**[EC2 (Elastic Compute Cloud):](https://aws.amazon.com/ec2/)**

* **Amazon EC2 (Elastic Compute Cloud)** is a web service interface that provides resizable compute capacity in the AWS cloud. It is designed for developers to have complete control over web-scaling and computing resources.

nstance types:

Remember instance types for the exam.

[EC2 Instance Types](https://aws.amazon.com/ec2/instance-types/)

|  |  |
| --- | --- |
|  |  |
| M | General Purpose i.e. app server (default) |
| T | Micro instances - Low cost, general purpose, web servers |
| C, D, I | CPU/IOPS optimised, good for memory intensive compute |
| R, X | Memory optimised |
| G, P | GPU |
| F | FPGA - Field Programmable Gate Array, for hardware acceleration of code |

Useful mnemoic: Dr Mc Gift Px

Instance Options

* **On Demand Instance** - pay a fixed rate per hour. Good for apps where compute needs scaling up/down - i.e. usage might increase 10x during certain hours of the day, or certain times of year.
* **Spot Instance** - available via bidding against other customers.
  + Can be extremely cheap
  + Can be terminated by you OR AWS at any time
  + Best for jobs which can be terminated at any time i.e. certain types of batch processing
  + Not charged for partial hour if your instance is terminated by AWS. Charged for the FULL hour if YOU terminate your instance.
  + Good for massively parallel computations, or high-compute batch jobs, due to the fact that you can get spot instances for often 50-90% less than on-demand instances, you can massively increase your compute capacity by 2-10x for the same budget.
* **Reserved Instance** - fixed compute, reserved for a certain period of time. Cheaper than on-demand if you have predictable long term usage.
* **Dedicated Host** - physical EC2 server available only to you. No shared. i.e. if a regulatory body says that you must not be using multi-tenant computing.

## **EBS (Elastic Block Storage)**

EBS volumes appear as native block devices, similar to a hard drive of other physical device.

EBS volumes can only be scaled up, not down.

All EBS volumes can be changed on the fly, except for Magnetic (standard), however if you do this, [you’ll need to wait 6 hours before making any further changes to the volume](http://docs.aws.amazon.com/AWSEC2/latest/APIReference/API_ModifyVolume.html).

For changing a volume, the best practice is to first stop the EC2 instance it’s attached to.

Instance store volumes is sometimes called ‘ephemeral storage’. If the underlying host fails or stops, all data will be lost. No data is lost on reboot.

When you create an EBS volume in an AZ, it it automatically replicated within that zone to prevent data loss due to the failure or any single hardware component.

EBS can’t tolerate an entire AZ failure - EBS volumes are only replicated within the AZ, so S3 recommends always keeping a snapshot of your EBS volumes in an S3 bucket for high durability.

If the root volume of an EC2 instance fails, and you need to recover data from it, you can:

* Detach the volume
* Attach it to another instance as a data volume
* Fix issues in the files, copy data out if necessary
* Re-attach to the original instance, and restart

### EBS Volume Types

[EBS Volume Types](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSVolumeTypes.html)

#### **General Purpose SSD (gp2)**

The performance of GP2 volumes are tied to volume size - larger volumes perform better. 3 IOPS per GB, up to 1000 IOPS. Can burst up to 3000 IOPS for extended periods of time.

Bursting and I/O credits are only relevant for volumes under 1TB, where burst performance exceeds baseline performance. i.e. Burst is 3000 IOPS per second, baseline performance at 1TB is 3000 IOPS per second - you are always getting burst performance.

#### **Provisioned IOPS SSD (io1)**

Highest performance SSD. Intended for apps which require more than 10000 IOPS per volume or 160 MiB/s of throughput per volume.

IOPS must be between 100 and 20000.

**The maximum ratio of between IOPS and Volume Size is 50:1**. This means that for an 8GiB volume, you can have a max IOPS of 50\*8 = 400 IOPS.

## **Security**

### Logging in to an instance

[EC2 Key Pairs](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs.html)

Amazon EC2 uses public key cryptography to encrypt and decrypt login information.

To log in to your EC2 instance, you must create a key pair, and provide the private key when connecting to the instance.

Linux instances have no password and you must use a key pair when logging in via SSH.

For Windows instances, you use the key pair to obtain the administrator password, then log in using RDP.

