AWS Certified CP Course – CLF C02

Section 3

– Traditional IT overview :

what is a server composed of ?

→ Compute – CPU

→ Memory – RAM

→ Storage – Data

→ Database

→ Network : Routers, Switch, DNS

There are problems with physical data centers.

like – pay for rent, power supply, limited scaling, hiring ppl to manage, etc.

Cloud Computing → on demand delivery of resources

→ pay as you go

→ provision exactly right type & size

→ access resources within seconds

Deployment models of cloud

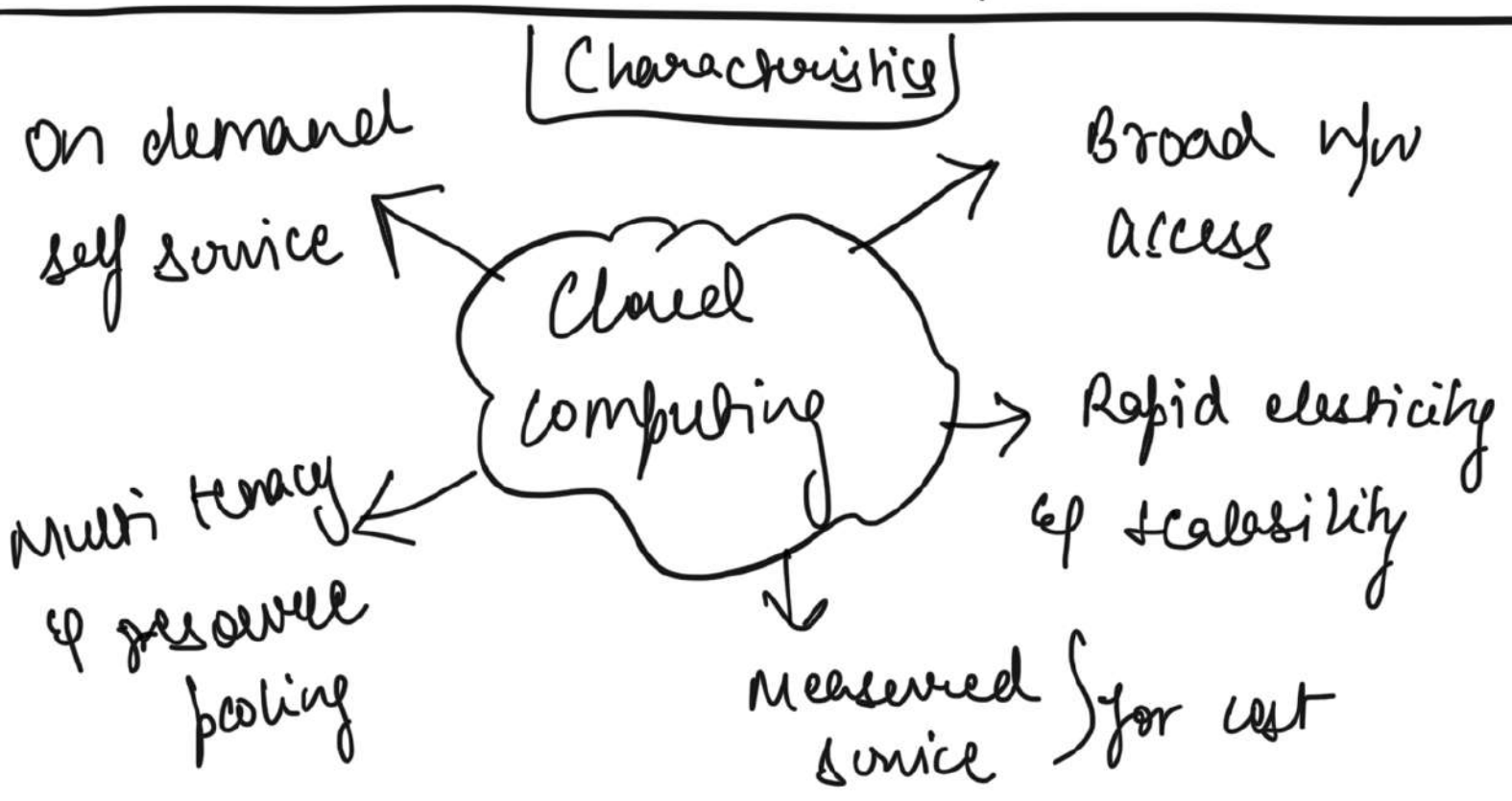
Deployment models of cloud

Private
↓
used by single org.
complete control
security for sensitive apps
for specific business req.

Public
↓
azure, gcp, aws
operated by someone else

Hybrid
↓
Keep some servers on premises & extend some capabilities to cloud
control over sensitive infra in pvt infra
flexible & cost effective like public cloud

Characteristics



Advantages

(1) Trade CAPEX for OPEX

① Trade CAPEX for OPEX

② Benefit from massive economies of scale

③ Stop guessing capacity

④ increased speed & agility

⑤ Stop spending money on data centres

⑥ go global in mins

Types of Cloud Computing

① IaaS – Terraform, EC2 (Infra) ← manage Apps, data, runtime, OS, db

② Paas – (Platform) – Beanstalk ← classic ↗ manage app, data

③ Saas – (Software) – many AWS services - gmail, zoom ↓ manage nothing

Pricing on cloud:

Compute

Pay for compute time

Storage

Pay for data stored in cloud

data transfer OUT of Cloud

data (IN) is free

How to choose AWS regions?

→ Compliance – with data governance & legal requirements

→ Proximity to customers to reduce latency

→ available services within a region

→ Pricing

AWS availability Zones

→ each region has AZ → 3 to 6

→ each AZ is one or more discrete data center with redundant power, networking & connectivity

→ they're separate from each other, so isolated from disaster

→ if something happens in 1 AZ, that'll not be replicated to other AZs

→ They're connected with high bandwidth, ultra low latency n/w

Shared Responsibility Model

Customer is responsible for security (in) the cloud

→ data
→ platform, apps, IAM
→ OS, n/w & firewall config
→ client side data encryption, & data integrity auth, server side encryption, networking traffic protection,

AWS responsible for: security (of) the cloud

→ software — compute, storage, db, n/w ing
→ Hardware / infra — regions, AZ, edge locations

IAM - identity access management

- global service
- groups only contain users not grps
- user can belong to 0 or n grps
- users / grps can be assigned policies
- in aws - you apply [least privilege principle]

- Iam policy structure: (op)

Consists of :→ version, id, statement

statement ↓

Consists of → id, (optional)
- effect, ← allow / deny
- principal, ← ace / user / role
- Action, ← list of actions
- resource, ← list of resources to which action applies
- condition (optional)

- Iam Password Policy – like min len, letters, numbers ?, every 90 days. etc.
no reuse

- MFA – password + security device
  - U2F key
  - H/w key fob device
  - Key fob id

To access AWS

1. aws mgmt console – mfa
2. cli – access keys
3. SDK – ''

- user manage their own access key
- key generated through console

→ Iam roles for services

- assign permissions to aws services with Iam roles
- ex.: ec2 instance roles, lambda roles
- some services need permission to do things on our behalf

→ Iam security tools

- credentials report (account level)
  ↳ report that lists all account users & status of their various credz
- access advisor (user level)
  ↳ shows service permission granted to user & when these services were last accessed

→ IAM best practices

- don't use root acc
- assign users to groups & assign permissions to group
- create a strong pwd policy
- use & enforce MFA
- create & use roles for giving permissions to

— create & use roles for giving permissions to aws services

— use access keys for cli/sdk

Shared Responsibility for Iam

aws

- infra (global nw security)
- config & vulnerability analysis
- compliance validation

users

- user, group, roles, policies mgmt & monitoring
- enable mfa on all acc
- rotate all keys often
- use iam tools
- analyze access patterns & review permissions

EC2 – elastic Compute Cloud

– IaaS

ec2 sizing & config options : (can choose)

→ OS : linux, windows or macOS

→ CPU? → RAM? → storage space?

→ nw? → hw

→ n/w: speed of card
→ Public IP

n/w attached (ebs / efs)

n/w (ec2 instance store)

→ firewall: security grp

ec2 instance types → (7) types

	VCPU	mem	n/w performance
t2.micro	1	1	low to moderate
t2.xlarge	4	16	moderate
c5d.4xlarge	16	32	up to 10 gbps
r5.16xlarge	64	512	20 gbps
m5.8xlarge	32	128	10 gbps

If you stop an instance & start it again, AWS may change it's public IP address

naming convention: m5.2xlarge
- m: instance class
- 5: generation
- 2xlarge: size within instance class

General Purpose instance types

- great for diversity of workloads
- balance b/w: compute | memory | networking

Compute optimised

- best for compute intensive tasks that require high performance processors
- ex: batch processing workloads
  media transcoding
  high performance web servers
  " " computing etc

Memory optimized

→ fast performance for workloads that process large data set in memory
→ ex: high performance db
  in memory db for BI
  distributed web cache

R X M ∃ naming

Storage optimized

→ great for storage intensive tasks that require high sequential read & write access to large data sets on local storage
→ ex: relational & no-sql db
  high ∨ online transaction processing
  data warehousing
  distributed file system

in memory cache

Security Groups in AWS

- fundamental of n/w security in AWS
- firewall around ec2 instance
- control traffic flow - in & out of ec2
- security groups only contain allow rules
- can reference IP or SG
- SG regulate:
  - access to ports
  - authorized IP ranges - IPv4 & IPv6
  - control of inbound & outbound n/w
- can be attached to multiple instances
- locked down to a region/VPC combo
- dies live outside EC2 - firewall
- ⭐ good to maintain separate SG for SSH
- all inbound traffic is blocked by default
- if app time out - SG issue
- if connection refused - app error, SG worked
- all outbound traffic is authorized by default

Classic ports to know

1. 22 = SSH (secure shell) - log into linux instance
2. 21 = FTP (file transfer protocol) - upload files into file share
3. 22 = SFTP (simple file transfer protocol)

