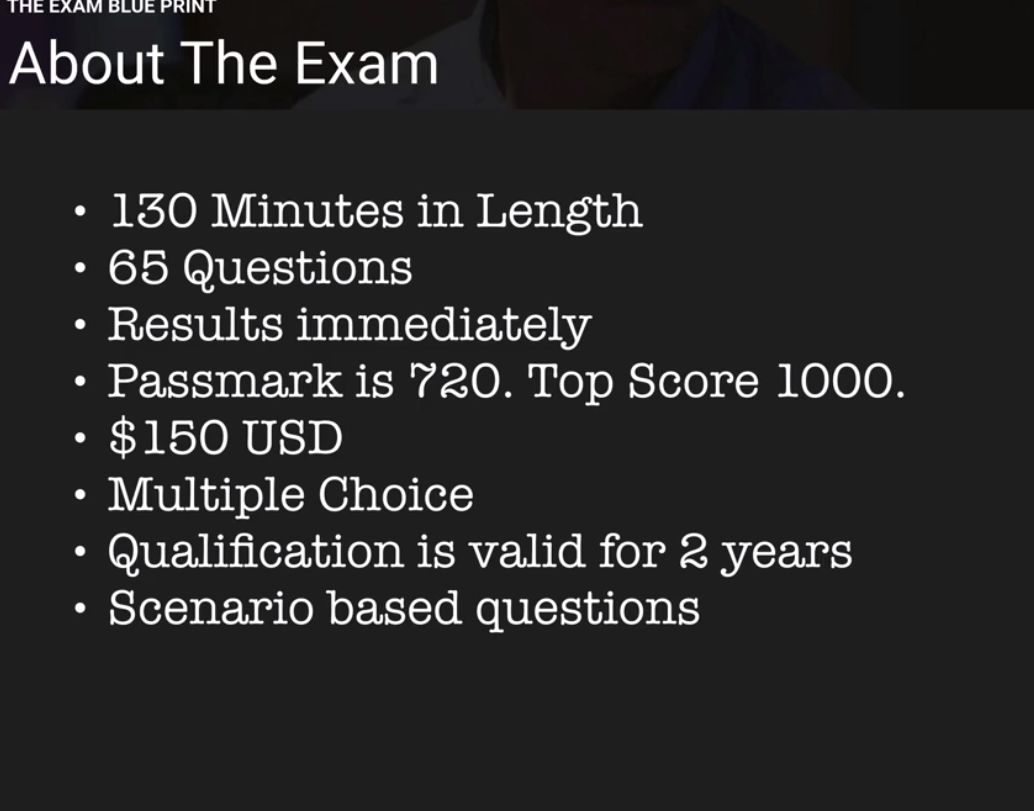
**AWS DEVELOPER CERTIFICATION**



**IAM – IDENTITY ACCESS MANAGEMENT**

**IAM is to manage the users , roles and permission through AWS console**

**IAM gives**

1. **Centralized control of AWS account**
2. **Shared access to AWS account**
3. **Granular permissions**
4. **Identity Federation (including AD, FB LinkedIn) which means it enables to authenticate the users within the AD or facebook or LinkedIn etc**
5. **Multi-factor authentication if the company want to give access to the user only when multiple ways to authenticate into the account.**
6. **Provides temporary access to the resources. For example if the user need access to sql db for a while and then dynamodb after that**
7. **Allows for password rotation policy.**
8. **Integrates with many different AWS services. You will have many AWS services**
9. **Supports PCI DSS compliance.**

**Terms**

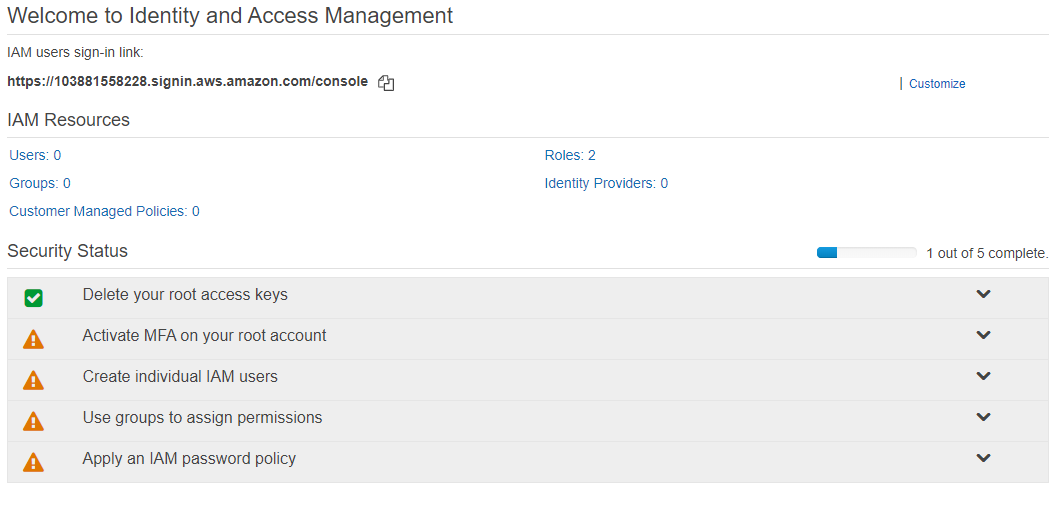
**Users – who are loggin into AWS account**