SSH uses port 22, and RDP uses port 3389. If you can’t connect to your instance, check your Security Group and NACL configurations.

### Security Groups and NACLs

Security Groups:

* Block all traffic by default; no traffic rules exist in a freshly created security group.
* Can be used for configuring inbound and outbound traffic rules. i.e. HTTP, SSH, etc.
* Changes are applied immediately.
* Are stateful. This means that inbound rules are also allowed out ([unlike NACLs](http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_ACLs.html)).
* Do not allow blocking of specific ip addresses (use NACs for this instead)

All instances in a security group can communicate with all other instances in that same security group by default.

To change the security group of an instance, right click on the instances, select ‘networking’, and select ‘change security group’

It’s possible to have multiple security groups associated with an instance.

NACLs

* Are stateless. If you have inbound traffic - in order for responses to inbound traffic to be received, any inbound traffic rule will require a corresponding outbound rule.

### Roles

It’s possible ot set role in instance creation and add/remove roles while the instance is running.

Using IAM roles means that you don’t need to store credentials (such as AWS Secret Key and Access Key) in the EC2 instance itself.

i.e. if you want to give your EC2 instance full access to S3, you can use the AmazonS3FullAccess IAM role. You can then run s3 commands such as ‘aws s3 ls’ within the EC2 instance.

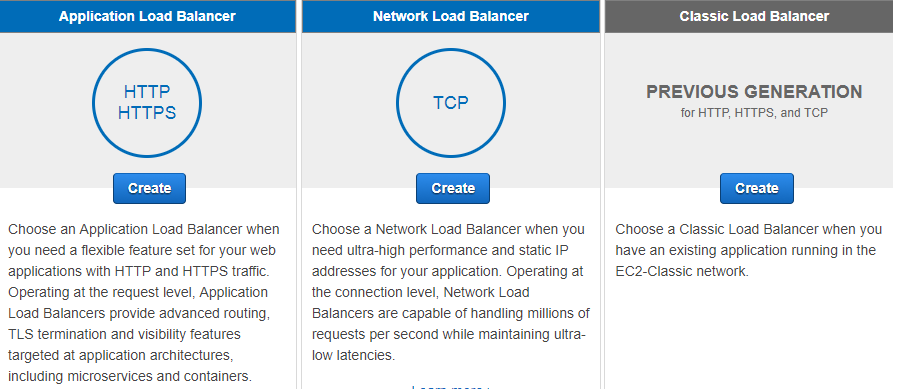
IAM roles are created globally

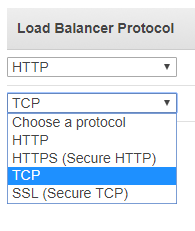
## **ELBs (Elastic Load Balancer)**

[ELB FAQ](https://aws.amazon.com/elasticloadbalancing/faqs/)

ELB supports Perfect Forward Secrecy.

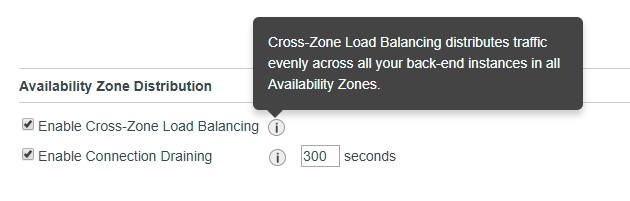
Types of ELB:



* **Classic Load Balancer** is the previous generation Load balancer for HTTP, HTTPS, and TCP traffic.
  + Can load balance HTTP/HTTPS applications
  + Can use layer 7-specific features such as X-Forwarded and sticky sessions
  + Can also use strict layer 4 load balancing for applications that rely purely on the TCP protocol.
  + When using the classic load balancer, you have the option of selecting which protocols it supports:
* **Application Load Balancer** is the current generation load balancer for HTTP, and HTTPS traffic.
  + Can be used instead of the Classic Load Balancer when using exclusively HTTP/HTTPS traffic. Do not use if the aplication depends on the TCP protocol.
  + Operates at the request level
  + Made available half way through 2016.
* **Network Load Balancer** is the current generation load balancer when using exclusively TCP traffic.
  + Can be used instead of the Classic Load Balancer when using TCP traffic.
  + Operates at the connection level
  + The Network Load Balancer is suitable when you need ultra-high performance, and have static IP addresses for your application.

