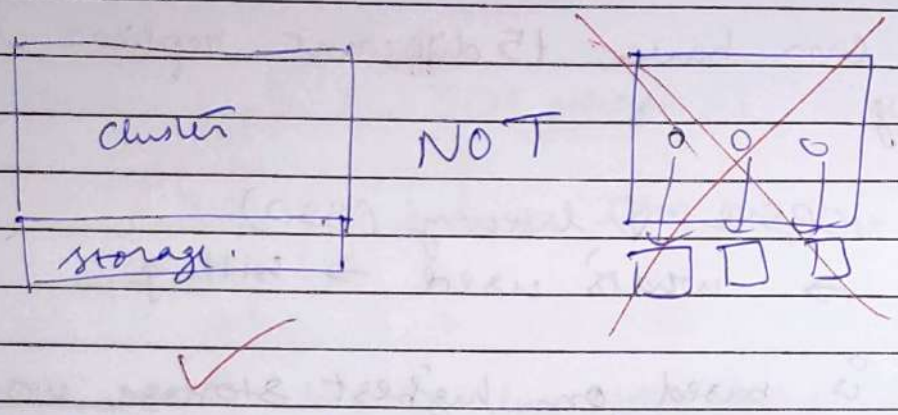


Restore will create brand new cluster

- BACKTRACK - restore SAME cluster to previous point in time.

Fast clone - make new database very fast but point to same storage and only create storage for difference (incremental).

Storage is for entire cluster, not instance



can set write & read rep' in different region

Remember that images are stored on EC2 locally & not on DB.

Aurora Serverless

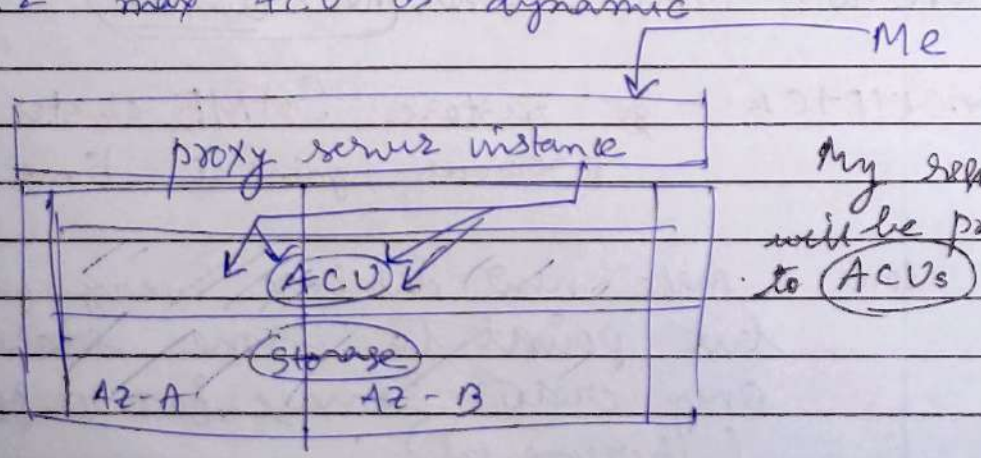
ACU = Aurora capacity units

min & max ACU

0 = paused cluster (billing = storage)

6 copies across AZs

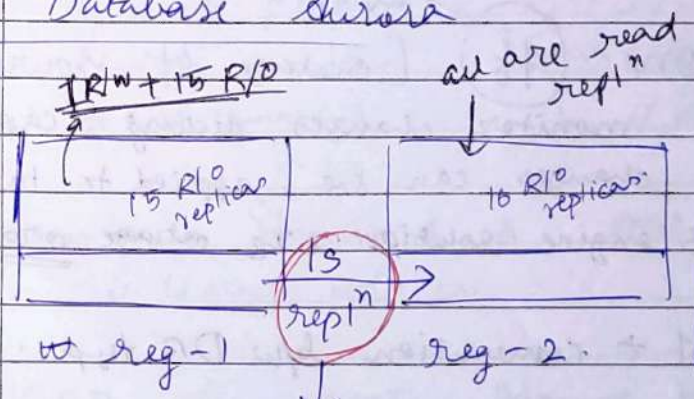
min & max ACU is dynamic



Use case

1. Infrequent ~~of~~ access apps.
2. New app
3. Apps whose DB load we dont know
   Variable work, Unpredictbl
   Dev & test DB , Multi tenant

Global Database Aurora



at STORAGE level.

→ ~~Glob~~ Cross region DR
→ Global low latency
→ 1s or less b/w primary to secondary region

Multi Master Write (*)

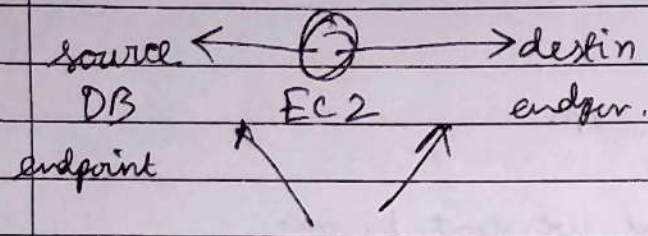
All ~~instc~~ DI are R+W enabled

No ~~A~~ Load balancing endpoint

Fault TOLERANT !!!!!! Fucking imp.

DMS migration service.

Managed service.



one of the 2 endpoints needs to be in AWS

Full load → one off → everything at once. [outage of source]

Full load + CDC → monitor changes during. captured changes can be applied to target

CDC only → DB engine solution like ~~orage~~ oracle.

Schema Conv. Tool (SCT) → conversion b/w DB type.

from on premise to AWS. [NO DOWNTIME migration].

→ on premise MS SQL → RDS MySQL

→ on premise Oracle → Aurora

For TB/PB scale, use Snowball + DMS

~~sto~~ change DB ~~from~~ and extract it locally with SCT (Schema Conv Tool)

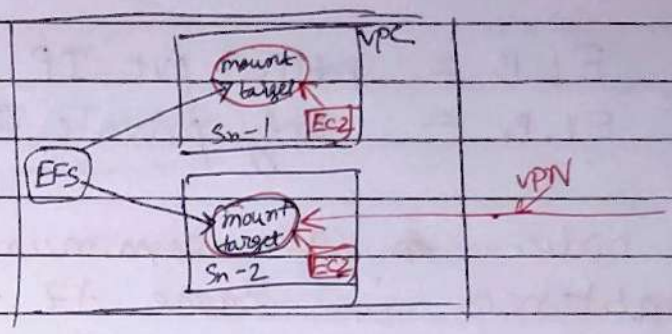
→ [we are changing it to generic file format]

DB engine change → only use case of SCT

Then store in Snowball & send it to AWS

EFS

Linux only.

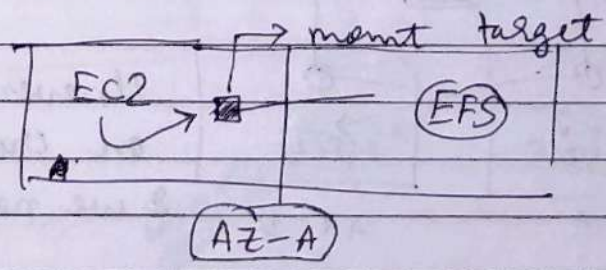


Modes - General purpose & Max I/O
(latency sensitive eg webservers) (big I/O but ↓ latency like & // workloads)

Storage class → Standard & IA
& Lifecycle policies.

We NEED to create Mount target in same AZ if we are trying to connect to EFS from EC2 in same AZ

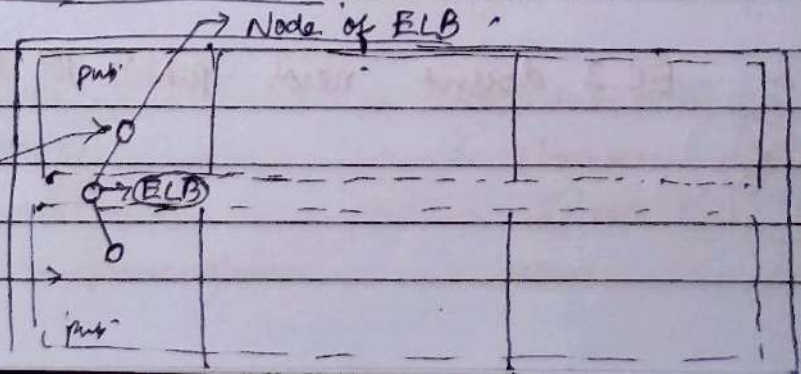
so



∴ Mount target in each AZ

Elastic Load Balancer

Select AZs and select 1 Subnet in that AZ



ELB = A record that points to individual nodes.