**Groups- Collection of users under one set of permissions. In order to do this create a group with set of permissions and assign the users to that group.**

Roles:

Policies: a **document that defines on or more permissions.**

**This is default view when using IAM for the first time. We normally not supposed to use the root account unless it is so needed. We create the IAM user and use that account to login.**



1. **Activate MFA on root account** 
   1. **You will have two options A virtual MFA device, U2F Security Key and A hardware MFA device**
   2. **For virtual device user need to have the MFA compatible device.**
2. **Adding users:**
   1. **For adding user there will be two access type options**
      * 1. **Programmatical access: After adding user , it will give Access key and Network secret key, The user need to have both to access the AWS programmatically. These credentials only show once for the first time. SO it is better to download as CSV file.**
      1. **Console access : It gives Alias name and password.**
3. **Role: Role can be created an assigned to instance. For example create a role with one or more permissions (Like S3 permissions) and assigned to an EC2 instance.**
4. **Policy: Policy is permission document which is in Json format. We can create our own new policy too.**



**EC2**

**EC2 Elastic Cloud computing is a webservice that provides to resize the computer capacity with in minutes. In olden days this task takes several days to finish as it needed to set the cables connect the new server etc.**

**This enables that you only pay for the capacity what you use for. And also it provides the developers not to worry about the common network or server failures scenarios.**

**EC2 options**

1. **On Demand: You pay per hour or by second and there no commitment required depending on the instance. For example as of now windows need per hour pay and linux server need per second pay. This may change .**

**This is perfect for the users whose applications are spiky, flexible loads and even at the time of development.**

1. **Reserved : Provides an option to reserve the capacity with the commitment with hug discount over the OnDemand option. The contracts usually be 1 to 3 year contracts.**

**Perfect for stable and predictable usage applications. If you pay upfront there will be great discounts.**

**Standard Reserved instances will have upto 74% off On demand.**

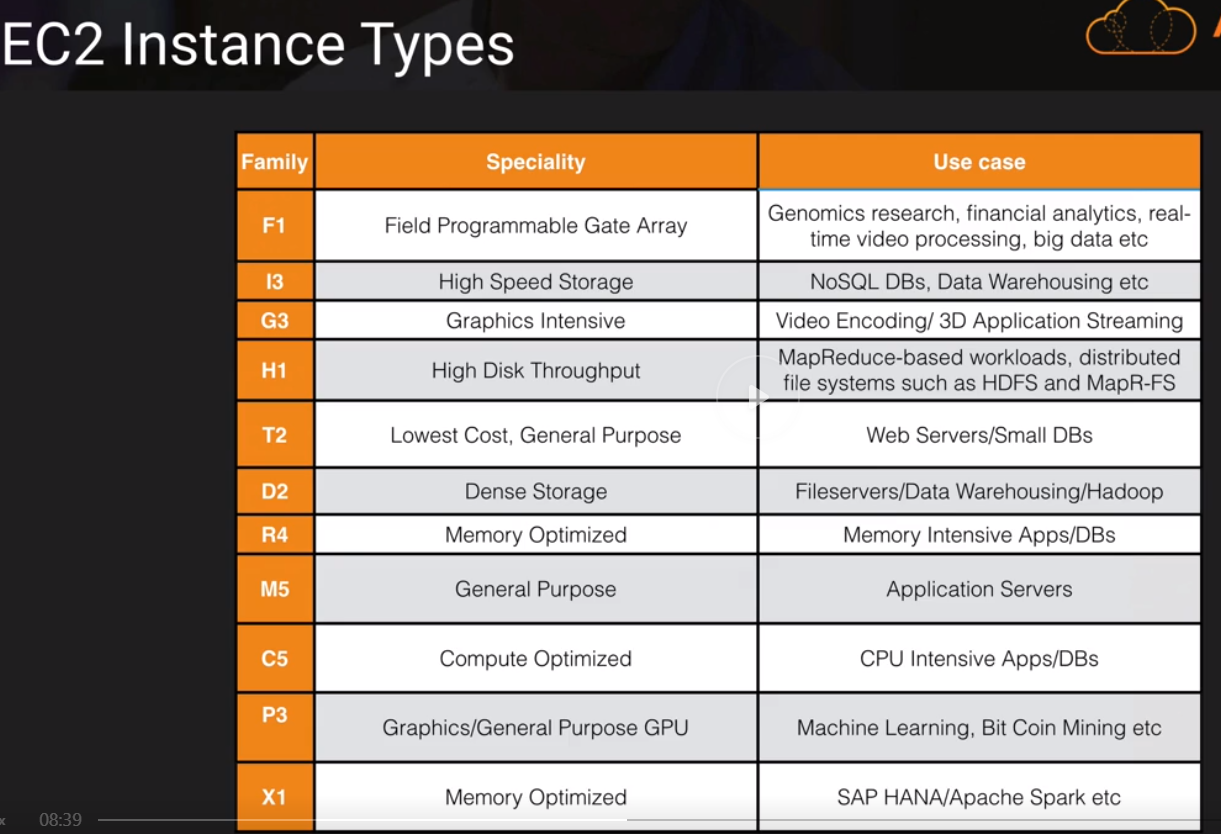
**Convertible RI s will have 54% off discount. For example if you want to go CPU intensive RI to Memory intensive RI.**

