SAA-PRO advanced concepts

① Kinesis: Can't have more than one consumers | Huge no. of producers
Nor can't look back in time | Name, Partition Key, Data

- EC2 running KCL
- λ
- Firehose

Server Side using KMS, at rest

Adding / Dropping Shards:

- for a given stream, 24 hour retention

① Kinesis Data stream
② Kinesis Shard
③ Kinesis Data Record

* Can also log shard-level metrics

Kinesis Firehose: Take data from a Kinesis stream | (PUT directly) → IoT → CloudWatch Logs → CloudWatch Events
↳ restructure data as it is entered:

* S3, RedShift, Elasticsearch, Splunk | When using stream already encrypted | (IoT → S3)) delivery of data to certain loc

---

Kinesis Data Analytics: → Real-time SQL-Like analytics

Kinesis Data An

(Stream) Source → (Firehose) → ((in-application input stream)) → Inside Analytics ((in-application output stream)) → (Can also pre-process with λ)

Redshift: ⇓ S3 best DR (CRegion)

- Automated Backups enabled

(Compute Node has slices where data is stored)

IoT: Device Shadow ↑ Device G/w

MQTT Topics x.509 | Pub/Sub ↳ Hierarchical

IoT Rule:

Basic ingest rules to trigger

directly runner than msg

Quicksight: → BI
Visualization

Elasticsearch: No SQL Database
Rapid Data Retrieval

CloudLogs

(Document DB)

ELK → Visualization of Logs / Metrics

shards to improve throughput & resiliency

Multi-cores ✓ Best

SNowBall

↔ S3 >10TB

50TB – 80TB of data

((at-rest / on transit)) ← You manage keys

S/Ball Edge :

Local compute instances

IoT 13

Analytical, compute optimized

process and store

Computing

SnowMobile:

10PB of storage

upto 100PB of data stored

SQS: decouple

(Best effort / atleast once)

Visibility timeout → if we do not explicitly delete

Short polling → $, lot fo API calls

Long Polling

256KiB

delivery delay → added and visible to people

Redrive Policy

CMK KMS

Scale the components based on queue-size

(Not for multiple consumers xReplay)

SNS: sending to a topic

CW and SNS
Metrics & Alarms

Push Notifs

Application
S3 Events
CW

λ can be subscriber

HTTP/S endpoint

Fan out:

Single message to multiple processing

SQS queues as independent subscribers

(Notif / messages to endpoints
Not a queue)

AWS MQ: Message Broker (AMQP, STOMP, OpenWire)
(only inside VPC)
Nlw of brokers

Queues, Topics and Virtual Topics

Step Fn: orchestrate Work Flows
WorkFlow retention of 1 year
Can define steps and flow

serverless

Task → Choice → Fail/Success
Pass → Wait → Parallel
State machine

Beanstalk:

Web Server & Worker
can have multiple env. in EB
Blue-Green swapping environments for good deployment

Opsworks:

Chef & Puppet

Data Pipeline:

CloudWatch

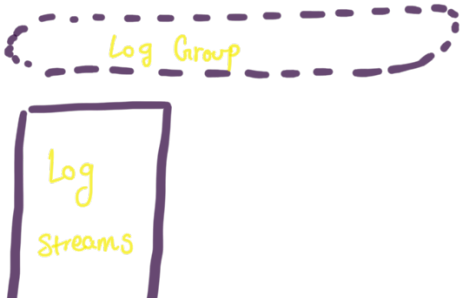
Very high frequency: low retention

Metric: Time-ordered set of data-points

Namespaces: Metrics grouped together

CW alarm can define actions to be taken based on metric breach: → SNS Topic

CW Logs:



Alarm changes → EC2 actions → AutoScaling

All state is stored on local EC2
if we switch, wer interruption

Trail → allows logs to be sent to S3/λ

((Stateless mein look-up available))

Billing Models:

On-Demand is default:
↓
Don't mind if EC2 are shut down

Spot Fleet: → Same type
Same AZ

- Same base instance
  AZ or Region →
  Base Load
  Cover

absolute priority

ASG

- → AMI Baking
- → Bootstrapping (Base image uske upar yeh configuration)

Launch Configuration
(On-demand or spot)
(Immutable object)

Launch Template

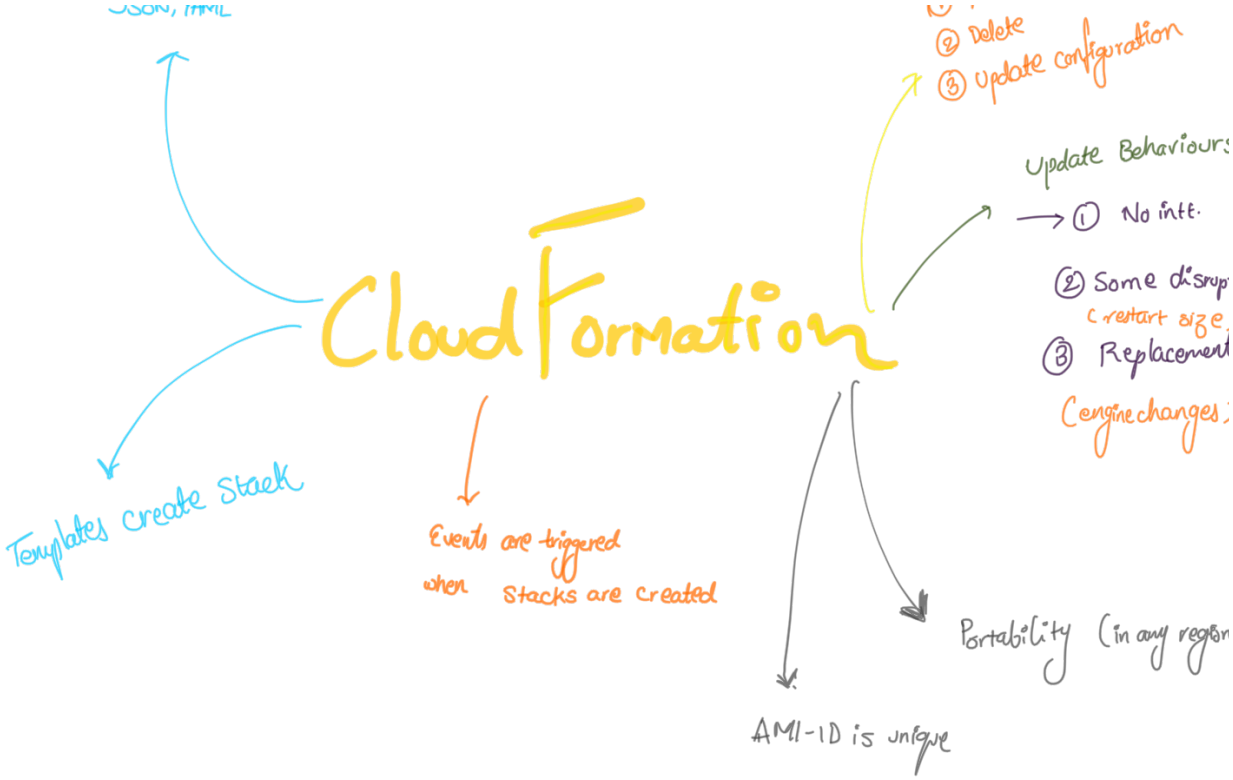
Scaling policy: 20 ↓ → 1
90 ↑ → 4 or 5
) Specify target value

Cross Zone
Load Balancing

ELB

→ in public-facing
get public IP

For update
① Add


JSON, YAML
CloudFormation
Templates create Stack
Events are triggered when Stacks are created
② Delete
③ Update configuration
Update Behaviours
① No intt.
② Some disrup
restart size
③ Replacement
(engine changes)
Portability (in any region
AMI-ID is unique