Nodes are ~~in~~ HA

Public ELB = public + pvt IP
private ELB = only private IPs

ELB nodes ~~in~~ can communicate with other instances in same AZ - be IT public or private subnet

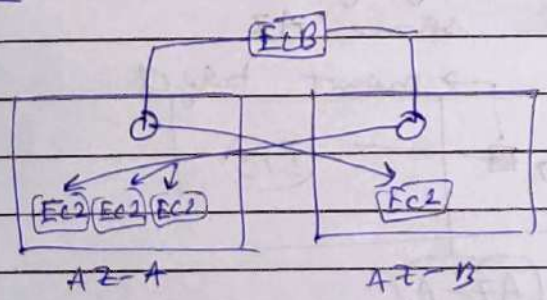
ELB needs ≥ 8 free IPs.

→ /28 = $2^4 - 5 = 16 - 5 = 11$ ✓
/27 = $2^5 - 5 = 32 - 5 = 27$ ✓

→ min recommended by AWS but /28 is also correct

Cross zone LB

enabled by default for ALB



Uneven distributi on cross AZs & we need to fix

Internet facing LB can ~~com~~ connect with both public & private EC2

& EC2 doesnt need public IP to ~~commun~~ work with LB

ALB vs NLB-

ALB → health check

→ layer 7 (HTTP & HTTPs.)

→ No other protocols.

→ NO UNBROKEN SSL connection.

connection will terminate on ELB & new connection will be created

→ ALB must ~~nee~~ have (SSL certs for HTTPs)

→ slower than (NLB)

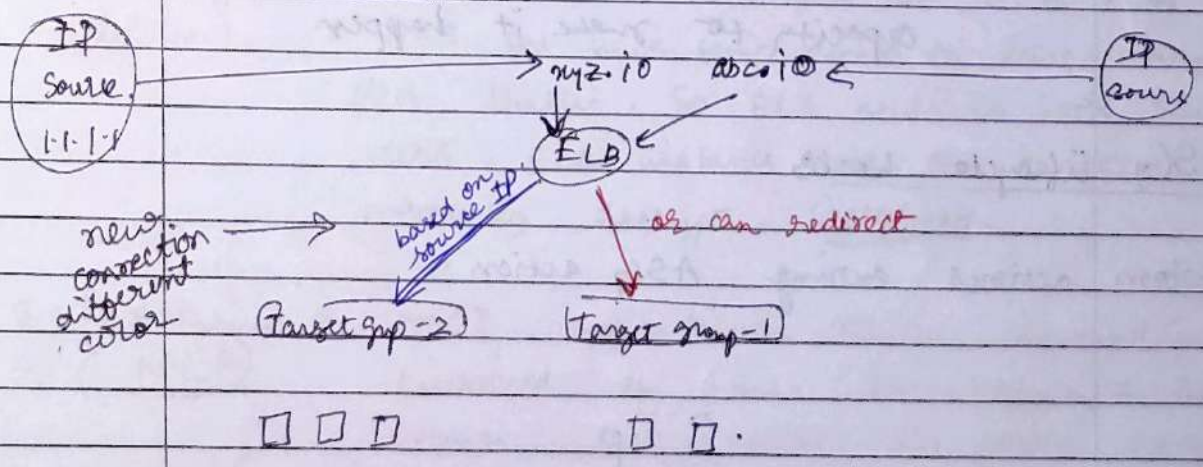
they have rules (what to do with the connection).

priority order.

Default = catch all.

Can use condition = header, method, query string, IP src etc.

Action = can forward request, redirect, fixed-response etc



NLB

→ TCP, UDP, TLS

→ No HTTP(S)

→ FAST AF

→ SMTP, SSH, game servers

→ Static IP for whitelist

→ Unbroken encryption.

→ Private link to provide service to other (VPC)

(LC) (LT) __/__/__

Launch Config & Launch Template (not editable)

Allows to define config of EC2 in advance

LC = no versions. LT = versions (new) ← (*) recommend

LC = used by ASG
LT = used by ASG + can launch EC2 from console.

Auto Scaling Group policies.

Manual + Dynamic

Simple scale = in alarm state. Static amount → same action for extreme variation

(*) Step scale = step adjust; upper & lower bound (more flexible) → recommend

target track = I want 50% CPU on average. Adjust capacity to make it happen → add instance.

ASG lifecycle hooks.

Custom actions during ASG action.

pending → in service.

wait → proceed

→ do action like load some data

terminating → terminated

wait → proceed

→ maybe backup, or purge logs.

ASG health check.

1. EC2 (~~default~~ → all except reboot = unhealthy.

2. ELB → elb is healthy & passing ELB layer 7 health check

3. Custom → external tool.

grace time = 5 min (default) ← [~~at~~ health check will start after that time].

↓ bootstrapping & ~~to~~ other changes needs to be done within ~~this~~ time.

SSL offload & session stickiness.

1. Bridging (Default) → ssl cert is changed in ELB. → AWS ~~can~~ have temp access to that unencrypted traffic. So ELB needs to have SSL cert. EC2 instance needs SSL & they manually decrypt. Overhead.

2. Pass-through (NLB) → TCP. ~~not~~ never touches encryption. Controlled by me. No layer 7 load balancer. EC2 needs SSL crt

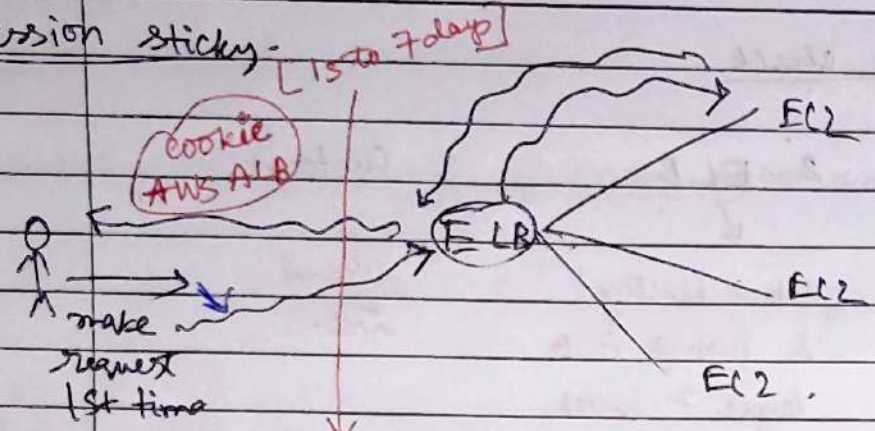
3. Offload →

me ═══ ELB ═══ EC2

↑ encrypted — SSL crt

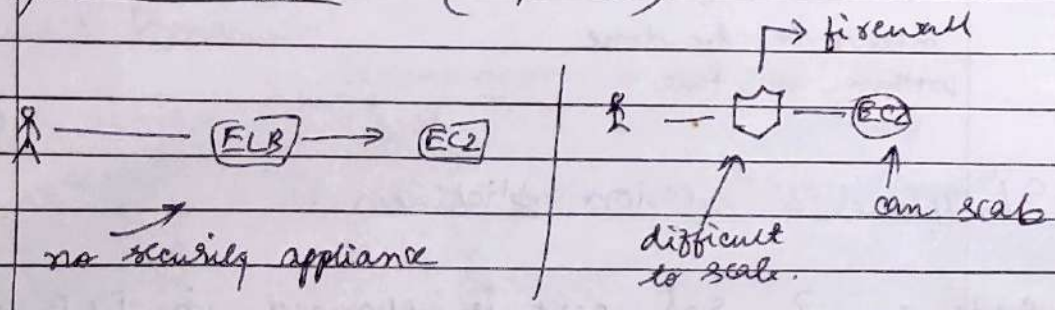
↑ not encrypted — no encryption, no SSL crt

Session sticky.



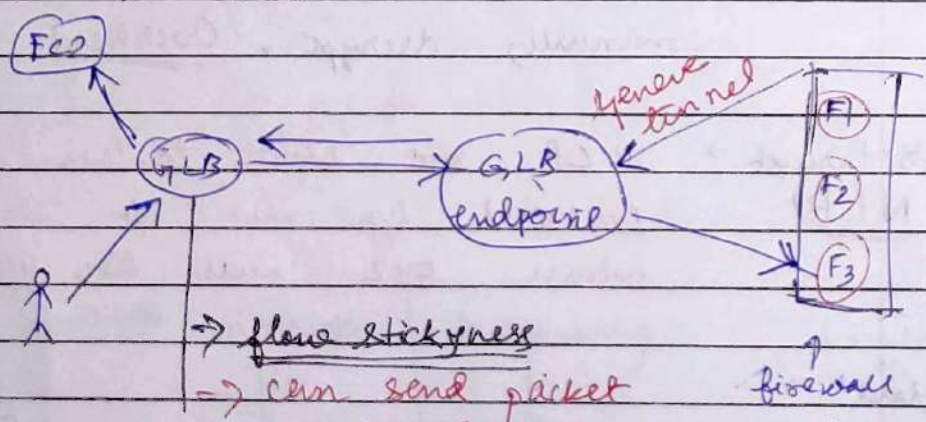
if cookie is present,
~~Ao~~ ELB will always send → uneven load.
traffic to same EC2

Gateway Load Balancer (GWLB).