- upload files using SSH

4. 80 = HTTP — access unsecured websites
5. 443 = HTTPS — access secured
6. 3389 = RDP (remote desktop protocol) — log into windows instance

EC2 instance purchasing options

- On demand instances
- Reserved (1 & 3 yrs) → reserved
  convertible
- Saving plan (1 & 3 yrs) — commitment to amt of usage, long workload
- Spot instances - short workload, cheap
- dedicated host — book entire physical server
- dedicated instances - no other customer will share your h/w
- Capacity reservations

Shared Responsibility Model for EC2

AWS	User
- infra - isolation on physical host - Replace faulty h/w - compliance validation	- SG rules - OS patches & updates - S/w & utilities

-

installed on ec2
- iam roles
- data security on instance

EC2 → AMI (OS) + instance size (CPU + RAM)
+ storage + sg + EC2 user data

EC2 instance storage : EBS

EBS volume – elastic block store volume
– n/w drive attach to instance
– persist data even if instance terminate
– 1 ebs to 1 ec2
– bound to AZ
– have a provisioned capacity
– delete on termination attribute

By default : for root → enabled
for iam disabled

ebs snapshots – backup
– can copy snapshot across AZ

snapshot archive → ¾ cheaper
→ 24hr to 72hr to restore

AMI – Amazon Machine Image

- customization of an ec2 instance
- built for specific region & can be copied across regions
- launch ec2 instances from:
  - public ami
  - own ami
  - aws marketplace ami

EC2 image builder

→ automate creation, maintainence, validation & test for ec2 amis
→ free service (pay for underlying resources)

ec2 instance store → for high performance h/w disk

→ ebs volumes are n/w drives with good but limited performance

efs → elastic file system
→ managed n/w file system
→ can be mounted on 100s of ec2
→ works with linux ec2 instance in multi AZ
→ highly available, expensive, scalable

ebs ↓ vs efs ↓

1 AZ ↔ many instances in
1 instance many AZ

efs infrequent access – cost optimized
– no one used these
files in last 60 days

Shared responsibility

aws
- infra
– replication of data
– replace faulty h/w

user
– backup / snapshot
– data encryption
– responsibility of any data on drive

Amazon fsx – launch 3rd party high performance file system on AWS

– fully managed service
– fsx for lustre,
for windows file server,
for NetApp ONTAP

lustre
– built on windows file server
– supports SMB protocol & windows NTFS
linux + cluster
HPC

EC2 Instance Storage - Summary

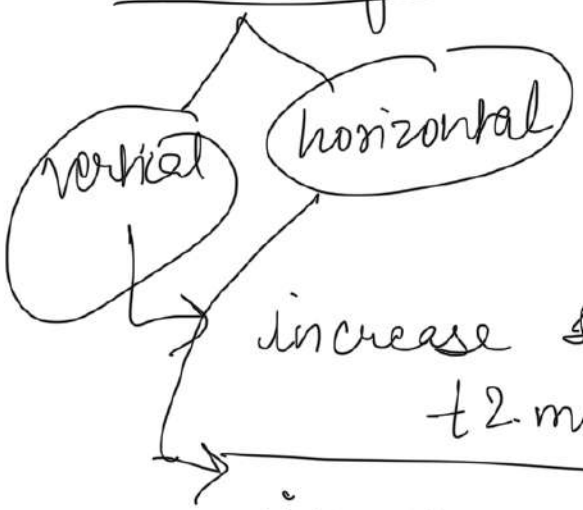
- EBS volumes:
  - network drives attached to one EC2 instance at a time
  - Mapped to an Availability Zones
  - Can use EBS Snapshots for backups / transferring EBS volumes across AZ
- AMI: create ready-to-use EC2 instances with our customizations
- EC2 Image Builder: automatically build, test and distribute AMIs
- EC2 Instance Store:
  - High performance hardware disk attached to our EC2 instance
  - Lost if our instance is stopped / terminated
- EFS: network file system, can be attached to 100s of instances in a region
- EFS-IA: cost-optimized storage class for infrequent accessed files
- FSx for Windows: Network File System for Windows servers
- FSx for Lustre: High Performance Computing Linux file system

Stephane Maarek

So to say for this section, I hope you liked it

ELB & ASG – elastic load balancing & auto scaling groups

Scalability



increase size of instance
t2.micro → t2 large
} common in non distributed system

increase no. of instances
applies to distributed systems

High availability – goes hand in hand with horizontal scaling

→ running app in atleast 2 AZ
→ to survive a data center loss

scalability vs elasticity vs agility

Scalability : ability to make h/w stronger
by making h/w stronger (scale up)
or by adding nodes (scale out)

elasticity : if system scalable, elasticity means
there will be "auto scaling"
→ cloud friendly

Agility : new IT resources are a click

away → less time taken to make resources available to dev

Load Balancing

load balancers are servers that find internet traffic to multiple servers (ec2) downstream

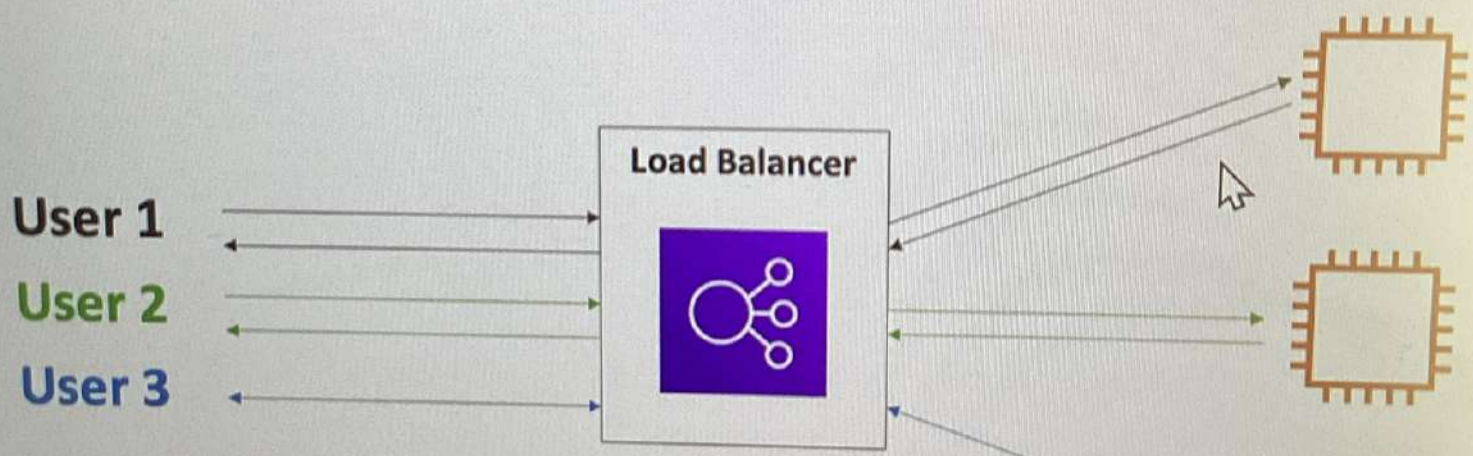
GitHu | Cloud | AWS-C | AWS S | AWS C | Cloud | AWS S | Search | AWS C

/course/aws-certified-cloud-practitioner-new/learn/lecture/20055864#notes

WS Certified Cloud Practitioner CLF-C02

What is load balancing?

- Load balancers are servers that forward internet traffic to multiple servers (EC2 Instances) downstream.