**Scheduled RIS : If you see a predictable usage in a day or month , you can schedule the RI to have that time to be peak. This option allows to match your capacity reservation to a predictable recurring schedule.**

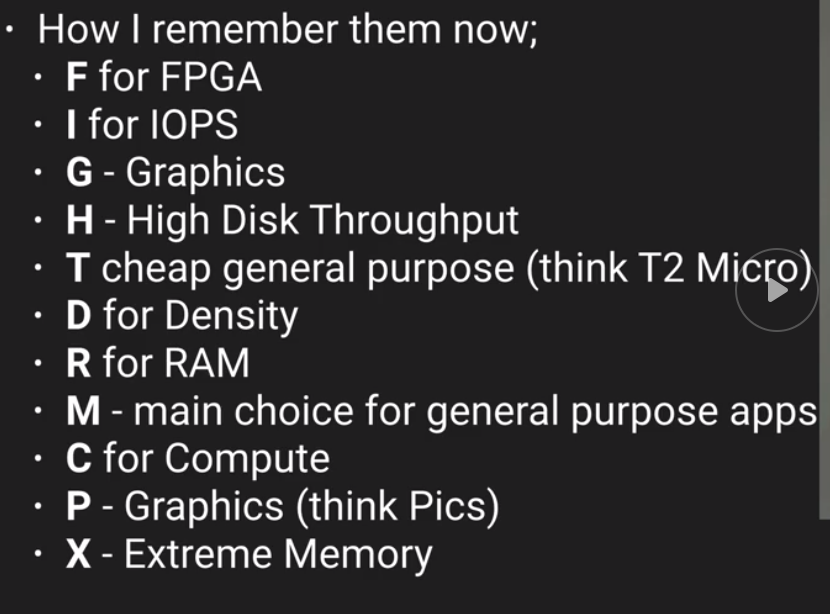
1. **Spot**: This is you can bid for the price. Spot prices will vary time time. But the users need to be flexible with start and end times.
2. **Dedicated Hosts** : Dedicated hosts are physical EC2 servers where you can bring your software licenses to AWS and use them. Examples are VMWARE servers, Oracle servers and SQl servers etc.

**Useful for regulatory requirements that may not support multi tenant virtualization**.

**You don’t remember all these options for this exam may be just the family names should be good memorize**



**FIGHT DR MC PX**



**EBS (Elastic Block Storage)**

**EC2 is virtual instance and EBS is virtual disk in the cloud. EBS allows to create a virtual disk and attach to EC2 instance. Once it is attached we can run database , file systems or application on top of it. EBS volumes placed in a specific availability zone to prevent single point of failure. So it is not just exist in one physical disk rather it is spread out availability zone.**

**EBS volume types**

1. **General Purpose SSD / GP2 : balanced b/w price and performance.**
   1. **It well suits for ratio of 3 IOPS (Input Output Per Second) per GB with up to 10000 IOPS and the ability to burst upto 3000 IOPS for extended period of time for volumes at 3334 GB and above.**
2. **Provisioned IOPS SSD / 101:** 
   1. **Suits for high intensive applications such as large relational or NOSQL Databases.**
   2. **Use if you need more than 10000 IOPS.**
   3. **Can provision upto 20000 IOPS**
3. **Throughput Optimized HDD / ST1 :** 
   1. **For Big Data/ Data warehouses / log processing**
   2. **Can not be a boot volume.**
      1. **Boot volume mean the portion of the hard drive that contains the operating systems. So in this case ST1 can be additional drive on the server just like we have D , E drives in the windows servers.**
4. **Cold HDD / SC1**
   1. **Lowest cost storage for infrequently accessed workloads. Typically use for file systems.**
   2. **Can not be a boot volume either.**
5. **Magnetic EBS**
   1. **Lowest cost per GB of all EBS volume types that is bootable.**
   2. **This EBS type is ideal for where data is accessed infrequently and where the lowest storage cost is important. Best example where you need a database for development and testing environments**

**Exam Tips:**

**HDD EBS storage volumes can not be boot volumes.**

**If the spot instance is terminated by Amazon EC2 you will not be charged for partial hour. But if you terminate the instance you need to pay for the full hour.**

