AWS - IAM - Identity Access Management

AWS - IAM - Identity Access Management

Notes:

- permissions are stored with JSON language
- this is globally, so everything you define is for all regions.

Key Terminology:

- Users
  - end users, such as employees, contractors etc
- Groups
  - collection of users. each user will inherit the group permissions
- Roles
  - you attach permissions to roles and then you can assign roles to users or groups
- Policies
  - called policy document, its simply json files that define the permissions for user/group/role

Power User - has access to all aws services except management of groups and users in IAM

AWS - Cognito & Web Identify Federation

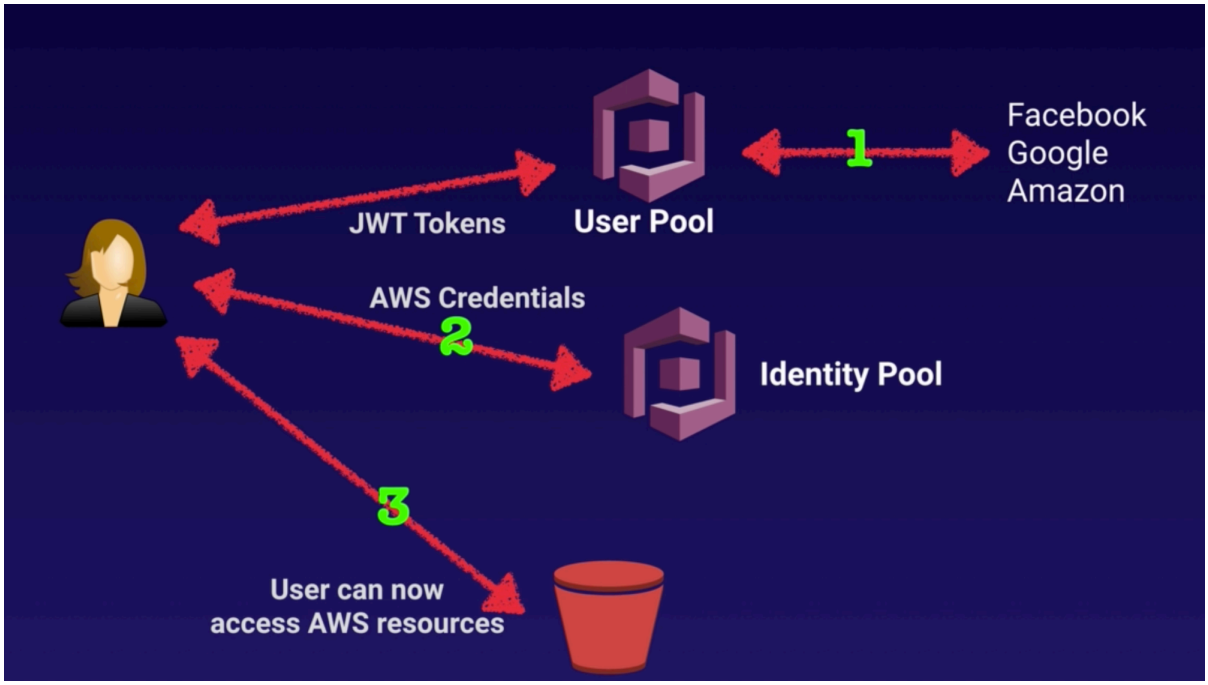
Web identity federation is a way to use login with other service providers like Facebook login, Gmail login.

Cognito will do the heavy lifting for you. it will reach to the providers and handle the process.

- login with Facebook
- you get the auth id from them
- you give the id to cognito
- cognito respond with a token that grants access to whatever aws resources you allowed

User Pools vs Identity Pools

- User Pools
  - here we store all the user data, it could be email, name and others
- Identity Pool
  - where you grant the permissions to users



Exam Tips:

- Federation allows users to login with web identity provider (Google, Facebook, Amazon)
- The user first authenticates with the web identity provider and gets an auth token which then exchange with cognito to temp aws credentials
- Cognito is identity broker which handles the interaction with the identity providers so need to write your own code
- User pool is user based. It handles things like user registration, authentication and account recovery.
- Identity pool authorizes access to your aws resources

AWS - Kinesis

Exam Tips:

- the difference between stream and firehose
    - shards or data persistence is streams
    - if you want to analyze on the fly, use firehose
- if you want to analyze your data inside kinesis use analytics

What is Streaming Data?

- data generated continuously by thousand of sources in the same time with small size
- examples for sources:
    - purchases from online store (ie amazon)
    - stock prices change
    - game data when player plays
    - social network actions, views etc
    - geo data, like uber
    - iot data

Kinesis is a place to send your streaming data to.

Types of Kinesis:

- Streams
    - stores data for 24H by default, but can get extended to 7 days
    - data is stored in shards and consumers can get the shard data
    - each shard
        - 5 transaction per sec for reads and 2mb data
        - 1000 writes per sec and 1mb data
- Firehose
    - no persistence for for the incoming data
    - as soon as you get new data you need to handle it
    - it can be start a lambda function, forwarded to s3, redshift etc
- Analytics

AWS - API Gateway

Service to publish, maintain and manage your APIs in any scale.
Think of it as a doorway to your aws services.