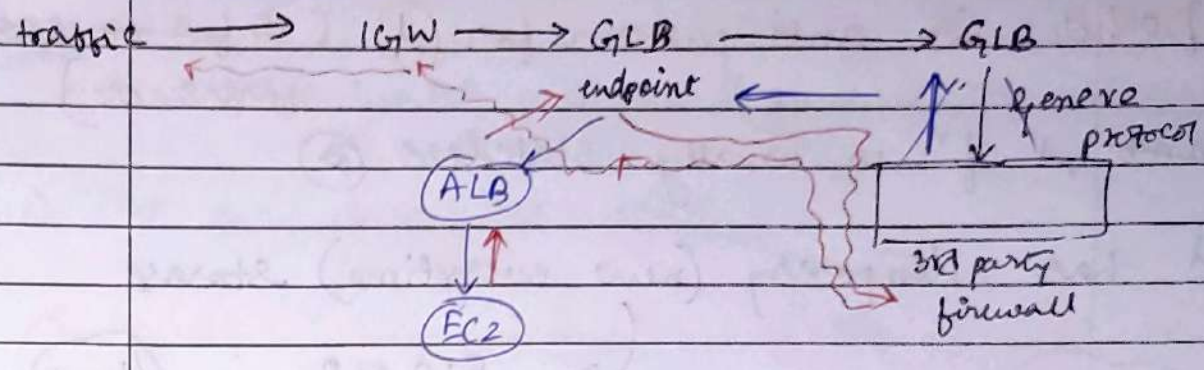


GLB → scale 3rd party security appliance - (firewall, IDS)

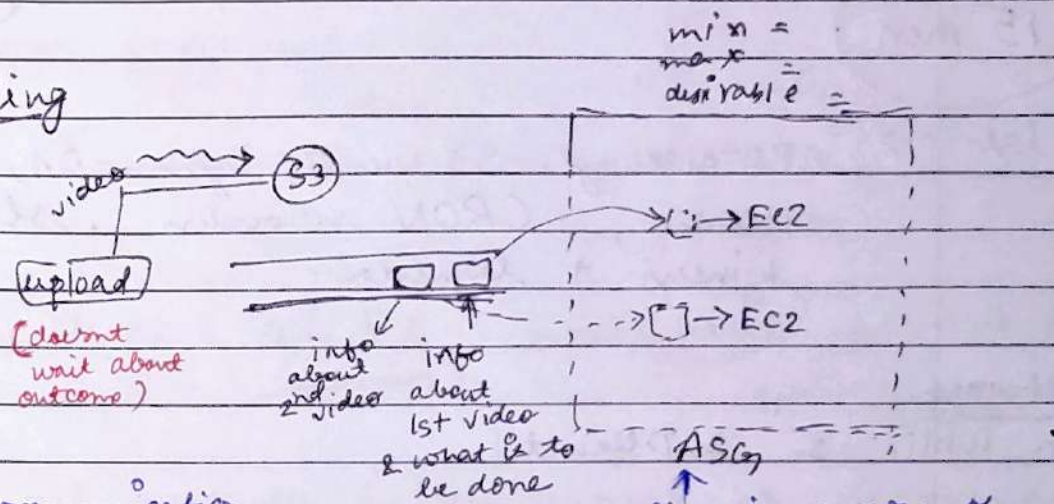
traffic inspection & protection.



→ flow stickyness
→ can send packet to this GLB via route table



Decoupling



Async communication

Producer —
Consumer — not constantly
Both — waiting.

Event driven architect (Not constantly waiting).

Producer generate event when something occur
eg - clicks, errors, --- etc

AWS lambda.

Runs our code. (FaaS)
We need to allocate Memory

Docker is anti pattern for λ [its for ~~to~~ ECS fargate etc]

~~Lamb~~ λ f^n is usually stateless

λ has temporary (new everytime) storage
↳ 512 MB in /tmp

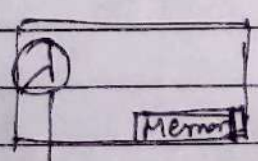
15 min

Uses → API gateway, S3 events, DynamoDB, streams, CRON scrolers, stream Kinesis + lambda.

Network
1. Public & 2. Default

1. Public λ → cannot access resource in VPC unless public IP is given & SG allows

2. VPC λ = λ cannot access public services without VPC network config
Can use VPC gateway endpoint to access that



needs role to access resources ← Execution Role.

λ resource policy → what service & acc can invoke λ f^n

__/__/__

Logs

λ use Cloudwatch, CW logs & X-ray. & metrics

X-ray for distributed tracing.

λ needs Policy in exⁿ role to do logs in CW

Invocation types:

Sync — Async — Event source

Sync client (cli/API gateway) will wait for λ for response.

client needs to handle errors or re-run λ

Async

image

S3 ——→ λ S3 event

S3 doesnt wait after it sends event to λ

λ will retry (0-2 times)

Code needs to be idempotent → same outcome everytime like running GET requests twice or many times

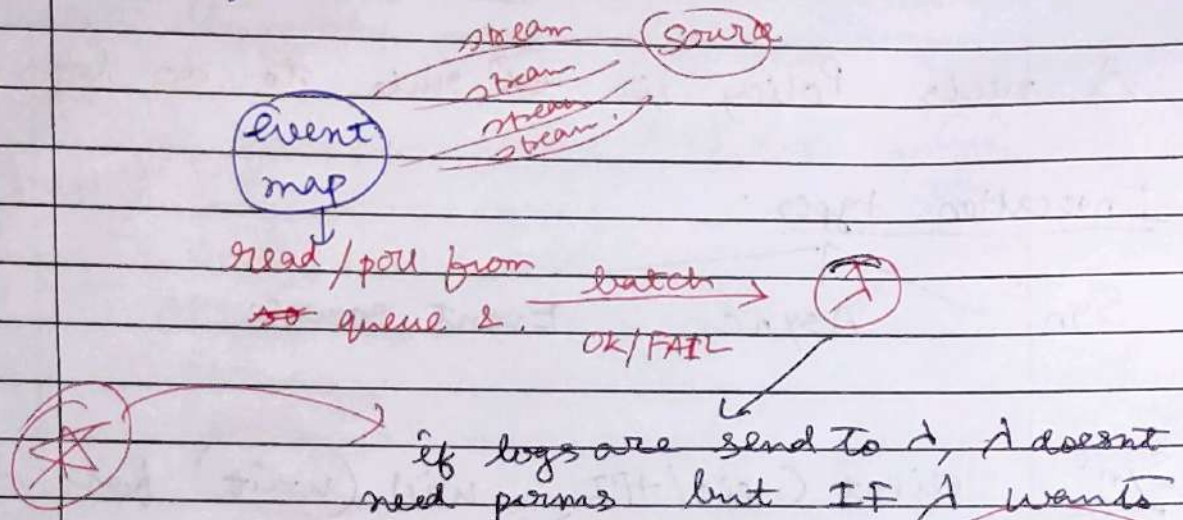
→ errors & events → Dead Letter Queue. for processing

→ can set destination → SQS, SNS, event bridge

__/__/__

Event Source

Which dont generate event (kinesis stream, Dyn DB stream).



BUT Event map needs a perms to retrieve data on behalf of λ

Version

= (code + config) of λ. Immutable (has ARN) never changes.

- can have alias

Execution context

context = env where λ runs. First time = cold start
↳ λ creates new context for 1st time.

Warm start = same context is reused (fast ~~st~~ cold start)

If not used frequently = context will be removed.
Concurrent = multiple context (new).

Provisioned concurrency = keep context ready beforehand.