**EC2 LAB**

**\*the cutting edge technologies will be available in Virginia region.**

**Launching web server.**

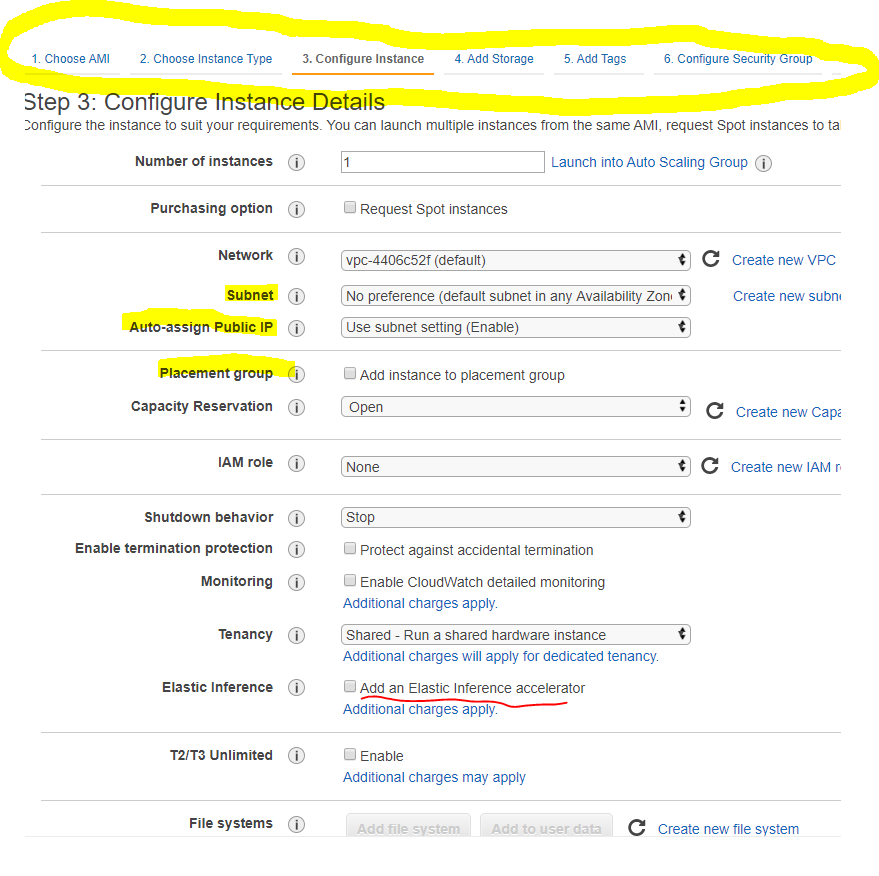
**You can create any of the Amazon Machine Images such as Linux, windows redhat, SUSE, Ubuntu etc**

**You can have build a webserver from AWS market place (?)**

**You can build webserver from Community AMIs such as Barracuda, F5. These web servers have default server settings that you can use.**

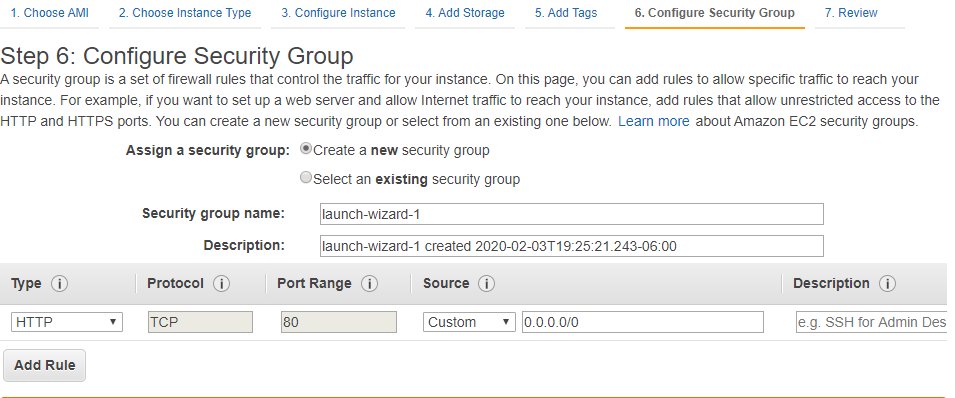
**VPC: Virtual private Cloud. You can create your own VPC to select IP range, create own subnets. Configure route tables etc**

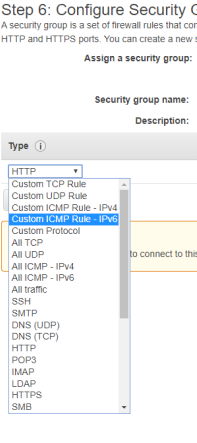
**Tenancy : Possession of being tenant**



**Configure security group Tab: Here you can give permission who can access the web server. So security group in AWS is a set of firewall rules that control the traffic for your instance**