What it can do?

- expose HTTPS endpoints to define RESTful API
- connect to serverless service lambda/dynamo
- send each endpoint to different targers
- Runs efficiently with low cost
- scale effortlessly
- track and control usage by api keys
- throttle requests to prevent attacks
- connect to cloudwatch to monitor requests

Caching:

- you can tell AG to use cache instead of doing another call to your targets
- the cache used will be served depends on the TTL your define

AWS - Test Items

Exam Tips:
AWS practice test (84%)
Topic Level Scoring:
1.0 Design Resilient Architectures: 100%
2.0 Define Performant Architectures: 71%
3.0 Specify Secure Applications and Architectures: 83%
4.0 Design Cost-Optimized Architectures: 50%
5.0 Define Operationally-Excellent Architectures: 100%

AWS - Elastic Transcoder

media transcoder in the cloud.

Exam Tips:

- takes media files
- and converts them in to different formats for different devices: iPhone, tablets, PCs etc.

AWS - SNS

Simple Notification Service.

A way to send notification based on events and criteria.
You can send emails, smses, pushes, hooks etc.

Exam Tips:

- push based, not pull base (like SQS)
- you create topics
    - like billing over 10$ a month
    - instance cpu over 90%
- then you create subscribers that listen to the topics

AWS - SWF

Simple Workflow Service

Exam Tips:

- a way to coordinate tasks between humans, computers etc
- (example) Amazon uses that in their order process
    - order created
    - data sent to the worker in the warehouse
    - the worker packages and moves the package
    - next script runs to update status

AWS - SQS

Queue service.

Exam Tips:

- pull based, not push base (like SNS)
- data retention in queues. Default is 4 days. But you can select from 1min to 14 days
- msg size between 1-256 kb. For larger msgs, better using S3 as storing the msg.
- visibility timeout

        - this is the time where the queue waits for a msg deletion after its processed by a worker
        - timeout maximum is 12 hours
- polling
    - short, means you ask for a msg and get a reponse
    - long, you open a “connection” and wait until a new msg comes or the connection times out
- msg limitation in queue
    - unlimited
    - but if msg was sent to the consumer and waits for deletion (in flight)
        - standard queue is 120K
        - FIFO queue is 20K
- Queue types:
    - standard
        - has very high throughput
        - will “generally” keep messages order but not promised
        - msgs can be delivered more then once
    - FIFO
        - limited to 300 transactions per second
        - msg order is uphold
        - msg will be sent once

AWS - VPC

virtual private network

Exam Tips:

- a subnet can be only in one AZ. But We can have several subnets in one AZ.
- one internet gateway can be attached to one vpc
- default vpc creation comes with: route table, network ACL & a security group
    - it won’t create any subnets or internet gateway
- AZ on different AWS accounts are randomized. AZ-1 can be different from another account AZ-1
- AWS reserves 5 ip addresses within your subnets
- security groups are per vpc
- route tables can have several subnets associated

vpc - virtual private cloud with cidr ip range that contains subnets
subnet - an “area” in a vpc with specific cidr that can contain instances
network ACL - another network layer above your subnet that can help with security
route table - points ips to destinations
security group - a group of rules for inbound traffic attached to instances

NAT instances & NAT gateways:

- network address translation
- instance is a single ec2 instance
    - create ec2 instance from community of “nat"
    - must be in the “public” subnet
    - disable Source/Dest. Check on the instance
    - go to vpc route tables and select the main route table for the vpc
    - add 0.0.0.0/0 and select target as the nat ec2 instance created above
    - be sure to have a security group that is open to the
    - internet bandwidth depends on your instance size
- gateways is a service spread on multi AZ. allows access to the internet
    - in vpc admin, go to “NAT Gateways” and create one
    - select the “public” subnet
    - and select or create the elastic ip
    - internet bandwidth scales automatically
    - no need to take care of the gateway
    - no need to associate to SG
    - automatically gets an ip address

ACL - Access Control List:

- unrelated new network ACL by default denies everything
- when a new vpc gets created you get a new acl
    - this ACL by default allows all inbound and outbound traffic
    - all subnets in vpc get defaulted to that ACL
    - a subnet can be attached to a single ACL only
    - ACL rules are evaluated from bottom up by rules numbers. So first rule will get precedence on next rules
    - ACL’s act before you even reach security groups
- if you want to block specific ip addresses use ACL
- ACL can have several subnets associated
- so the ports you need to consider:
    - 80/443 inbound and outbound to have web network
    - 22 inbound so devs could ssh to the machine
    - 1024-65535 outbound (Ephemeral ports) those are ports the server use while there is a connection to a client
- https://medium.com/awesome-cloud/aws-difference-between-security-groups-and-network-acls-adc632ea29ae