across multiple EC2 instances

phane Maarek

Notes | Announcements | Workspaces ★ Pro | Reviews | Learning tools

Create a new note at 1:08

All lectures | Sort by most recent

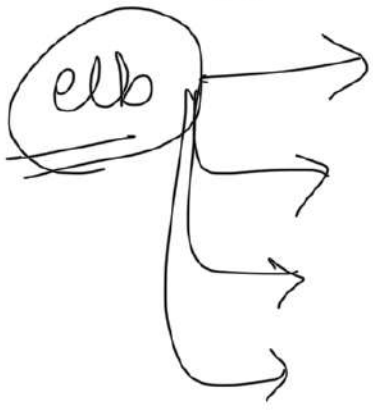
Search

DELL

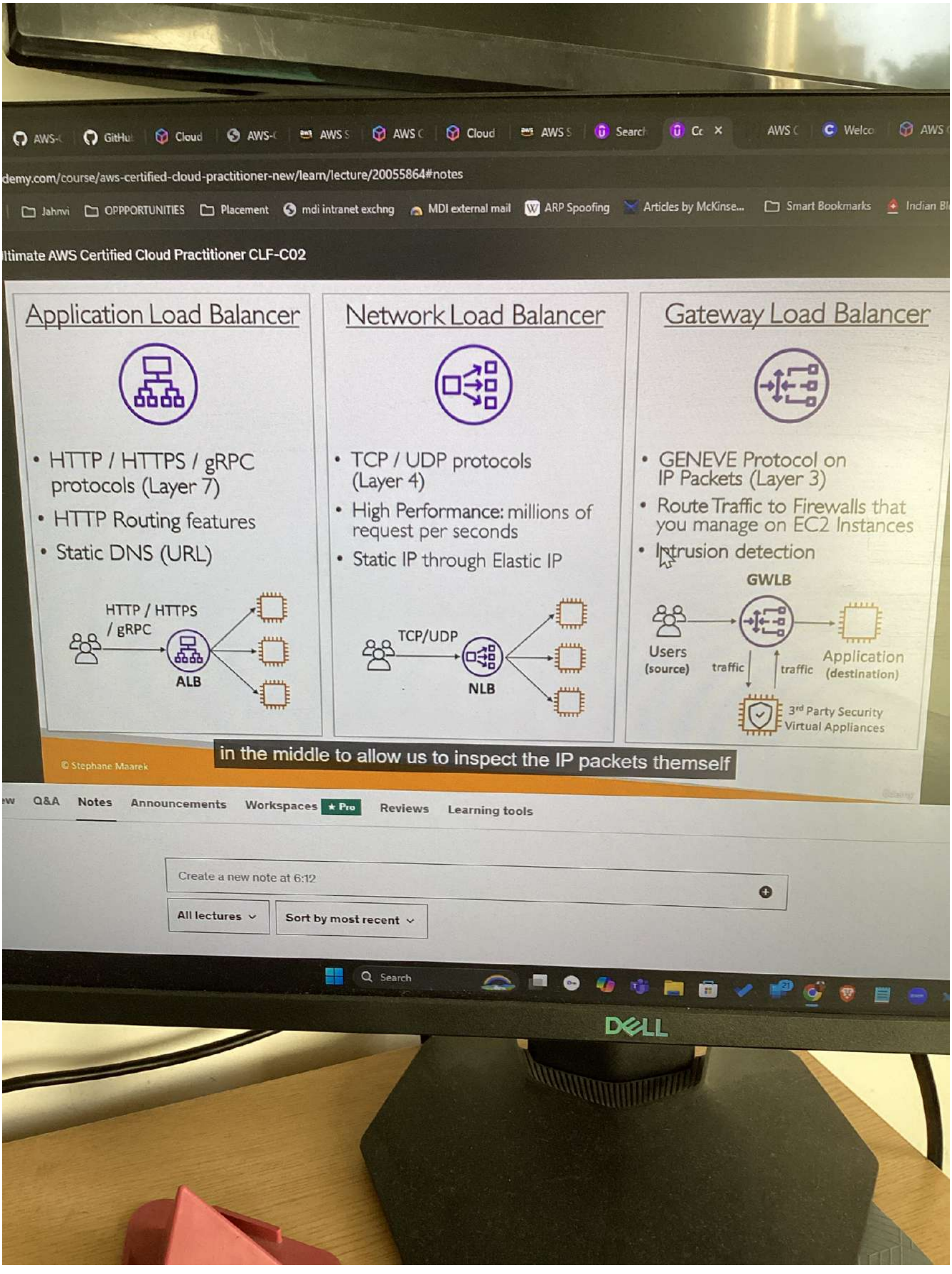
Why to use load balancer?

- Spread load across multiple downstreams
- expose a single point of access (DNS) to your application
- seamlessly handle failures
- do regular health check to your instances
- provide ssl (https for your website
- high availability

Elastic load balancer → can be multi AZ
- managed load balancer
- aws guarantees everything
- cost less to set up own elb but then all maintenance upto you
- supports health checks



- application lb - layer 7 (http/ https)
- n/w lb - layer 4 (ultra high perf, allow TCP)
- gateway lb - layer 3
- classic lb - layer 4 & 7 (retired)



Low latency in nlb

Auto Scaling Groups

→ goal is to scale out/ scale in automatically
& ensures to run a min/max instance at a time
→ replace unhealthy grps
→ automatically register new instance
→ Cost optimized

min size | actual / desired capacity | max size

ASG strategies of scaling

① Manual scaling – update size of ASG manually
② Dynamic scaling – respond to changing demand
   i) simple / step scaling
      → if alarm triggered for threshold increase, add 2 units
      → if alarm triggered for less, remove unit
   ii) target tracking scaling
      → asg will scale up/down automically to cater to set thresholds
   iii) scheduled scaling
      → anticipate scaling based on known usage pattern
   iv) Predictive scaling → uses ML to predict

future traffic based on past traffic

→ automatically provisions right # EC2 in advance

(Buckets)

Amazon S3 – simple storage service

– infinitely scalable storage

Use cases – backup & restore
– disaster recovery
– archive
– hybrid cloud storage
– application hosting, media hosting
– data lakes & big data analytics
– s/w delivery
– static website

[Nasdaq stores 7 yrs of data into S3 glacier
[Sysco runs analytics on its data & gain insights

→ store (files) (=objects) into "buckets"
→ buckets must have globally unique name across all regions all accounts
→ defined at region level

Naming convention : no uppercase, no underscore
3-63 char long
not an IP

must start with lowercase letter/no.

prefix ≠ xn--

suffix ≠ -s3alias

S3 objects → - are files & have a key

- key = FULL path = prefix + obj-name
- no concept of directories
- everything is a key

Object values are content of body:

i) max object size = 5TB

ii) for size > 5TB, use multipart upload

Metadata ⇒ list of key value pairs

Tags ⇒ unicode key value pair — useful for security / lifecycle

version ID ⇒ (if versioning enabled)

S3 - security

user based → IAM policies

resource based → bucket policies (common)

→ object access control list

→ bucket access control list

Iam principle can access an S3 object if the user Iam permission allow it or resource policy allows it and there's no explicit deny

S3 bucket policies → json based policy

(1) resource block → Specify bucket / objects

(2) effect → allow / deny

(3) principal → acc / user to which policy applies to

by default → all public access is blocked to s3

→ can be set at acc level

S3 — static website hosting

→ website URL will depend on region

→ enable public reads on this bucket

versioning in S3 — bucket level

— same key overwrite will change version

— best practice to version your buckets

— easy to roll back

— protect against unintended deletes

— restore any version

— any file that's not version is versioned "null"

— suspending versioning doesn't delete previous version

S3 replication — CRR & SRR

(cross) (same)

- async replication — enable versioning
- buckets can be in diff aws acc
- iam read & write permit repl

use cases

(1) CRR: compliance, lower latency access, replicate across acc

(2) SRR: log aggregation, live replication b/w prod & test acc

S3 storage classes

→ S3 general purpose
→ S3 infrequent access (IA)
S3 one zone - IA
S3 glacier instant retrieval
glacier flexible retrieval
glacier deep archive
intelligent tiering

S3 durability & availability

durability — high durability of obj across multi AZ
— same for all storage classes
— lose 1 obj every 10k yrs

availability — measures how readily available a service is

- varies depending on storage class
- 99.99% availability
- not available 53 min/yr

S3 standard
- general purpose
- 99.99% availability
- used for frequently accessed data
- low latency + high throughput
- sustain 2 concurrent facility failures

S3 IA
- require rapid access when needed
- lower cost than S3 standard

standard IA
- 99.9% availability
- DR backups

one zone IA
- high durability - single AZ
- data lost if AZ destroyed
- 99.5% available
- store secondary copy of data or data that can be recreated

S3 glacier storage classes

- meant for archive / backup
- low cost object storage
- pay for storage + object retrieval cost

(1) glacier instant retrieval

- millisecond retrieval — once a objt access
- min storage for 90 days

(2) S3 glacier flexible retrieval
- expedited, std, bulk — free (5-12 hrs)
- 90 day

(3) glacier deep archive — for long term storage
- std, bulk (48 hrs)
- 180 days

S3 encryption — may be 1 ques

server side encryption
→ a S3 secures that objects that you upload
→ by default ON

client side
→ client encrypts the file before uploading file

Iam access analyzer for S3

- ensures that only intended ppl have access to your S3 buckets
- dashboard for bucket policies kinda

Shared responsibility for S3

aws
- infra
- config & vulnerability analysis

user
- S3 versioning
- bucket policies

- compliance validation
- replication setup
- logging & monitoring
- storage classes
- data encryption at rest & in transit

AWS snow family

- highly secure devices to collect & process data at edge & migrate data into & out AWS

- snowcone
  - small device + small storage
  - 8TB - 14TB
- snowball edge
  - 80TB - 210TB
  - PB data

challenges → limited h/w, connectivity
high n/w cost
shared b/w
Connection stability

- offline devices to perform data migrations
- if it takes over a week to transfer over n/w - use snowball devices

edge computing - preprocess data
ML, transcoding

Storage gateway

(1) AWS is pushing for hybrid cloud
- Some infra on prem & some on cloud

Can be due to:
- long cloud migration
- IT strategy
- Security req
- compliance req

- S3 is proprietary storage tech
- use storage gateway to export data on prem (*)

Storage cloud native options

block	file	object
amazon ebs	amazon efs	S3, glacier
EC2 instance storage		

(*)
→ Storage gateway bridge on prem data of S3 cloud data
↳ file, volume, tape

Amazon S3 – Summary

- **Buckets vs Objects:** global unique name, tied to a region
- **S3 security:** IAM policy, S3 Bucket Policy (public access), S3 Encryption
- **S3 Websites:** host a static website on Amazon S3
- **S3 Versioning:** multiple versions for files, prevent accidental deletes
- **S3 Replication:** same-region or cross-region, must enable versioning
- **S3 Storage Classes:** Standard, IA, 1Z-IA, Intelligent, Glacier (Instant, Flexible, Deep)
- **Snow Family:** import data onto S3 through a physical device, edge computing
- **OpsHub:** desktop application to manage Snow Family devices
- **Storage Gateway:** hybrid solution to extend on-premises storage to S3

Databases & Analytics

1. storing data on disk can have its limits
2. can structure the data
3. build indexes to efficiently query / search through data
4. define relationships b/w datasets

(I) Relational Databases
- looks like excel spreadsheets with links
- Can use SQL queries

(II) NOSQL db - non relational db
- built for specific data models & have flexible schemas for building modern apps
- Benefits :
  i) flexibility - easy to evolve dbs
  ii) scalability - scale out by using distributed clusters
  iii) high performance - optimized for specific data model
  iv) highly functional -

ex: key value, document, graph db
Json

Shared responsibility on AWS

- AWS offers use to manage dfg databases
- benefits include:
  - → Quick Provisioning — high availability
    vertical scaling
    horizontal scaling
  - → Automated backup & restore, operations
  - → OS patching is handled by AWS
  - → Monitoring + alerting

(iii) ## RDS & Aurora

- managed db service — use SQL
- allows you to create dbs in cloud managed by AWS

dbs: Postgres, mysql, aurora, mariadb

Adv of RDS over DB on ec2

- managed service
- automatic provisioning
- os patching
- continuous backup & restore
- monitoring
- read replicas
- multiAZ for DR
- scaling
- storage backed by ebs
- NO SSH
- multi AZ