**\* DMZ or demilitarized zone (sometimes referred to as a perimeter network or screened subnet) is a physical or logical subnetwork that contains and exposes an organization's external-facing services to an untrusted network, usually a larger network such as the Internet. The purpose of a DMZ is to add an additional layer of security to an organization's local area network**

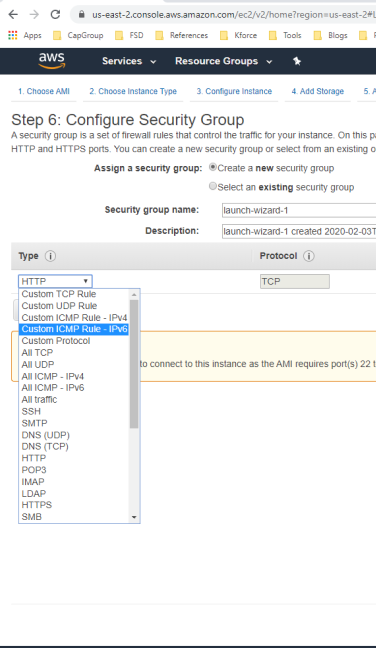


 **Need to know all these types**

**SSH is the protocol to connect to Linux**

**RDP is for Windows :**

**These are the protocols to just connect remotely and access the EC 2 instance.**

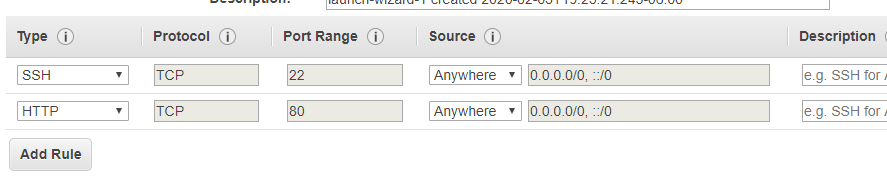


**In this example we can remote desktop into this instance using 22 port. And the web traffic comes from 80 port.**

**We mentioned source to Anywhere (this is not secure)**

**0.0.0.0/0 for Ipv4 and ::/0 is for Ipv6 (????) cider notation**

**This I basicall y saying connect any one using SSH notation**



**Then Review and launch**

**It asks to download the public and private key file with extension .pem .**

**Run the EC2 instance from MAC**

**Go to Terminal**

**Cd <the location where you downloaded the public and private key>**

**Chmod 400 MyKeypair.Pem (This will change the permissions and lock down the file)**

**Ssh ec2-user@<ip address of the ec2 server> - i <keyname.pem> (This is to connect to EC 2 server through SSH) . and add the computer to the list of known hosts**

**Sudo su (this change to run the programs as another user in this case su (super user) which means you are logged in as super user in the EC 2 server)**

**Yum -update -y (**Yellowdog Update, Modified**(YUM) is a program that manages installation, updates and removal for**[**Red Hat package manager**](https://www.bing.com/search?q=RPM+Package+Manager&filters=sid%3a6a8c1e1a-bf60-2387-900d-349f7cf9dbbf&form=ENTLNK)**(RPM) systems. YUM allows the user to update groups of machines without having to update each RPM separately.) with this command it updates the sever with all the latest packages**

**Yum install httpd -y (this install apache in the server, Apache turns the EC 2 server into webserver)**

**Service httpd start (this starts the apache server)**

**Chkconfig httpd on (this is the configuration setting that we are doing. In case ec2 server starts , this little configuration will let the apache to run automatically rather we start again manually)**

**Service httpd status**

**Creating web page from terminal**

**Cd /var/www/html**

**ls (check the directory)**

**nano index.html (creates the file)**

**Elastic Load Balancer EBL**

**To balance the load between the web servers. If one webserver is overwhelmed ,**

**Three types of load balancers**

**Application load balancer**

1. **Work at OSI layer 7**
2. **Best suited for *HTTP* and *HTTPS* traffic**
3. **You can do advance request routing, sending specific requests to specific web servers. For example, if you need to request the sales.domain.com., we can do it to go to sales web server.**
4. **Model x**

**Network load balancer**

* 1. **Best suits for load balancing of *TCP* traffic where extreme performance is required.**
  2. **Operates at layer 4.**
  3. **Most expensive and high performant**
  4. **Network load balance is capable of handling millions of requests per second.**

**Classis load balancer \***

1. **Classic load balances are the legacy elastic load balancer. Even though Amazon is not pushing these loadbalncers, exam mainly focus on this.**
2. **Use to load balance *HTTP / HTTPS* applications and use layer 7 specific features, such as X-forwarded and sticky sessions.**
3. **Also use strict layer 4 load balancing for applications that rely purely on the *TCP* protocol**

**Load Balance erros**

**If there is an error 504 error code, it means there is an issue with application layer or database layer. Which means this is nothing to do with load balancer.**