VPC Flow Logs:

- feature that enables capturing the incoming and outgoing traffic to cloudwatch
- flow logs can be created on 3 levels: vpc, subnet, network interface level
- create a flow log:
    - you will first need to create a role with the proper permissions
    - you will need to create a destination, either:
        - cloud watch destination
        - s3 bucket
    - you will go to the level you want flow logs on, and add it on actions
- notes:
    - you can't enable flow log with a peer vpc, unless its within your aws account
    - you can't tag a flow log
    - after you created a flow log you can't edit its configuration
    - not all ip traffic is monitored:
        - traffic that contact AWS dns service, unless you use your own dns server
        - traffic generated by windows instance for aws windows license activation
        - traffic to and from 169.254.169.254 (instance data ip)
        - DHCP traffic (Dynamic Host Configuration Protocol)
        - traffic to the default vpc router

Bastion Hosts:

- NAT gateway on instance provides internet traffic to private subnets
- Bastion securely administers a way to ssh to your private instances
- you can't use NAT gateway as Bastian host

Direct Connect:

- directly connects your data centers to AWS
- useful for high throughput workloads (ie lots of network traffic)
- or if you need a stable and reliable connection

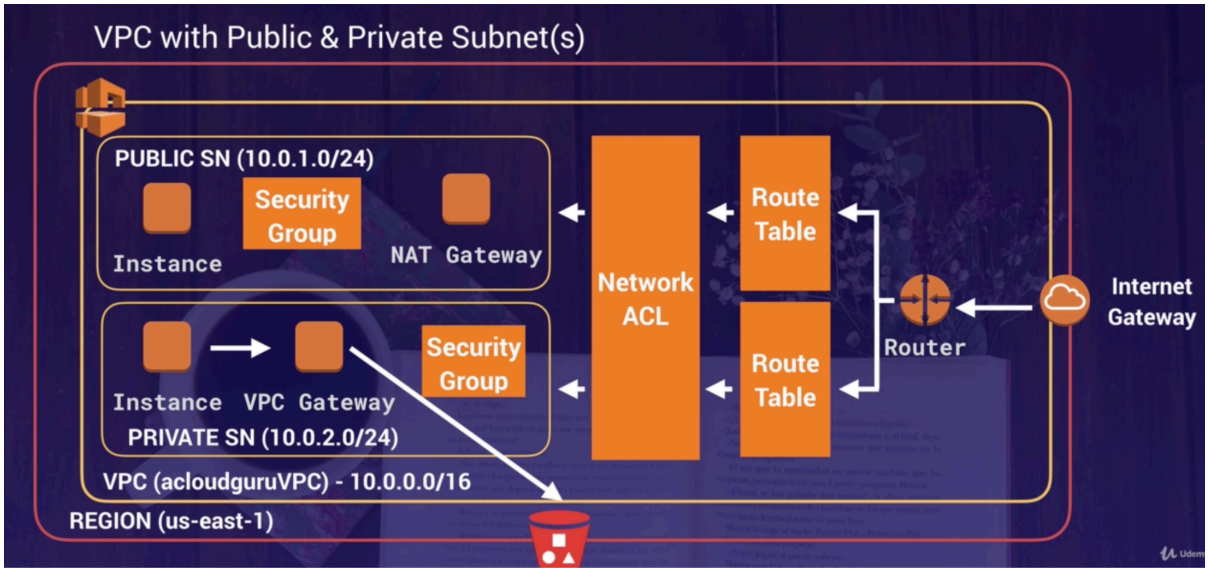
VPC Endpoints:

- allows you to privately connect your VPC to supported AWS services without requiring internet gateway.
- you don't go through the internet by go through inside aws network

- two types:
    - interface endpoint
    - gateway endpoint
        - supports S3 and Dynamo

Create a new VPC:

- create new with desired IPv4 CIDR
- create your subnets with desired IPv4 CIDR
    - if any public subnets, enable auto assign for public ips
- if you need internet access.
    - create internet gateway and attach it to the vpc (IG)
    - create route table with routes destinations => 0.0.0.0/0 and ::/0 that attach to the above IG
    - associate the subnet that needs web access to the route table
- then you need to create EC2 instances
    - make sure to assign them to the vpc and the right subnet
    - if the subnet is connected to IG and you want to have access from the internet
    - you will need to create a new security group that enables that
    - if you want to connect instances from subnet to another, you need to define it on security group



Step By Step Guide with sample values:

- create vpc

        - cidr block: 192.168.0.0/16
- create two subnets
    - select our above vpc
    - select AZ for each
    - first one:
        - cidr block: 192.168.1.0/24
        - this one will be our “public” subnet so be sure to enable auto assignment for IPs
    - sec one:
        - cidr block: 192.168.1.0/24
- create internet gateway if needed to access to the web
    - attache new IG to the vpc
- create route table
    - this will connect a subnet to the internet
    - select the vpc
    - associate the subnet that will have access to the internet
    - add route with 0.0.0.0/0 and select the target as the IG we created above
- create security groups
    - open relevant ports, ie: 22, 80, 443
- create your first “public” instance => EC1
    - select the vpc
    - select the “public” subnet
    - attach proper SG
- create your first “private” instance => EC2
    - select the vpc
    - select the “private” subnet
    - attach proper SG
- now EC1 has access to the internet but EC2 doesnt
- to let EC2 access the internet, we will create NAT gateway
    - select the “public” subnet that has access to the internet
    - attach an elastic ip existing or create a new one
- now go to the route table for our “private” subnet and add a route for 0.0.0.0/0 with the above NAT as the target
- to secure your vpc more you can use ACL
    - for your “public” ACL
        - inbound rules:
            - 80, 443, 22, 1024 - 65535, ALL ICMP
        - outbound rules:
            - 80, 443, 1024 - 65535, ALL ICMP

AWS - EC2 - Elastic Compute Cloud

Exam Tips:

- you can now encrypt the root device volume and additional volumes
- termination protection is turned off by default
- on ebs backed instance the default will delete root volume on termination. Any additional volumes won't be deleted by default
- get data about your instance
    - curl http://169.254.169.254/latest/meta-data
        - get public ip, local ip etc
    - curl http://169.254.169.254/latest/user-data
        - the user bootstrap script

Pricing Models:

- On demand
    - pay by the hour (or by the second)
- Reserved
    - reserve capacity with contracts for 1-3 years.
    - Types:
        - standard
            - up to 75% off just pay up front
            - locked to the reserved instance
        - convertible
            - up to 54% off
            - you are able to change instance attributes as long its the same or larger
        - Scheduled reserved
            - set specific time like Sunday 2pm for 4 hours
- Spot
    - option to bid on capacity. As long as your apps has flexible start, end times
    - if your instance gets stopped in the middle of the hour by aws, you won't be charged for that partial hour. But if you stopped, you will.
- Dedicated hosts
    - Physical EC2 instances dedicated for your use
    - good for regulations where you can't share your instance
    - licensing, when you are not allowed to share your instance
    - can be on demand or reserved

Security Groups:

- changes take affect immediately
- all inbound traffic is blocked by default
- all outbound traffic is allowed
- you can allow rules but can't deny rules
- you can't block specific ips. But you can do that with network access control lists

EBS - Elastic Block Store

- can't be shared between EC2 instances
- each volume is replicated in its availability zone
- 5 different types:
    - general purpose (ssd)
    - provisioned IO{S (ssd)
    - throughput optimized harddisk drive
    - cold hard disk drive
    - magnetic
- volumes will always be in the same availability zone of the EC2 instance
- to move a volume between availability zones. You should take a snapshot, create an AMI and create the volume on the AZ.
- to move a volume between regions. You should take a snapshot, create an AMI, copy AMI to the other region and create the volume on that region.

EFS - Elastic File System

- can be shared between EC2 instances
- supports the Network File System Version 4 (NFSv4) protocol
- you only pay for the storage you use
- can scale up to petabytes
- can support thousands of concurrent NFS connections
- data is store across multiple AZ's within a region
- read after write consistency

Placement Groups

- 3 different types:
- clustered placement group
    - grouping instances in the same AZ
    - low network latency, high network throughput or both
- spread placement group
    - protects from hardware failure
    - each instance is on different hardware
    - either on the same AZ or different AZ depends how you configure it
- partitioned
    - similar to spread placement group
    - but here you can partitioned group of instances that will be together on the same hardware

Instance Store

- instance store volumes are sometimes called “Ephemeral storage"
- They can be stopped. If the underlying host fails, the data is lost.
- You can’t leave the volume in case you want to on host deletion

Getting Instance Data

- curl to http://169.254.169.254/latest/meta-data/ (public ip, private ip, etc)
- curl to http://169.254.169.254/latest/user-data/ (bootstrap script if you added it)

EBS Types:

	Solid-State Drives (SSD)		Hard disk Drives (HDD)		
Volume Type	**General Purpose SSD**	**Provisioned IOPS SSD**	**Throughput Optimized HDD**	**Cold HDD**	**EBS Magnetic**
Description	General purpose SSD volume that balances price and performance for a wide variety of transactional workloads	Highest-performance SSD volume designed for mission-critical applications	Low cost HDD volume designed for frequently accessed, throughput-intensive workloads	Lowest cost HDD volume designed for less frequently accessed workloads	Previous generation HDD
Use Cases	**Most Work Loads**	**Databases**	**Big Data & Data Warehouses**	**File Servers**	**Workloads where data is infrequently accessed**
API Name	gp2	io1	st1	sc1	Standard
Volume Size	1 GiB - 16 TiB	4 GiB - 16 TiB	500 GiB - 16 TiB	500 GiB - 16 TiB	1 GiB-1 TiB
Max. IOPS**/ Volume	16,000	64,000	500	250	40-200