CW Events (now Event Bridge) → near real time

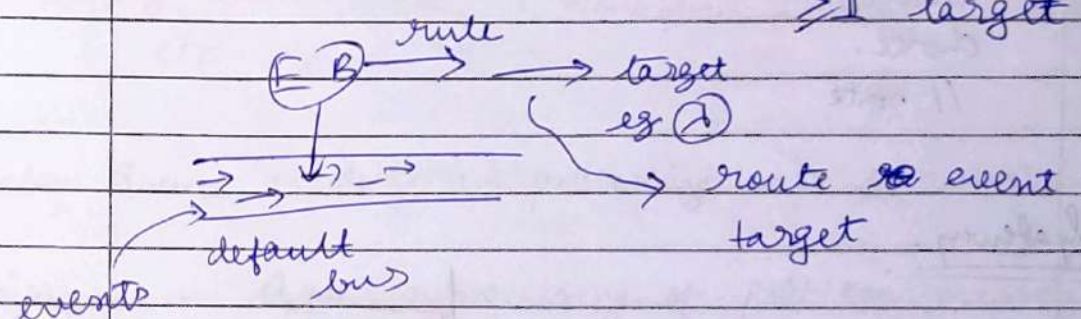
(if x happens, do Y)

EB is new

CW Event 1 bus

EB >1 event bus.

→ Rule match an event & routes event to ≥1 target



event = JSON

SNS

Status of delivery retries — HTTP, λ, SQS
— reliable delivery.

↳ HA & scalable in that region

SNS – SSE

Topic – cross account (resource policy).

Step f^n

problem with λ. λ runtime = stateless. non persistent

State Machine (make workflow)

· Start → state → end.

State → things that occur.

Duration = 1 year

Standard & Express
↓
1 year (default) (high workflow) → 5 min

Status

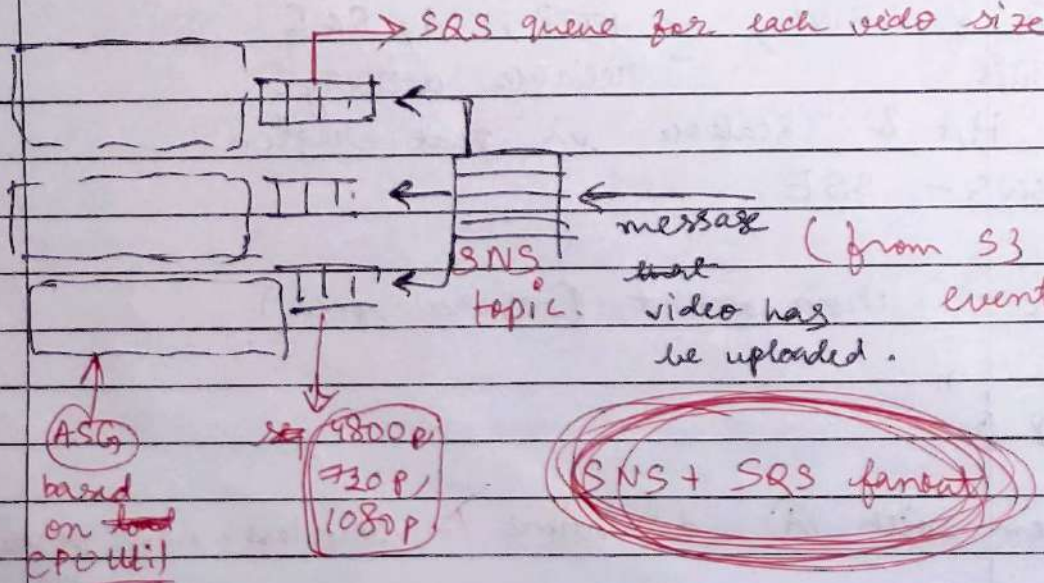
Succed / fail
wait
choice.
// state

API Gateway.

Edge opt - nearest cloudFront
Regional - same region.
Private - only in VPC

Caching
↳ can be encrypt
500 MB to 237 GB.
↳ reduce cost & load

503- service unavailable
504- integration timeout (29s)



Standard Queue	FiFO
Multi lane highway.	Single lane
Unlimited transaction/sec	300 - 3000 T/sec.
Ordering - no guarantee (best effort)	→ fifo suffix
No guarantee preservation order.	→ (1) processing no duplicates → Order preserved
De coupling batch work etc.	workflow, command admin's etc

Delay Queue - delay in processing into our app.

Visibility timeout - Auto reprocessing of problem message. timeout to prevent consumer from receive/process message

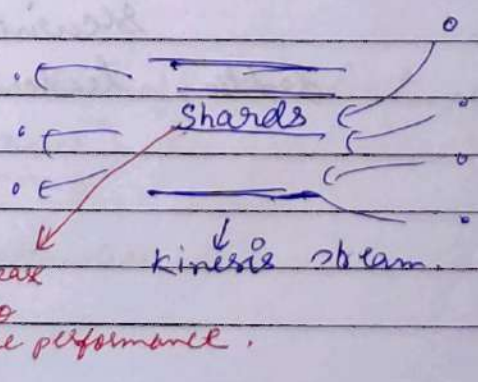
Dead letter Queue (DLQ) - Isolated diagnostic of particular message. Original ~~timestam~~ retention time for message is not change.

Kinesis Data stream

Data ingestion (stream ingestion) from multiple sources. Public service.

24 hr retention

ingestion, analytics, monitor



Kinesis Data firehose

→ fully serverless. (Near real time = 60 sec)

bil - data vol.

Kinesis stream = realtime

→ can do data transform (buffer)

can send to — splunk, redshift, Opensearch, S3 bucket

NOT REALTIME

→ provide permanent data stream storage.

Kinesis Data Analytics

Real time : using SQL queries from stream or firehose

DA → stream
→ firehose

can be S3 static data

~~Near~~ Real time SQL analysis
E sports dashboards.
security metrics
} Real time

~~dashbo~~ leaderboard = latest = SQL (DA)

Kinesis video stream.

See. camera, thermal depth, RADAR
(used with Rekognition)

Persist & encrypt data

Cant access data directly from S3/EBS etc
↳ only API

Cognito

User pool - sign in & JWT token, manage users
↳ also social sign in

ID pool - temp AWS credentials, unauthenticated IDs,
(swap ID tokens) ~~also federated external IDs~~

JWT tokens cant be used to access resources

ID pool will use google / FB etc token and with the help of a role, give it temp AWS credentials

Unlimited no. of users

ETL (serverless)

→ serverless extract, transform / load, ad-hoc
→ datapipeline (use server)

→ data catalogue. with crawlers.

Amazon MQ

(Sns + sqs) → public services.
Open source message broker on Apache.
(both Queue + topics)

NOT public - VPC only. / PRIVATE

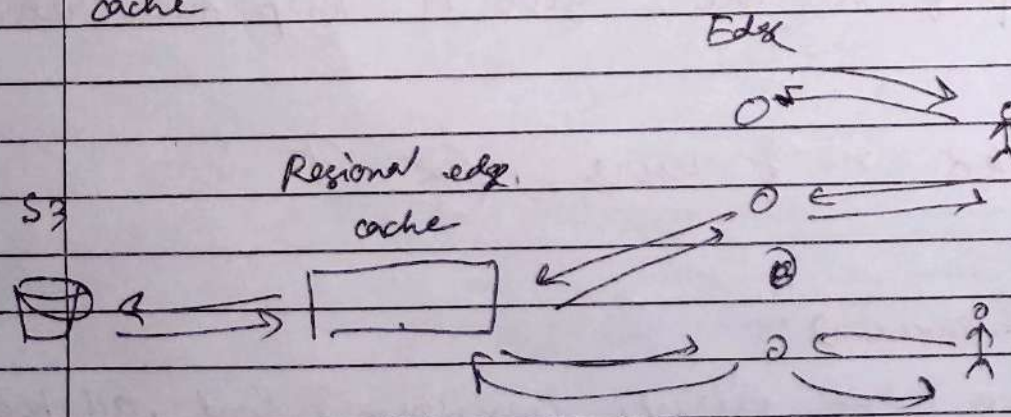
MQ - migrate from existing system & APIs

App flow

middleware - integration service.
connector, aggregate data.

CloudFront

Edge loc - local cache
Regional edge cache - larger data than edge location



Cloud front → Only for download
→ only (Re) caching

TTL = after this time object in Edge location will expire

default = 24hrs.

Cache invalidation = to all edge locations

Versioned file names = all object has diff file name
(cost effective)

- ACM (certificate Manager)

can generate, import cert

If ACM used - cert can be regenerated

Not all support ACM (Cloudfront, ALB, EC2 ✗

* Cert will NEVER go out from 1 region

CloudFront - use us-east-1 for ACM

US-EAST-1

origin ← cloud front ← 🚶 } 2 SSL public cert

cert needs to MATCH DNS NAME

NOT self signed

SNI - run >1 HTTPS service in same IP
(free of charge)

Distribution

we create everything here - origin policy, logging, etc + cache control etc.