**Some important points**

**The**X-Forwarded-For**(XFF) header is a de-facto standard header for identifying the originating IP address of a client connecting to a web server through an HTTP proxy or a load balancer. When traffic is intercepted between clients and servers, server access logs contain the IP address of the proxy or load balancer only. To see the original IP address of the client, the X-Forwarded-For request header is used.**

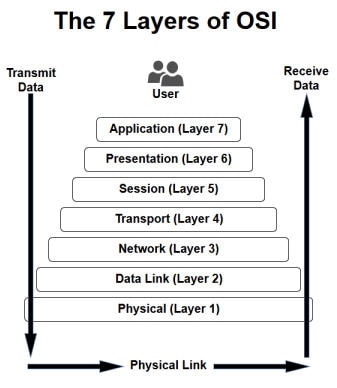
***TCP vs HTTP***

**TCP provides the delivery of a stream of bytes from a program from one computer to another computer.**

<http://www.differencebetween.net/technology/internet/difference-between-tcp-and-http/>

**OSI model is not a physical function or it doesn’t perform any functions. It is a conceptual framework to understand the complex interactions that are happening.**

**Layers 1-4 are considered the lower layers, and mostly concern themselves with moving data around. Layers 5-7, called the the upper layers, contain application-level data**



**Route 53**

**If there is an error 504 error code, it means there is an issue with application layer or database layer. Which means this is nothing to do with load balancer.**

**Route 53 is Amazon Domain Name service**

**It maps the domain name for**

1. **EC2 instances**
2. **Load balancers**
3. **S3 Storage / Buckets**

**Under the services 🡪 Network 🡪 Route 53**

**If you don’t have registered domains to create hosted zone, create one domain.**

**Mine is alienbot.net**

**Once we register domain name AWS created hosted zone for us.**

**We need to create ‘A record’ for the hosted zone. Typically that would be a load balancer.**

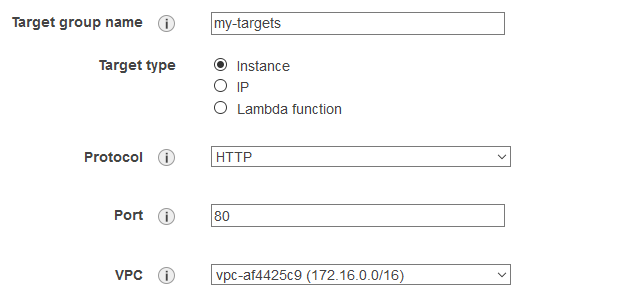
**A record : An**A record**maps a domain name to the IP address (Version 4) of the computer hosting the domain. An A record is used to find the IP address of a computer connected to the internet from a name. The format is 32 bit.**

**AAAA record : This is A record but for Ipv6 protocol . The format is 128 bit**

**To create load balancer EC2 🡪 Load Balancer. While creating load balancer , we need to register the *target group* to the instances that we need to map to.**

**Target type could be instance or IP or lambda function**

**Target Group : Load balancer routes requests to the targets in this target group using the protocol and port that you specify and perform health check on the targets using these health check setting. Each target group can be associated with only one load balancer.**



**CLI DEMO Command Line interface**

**To access any service from CLI type aws <service name> ls**

**Here aws s3 ls**

**To create a bucket**

**Aws mb s3://personalbucket**

**Here mb is make bucket**

**Echo test text > test.txt**

**Aws s3 cp test.txt s3://personalbucket**

**Cp – copy this command copies test.txt to s3 bucket that specified**

**There are many commands to use with CLI. Check the documentation but we don’t need to remember those commands for the exam.**

**EC2 with Role**

**When you create a role to an EC2 instance for specific service (for example for S3) EC2 will talk to S3 with that role instead of using the specific user.**

**SO Roles allow you to not use access key ID s and secret access keys.. Roles always preferred for secured access over using individual users. You can change a policy on a role and it effects immediately on all the instances. You can attach and detach roles to runnind EC2 instances, you don’t need to stop and start instances. For exam if you have question about roles and accesskey ids always choose roles.**

**How to Encrypt an EBS Volume**

**When creating EBS volume, if you attach encrypted snapshot volume it will be encrypted, if you attach an unencrypted volume it will be unencrypted. If you are creating new volume you have an option to Encrypt the Volume. This can be a question in exam.**