Amazon (Aurora) → AWS tech not open source

- PostgreSQL & MySQL are both supported as

aurora dbs
- aurora is cloud optimized
- storage grow automatically in increment of 10gb
- not in free tier

aurora serverless — db instantiation automated
- both SQL supported
- least mgmt overhead
- Pay per second — cost effective

RDS deployments : Read Replicas, multi AZ

Read Replicas → scale read workload of dbs
(RR) — can create upto 15 read replica
— data written to main DB only

Multi AZ → failover in case of AZ outage
→ can only have 1 AZ as failover

Multi Region → RR in multi region
→ DR in case of region issue
→ local performance for global reads
→ replication costs

(IV) Elasticache → db type
↳ used to get managed Redis or Memcached
→ in memory databases with high perf & low latency
→ reduces load off dbs for read intensive workloads

V) DynamoDB – fully managed & highly available with replication across 3 AZ

- NoSQL database – distributed serverless db
- Fast & consistent in performance
- low latency retrieval, low cost, auto scale
- std & IA class
- key : value db

↓

- Primary key → Partition key + sort key
- schema defined per item

VI) Dynamodb Accelerator – DAX

- fully managed in memory cache for db
- if we want to access cached items
- DAX only with ddb

Global tables → makes ddb table accessible with low latency in multiple regions

→ Active Active replication (read/write to any AWS region)

VII) Redshift – based on Postgresql
– not for OLTP

Online transaction Protocol

RDS suitable ✗

OLAP – Online analytical processing
(analytics of data warehousing)
- load data once 1 hour
- Columnar storage of data
- Parallel Query Execution
- has SQL interface for performing queries
- BI tools integrated

(VIII) Redshift serverless

- automatically provision of scale data warehouse
- no need to manage underlying capacity

(IX) Amazon EMR – elastic map reduce
- helps creating hadoop cluster (big data)
to help analyze & process vast amt data
- Apache Spark, Presto, flink ...
- emr takes care of all provisioning & config
- auto scaling & integrated with spot instance
- use case: data process, ML, web indexing ...

(X) Athena – serverless query servie to perform analytics against S3 objects
- uses std SQL to query files

- support csv, json, ..

$5.00 / TB

use columnar data for cost saving

(*) analyze data in serverless S3 using athena

(XI) Amazon Quicksight → report & dashboards
- auto scale
- B) in AWS - embeddable
- integrated with so many db

(XII) Document db overview - aws version of mongo db
↓
mongo db
→ to store & query & index JSON data
- fully managed - replicate across 3 AZ
- automatically scales to workloads with millions of request per second

(XIII) Neptune - fully managed graph db
- highly available
- 3 AZ upto 15 RR
- used for highly connected db
- can store upto billions of relations & query graph with ms latency
- great for Knowledge graph, fraud detection

(XIV) Timestream — for time series data
— for data evolving overtime
— store trillions of data
— built in analytics func

(XV) QLDB — Quantum ledger DB
— ledger : book recording financial transactions
— review history of changes made to your app data over time
— immutable system
— helpful for financial transact
— no decentralization component
— have central authority

(XVI) Amazon managed blockchain
— decentralization here unlike QLDB
— no central authority

(XVII) Amazon glue
— managed extract, transform & load (ETL) service
— transform data for analytics
— serverless

(XVIII) Data migration service — DMS

- Quick & secure db migration
- Source is available during migration

Databases & Analytics Summary in AWS

- Relational Databases - OLTP: RDS & Aurora (SQL)
- Differences between Multi-AZ, Read Replicas, Multi-Region
- In-memory Database: ElastiCache
- Key/Value Database: DynamoDB (serverless) & DAX (cache for DynamoDB)
- Warehouse - OLAP: Redshift (SQL)
- Hadoop Cluster: EMR
- Athena: query data on Amazon S3 (serverless & SQL)
- QuickSight: dashboards on your data (serverless)
- DocumentDB: "Aurora for MongoDB" (JSON – NoSQL database)
- Amazon QLDB: Financial Transactions Ledger (immutable journal, cryptographically verifiable)
- Amazon Managed Blockchain: managed Hyperledger Fabric & Ethereum blockchains
- Glue: Managed ETL (Extract Transform Load) and Data Catalog service
- Database Migration: DMS
- Neptune: graph database
- Timestream: time-series database

RDS multi AZ deployment main purpose is high availability

RR main purpose is scalability

Multi region deployment main purpose is DR & local performance

---

Docker - S/w development platform to deploy apps

- apps are packaged in containers that can be run on any OS
- apps run same no matter "where"
  - any machine - no compatibility issues
  - predictable behaviour
- Scale containers easily

docker images are stored in docker repo or amazon ECR (pvt)

ECS - elastic container service
- launch docker container on AWS
- You must provision & maintain infra
- AWS takes care of starting / stopping container
- has elb integration

Fargate - launch docker on aws
- serverless

— AWS manages infra
— AWS run containers

ECR — elastic container registry
— Pvt registry docker

serverless = function as a service (FaaS)

AWS Lambda — virtual func : no server to manage
— limited by time : short executions
— run on demand
— Scaling automated

Benefits of aws lambda

→ easy pricing
— multi language
- easy monitoring
- cheap

→ event driven (func get invoked when needed)
- easy to get more resources per functions

Amazon API gateway — fully managed
— easily create, publish, maintain, monitor & secure API
- serverless & scalable

AWS Batch — fully managed batch processing at any scale

- batch job start & end
- batch will dynamically launch ec2 instances or spot instances
- provisions the right amt of compute/memory
- batch jobs are defined as docker img & run on ecs

Batch:
- No time limit
- Any runtime as long as it's packaged as a Docker image
- Rely on EBS / instance store for disk space
- Relies on EC2 (can be managed by AWS)

Lambda:
- Time limit
- Limited runtimes
- Limited temporary disk space
- Serverless

Lightsail — virtual servers, storage, db & n/w
- low & predictable pricing
- alternative to ec2, rds, elb, route 53
- no cloud exp can start with this

most likely @ distractor in exam

Cloud integrations (A)

Amazon SQS – Standard Queue

- Oldest AWS offering (over 10 years old)
- Fully managed service (~serverless), use to **decouple** applications
- Scales from 1 message per second to 10,000s per second
- Default retention of messages: 4 days, maximum of 14 days
- No limit to how many messages can be in the queue
- **Messages are deleted after they're read by consumers**
- Low latency (<10 ms on publish and receive)
- Consumers share the work to read messages & scale horizontally

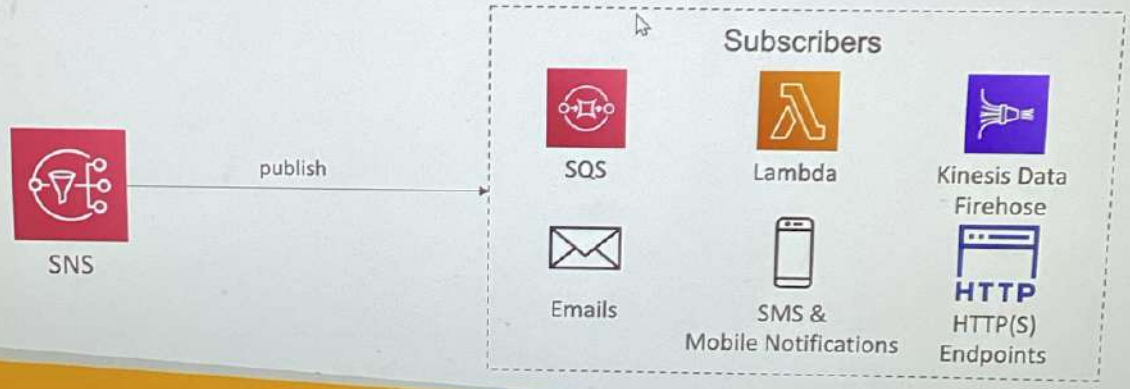
Amazon Kinesis

- For the exam: Kinesis = real-time big data streaming
- Managed service to collect, process, and analyze real-time streaming data at any scale

- Too detailed for the Cloud Practitioner exam but good to know:
  - Kinesis Data Streams: low latency streaming to ingest data at scale from hundreds of thousands of sources
  - Kinesis Data Firehose: load streams into S3, Redshift, ElasticSearch, etc…
  - Kinesis Data Analytics: perform real-time analytics on streams using SQL
  - Kinesis Video Streams: monitor real-time video streams for analytics or ML

Amazon SNS

- The "event **publishers**" only sends message to one SNS topic
- As many "event **subscribers**" as we want to listen to the SNS topic notifications
- Each subscriber to the topic **will get all the messages**
- Up to 12,500,000 subscriptions per topic, 100,000 topics limit



Amazon MQ

- SQS, SNS are "cloud-native" services: proprietary protocols from AWS
- Traditional applications running from on-premises may use open protocols such as: MQTT, AMQP, STOMP, Openwire, WSS
- When migrating to the cloud, instead of re-engineering the application to use SQS and SNS, we can use Amazon MQ
- Amazon MQ is a managed message broker service for

RabbitMQ™ ACTIVEMQ

- Amazon MQ doesn't "scale" as much as SQS / SNS
- Amazon MQ runs on servers, can run in Multi-AZ with failover
- Amazon MQ has both queue feature (~SQS) and topic features (~SNS)

Deployment & Managing infra at scale

Cloud formation

→ declarative way of outlining AWS infra for any resources
→ you define need in CF template & CF creates those resources in right order

Benefits

(1) IAC
(2) cost → each resource is tagged with an identifier
→ estimate cost with template
→ automation of deletion / creation
(3) Productivity – auto generation + declarative programming
(4) don't re-invent wheel
(5) support

For templates, used application composer
→ see all relation of resources
→ helpful when re-create resources in diff env

AWS cloud dev Kit (cdk)

→ define cloud infra using familiar lang.
→ then code compiled into CF template (json/

yaml)

→ deploy runtime + app code together
  – great for lambda / ecs / eks

elastic
Beanstalk → typical arch : web app 3 tier

→ developer centric view
→ uses all component of AWS – full control over config
→ Platform as a service (PaaS)
→ free but pay for underlying services
→ managed service
→ user responsibility = code only