EC2 Instance Types:

Family	Speciality	Use case
F1	Field Programmable Gate Array	Genomics research, financial analytics, real-time video processing, big data etc
I3	High Speed Storage	NoSQL DBs, Data Warehousing etc
G3	Graphics Intensive	Video Encoding/ 3D Application Streaming
H1	High Disk Throughput	MapReduce-based workloads, distributed file systems such as HDFS and MapR-FS
T3	Lowest Cost, General Purpose	Web Servers/Small DBs
D2	Dense Storage	Fileservers/Data Warehousing/Hadoop
R5	Memory Optimized	Memory Intensive Apps/DBs
M5	General Purpose	Application Servers
C5	Compute Optimized	CPU Intensive Apps/DBs
P3	Graphics/General Purpose GPU	Machine Learning, Bit Coin Mining etc
X1	Memory Optimized	SAP HANA/Apache Spark etc
Z1D	High compute capacity and a high memory footprint.	Ideal for electronic design automation (EDA) and certain relational database workloads with high per-core licensing costs.
A1	Arm-based workloads	Scale-out workloads such as web servers
U-6tb1	Bare Metal	Bare metal capabilities that eliminate virtualization overhead

AWS - Databases Aurora

Mysql compatible engine that provides the price of opensource db with the cost commercial databases.
Up to 5 time better performance then mysql DB and 1/10 of a commercial DB.

Exam Tips:

- 2 copies of your data is contained in each availability zone, with a minimum of 3 availability zone. 6 copies of your data.
- you can share aurora snapshots with other AWS accounts
- 2 types of replicas available. Aurora replicas and mysql replicas. Automated failover is only available with aurora.
- aurora automated backup us turned on by default

Things to know:

- start with 10gb and has increments of 10gb (up to 64tb) with auto scaling
- compute resource can go tup to 32vcpu and 244gb of memory

Scaling Aurora:

- transparently handle:
    - loss of up to two copies with no effect to write availability
    - loss of up to three copies with no effect to read availability
- disks are self healing, so they are repeatedly scanned and fixed

Replica types:

- aurora replicas (currently 15)
- mysql read replicas (currently 5)

AWS - Route53

AWS dns service.

Exam Tips:

- ELB's do not have predefined IPv4 addresses; you always resolve to them using a dns name
- it can take up to 3 days for domain registration
- diff between cname and alias
- health checks
    - you can add health checks to individual record sets
    - if a record fails a health check, it will get removed from route53 until its healthy again
    - you can set sns notifications to alert you if health checks fail
- you have a limit f managing 50 domains. But you can contact aws for more

DNS converts human friendly names to ips. IPv4 & IPv6
IPv4 - 32 bit field
IPv6 - 128 bit field

- you register domains on domain registrars. Like AWS, godaddy etc.

DNS:

- SOA - start of authority
    - dev in charge
    - ttl (time to live)
    - dns records

NS - Name server

How browser finds the ip?

- it gets the domain name: "tzook.com"
- it goes to the top level domain server, here its ".com"
- and give the requested domain
- then it gets back the name servers for that domain. like "ns.awsdns.com"
- then it goes to query for NS records on "ns.awsdns.com"
- NS records will return the SOA

dns types:

- A
    - points to IP or alias
- CNAME (canonical name)
    - resolve one domain name to another domain name
- Alias
    - similar to CNAME but here you can point to a different target dns like S3, load-balancers, cloudfront etc
    - CNAME difference as it can point to "naked" domain name like "http://tzook.com"
- NS - name server records
- MX for mail
- PTR reverse of A record

Different Routing Policies:

- simple
    - you can have several ips and the user will get one of them each time in random order.
- weighted
    - send traffic by the weight you give each ip. So it its 80 to A, 20 to B. 8 time of 10 A will get the traffic
- latency

    - route53 will check which ip has less latency and return it back
- failover
    - you will have active an passive ips. When active is not working, passive will kick in.
- geolocation
    - redirect users by the location you define to the attached ip
- geoproximity
    - considers the users location and the resources location. Can get very complicated. You must use route53 traffic flow
- Multivalue answer
    - similar to the "simple" policy but you can add health checks that you cant add on "simple policy". It will return all the healthy records and let the requester decide.

Why the name?
First route from coast to on the US was route 66.
DNS is on port 53. So that's why.