**EBS volumes should be created in the same availability zone that EC@ instance is created.**

**Create EBS volume with encryption enabled**

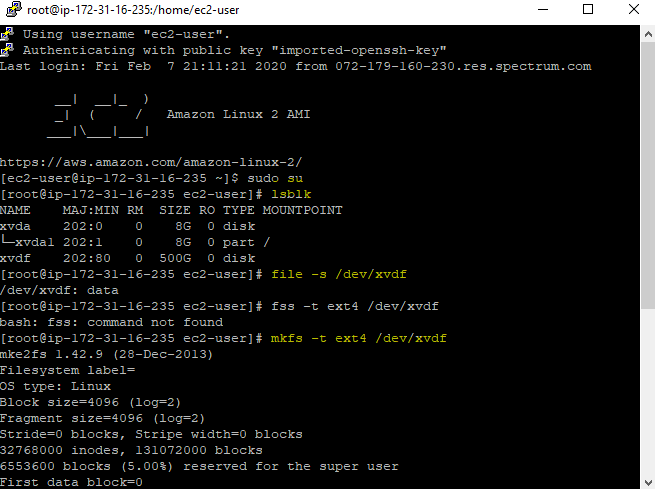
**SSH into server**

**lsblk – list all the drives on the server.**

**The one drive with / is the root drive**

**file -s /dev/xvdf -- to check if there are any file systems already exists**

**mkfs -t ext4 /dev/xvdf -- to make file system(mkfs) of type (-t) ext4 on /dev/xvdf**



**we haven’t mounted the file system yet. To mount this**

**cd / -- go to root directory (which means where the OS is installed)**

**mkdir alienbotfilesystem -- make a directory in the root directory**

**mount /dev/xvdf alienbotfilesystem -- now mount the EBS volume**

**After this we unmount the EBS volume , detach the volume from console, create a snapshot of the volume , Delete the volume. Then create volume from the snapshot**

**umount -d /dev/xvdf -- to unmount the EBS volume from the root directory.**



**File system Types – No need for exam just for information**

* auto - this is a special one. It will try to guess the fs type when you use this.
* ext4 - this is probably the most common Linux fs type of the last few years
* ext3 - this is the most common Linux fs type from a couple years back
* ntfs - this is the most common Windows fs type or larger external hard drives
* vfat - this is the most common fs type used for smaller external hard drives
* exfat - is also a file system option commonly found on USB flash drives and other external drives

**RDS**

**Relational Databases that AWS provides**

1. **SQL**
2. **MySQL**
3. **Oracle**
4. **PostgreSQL**
5. **Amazon Aurora : this amazon own database that similar to MySQL Amazon Aurora don’t have free tier template like others. (This could be exam question)**
6. **MariaDB**

**DataWarehousing**

**Use Datawarehouse to pull in very large and complex datasets and run queries against it to create reports. Examples CosmoDB, JasperSoft, SSRS, Oracle Hyperian etc.**

**OLTP vs OLAP  
OLTP : Online Transaction Process is just adding minimal information when a transaction occurs like order is placed.**

**OLAP : Online Analytical Process have much more data than OLTP. We user DWH for this purposes. DWH have different architecture and infrastructure pattern than regular Databases..**

**RedShift**

**ElasticCache**

**Elastic chache is a web service that makes it easy to deploy operate and scale an in-memory cache in the cloud. The service improve performance of web application by not pulling data from production databases but cache.**

**Elasctic cahe supports two open source Caching engines**

* **MemCached**
* **Redis**

**Creating RDS**

**After creating RDS amazon gives us the DNS name of the database. It will look like this**

**Endpoint**

**alienbotdb.cl2heusaibwc.us-east-2.rds.amazonaws.com**

**Port**

**3306**

**The exam tip is \*\***

**If EC2 instance is in one security group and RDS is in another security group, it can not communicate with each other it gives ‘unable to connect to RDS’. To resolve this we need open up the rds security group and open up the inbound rules and add the webserver security group to allow to connect on the RDS port. (3306)**