3 architecture models
  → single instance deployment (for dev)
→ db + asg → prod or preprod
→ ASG only → for non web app in prod

→ support many lang / platforms
→ have health monitoring – publish health event to cloudwatch

AWS CodeDeploy → to deploy app automatically

→ works with ec2 or onprem servers (hybrid service)
→ servers / instances must be provisioned & configured ahead of time with codedeploy

agent

Code Commit → discontinued

→ way to store app code before pushing to server
→ like git repo
  → easy to collaborate
→ fully managed + private to account

AWS Code Build → build code on cloud

Code pipeline → define a way to have code automatically pushed to prod
  → CI/CD

Code artifact → storage & retrieval of dependencies like fw packages is called artifact management

AWS Systems manager (ssm)

→ manage ec2 + on prem system ⇒ hybrid
→ get operational insights about state of infra
→ imp features:
  → patching automation for enhanced compliance
  (A) → run commands across entire fleet of servers
  → store param config with ssm param store
→ linux, windows, mac OS, raspbian

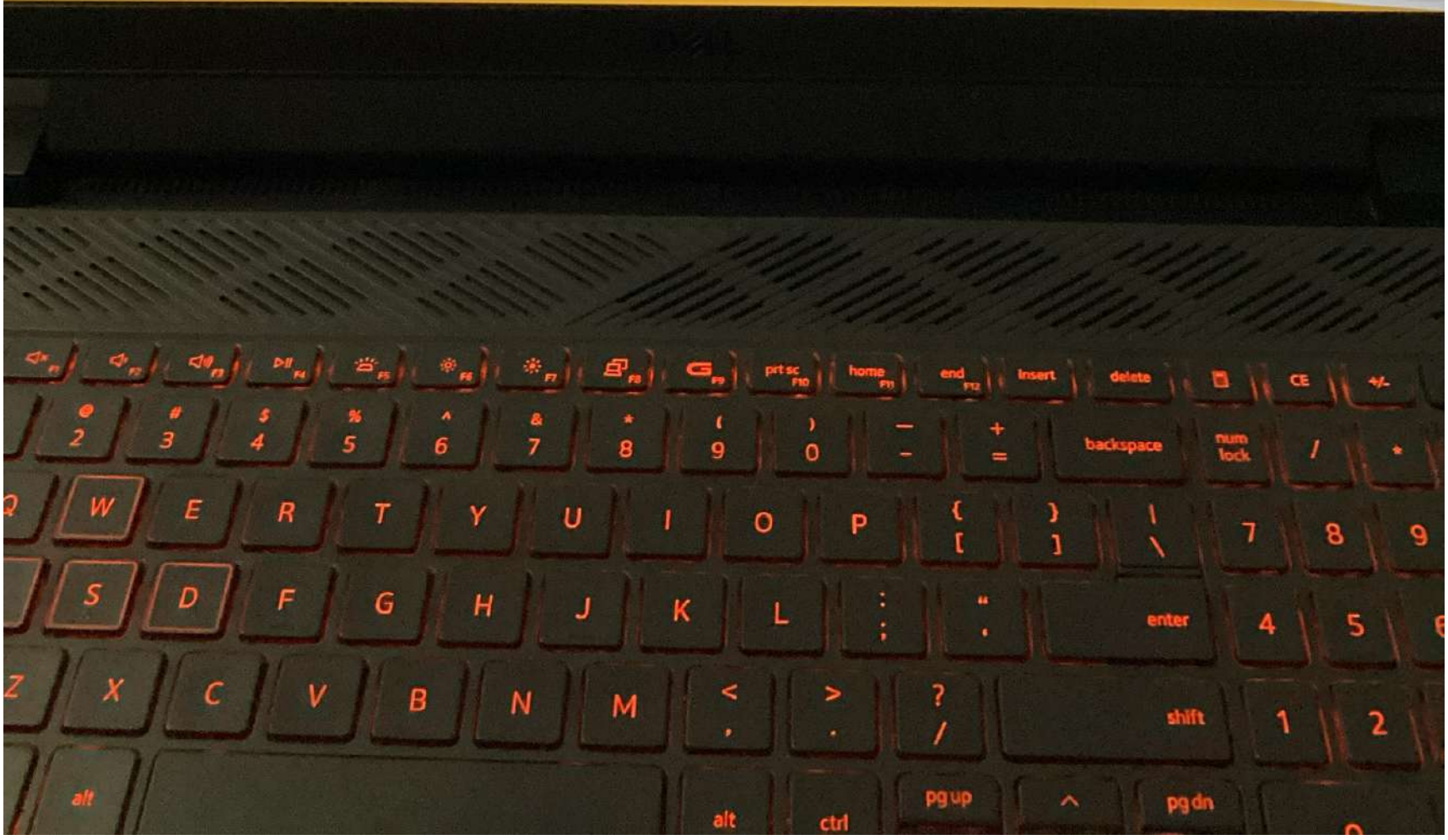
SSM Session manager

→ allow to start secure shell on your ec2
  & on prem server
→ no ssh access, no port 22
→ linux, macOS & windows
→ send data to S3/ cloudwatch

SSM Param store – secure storage for config & secrets
  – store API keys, pwd, config
  – serverless, scalable, durable, sdk
  – control access permit using iam
  – version tracking & encryption

Deployment - Summary

- CloudFormation: (AWS only)
  - Infrastructure as Code, works with almost all of AWS resources
  - Repeat across Regions & Accounts
- Beanstalk: (AWS only)
  - Platform as a Service (PaaS), limited to certain programming languages or Docker
  - Deploy code consistently with a known architecture: ex, ALB + EC2 + RDS
- CodeDeploy (hybrid): deploy & upgrade any application onto servers
- Systems Manager (hybrid): patch, configure and run commands at scale



Developer Services - Summary

- CodeCommit: Store code in private git repository (version controlled)
- CodeBuild: Build & test code in AWS
- CodeDeploy: Deploy code onto servers
- CodePipeline: Orchestration of pipeline (from code to build to deploy)
- CodeArtifact: Store software packages / dependencies on AWS
- AWS CDK: Define your cloud infrastructure using a programming language

Global Infrastructure

→ global app is deployed in multiple geographies
→ on regions / edge locations
→ decreased latency - deploy app closer to users
→ disaster recovery - increase availability of your app
→ attack protection - distributed global infra is hard to attack

1) global DNS - route 53
   → great for DR → route users to closest deployment with least latency
2) global content delivery n/w (CDN): cloudfront
   → replicate part of app to aws edge location
3) S3 transfer acceleration
4) aws global accelerator - improve global app availability

Route 53 - managed DNS

- DNS = rules & records which helps client understand how to reach server through url
- most common records in aws:
  1) website to IPV4: A record
  2) website to IPV6: AAAA record
  3) hostname to other hostname: CNAME
  4) ... to aws record: Alias

Routing Policy (RP)

1) simple RP – no health checks
2) weighted RP – allow to distribute traffic (load balancing) across multiple instances
   – health checks ✓
3) latency RP – look where user is located & redirect them to nearest server
4) failover RP – DR
   – health check on primary, in case of failure, switched to secondary

Cloud Front – CDN

– improves read performance, content is cached at edge
– improves user exp
– 216 points of presence globally (edge locn)
– DDOS protected ↓ integration with shield, aws WAF (firewall)

Cloud front origins

1) S3 bucket – for distributing files & caching
   – enhanced security with CF Origin access control (OAC)
   – CF can be used as ingress (to upload files to S3)

2) Custom origin (http)
   - application load balancer
   - ec2 instance
   - S3 website - any http backend

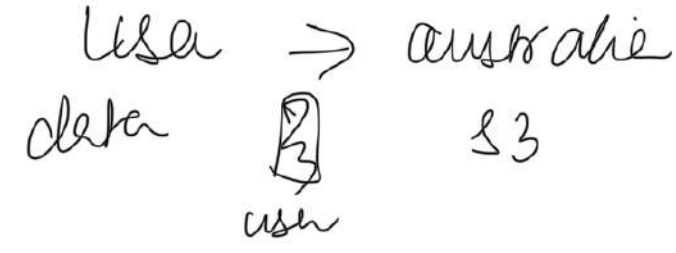
CF vs S3 CRR

CF
→ global edge n/w
→ files are cached for ttl
→ great for static content that must be available everywhere

S3
→ set up for each region where you want replication
→ read only
→ file updated in real time
→ no caching
→ great for dynamic content that needs to be available at low latency in few regions

S3 transfer acceleration

→ increase transfer speed by transferring content to nearest edge location which find files to S3 in nearest region



AWS global accelerator - improve global app availability & performance

(gw)

using pvt aws n/w

vs CF → both use edge location & pvt aws n/w
→ both DDOS protection

CF is CDN - caches content
- content served at edge

ga → no caching
→ proxy pkt at edge
→ improves performance over tcp/udp
→ good for dtlp usecase

AWS Outposts - server racks that offer same aws infra, services, API to build your own app on prem like cloud

Benefits → low latency
local data processing
data residency
fully managed
easier migration

Some services that use outposts:
ec2, s3, ebs, eks, ecs, rds

AWS wavelength (↑)

→ Zones are, infra deployments embedded within

telecomm providers' datacenters at edge of 5G n/w

- high b/w & secure connection to parent region
  - very close to edge

use cases: ar/vr, gaming, smart cities

AWS

local zones — allows you to place services closer to end users to run latency sensitive apps
- extend aws regions

global app architecture

single region, single AZ

single region, multi AZ

multi region, active passive (only read in passive)

" " active active (read/write all in passive & active)

bal Applications in AWS - Summary

- Global DNS: Route 53
  - Great to route users to the closest deployment with least latency
  - Great for disaster recovery strategies
- Global Content Delivery Network (CDN): CloudFront
  - Replicate part of your application to AWS Edge Locations – decrease latency
  - Cache common requests – improved user experience and decreased latency
- S3 Transfer Acceleration
  - Accelerate global uploads & downloads into Amazon S3
- AWS Global Accelerator
  - Improve global application availability and performance using the AWS global network

