

AWS Basics

IAM 101 → best place to start

What's IAM? [Identity Access Management]

↳ allows for management of users / privileges for aws console

• What's IAM give you?

- ① centralized control to AWS account
- ② shared access to AWS account
- ③ granular permissions
- ④ Identity Federation
- ⑤ MFA
- ⑥ temp access/control for users / devices and services as needed
- ⑦ integration w/ many diff aws services
- ⑧ PCI dss compliance [something abt payments]

• important terms?

- users = end users [ppl]
- Roles = you create roles and assign them to AWS resources
- policies = defines one or more permissions

EC2 101 → resizable = capacity you want when needed

What's EC2? [Elastic Compute Cloud]

- ↳ secure, resizable compute capacity in the cloud
- basically a VM only hosted in AWS instead of ur own data center

①

Why's EC2 so cool?

- first public cloud provider, only pay for what you use
no wasted capacity
 - before EC2 capacity had to be estimated which costed a lot more money upfront
 - takes minutes to get up and running

EC2 summary

What are 3 types of load balancers?

- ① App load balancers
- ② Network load balancers
- ③ classic load balancers

What's a 504 error mean?

↳ gateway timeout = app not responding w/in idle timeout period

- Troubleshooting: its either the web or DB server

What's Route 53? → Amazon DNS service

↳ allows for mapping of domain names to ...

- ① EC2 instances
- ② Load balancers
- ③ S3 buckets

CLI tips? → standard least privilage stuffs

OS level encryption possible

What are the DB types in AWS?

- ① RDS - OLTP → what EC2 can do? [not sure abt this]

- Ⓐ SQL
- Ⓑ MySQL
- Ⓒ PostgreSQL
- Ⓓ Oracle
- Ⓔ Aurora
- Ⓕ MariaDB

- ② DynamoDB → no SQL

- ③ Redshift - OLAP → data warehousing service

- ④ ElastiCache → in memory caching
consists of ...

- Ⓐ memcache - Ⓓ
- Ⓑ redis

prob don't
need to
know use
case for this
app

②

S3 101

What's S3? [simple storage service] → obj based storage

- official amazon definition

- ↳ provides devs w/ secure, durable, highly scalable obj storage

- simple def?

- ↳ place to put your files/pictures/videos

Basics of S3? → is obj-based [allow you to upload files]

- file size 0 Bytes to 5TB • files are stored in buckets

- unlimited storage • S3 is a universal namespace

- uploading S3 file results in HTTP 200 code

What does being obj based mean?

- Objs = files consisting of ...

- ① Key: name of the obj

- ② Value: the data

How does data consistency work for S3?

- ① if you're writing a new file and reading to it immediately, you'll be able to view that data

- ② if you update an existing file / delete a file and read it immediately, you may get the older version, or you may not

- ↳ Δ's to obj take time to propagate

- ④ read after write consistency for puts of new obj

- ② Eventual consistency for overwrite PUTS and DELETE can take time to propagate

- ↳ what was initially said abt the data consistency

③

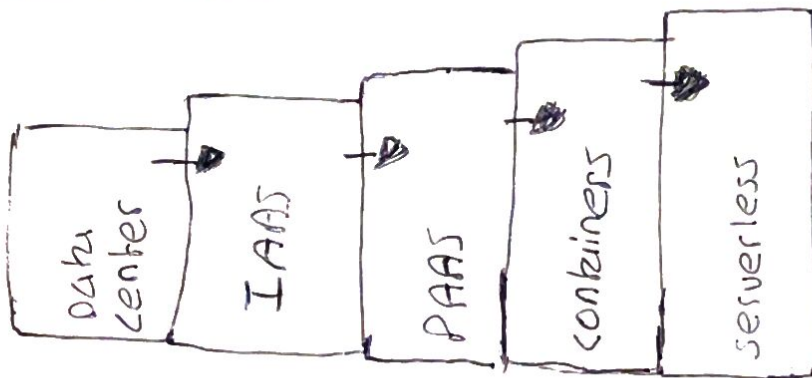
Features of S3?

- ① Tiered storage
- ② Life cycle management
- ③ version control
- ④ encryption
- ⑤ secure your data through Access control lists and Bucket policies

What are the S3 storage classes?

- ① S3 standard
- ② S3 - IA - quick-access but only needed every once in a while
- ③ S3 one zone - IA - 2 but stored only in one place
- ④ S3 - intelligent tiering - ML based
- ⑤ S3 Glacier
- ⑥ S3 Glacier Deep Archive

serverless 101



serverless = code being run on the cloud

Lambda

What's Lambda?

- compute service where you can upload your code and create a Lambda Fn. Lambda takes care of provisioning/managing the servers that you use to run the code

How can Lambda be used?

- ① event driven compute service
 - ↳ AWS runs your code in response to events
- ② compute service
 - ↳ codes run in response to HTTP requests

④

What langs can be used for lambda?

• Node.js, Java, Python, C#, Go

API Gateway] → think waiter in restaurant

What's an API? [Application programming interface]

↳ middleman that takes requests for data from the user, communicates w/ another service and then returns the result of the communication to the user

Types of API?

- ① REST API → uses JSON] → what API gateway uses
- ② SOAP API → uses XML

What can API gateway do?

- ① expose HTTPS endpoints to def a REST API
- ② serverlessly connect to services like lambda and dynamo db
- ③ send each API endpoint to diff target
- ④ easy tracking / scaling

How do I config API gateway?

- ① define API
- ② define resources and nested resources (URL Paths)
- ③ for each resource you need to ...
 - ① select supported HTTP methods (verbs)
 - ② set security
 - ③ choose target → EC2 / Dynamo, etc.
- ④ deploy API to stage
 - ① uses API gateway domain by default

API catching isn't important for your apps imo

same origin policy prevents XSS

DynamoDB

→ fully managed DB that supports both key-value and document data models

What's DynamoDB?

• A fast and flexible noSQL DB for all apps that need consistent single digit millisecond latency @ any scale

• Characteristics of DynamoDB?

① stored on SSD storage

② spread across 3 geographically distinct data centers

③ 2 consistency models

④ eventual consistent reads [default]

↳ consistency of the data is usually reached w/in a second. Repeating a read after a short time should return updated data [best read performance]

⑤ strong Consistent Reads

↳ returns a result that reflects all writes that recieved a successful response prior to the read

What's DynamoDB made out of?

• tables

• items [think row of data in table]

• Attributes [columns of data in a table]

• supports K-V and document data structures

- key = name of data, - value = data itself

can be written in JSON, HTML or XML

How does DynamoDB store and retrieve data? → based off the primary key

• What are the 2 types of primary keys?

① Partition key: unique attribute (eg. userID)

② value is input to internal hash fn for determining location for the data being stored

③ no 2 items can have the same partition keys

20/03/21

timestamp of post

② composite key = Partition Key + Sort Key

- Ⓐ 2 items may have the same partition key but they must have a diff sort key
- Ⓑ all items w/ the same partition key are sorted together, then sorted according to the sort key value
- Ⓒ multiple items can be stored w/ the same partition key

DynamoDB example

{

"UniqueID": 1475 → Partition Key
"FirstName": "Allan"
"Surname": "Brown"
"Phone": 555-2323

}

{

"uniqueID": 1476 → Partition Key
"FirstName": "Riad"
"Surname": "Ramanou"
"Course Name": "AWS Developer Associate" → sort key
"Addy": {
 "Num": 123
 "Street": "River Rd"
}

}

How's Access controlled? → through AWS IAM