Distribution will give me URL which will point to same S3 bucket website from edge location

(- - - - . cloudfront . net) → domain

We should use invalidation tab to invalidate objects/ whole domain code + resource etc if we upload different object with same name

we can use paths like (/*)

Default SSL cert is given ∀ (cloudfront.net domain)

A

Can add SSL cert to our domain in R53 & use that to point to cloudfront distribution (custom distribution name)

1. Create ~~subdomain~~ <anything>. mydomain. com <alternate domain>
2. Create a certificate for that subdomain in ACM. <must have same name>.
3. then in certificate add CNAME record to domain in R53
4. Add that cert in distribution CF.
5. Create simple routing ALIAS to CF (domain name is subdomain name in cert)

DONE

OAI

Origin Access identity

(*) CF distrb becomes to OAI and that can be explicitly allowed in S3 bucket access and deny the rest. (ONLY for S3)

for Non S3 = Make Origin to send that (Custom Header)
custom origin

Signed URL = 1 object
signed Cookies = group of objects

Lambda@ Edge

- S3 origin request migration, content based on device

(*) Country URL redirect @ Edge location

Global Accelerator (no caching)

Connections enter edge using Anycast IP
(Anycast - 1 IP to multiple device)

> for TCP/UDP/~~[illegible]~~ etc (network data)

Cloudfront - only (http/s) + caching

__/__/__

Advanced Networking

VPC flow logs.

→ packet metadata (not packet content)

can be of
- → VPC (act ENI)
- → subnet
- → ENI

NOT REALTIME

Destination — S3 + CW Logs.

Metadata traffic is not recorded (169.254.169.254)

Egress only IGW

Outbound only for IPV6 (like NAT but for IP V6)

VPC endpoints
- → gateway endpoints
- → interface endpoints

(1) Gateway - S3 & Dynamo DB.
(SDG)
(HA)

allow private VPC resource to use S3 and DynamoDB.

↳ we set which subnets can use it. That are same region only.

(2) Interface

Private VPC → can access all service (other than Dynamo DB)

Interface - NOT HA use DNS

need 1 IE per AZ

TCP & IPv4

Can use SG

Use Private link

provider: DNS

Regional DNS, Zone DNS

gateway = PL = prefix list (we put in route table)

Interface = specific subnet

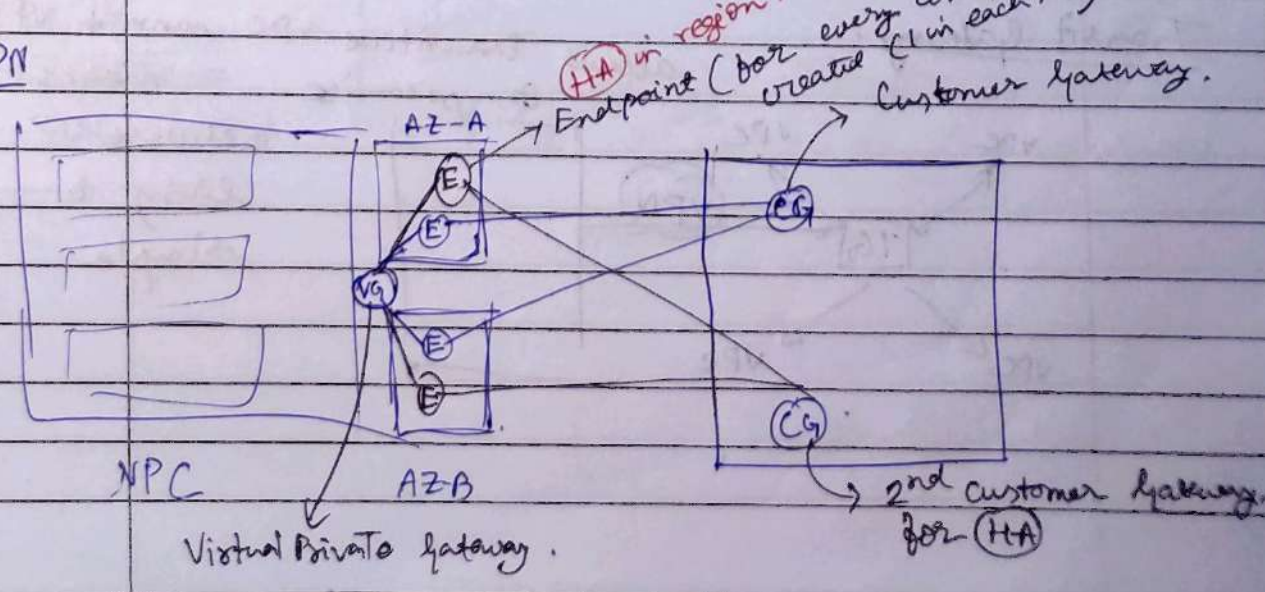
↳ Need to enable (private) DNS or put manually.

BGP

Default = shortest route is preferred

but ~~the~~ if shortest route is bad/slow, then it can be modified in ASPATH = we can make it look ~~too long~~ by changing ASPATH and adding ASN (more) number in that path

VPN



both use IPsec

Static VPN	Dynamic VPN
Static route to route table on both side	Used BGP (path)
NO load balancing	Multiple VPN con^n
Simple	HA traffic distribution

Route propagation - No need to add route manually. It will learn and add them in RT

DX

Not owned by AWS
AWS rents this

→ Me → DX region ← AWS region

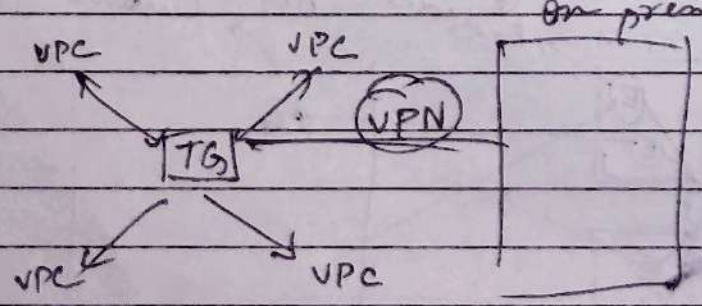
Allows con^n b/w AWS VPC & Public service
→ NO INTERNET by default

DX is basically a port allocation.

NOT RESILIENT BY DEFAULT

Transit Gateway.

allows transitive VPC conn + VPN = makes network easy & simple



Storage Gateway.

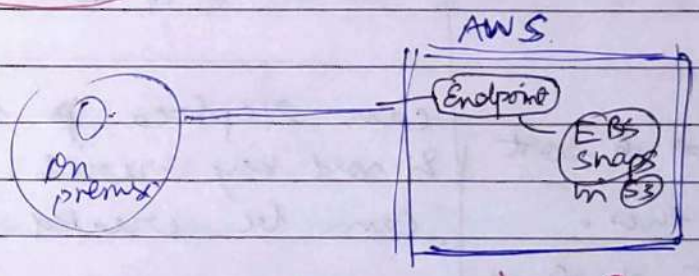
→ Integrates with EBS, S3 + Glacier

Migrations, Extensions, Storage Tiering, backup system

Volume
(Stored)

1. All stored locally : Full disk backups.

Raw disk



(*) Doesn't help on low capacity issues (*)

(*) cached

- Data stored in (S3) & cached locally (*)
  Managed S3 bucket

→ DC extension (*) (*)

Tape	File
• Tape backup	file → S3

Dynamo DB

Query = only affected data consumed

Scan = Entire table consumed (Σ all data = RCU)

filter = entire row

LSI Local secondary	Global secondary (GSI)
can replace sort key with this. Must create during table create	can replace both Partition & sort key with this. can be created anytime.

both are sparse. (if index is not present then that is omitted)

WCU = 1KB / sec
RCU = 4KB / sec

DAX

primary – R+W
replica – R

Write caching also possible

Elastic Cache

Session data storage

Notes revision (Adrian)

RDS

Multi AZ (sync) - Same Region

Read replica (async) - Cross Region (global DR & resilience).

Aurora

Disk failure - self healing.

Storage ~~@~~ (shared, high use bill, ~~less on map~~)

Fast clone of entire prod DB for testing (Copy on write).

Aurora Serverless

very flexible - unpredictible

ACU = 0 means no load (less bill)

multi tenant (many consumer - Netflix 4K more exp than HD)

Aurora GD

→ can replicate entire DB of 1 region to other secondary regions.
low latency, global DR, resilience.