Global Applications in AWS - Summary

- AWS Outposts
  - Deploy Outposts Racks in your own Data Centers to extend AWS services
- AWS WaveLength
  - Brings AWS services to the edge of the 5G networks
  - Ultra-low latency applications
- AWS Local Zones
  - Bring AWS resources (compute, database, storage, ...) closer to your users
  - Good for latency-sensitive applications

Cloud Monitoring

Amazon CloudWatch Metrics

→ for every service → timestamps
→ dashboard of metrics to see all at once

important metrics → EC2 instances : CPU, status check, N/W
default : 5 min

ebs volumes : disk read/writes

S3 buckets : bucket size bytes, no. of objects, all requests

billing : estimated charge (us east 1)

service limit

custom metrics

Cloudwatch alarms — to trigger notifications for any metric

- billing alarm
- [OK | INSufficient_Data | Alarm] type
- options : sampling, %, min, max

CW logs : can collect logs from:

elastic beanstalk

ecs

lambda

cloudtrail

hybrid **cw log agents** : on ec2 m/c or onprem

route 53

- real time monitoring
- adjustable for log retention

Eventbridge (cw events earlier)

- schedule: cron jobs (scheduled script)
- event pattern: event rules to react to a service doing something
- trigger lambda, send sqs/sns

default event bus / partner event bus / custom event bus

events from aws services	events from aws partners (zendesk, datadog)	own custom apps send own events

→ schema registry : model event schema
→ archive events sent to bus
→ replay archived events

Cloudtrail - provides governance, compliance & audit

for your aws account

- enabled by default
- get history of all API calls in AWS acc
- can put logs from CT to CW or S3
- trail can be applied to all regions or 1 region

X Ray

→ debugging in Prod — good old way is to test locally, print statements, redeploy

→ no way to see common view for entire arch

→ log analysis hard

⇒ use XRAY → do tracing + visual analysis for your app

Advantages
- troubleshooting performance
- understand dependencies
- pinpoint service issues
- find error & exceptions
- sla met?
- throttled?
- which service, what users?

Code Guru — ml service for automated code review + app performance recommendation

① Review static code analysis

reviewer (review)

- Java & python

profiler (recommendations)

→ app in prod

identify code inefficiency

anomaly detection

min overhead on app

AWS health dashboard

- show all regions + all service health
- historical info

for account :

- earlier Personal health dashboard
- provides alert & remediation guidance
- notifies to proactively look into scheduled activities
- global service
- alerts

Monitoring Summary

- CloudWatch:
  - Metrics: monitor the performance of AWS services and billing metrics
  - Alarms: automate notification, perform EC2 action, notify to SNS based on metric
  - Logs: collect log files from EC2 instances, servers, Lambda functions…
  - Events (or EventBridge): react to events in AWS, or trigger a rule on a schedule
- CloudTrail: audit API calls made within your AWS account
- CloudTrail Insights: automated analysis of your CloudTrail Events
- X-Ray: trace requests made through your distributed applications
- AWS Health Dashboard: status of all AWS services across all regions
- AWS Account Health Dashboard: AWS events that impact your infrastructure
- Amazon CodeGuru: automated code reviews and application performance recommendations

VPC & networking

IP addresses (ec2) (low) — IPV4 — Public + Pvt
— IPV6

⇒ elastic IP : allow you to attach a fixed public IPV4 to ec2

VPC — (pvt. nw) to deploy your resources (regional)

↓

within VPC — can have (subnets) : allow you to partition your nw inside VPC (AZ resource)

public subnet → accessible from internet
private → not " " "

route tables → to define access to internet & b/w subnets

internet gateway : help vpc instances connect with internet

Public subnet ⇒

NAT gateway / instances allow Pvt subnet to access internet

Network ACL & SG

(NACL)

→ NACL → firewall which control traffic from & to subnet

→ can have allow/deny rules

→ attached at subnet level

→ rules only include IP addresses

→ Security groups → firewall that control traffic to & from an ec2 instance

→ can have allow rules

→ rules include IP address + other SG

letwork ACLs vs Security Groups

Security Group	Network ACL
Operates at the instance level	Operates at the subnet level
Supports allow rules only	Supports allow rules and deny rules
Is stateful: Return traffic is automatically allowed, regardless of any rules	Is stateless: Return traffic must be explicitly allowed by rules
We evaluate all rules before deciding whether to allow traffic	We process rules in number order when deciding whether to allow traffic
Applies to an instance only if someone specifies the security group when launching the instance, or associates the security group with the instance later on	Automatically applies to all instances in the subnets it's associated with (therefore, you don't have to rely on users to specify the security group)

https://docs.aws.amazon.com/vpc/latest/userguide/VPC_Security.html#VPC_Security_Comparison

ane Maarek

VPC flow logs

→ Capture info about IP traffic going into your interfaces:
- VPC flow logs
- Subnet flow logs
- Elastic n/w interface flow logs

- helps to monitor & troubleshoot connectivity issues
- Capture n/w info from AWS managed interfaces
- Vpc flow logs data can go to S3, CW logs, ..

VPC peering → connect to two VPC privately using AWS n/w
→ make them behave as if they were in same n/w
→ must not have overlapping CIDR
→ not transitive

VPC endpoints → allow to connect aws services using pvt n/w instead public
→ enhanced security + low latency
→ vpc endpoint gateway : S3 + dDB
" " interface : rest

AWS **PrivateLink** → secure + scalable
→ does not require vpc peering / NAT

→ talk to 3rd party vpc
→ requires a load balancer + customer vpc (service vpc)

site to site vpn & direct connect (DX)

→ on prem data center to vpc connect
→ connection automatically encrypted
→ goes over public internet

DX → establish physical connect b/w
→ takes a month at min
→ pvt, secure & fast
→ goes over pvt n/w

Ques
can ask dif b/w

site to site vpn requirements

on prem must use customer gateway (cgw)
aws vpc must use virtual pvt. gateway (vgw)

AWS client vpn

idea : connect computer using openVPN to pvt n/w in AWS & onprem
→ allows you to connect ec2 instances over a pvt IP

→ goes over public internet

Transit gateway → for having transitive peering b/w thousands of vpc of on prem, hub of spoke (star) connection

→ one single gateway to provide this

→ works with DX, vpn connections

VPC Closing Comments

- VPC: Virtual Private Cloud
- Subnets: Tied to an AZ, network partition of the VPC
- Internet Gateway: at the VPC level, provide Internet Access
- NAT Gateway / Instances: give internet access to private subnets
- NACL: Stateless, subnet rules for inbound and outbound
- Security Groups: Stateful, operate at the EC2 instance level or ENI
- VPC Peering: Connect two VPC with non overlapping IP ranges, nontransitive
- Elastic IP –fixed public IPv4, ongoing cost if not in-use

VPC Closing Comments

- **VPC Endpoints:** Provide private access to AWS Services within VPC
- **PrivateLink:** Privately connect to a service in a 3rd party VPC
- **VPC Flow Logs:** network traffic logs
- **Site to Site VPN:** VPN over public internet between on-premises DC and AWS
- **Client VPN:** OpenVPN connection from your computer into your VPC
- **Direct Connect:** direct private connection to AWS
- **Transit Gateway:** Connect thousands of VPC and on-premises networks together

Security & Compliance

AWS Shared responsibility model

AWS Responsibility – security of cloud
- Protecting infra
- Providing services

Customer – security in cloud
- how to use service
- encrypting data
- Management of service

Shared controls – Patch mgmt, config mgmt, awareness & training

DDOS attacks – distributed denial of service

Protection on aws → ① aws shield standard
② aws shield advanced
③ WAF – web application firewall
④ cloudfront & route 53

AWS Shield
- free service
- Protection from attacks basic] standard by default

advanced shield
- optional DDOS migration service
- Protection against more sophisticated attacks

AWS WAF → protect from common web exploits (layer 7)

→ deploy on ALB, API gateway, CF

→ Define web access control list (ACL)

- ~~rules~~ rules
- Protect from common - SQL injection or cross site scripting
- geo match
- rate based rules - ddos protects ex. 8 request ⇒ 5.

AWS N/W firewall - to protect vpc overall
- layer 3 to 7
- any direction protect

AWS firewall manager

- manage all security rules in all accounts of an AWS org
- manage vpc SG across all acc in org
- waf rules
- shield - firewall

- rules are applied to new resources as they are created across all of future acc in org