AWS - ELB - Elastic Load Balancer

Elastic Load Balancer

3 types:

- Application
    - best suited for http/s traffic
    - operate at layer 7 and are application aware
    - intelligent and you can define rules on how to balance
- Network
    - best suited for TCP traffic where extreme performance is required
    - operate at layer 4
    - able to maintain millions of requests per second with super low latency
- Classic
    - the legacy elastic ELB
    - you can load http/s traffic and use layer 7 features like X-Forward and sticky sessions
    - you can even use for layer 4 TCP traffic
    - in case your application/db has errors it will return 504

X-Forwarded-For:

- when a user from ip 12.14.12.14 visits your site
- he gets yo your ELB
- that has the local ip of 10.0.0.3
- when the request gets to your application
- the app thinks the ip of the user is 10.0.0.3
- to resolve that and know the real user ip, use: "X-Forwarded-For" header

Sticky Session:

- when you want a specific user to keep on interacting with the same server or server group on the same session
- could be for writing a file to the server
- or when sessions are store locally on the server etc
- checks:
    - when one server gets no traffic you can disable sticky session and check

Cross Zone Load Balancing:

- if enabled
    - your load balancer can split traffic between instances in multiple zone
- if not
    - the ELB will stick with instances on its zone

Path Patterns:

- you can define rules by the url paths
- that lets say all routes will go to a group of instances
- and for path `/images` the traffic will go to a different AZ

AWS - S3 - Simple Storage Service

Exam Tips:

- Object based. allows to upload files
- files can be from 0 bytes to 5 TB
- unlimited storage
- files are stored in buckets
- s3 names are universal

- recognize : https://s3-**REGION**.amazonaws.com/**BUCKET_NAME**
- upload an objects gets 200
- key - value pair
- the data consistency on write or update
- 6 different storage types
- you can enable MFA to block file deletion
- go through S3 FAQ -> https://aws.amazon.com/s3/faqs/
- default buckets amount - 100

- safe place to store your files
- object based storage
- file size from 0 bytes to 5 TB
- unlimited storage
- files are stored in buckets (buckets are like folders)
- bucket name should be unique globally
- example url:
- https://s3-**eu-west-1**.amazonaws.com/**acloudguru**
- when you upload a file you always get 200 on success
-

How Data is stored:

- key - simply the name of the object
- value - the data of sequence of bytes
- version ID (important for versioning)
- metadata - data about what you are storing
- subresources - access control lists, torrents

Data Consistency:

- when you write a new file, you will be able to read it immediately
- when you update an existing file, when you read it, you may read the new version or the older one
- same goes for deletion, you may still have access to the old file that you deleted

AWS Guarntees:

- durability 99.999999999% (11 x 9s)

S3 features:

- tiered storage
- lifecycle management
- versioning
- encryption
- **access control** lists an **bucket policies** s

S3 storage classes:

- Standard
  - 99.99% availability
  - 99.999999999% durability
  - design to sustain the loss of two facilities concurrently
- IA - Infrequently Accessed
  - 99.9% availability
  - access less frequently but you still need rapid access. You are charged a retrieval fee
- IA one zone
  - 99.50% availability
  - same as the above but on one zone
- Intelligent Tiering
  - will move data automatically with machine learning to the best spot
- Glacier
  - 99.99% availability
  - very cheap option but retrieval takes between minutes to hours
- Glacier deep archive
  - 99.99% availability
  - 12 hours retrieval is acceptable

LifeCycle Management:
You can define that after X days a file will move from standard to IA and then glacier etc.

Versioning:
you can have versions of the same file each time it updates

Encryption:
select your own encryption fo the stored files

How Are you charged?

- storage
- requests
- storage management pricing
- data transfer pricing
- transfer acceleration
- cross region replication

transfer acceleration:
This means you will use AWS own network for transfers. So if your bucket is in NY and your uploading user is in AU. If we use the acceleration, the user will upload to the edge location on AU. Then the transfer to the bucket will go through AWS internal network.

Cross region replication:
Simply means you have a primary bucket that every time it changes it will get replicated to another bucket.

How re you billed:

- storage
    - the amount of data you store
- requests
    - how many requests you use for fetching/updating/deleting
- storage management pricing
    - the tier you use for storage
- data transfer pricing
    - the amount of bytes you transfer through the network
- transfer acceleration
- cross region replication

Encryption

- Encryption in transit. Achieved by ssl/https
- Encryption at rest. Stored encrypted on the server with three options:
    - S3 managed keys - SSE-S3
    - AWS key management service, managed keys. - SSE-KMS
    - server side encryption with customer provided keys. SSE-C
    - —— client side encryption. simply means you upload encrypted files

Versioning

- versioning can be turned on or suspended only
- you can add security for file deletions with MFA
- file deletion would just add a “delete” flag, but you can still delete specific versions

Lifecycle

- automates the moving of your files between different tiers
- can be used with versioning

Cross Region Replication

- versioning must be enabled on both buckets
- regions must differ
- files in existing bucket not automatically synced
- delete markers or deleting versions are not replicated

transfer acceleration

- instead of uploading directly to your bucket, you can upload to AWS edge location and then AWS will use its internal network to forward the upload to your bucket

Cloudfront

- CDN - content delivery network
- Edge location is where your files are cached
- origin this is the original source of your files, it can be s3, ec2 etc
- distribution - is the given name for the can that consist a collection of edge locations
- TTL - is the time to live for cached objects (configurable)
- you can invalidate the cached objects, but you will be charged
- two types: “web” for files & RTMP for streaming

Snowball
a big physical computer that get delivered to you. You then can upload files and as AWS to come back and pick it up. What ever you store you can tell AWS what to do with when its gets back to them.