∴ Aurora GD vs RR RDS (both global cross reg)

no extra time — same time

(*) (Aurora ~~reader~~ ~~cross region read replica~~ can be cross region) (*)

TD tests

1. MGN (App migration service), DMS, Data sync.
2. ECS, EKS, Grafana
3. Dynamo DB stream / DB transactions
4. S3 object lock

Dynamo DB stream - Actions will be streamed. Lambda will run on ~~a~~ any change in DB. It will make a CW log & show (keys only, old, new or both states)

Dyn. DB Transaction - bunch of operations at same time. (group of tables / group of rows)
. If any one fails - ALL FAIL.
ALL / NOTHING.
ACID compliant

Lambda: Invoke f^n = this acc user can invoke lambda.
It is resource based policy.

S3 object lock = once enabled - cannot suspend versioning

& CANNOT DISABLE lock.

governance
special perms needed by some users

legal hold
It doesn't have time

retention
after this time lock will not take place

compliance
NOONE can delete / overwrite

CloudTrail Insights = Will tell unusual API activity.

Inspector	GuardDuty
Vulnerability scan	continuous monitor AWS use AI / ML for malicious activity

App discovery service - migration tracking of VMS, on prem servers during migration.

(*) Glue + Athena used together. (Glue for catalogue)

Glue (catalogue) → Athena SQL queries → Quisight visualise.

Q. Redshift spectrum, Recovery manager, oracle, Real App Cluster, SQS target group & processing system,

native stored procedure, short/long polling, CPU Util for RDS + Cloudwatch, ECS cluster config

RDS & (Aurora) → endpoint, API gateway, Redis & Elastic cache.

CloudTrail trail encryption, Enhanced monitoring

Redshift spectrum = Query on (S3) (not serverless).

(OLAP)

petabyte scale data warehouse

RDS metrics	CW	Performance Insight	Enhanced Monitoring
	1min interval.	visualise DB load, Aggregates perf data	monitor OS, process threads that use CPU (no extra software).

Dynamo DB events - CRUD + ~~many~~ DB changes too
these can be used by lambda
& CW logs.

BUT

RDS events → DB inst event, Parameter groups,
Sec events, snap events.

NOT (CRUD) operation

Native f^n / Stored procedure
↓
they can invoke (λ)

~~Poll~~ Messages are not deleted after visibility timeout in SQS. Consumer needs to delete manually.

Cost cutting

long poll	short poll
Not 0. Atleast 1	default
	Can return 0 message
ReceiveMessage > 0	ReceiveMess = 0

RDS auth = IAM DB ONLY

Protect backend = Throttling in API gateway + cache

Cloudfront doesnt protect backend

Aurora → custom endpoints can point to different instance sizes

__/__/__

ASG - Cooldown period - after this period, scaling will start.

Constant resources - target tracking

Multi AZ RDS - AZ failure
RR - performance

Object lock - bucket wide

Session data = Elastic Cache + Dynamo DB

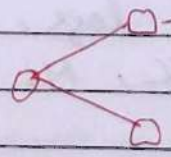
AWS AD connector + IAM roles =

Lambda can handle sudden burst of traffic

RDS proxy - prevents "Too many con"
↳ prevents large number of open con to DB. Use existing con. η & scalable. share connections

Max SQS message retention - 14 days

NLB →



→ node → this can have static Elastic IP so that the EC2 instances firewall can be configured. (static so fire wall config doesn't need to change)

__/__/__

Data sync

huge scale.

on prem → S3, Glacier, Deep arch } directly

use NFS, SMB

Move cold data

(TBs of data move)
not hybrid

storage gateway

low latency storage & access to data + store securely on S3.
Optimize, compress

DC extension.
(storage + access)
→ optimization
hybrid.

can google drive

File gateway
→ NFS & SMB
↳ can mount and send to S3

Vol gate → S3 but like E drive

Dynamo DB backup can't be → other region / other Account

For this use AWS Backup
→ copy backups
→ cost allocation task
→ transition to cold storage.

CW normal - 5 min (free)
CW detailed - 1 min (cost)

EMR - vast amount of logs, log analysis, ETL, ML etc.

To allow/deny (A), we need to use execution role (generally IAM role)

Not function ARN

App sync - GraphQL.

App flow - transfer data securely b/w SaaS applications.

Aurora $\xrightarrow[\text{Service}]{\text{Data Migration}}$ Aurora serverless

RDS - auto scaling storage (~~stor~~ storage will grow & sink)

Limit - EC2 instance - VCPU Region limit

RDS point in time = 5 min. (Automated backups)

Cloudfront can mitigate DDOS

EFA - Fabric Adapter in EC2 - HPC

EC2 billed if hibernating with stopping state

↓

~~for EBS + EIP if applicable~~

VPC

/16 to /28

NOT STOPPED

↓

If stopped then charge for EBS + EIP

Intelligent tiering - unpredictable load

Raid 0 - IOPS
Raid 1 - fault tolerance.

Elastic IP for NLB, Not ALB

NLB → attach EIP (for white listing)
↓
ALB

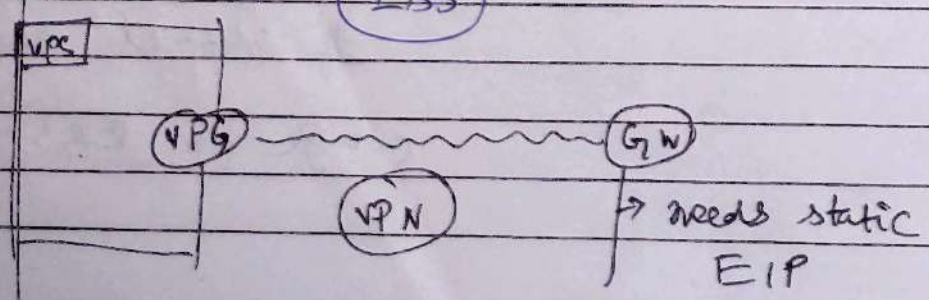
Origin failover is supported by cloudfront
↳ HA

IAM DB auth - dont need pwd (only auth token)
TDE = stored data encrypt (not transit)
↳ Ms SQL server & Oracle DB > can use CloudHSM

Redshift - structured + semi structured data

SWF = workflow service. coordinate across decoupled / distributed service.

Data lifecycle manager - auto creation & manage snaps for EBS



Gateway / Interface endpoint policy → can control access to S3 / dynamo DB etc.

CW alarm can be sent to SNS.

ECS cluster = batch processing & // workload

EMR = ML, logging, petabyte scale - huge frame works

EC2 - transfer cost if different AZ
↓
inter AZ cost

same = no cost

Data lifecycle Manager	AWS backup
Auto create, retention, deletion of EBS Snapshot	Manage, monitor backup across many service. ↓ from 1 place (if you want to take backup to another region)

SSE - S3 → Regularly rotates

CW alarm - stop / terminate / reboot → EC2 instance

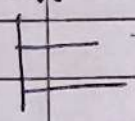
RDS - RR - each RR has separate CNAME (R53 weighted routing)

__/__/__

Aurora

DB

cluster

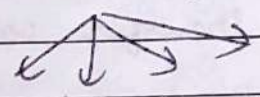


Type

Writer (one CNAME)

Reader (one CNAME)

Any RR that we create, will be added inside Reader endpoint (so 1 Reader endpoint (LB)



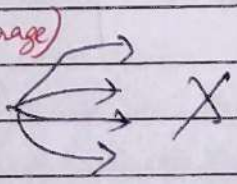
many readers)

(*) Can create custom endpoint and add many readers & writers (LB)

Manual failover to Read Replica (it becomes primary) possible.

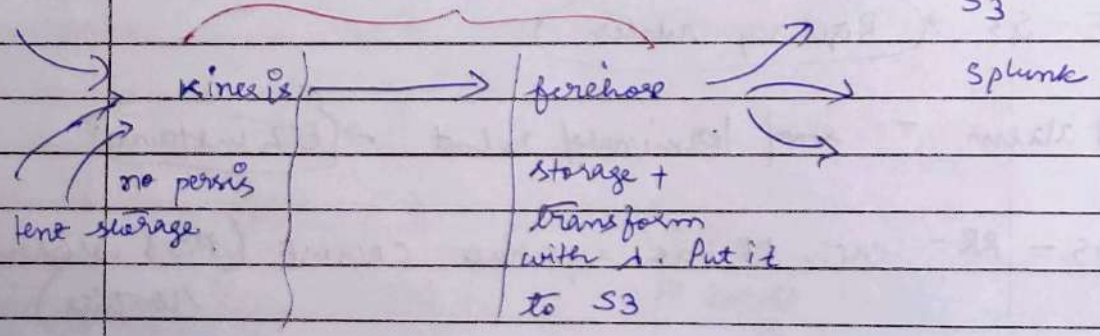
RDS (auto scaling storage)