Penetration testing

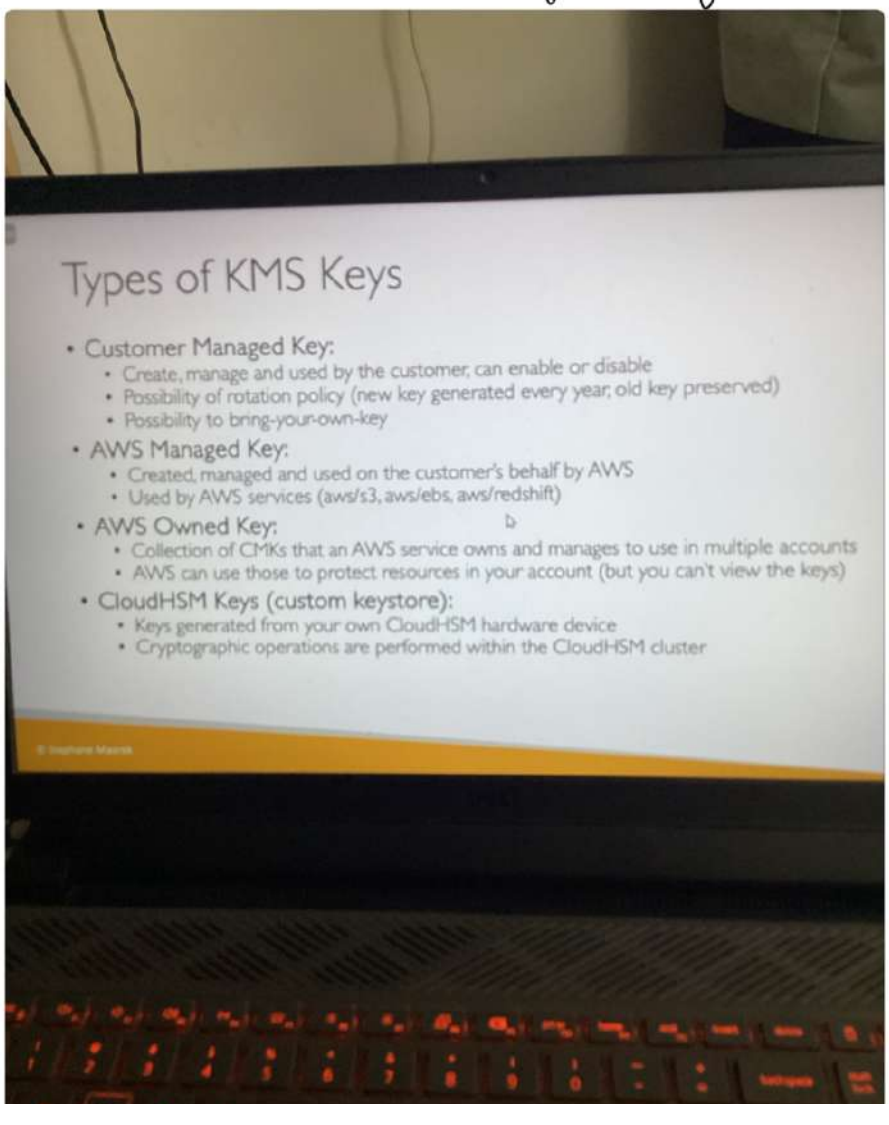
→ Customer can carry out security assessment for 8 services without prior approval

→ Prohibited: → dns zone walking
→ dos / ddos
→ port flood
→ protocol "
→ request "

Encryption → data at rest
→ data in transit

KMS - aws managed keys

Cloud HSM — aws provisions h/w but key managed by customer



AWS certificate manager (acm)

- to easily provision, manage & deploy ssl/tls certificates
- provide in flight encryption for websites (https)
- support both public & pvt tls certs

Secrets manager – force rotation of secrets every X days
- automatic generation of secrets
- encrypted by KMS
- for RDS mainly

AWS artifacts – portal that provides customers with on demand access to AWS compliance documentation & AWS agreements

Artifact reports – allows you to download security & compliance docs from third party auditors

Artifact agreements – allow you to accept & track status of AWS agreements

→ Can be used to setup audit & compliance

AWS Guard Duty — intelligent threat discovery to protect your AWS account

- uses ML for anomaly detection
- 30 days trial

input data includes:

→ cloudtrail logs
→ VPC flow logs
→ dns logs
→ optional: S3, lambda, eks, rds

(A) can protect against cryptocurrency attacks

→ create eventbridge rule to trigger sns/lambda

Amazon Inspector — run automated security assessments

only for these

- for ec2:
  → leveraging aws system manager (ssm)
  → analyze unintended nw accessibility
  → analyze running os against vulnerabilities
- for container images push to amazon ecr:
- for lambda:
  - vulnerability in code/package identified
  - assessment of functions as they're deployed

- reporting & integration with aws security hub
- continuous scanning of ifra when needed

AWS Config — helps with auditing & recording compliance of your AWS resources
- record config & changes over time
- per region service

Amazon Macie — fully managed data security & data privacy service that uses ML & pattern matching to discover & protect your sensitive data in AWS
- alert you to sensitive data such as personally identifiable info (PII)

AWS security hub — central security tool to manage security across several AWS acc & automate security checks
- dashboard to show current security & compliance status to quickly take actions
- aggregate alerts from various service & partner tools
- can cover multiple acc at a time

Amazon detective — uses ML & graphs

- sometimes security findings require deeper analysis to isolate root cause & take action
- detective analyzes & quickly identifies root cause of security issues or suspicious acts
- automatically collect & process events

AWS Abuse — report suspected AWS resources used for abusive or illegal resources
- Spam / DDos / port scanning / malware

Root user priveledges (account owner)
- complete access to all AWS service & resources
- can do all the authoritative tasks

Iam access analyzer
- find out which resources are shared externally
- define zone of trust ⇒ aws acc / org
→ access outside " → findings

Section Summary: Security & Compliance

- Shared Responsibility on AWS
- Shield: Automatic DDoS Protection + 24/7 support for advanced
- WAF: Firewall to filter incoming requests based on rules
- KMS: Encryption keys managed by AWS
- CloudHSM: Hardware encryption, we manage encryption keys
- AWS Certificate Manager: provision, manage, and deploy SSL/TLS Certificates
- Artifact: Get access to compliance reports such as PCI, ISO, etc…
- GuardDuty: Find malicious behavior with VPC, DNS & CloudTrail Logs
- Inspector: find software vulnerabilities in EC2, ECR Images, and Lambda functions
- Network Firewall: Protect VPC against network attacks

Section Summary: Security & Compliance

- **Config:** Track config changes and compliance against rules
- **Macie:** Find sensitive data (ex: PII data) in Amazon S3 buckets
- **CloudTrail:** Track API calls made by users within account
- **AWS Security Hub:** gather security findings from multiple AWS accounts
- **Amazon Detective:** find the root cause of security issues or suspicious activities
- **AWS Abuse:** Report AWS resources used for abusive or illegal purposes
- **Root user privileges:**
  - Change account settings
  - Close your AWS account
  - Change or cancel your AWS Support plan
  - Register as a seller in the Reserved Instance Marketplace
- **IAM Access Analyzer:** identify which resources are shared externally
- **Firewall Manager:** manage security rules across an Organization (WAF, Shield…)

Machine learning

Amazon Rekognition — used to find objects, ppl, text in images & videos using ML

- Facial analysis & facial search to do user verification, ppl counting

Use cases → labelling, content moderation, text detection, face detection & analysis, etc.

Amz transcribe — convert speech to text
— uses deep learning process called automatic speech recognition (ASR)
— Removes PII using Redaction
— multi lingual audio

Amazon Polly — opp of transcribe
— convert text to speech

Amazon translate — language translation
— allow you to localize content & easily translate large volumes of text

Amazon (Lex) + connect

→ powers Alexa
→ automatic speech recog
→ natural lang understanding

→ virtual contact center
→ integrate with CRM

Amazon Comprehend — for NLP
— fully managed & serverless service
— uses ML to find insight & relation in text
— structured & unstructured data

Amazon SageMaker — fully managed service to build ML models

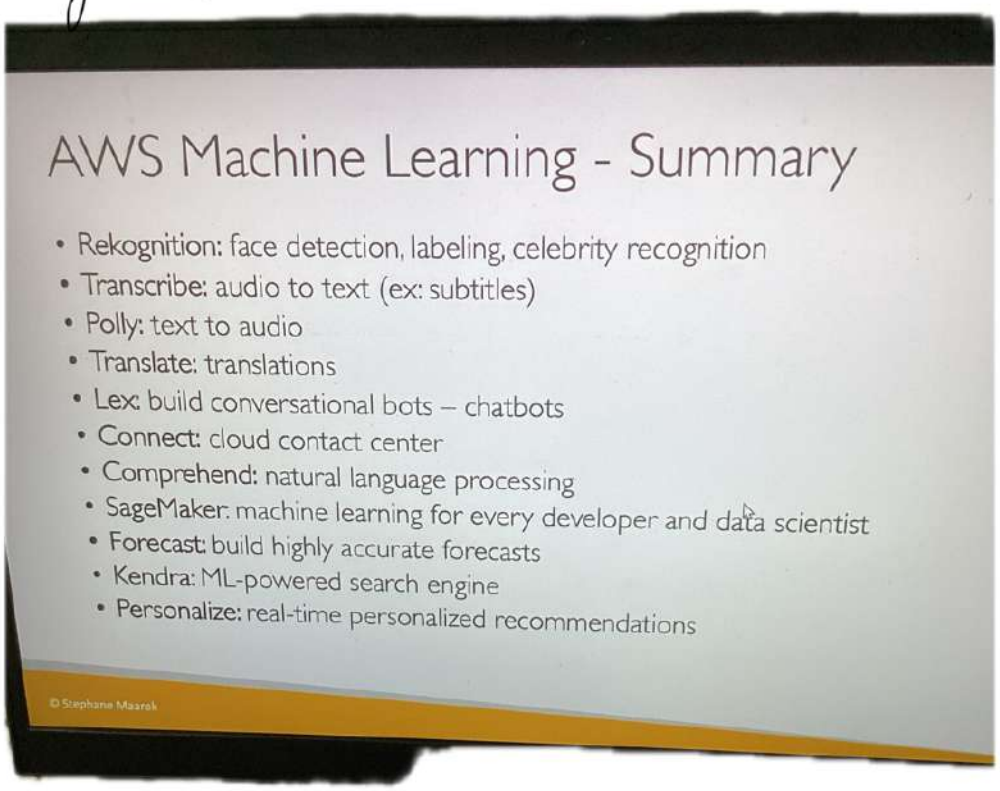
Amazon Forecast — fully managed service that uses ML to deliver forecasts

Amazon Kendra — fully managed document search service powered by ML
— extract ans from docs
— natural lang search capabilities
— incremental learning
— can fine tune results

Amazon Personalize — fully managed ML service to build apps with real time personalized recommendation

Amazon textract — extract text from handwriting or data from

any learned doc



Account Management, Billing & Support

AWS Organizations — manage multiple accounts
- consolidated Billing
  - pricing benefits from aggregated usage
- Pooling of reserved ec2 instances
- api to automate aws acc creation
- restrict acc priveledges using service control policies (scp)

Multi acc strategy — create acc per dept/per cost center, etc.
- use tagging for billing purposes
- enable cloudtrail on all accc to send logs to central S3
- send cw logs to central logging acc

Service control policies (scp)