Storage Gateway

a way to deliver data from your data centers to aws s3

Storage Gateway

- file gatway
    - stores flat files in s3
- volume gateway
    - stored volumes
        - entire datasets stored on site and async backed to s3
    - cache volumes
        - entire datasets stored on s3 and most frequent data store on site
- taped gateway
    -

AWS - Overview

Currently: (2019)

- 24 regions
- 72 availability zones

Region:
a geographical area that contains 2 or more availability zones

Availability zone:
is simply a data center inside a region.
may be several data centers but they are relatively close to each other.

data center:
just a building filled with servers

Edge location:
another smaller center that spread around the world. Those are used for caching data for users.
or for using the faster upload to amazon network.

AWS - Elastic Beanstalk

Exam Tips:

- you can quick deploy and manage your app without really knowing about AWS
- its built on top of cloud formation but you don't need to think about it
- it will take care of capacity, load balancing, scaling and health monitoring

AWS - CloudFormation

Exam Tips:

- is a way to completely script your cloud environment
- quick start is a bunch of cloud formation already built by AWS solutions architects

AWS - Databases RedShift

great service for data intelligence or data warehousing in the cloud.

Exam Tips:

- enabled by default with 1 day retention
- max retention is 35 days
- RedShift always attempt to have 3 copies of your data.
    - the original
    - replica on the compute nodes
    - backup on S3
- Redshift can also asynchronously replicate your snapshots to S3 in another region for disaster recovery
- great for OLAP (Online Analytical Processing)

Redshift Configure Options:

- Single Node (160gb)
- Multi-Node
    - leader node, manage client connections and receives queries
    - compute nodes, stores data and performs queries and computations. up to 128 compute nodes

Backups:

- enabled by default with 1 day retention period

- max retention period, 35 days
- RedShift always attempt to have 3 copies of your data.
    - the original
    - replica on the compute nodes
    - backup on S3
- Redshift can also asynchronously replicate your snapshots to S3 in another region for disaster recovery

How are you billed:

- compute node hours. 1 hour per node.
- you won't be charged on the leader node.
- backups
- data transfer (only within a vpc)

security:

- in transit with ssl
- at rest using AES-256
- by default aws manages your keys, but you can manage your own

Availability:

- currently only on one AZ
- can restore snapshots to a new AZ in case of an outage

AWS - Databases ElasticCache

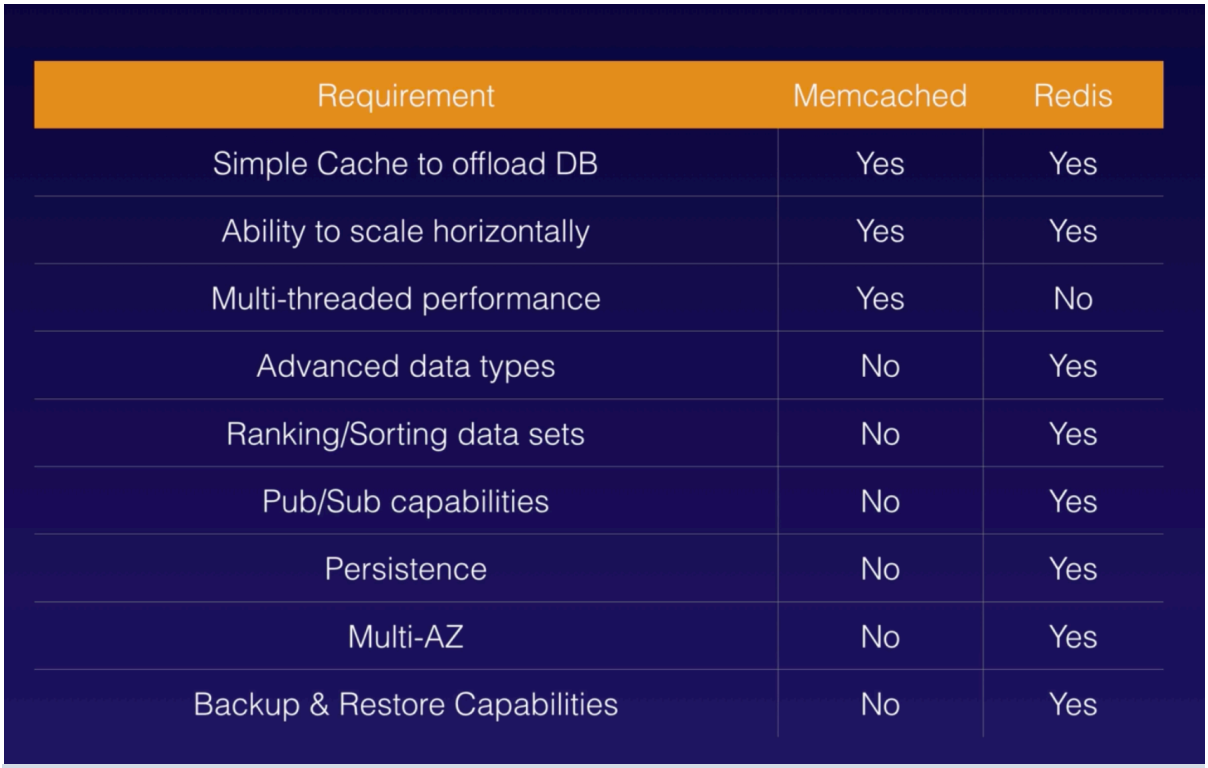
A web service that allows you to create an in-memory cache in the cloud. Instead of relying on slow disk-based DBs.