**Multi AZs and Read Replicas (Very important)**

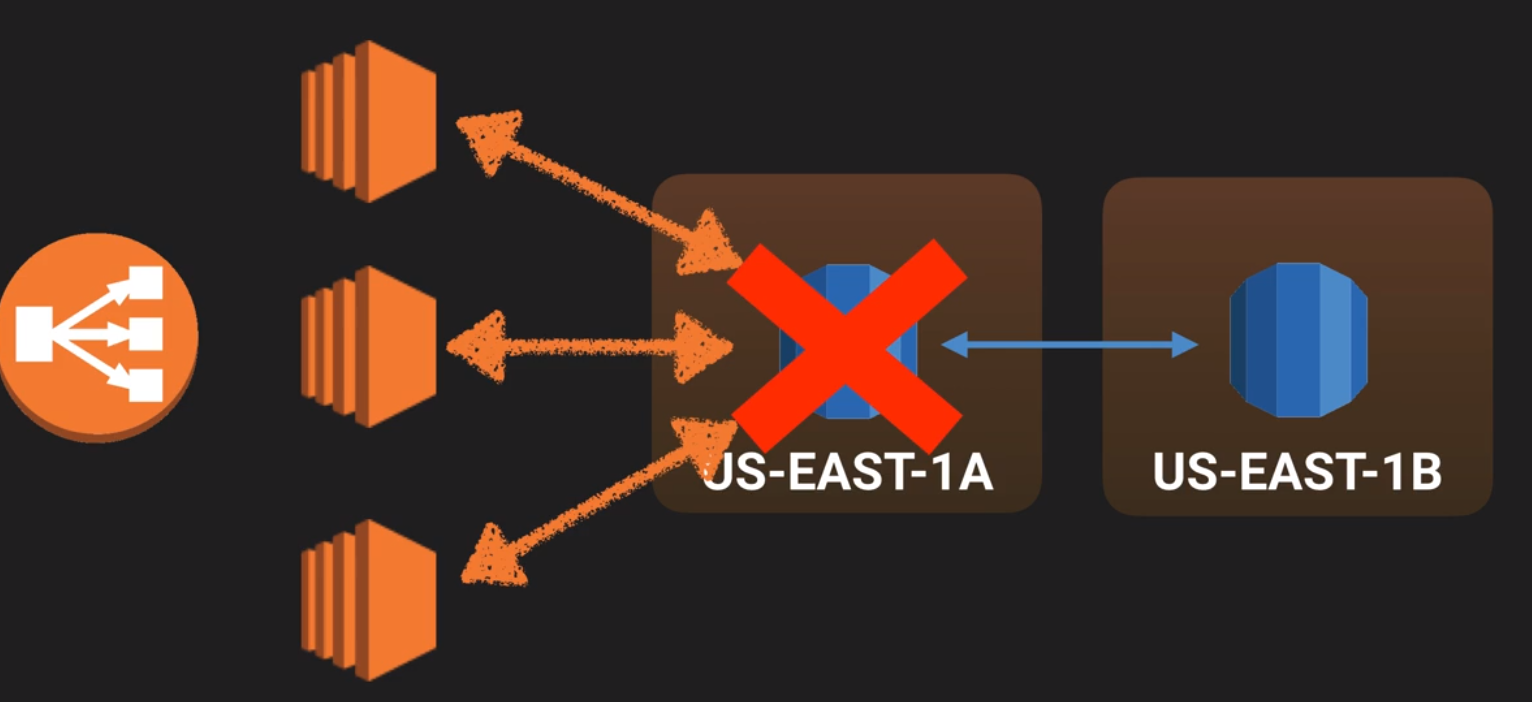
**There are 2 different types of backups**

1. **Automated backups: Allow you to recovers the database to any point in time within a retention period. The retention period can be anywhere from 1 day to 35 days. These backups will take full daily snapshot and also store transaction logs.** 
   1. **AWS gives the backup database size same as the database instance. Thee backup data is stored in S3**
   2. **When back ups are taken within a defined window storage I/o may be suspended. So latency is expected**
2. **Snap Shots : this is manual backup. They are stored even after you delete the original RDS instance, unlike automated backups.**

**Restoring backups: When you do the restore, the new restored version of the database will be new RDS.**

**Multi AZ**

**The RDS instance have a replica RDS in US-EAST-1B ..in case of failover the amazon automatically updates the database DNS to the new one.**



**Multi AZ allows you to have an exact copy of production db in another Availability zone. AWS handles the replication, so when the prod DB is written to, this write will automatically synchronized to the stand by database.**

**\*\*\* Important for Exam :Multi AZ is for Disaster recover only. It is primarily used for improving performance. IF we need performance we need Read Replicas. All the RDS (sql , oracle , my sql) provides multi AZ s. Amazon aurora has it by default.**

**Read Replicas : Read replicas allows you to have a read only copy of the prod DB. This is achieved by using Asynchronous replication from the primary RDS instance to the read replicas. These are basically designed for read only traffic. Like for reporting. These are allowed only for MySQL, Postgresql, MariaDB and Aurora. Not allowed for SQL and Oracle.**

1. **Must have automatic backups turned in order to deploy a read replica.**
2. **You can have up to 5 read replicas. You can have read replica of a read replica but watch out for latency.**
3. **Each read replica have its own DNS point.**
4. **You can have read replicas that have multi AZ or complete in another region. You can have the read replica in a second region.**
5. **Read replicas can be promoted to be their own database and this breaks the replication. For instance, if you have more than enough read replicas and you don’t need all of them. So you can have a read replica to have its own database and can do the work on it.**

**Elastic Cache**

**Elastic cache is a web service that makes it easy to deploy . operate and scale an in-memory cache in the cloud.**

**Types of ElasticCache**

1. **MemCacheD : A widely adopted memory object caching system. ElasticCache is compliant protocol with MemCached.**
2. **Redis** : **A popular open source in memory Key- Value store that supports data structures such as sorted sets and lists. ElasticCache supports Master/Slave replication and Multi-AZ which can be used to achieve cross AZ redundancy,**
   1. **SO if you want multi AZ redundancy, use Redis cache and if you don’t need multi-AZ use MemCached**

**Although both caches appear similar (as they are both in memory key stores) but they are different in practical. Because of the replication and persistent features of Redis, ElasticCache manages Redis more as a relational DB. Redis Elastic cache clusters are managed as stateful entities that include failover similar to Amazon RDS.**