- whitelist / blacklist IAM actions
- applied at OU or acc level
- doesn't apply to master acc
- SCP is applied to all users & roles of acc including root
- doesn't affect service linked roles
- by default - all "denied"

Use cases: - restrict access to certain services
- enforce PCI compliance by disabling service

AWS Consolidated Billing - AWS Org

Provides you:

1) combined usage: across all AWS acc in that org to share volume pricing, reserved instances & savings plan

2) one bill for all acc

AWS control tower - easy way to set up & Govern secure & compliant multi account AWS environment

benefits: - automate env set up in few clicks
- automate ongoing policy mgmt using guardrails
- detect policy violations

- monitor compliance

AWS Resource access manager (aws ram)

- share resources with other acc
- avoid resource duplication

AWS service catalog

- self service portal to launch a set of authorized products pre-defined by admins
- admin can select products from cf template, select their config
  cf Give iam permissions to access portfolio
- users can see products accessible to them & launch with configured resources to prevent wrongful resources creation

Pricing Models in AWS (4)

benefits
→ Pay as you go
→ save when you reserve
→ pay less use more

ways → free tier
→ compare pricing — ec2, lambda, ecs

→ Storage pricing – S3, ebs
→ Database pricing – RDS
→ Content delivery – cloud front
→ n/w Ring costs
↳ use pvt IP for good savings + n/w performance
→ same AZ for max savings

Savings plan – instead of reserving instances
↳ commit to $x / hour for 1 or 3 yrs
– long term commitments

Types

→ ec2 savings plan

– upto 72% discount as compared to on demand
– all upfront, partial, no
– commit to use individual instance families in a region

→ compute savings plan

– upto 66% discount compared to on demand
– regardless of family / region / size / OS ...
– compute options – ec2, fargate, lambda

→ ML savings plan : sagemaker, etc.

AWS compute optimizer

– Reduce cost & improve performance by

recommending optimal AWS resources for your workloads

- uses ML to analyze resource config & utilization
- available for : ec2, instances
  ec2 auto scaling groups
  ebs volumes
  lambda functions
- lower cost by upto 25%

Billing & Costing tools

1) Estimate cost in cloud - Pricing calculator
2) Tracking cost in cloud - Billing dashboard
   - cost allocation tag
   - cost & usage reports
   - cost explorer
3) Monitoring against cost plan
   - billing alarms
   - budgets

Pricing Calculator - estimate cost for your solution architecture

Tracking cost in cloud

1) AWS billing dashboard
2) AWS free tier dashboard

3) Cost allocation tags — used to track your aws costs on a detailed level

AWS generated tags:
- automatically applied to resources
- prefix = aws:

User defined tags
- prefix = user:

Tagging & resource groups

→ Tags are used for organizing resources
→ free naming
→ tags can be used to create/maintain/view resource groups

Cost & usage Reports

- dive deeper into your aws costs & usage
- most comprehensive set of aws cost & usage data available

Cost explorer

- visualize, understand & manage, AWS costs &

usage over time
- create & custom reports that analyze cost & usage data
- analyze data at high level
- forecast usage up to 12 months based on previous usage

Billing alarms in cloudwatch

- Billing data metric is stored in CW us-east-1
- billing data are for overall worldwide AWS costs
- it's for actual costs | not projected

AWS Budgets - send alarms when costs exceeds the budget
- 4 types: usage, cost, reservation, savings plan
- upto 5 sns notif per budget

AWS cost anomaly detection

- continuously monitors cost & usage data & uses ML to detect unusual spends
- send reports with root cause analysis

AWS service Quotas - notify you when you're close to service quota value threshold
- create CW alarms on service Quota console

AWS Trusted Advisor — gives high level acc assessment

- analyze AWS acc & provides recommendations on 6 categories :
  - cost optimization
  - performance
  - security
  - fault tolerance
  - service limit
  - operational excellence

Support plans for AWS

developer	business	enterprise on ramp	enterprise

1. **basic**
   - customer service & communities
   - 7 core checks of aws trusted advisor
   - aws personal health dashboard
2. **developer plan**
   - basic +
   - business hour email access to cloud support associate
   - unlimited case / contacts
   - same
     - general response time < 24 hrs
     - system impaired < 12 hrs

3 Business
- for prod workloads
- full trusted advisor
- 24×7 support access
- access to infra mgmt for additional fees

same [same + prod system impaired < 4 hrs
prod system down < 1 hr

4 enterprise on ramp - business + access to TAM
| - support team
| - Operation review team
same + business critical system down < 30 min

5 enterprise - all before + designated TAM
- used if mission critical workload
|
|
[same + business critical sys down < 15 min

useful CP cheatsheets | Cour | build your own X | terraform exam blog | How | how | how | how | 10 S | Log

nc.udemy.com/course/aws-certified-cloud-practitioner-new/learn/lecture/20056442#notes

course selection | scholarships | tickets | housing | terraform | visa | Jahnvi | OPPPORTUNITIES | Placement | mdi intranet exchng | MDI external mail

] Ultimate AWS Certified Cloud Practitioner CLF-C02

Account Best Practices – Summary

- Operate multiple accounts using Organizations
- Use SCP (service control policies) to restrict account power
- Easily setup multiple accounts with best-practices with AWS Control Tower
- Use Tags & Cost Allocation Tags for easy management & billing
- IAM guidelines: MFA, least-privilege, password policy, password rotation
- Config to record all resources configurations & compliance over time
- CloudFormation to deploy stacks across accounts and regions
- Trusted Advisor to get insights, Support Plan adapted to your needs
- Send Service Logs and Access Logs to S3 or CloudWatch Logs
- CloudTrail to record API calls made within your account
- If your Account is compromised: change the root password, delete and rotate all passwords / keys, contact the AWS support
- Allow users to create pre-defined stacks defined by admins using AWS Service Catalog

So that's it for this lecture.

rview | Q&A | Notes | Announcements | Workspaces ★ Pro | Reviews | Learning tools

Create a new note at 1:33

All lectures | Sort by most recent

Search

DELL

Billing and Costing Tools – Summary

- Compute Optimizer: recommends resources' configurations to reduce cost
- Pricing Calculator: cost of services on AWS
- Billing Dashboard: high level overview + free tier dashboard
- Cost Allocation Tags: tag resources to create detailed reports
- Cost and Usage Reports: most comprehensive billing dataset
- Cost Explorer: View current usage (detailed) and forecast usage
- Billing Alarms: in us-east-1 – track overall and per-service billing
- Budgets: more advanced – track usage, costs, RI, and get alerts
- Savings Plans: easy way to save based on long-term usage of AWS
- Cost Anomaly Detection: detect unusual spends using Machine Learning
- Service Quotas: notify you when you're close to service quota threshold

Advanced Identity

AWS STS (Securing token service)

- enables you to create temporary, limited privileges credentials to access your AWS resources
- short term credentials
- configure expiration period
- use cases :
  - identity federation
  - iam roles accesses for ec2 / cc2

AWS Cognito - provide identity to web/mobile app users (potentially million)
- instead of iam use cognito

Microsoft Active Directory (AD)

- database of objects - user/ acc / computers -
- centralized security mgmt, user acc

AWS directory service

- AWS managed microsoft AD
- AD connector
- Simple AD

AWS Iam identity center

- One login → for all AWS org acc
  - → business cloud apps
  - → saml 2.0 enabled apps
  - → ec2 windows instances

- Identity providers
  - built in identity store in iam identity center
  - 3rd party & AD, onelogin

Other services (rarely)

Amazon workspaces — managed desktop as service to easily provision windows/linux desktop

Appstream 2.0 — desktop app streaming service
- app delivered from within web

IoT core — easily connect IoT device to AWS cloud

Elastic Transcoder — convert media files stored

in S3 into media files in formats
req by consumer playback devices

AppSync — store & sync data across
mobile & web app
— graphQL used

Amplify — set of tools & services that helps
you develop & deploy scalable
full stack web & mobile apps

App composer — build serverless apps quickly
on AWS
- generates IaC using CloudFormation

Device farm — fully managed service that
test web & mobile app
against browser

AWS Backup — fully managed service to
centrally manage & automate
backup across AWS service
- on demand - scheduled - PITR
- CRR

DR strategies → Backup & restore (cheap)
→ Pilot light (just run core
(DB) & sync on cloud)

min critical func are available

→ warm standup — full version of app in cloud but at min size

→ multi site / hot site — full version, full size

→ failover to dfy AZ

AWS elastic DR — cloudendure DR
— quickly & easily recover your physical, virtual & cloud based server in AWS
— continuous block level replication for servers

Datasync — move large data from onprem to AWS
— replication tasks are incremental after first full load

Cloud Migration strategies — 7 Rs

Retire
Retain
Relocate
Rehost "lift & shift"
Replatform "lift & shift"
Repurchase "drop & shop"
Refactor / Re-architect

Application Discovery Service — plan migration projects by gathering info about onprem data centers

→ Agentless Discovery
→ Agent based

Application Migration Service

— Rehost soln which simplify migrating applications to AWS

AWS migration evaluator

AWS Migration Hub — central location to collect servers & apps inventory data for assessment, planning & tracking of migrations to AWS
— automate lift & shift

AWS fault injection simulator — fully managed service for running fault injection experiments on AWS workloads
— Based on chaos engineering
— helps to uncover hidden bugs

AWS Step Functions — build serverless visual

workflow to orchestrate your lambda func

Ground station — fully managed service that lets you control satellite comm^n, process data & scale your satellite ops

Pinpoint — scalable 2way marketing comms service
— ability to segment & personalize msg with right content to customer

↓ evolution of sns for marketing comp

AWS architecting & ecosystem

Well architected framework, general guiding principle

→ stop guessing your needs
→ test sys at prod scale
→ automate to make architectural exp easier
→ allow for evolutionary arc
→ drive your arch using data
→ improve through game days

Design principles → scalability
→ disposable resources

→ automation
→ loose coupling
→ services, not servers

6 pillars

operational excellence
security
reliability
performance efficiency
cost optimization
sustainability

} they are a synergy