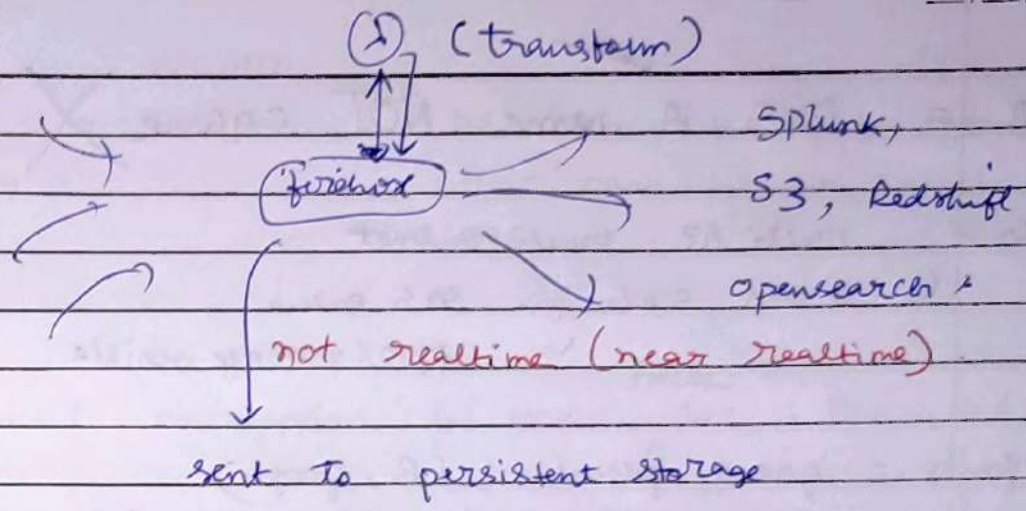
No 1 endpoint



Every RR has separate CNAME

Realtime





Stream cannot send data to Dynamo DB
Firehose can use λ to transform & then send to Dynamo DB

Firehose = Click stream
(load data).

Athena	Redshift
→ Serverless	→ Cluster
→ Simple queries	→ Complex queries
→ semi structured & structured data	→ petabyte scale data warehouse.

Elastic Beanstalk

→ Server logs are stored in (S3 OR CW)

Reserved instance = AZ not Region.

TDE = stored NOT transit

ELB = DNS A name NOT cname ✗

ASG = multi AZ. multi subnet
↳ can scale on SQS queue
↳ approx message visible.

Cognito - grant perms. (FB, google)
SSO - users already in AD & can federate with SAML 2.0

Q

DR
backup & restore,
acti..., warm, pilot light

DX - VIF (virtual interface), RDS encryption.

queue priority.

Dynamo DB & RDS provisioned throughput & (Auto scaling)

DR
Backup - RPO/RTO = hrs = Cheapest. After event start recovery.

Pilot light - RPO/RTO = 10 mins = Some resources are live. Data is live in DB. Service is idle.
infrastructure = ✓
functional elements = ✗

RPO/RTO

Warm standby = mins = has min live deployment that can handle small request

In transit encryption for RDS - Force SSL. Download & use that certificate will all connections from database

REQUIRE SSL

Dynamo DB scaling

Dynamo DB (WCU, RCU) → ASG

when cap exceed → Cloudwatch Alarm → ASG starts working

λ @ edge

Code closer to customer (reduce latency)
Origin replacement
Authentication (λ will run cognito)
SEO for bots (λ can deliver prestored static web page @ S3)
Route
Premium / payware content (filter not auth requests)
Prioritize users

AWS proton
Automatic infra as code for serverless, container based apps.
CICD, Staging, dev, prod teams

AWS lakeformation

__/__/__

hibernation - RAM contents will be saved in EBS (root volume)

λ will encrypt env variable with default KMS key.
~~People having~~

(*) but recommended to use encryption helpers with custom KMS key

(disable)
Guard Duty entry ?? guard duty delete logs

Aurora promotion tiers

Snapshot vs AMI
↓ ↳ all
of EBS (where there is an AMI, there is snapshot in same region).
↓
can be turned to boot vol from snapshot

Cloudfront will not cache proxy methods
↓
PUT/ POST/ DELETE ....

S3 bucket perf ↑ = prefix name

SQS = 300 default. Batch mode = 300 × (10) = 3000
↓
max

Throttle
→ API GW, λ, SQS (buffer), Kinesis (buffer)
(if concurrency max achieved)

CANNOT store data from kinesis to Dynamo DB directly

Kinesis DA can only take data from Streams & firehose

SSE-KMS = Audit trail

Aurora RR & Multi AZ standby.

Cognito ID pool vs user pool.

- S3 lifecycle transition rules

mi 30 days in standard → IA / 1Z IA

Aurora failover ??

Which metric on SQS to poll? (no. of messages)

SSO - org centrally manage SSO

Vol gateway (stored) - low latency to entire database
" (cache) - " " " data subsets

Fsx windows = AD integration (DFS)
file gateway = (NFS+SMB) AD " but not enough throughput
Directory service = integrate with on prem AD (microsoft AD)

WAF (web ACL) → Can block country IP & set captcha ... etc.

Cloudfront ⇒ Geo block → content block. prevent accessing content on country/ finer level

ASG lifecycle hooks ??

SQS metric

ApproxNumberOfMessageVisible = ≃ denotes length of queue
Age Of Oldest Message ≃ if message is in queue for a lot of time & you need to process it fast. Time sensitive

Most no of instance with AZ
↓
Oldest launch config
↓
next nearest billing hour

Warm up · ASG = Step Scaling (ALB)

EBS

If EBS is encrypted - snapshots will always be enc

If EBS is not enc - first snap will not be enc.

→ BUT we can create a copy of snapshot & encrypt that

→ BUT while creating ~~tot~~ vol - we can encrypt

Encryption in EBS → static encryption.

Once enc is enabled - cannot turn off.

AES 256 → Operating system is not aware

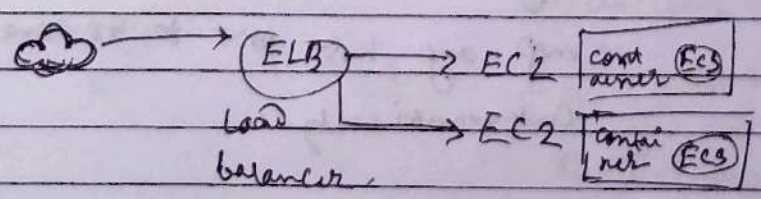
AMI → if we had 2 EBS vols, separate snapshots of both of them will be created for AMI

Copy of AMI to different region - new snaps will be created for that region for the AMI

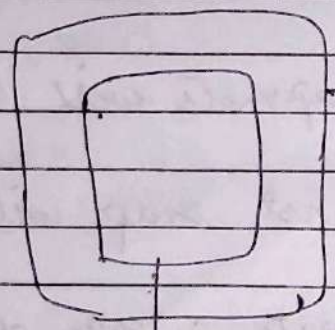
∴ AMI = (snapshot of EBS × no of EBS)

ECS

EC2 model = ECS agent needs to be installed.



ECS demo



task definition (can have >1 container)
- → network mode (bridged)
- → task size (400 MB RAM)
- → task CPU (1vCPU)

→ container definition ← add container
- → docker image
- → memory limit
- → port mapping
  - host : container
- → ~~Edit~~ =

→ Edit = create new version

→ EC2 launch type - task size & optional

Run task

- → lets say we have 2 EC2 inst
- → Each task = 1 container
- → 2 task = 2 container = 1 in each EC2
- → 4 task ✗ = task 1 = 80 : 80 (host : cont) — EC2 1
  - task 2 = 80 : 80 — EC2 2
  - task 3 = - - - ✗
  - 4 ✗

port mapping already consumed

Dynamic port

host : cont
0 : 80
↓
map any host port to 80 cont automatically.

Service definition.

→ its to keep HA & auto scaling for task.
If 1 task dies - service def^n min task will auto scale it to maintain min task (container)

Application LB can balance tasks (containers)
ALB has to be ~~bou~~ public but tasks can be in $\boxed{\text{private subnet}}$

RDS

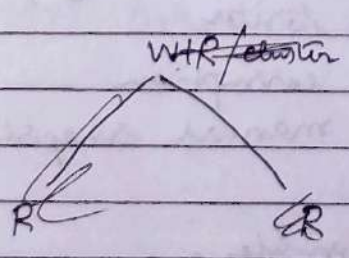
① Multi AZ DB instance

= primary + ONLY 1 standby in different (AZ)

$\boxed{\text{Only 1 CNAME endpoint}}$ (no r/w - only used when failover)

② Multi AZ DB cluster

1 primary + 1 readable standby
1 readable standby



W+R/cluster [1 instance endpoint + cluster endpoint]

R [1 instance endpoint]
R [1 instance endpoint]

1 reader endpoint

If writer fails : any of the readers will be promoted to writer & same old writer endpoint can be used

Can do read scaling ⊛

Snapshot / backup

Manual snap - anytime
Automated backup - on intervals. 35 days max retention
(5 min transaction log).

both are managed by AWS managed S3 buckets

Restore from snapshot = everything will be new

Storage Auto Scaling.