Exam Tips:

- use elastic cache to increase DB performance
    - Redis is multi-AZ
    - you can do backups and restores of Redis

Supports 2 OS engines:

- Memcached
- Redis



Requirement	Memcached	Redis
Simple Cache to offload DB	Yes	Yes
Ability to scale horizontally	Yes	Yes
Multi-threaded performance	Yes	No
Advanced data types	No	Yes
Ranking/Sorting data sets	No	Yes
Pub/Sub capabilities	No	Yes
Persistence	No	Yes
Multi-AZ	No	Yes
Backup & Restore Capabilities	No	Yes

AWS - Databases DynamoDB

noSQL DB that supports key-value and documents.
pay for what you use.

Basics:

- stored on SSD
- spread across 3 geographical distinct data centers
- consistent read
    - eventual consistent read (default)
        - could take more then a second to persist for future read
    - strongly consistent read
        - if u need to read up to a second after a write

AWS - Databases

Exam Tips:

- RDS
    - runs on virtual machines
    - you can’t login to those machines
    - patching the databases or the OS is AWS responsibility
    - this is not serverless
        - Exception - "Aurora serverless” is serverless

Types:

- RDS (OLTP - online transaction processing)
    - SQL
    - mySQL
    - postgresSQL
    - Oracle
    - Aurora
    - MariaDB
    - RedShift
        - business intelligence or data warehousing
- NoSQL
    - DynamoDB
    - ElasticCache
        - used for speed up performance of existing databases (frequent identical queries)
        - build on: Redis or Memcache

AWS - Databases RDS

Backups & Snapshots:

- backups
    - are done automatically every day for 35 days, and we do have daily transactions
    - so you can revert to a specific second in time
- snapshots
    - are done manually
- both are store on S3, and you have free storage as the size of your RDS
- whenever you restore a backup it will be on a new instance with new dns

Multi A-Z:

- AWS will handle replicating your DB to another AZ
- so in the case one DB fails, AWS will just used the the replicated DB
- no need to changed any DNS as it stays the same
- this is used for disaster recovery
- available for: sql, oracle, mysql, postgres, mariadb

Read Replica:

- its a copy of your main DB, and its replicated synchronously
- its replica has its own DNS
- you can have read replicas with multiple AZ
- you can have replicas for a multi-AZ source DB
- you can have a read replicas in a different region
- so you can tell your apps to use the main DB for writes and the replicas for reads
- you can have read replicas for read replicas and so on
- this used for improving performance
- available for: oracle, mysql, postgres, mariadb, Aurora
- Must have automatic backups on in order to turn on read replicas
- you can have up to 5 read replicas of any DB
- you can have read replicas of read replicas (but watch for latency)
- can be promoted to master, but it will break the read replica

AWS - EC2 - Placement Groups

Exam Tips:

- clustered placement can't span on multiple AZ
    - a spread and portioned can
- a placement group name must be unique in your AWS account
- only certain types of instances can be launched in placement groups
- AWS recommends homogenous instances within clustered placement groups
- you can merge placement groups
- you can move an existing instance to a placement group. What you can do, is create an AMI from the instance and recreate it into the placement group

3 different types:

- clustered placement group

    - grouping instances in the same AZ
    - low network latency, high network throughput or both
- spread placement group
    - protects from hardware failure
    - each instance is on different hardware
    - either on the same AZ or different AZ depends how you configure it
- partitioned
    - similar to spread placement group
    - but here you can partitioned group of instances that will be together on the same hardware

AWS - CloudWatch

Exam Tips:

- Easy to mix with CloudTrail. CloudTrail monitors API calls in the AWS Platform. CloudWatch monitors performance.
- With EC2 will monitor events every 5min by default
    - You can have 1min intervals by enabling detailed monitoring
- you can create CloudWatch alarms for triggering notifications

Service that monitors your AWS resources and the applications that you run on AWS. Basically watches the performance.

Host level metrics:

- CPU
- network
- disk
- status check

What can I do in CloudWatch:

- Dashboards - create dashboards to see what is happening with your aws env
- Alarms - setting alarms to for notifying on different thresholds
- Events - you can respond to defined events in the system
- Logs - you can aggregate, monitor and store logs