This is not available for multi AZ DB cluster. ~~Only for~~

Read replica
→ can give 0 RPO → it constantly in sync with writer.
very low RTO → can be quickly promoted to writer

they dont help in data corruption.
we still need to take manual snapshot

Cross region RR is also possible.

RDS security

Transit

SSL/TLS = can be made to be forced for users so users can be forced to use SSL/TLS

Static

Default KMS enc = handled by EBS

For KMS enc at rest - storage, logs, snaps/ replica all are enc

Enc CANNOT be removed

Security by DB engine itself

Microsoft SQL
Oracle (HSM) > TDE

E done by DB

IAM Authentication:

IAM role → user / EC2

User can authenticate if they have correct role.

Its like FGAC but here Authorisation is handled by DB Rules NOT IAM roles

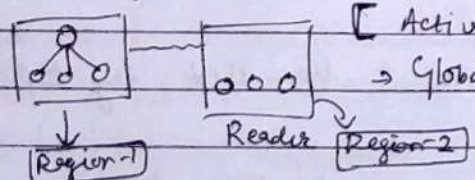
Aurora

Point in time recovery = Backtrack.

Fast clone = entire copy of database (copy-on-write)

~~Server~~ Serverless - proxy fleet + ACU auto scaling

Global database



[Active ~~active~~ passive rep^n]

→ Global DR (RPO - 1 sec, RTO - 1 min).

RDS proxy.

Reduce failover by 66% and pool application connections. Open and close connection fast maintain

Aurora Multi master writes = Active - Active rep^n

FAULT TOLERANT

Application Load Balancer

→ Have ~~client~~ SNI → can be used for multiple domains & SSL certs

→ Only Layer 7 (HTTP or HTTPS).

→ Broken SSL - terminate at ALB

→ ALB MUST HAVE SSL cert if HTTPS is used

NLB

→ Whitelist static IP

→ Private link

Launch Template & Config.

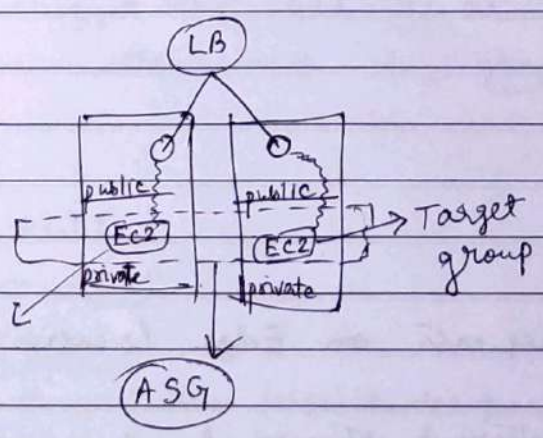
LC → ASG

LT → ASG + EC2 instance separately.

ASG

Grace period health check - default 5 min. Time after which first health check occurs after EC2 launches.

ALB

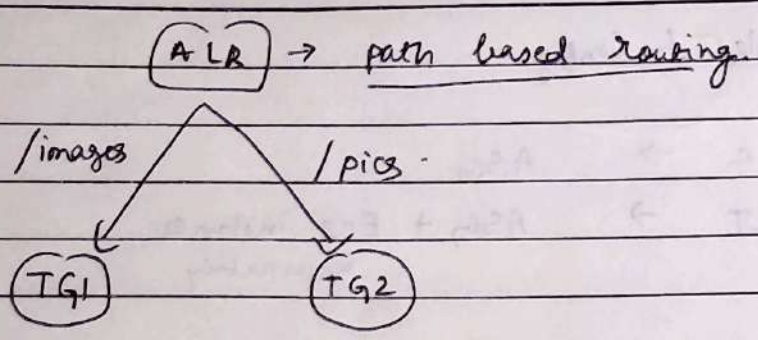


→ its a group of EC2 instance which the ELB will forward request to.

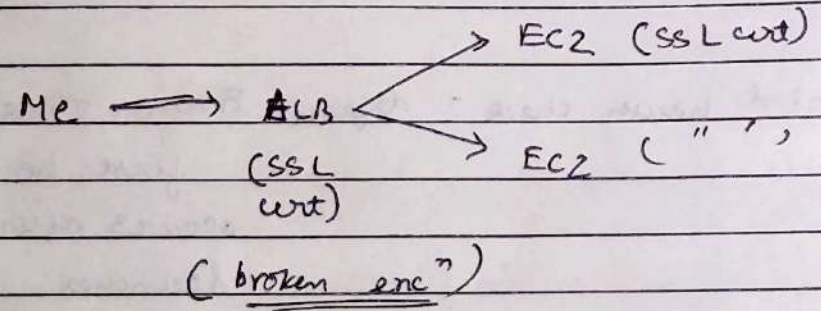
This can also be a target group

(~~when~~ When we create ASG, we can select a target group. ASG will launch instances in that target group.)

ASG cooldown → wait for cooldown before scaling activity
↳ prevent scaling action before the last scaling action is visible



ELB (ALB)



Lambda @ Edge

Run and check requests on Edge locations

Useful to
- → send different objects based on User-Agent
- → Authorise tokens.
- → redirect users

Lambda @ Edge	Api gateway + λ
Across all locations & edge locations. Not confined to single region.	Regional service

Dynamo DB

Read		Write
Eventually consistent	Strongly consistent	1 RCU = 1 KB
1 RCU = 8 KB	1 RCU = 4 KB More exp	

Dynamo DB streams

Any update here in table - create, update, modify, delete will be sent to Dynamo DB stream. Then in that stream we create trigger and add λ as trigger. Logs can be found in cloudwatch log group.

RDS - OLTP

Redshift - OLAP

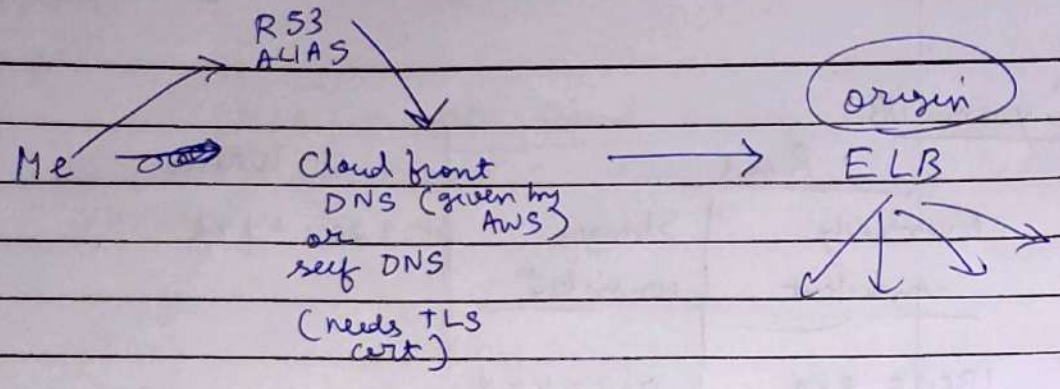
(cluster) Redshift spectrum - run large complex queries on S3 data.

Redshift ~~spectrum~~ - pull & join to structured data

(serverless) Athena - sample queries of S3 data (Athena replicate entire DB) CSV format

EMR - extremely large data process with hadoop/spark etc

ALB	NLB
X	Static IP / Elastic IP
web socket	Web socket
X	Private link
λ as target	X
SSL offloading	SSL offload
SNI	SNI
Authentication.	X



It can protect backend

SQS queue

maxReceiveCount - max times a message can fail. After this message move to DLQ

DLQ = same as original queue.

Delay queue	Visibility timeout
message hidden when added to queue. (default	message hidden when consumed (default = 30s)

Long	Short polling
wait > 0	wait = 0
0 only when polling wait time expires (20s) ↑ max	can be 0 result

else min 1 result

EC2 scale based on ApproxNoOfMsgVisible

Kinesis vs firehose

Kinesis = realtime

firehose = 1 min (near realtime)
input = realtime
output = 1 min

(more long term storage than ~~fire~~ ~~hose~~ data stream (24 hr))

Click stream

Capture, transform with λ → Dynamo DB

HTTP — splunk — OS — S3

Redshift

Lifecycle rules

S3 → 30 days → IA / 1 zone → 30 days → Glacier / Deep arch

S3 standard → Any days → Glacier / Deep archive