### Cross-zone load balancing

By default, the Classic Load Balancer distributes traffic across all EC2 instances regardless of AZ. If you want to balance evenly across AZ, make sure you enable Cross-Zone load balancing:



### Make sure you remember to shut down your ELBs if you’re not using them

The major reason that people exceed the free tier is because they forgot to terminate their ELBs. It’s a good idea to tag your ELBs so that they can be tracked via resource groups.

ELBs initially have a DNS name, but no public IP address.

When an ELB is monitoring instances, the instance can have one of two status:

* In service
* Out of service

Set Evalate Target Health to true, and enable Latency Based Routing for HA (High Availability)

## **Auto scaling**

Scaling Policy is a set of rules i.e. Increase if average CPU > 80% for a consecutive peroid of 5 minutes.

**Desired Instances** is adjusted based on the scaling policy, and won’t go below the miniumum or above the maximum size of the group.

Note that **there is a default maximum of 20 running on-demand EC2 instances** regardless of the min/max you set in your ASG scaling policies. You can request a limit increase by getting in touch with AWS… if your auto scaling triggers are firing, but you are not getting any more instances, check that you haven’t reached the default maximum.

Deleting an ASG will automatically delete any instances that it created.

**Launch configurations cannot be modified after creation**. If you need to make a change, create a new launch configation and update your auto scaling group to use it.

Launch configurations can belong to multiple Auto Scaling groups, however you can only specify one launch configuration at a time for an Auto Scaling group.

AMIs can be used with Auto Scaling groups.

[More info on launch configurations](http://docs.aws.amazon.com/autoscaling/latest/userguide/LaunchConfiguration.html)

The following scale out options are available:

* Scheduled scaling - adjusting the size of a group at a specific time
* Dynamic scaling - via creating a scaling policy to automatically adjust the size of the group based on a specificed increase in demanc
* Manual scaling - via manually increasing the size of the group

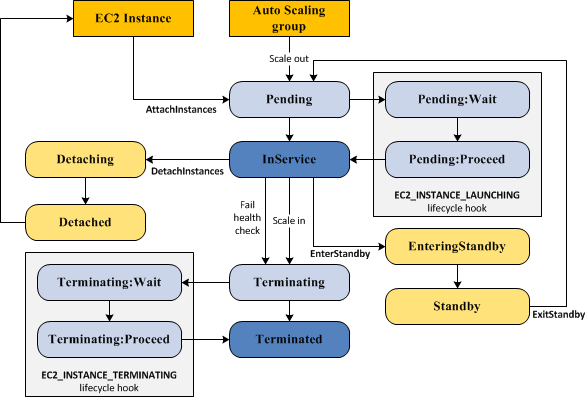
[More info on auto scaling lifecycles](http://docs.aws.amazon.com/autoscaling/latest/userguide/AutoScalingGroupLifecycle.html)

To attach EC2 instances to an Auto Scaling group, ensure that:

* The instance is in the running state
* The AMIs used to launch the instance still exist
* The instance is not a member of another Auto Scaling group
* The instance is in the same AZ as the Auto Scaling group

[More info on attaching instances to an Auto Scaling group](http://docs.aws.amazon.com/autoscaling/latest/userguide/attach-instance-asg.html)

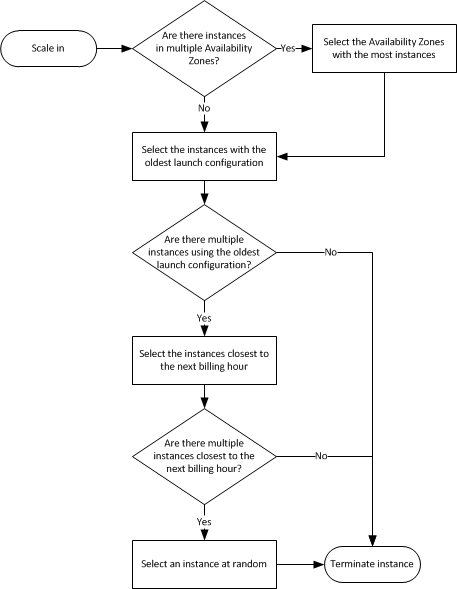
### Auto Scaling Lifecycle and Lifecycle Hooks



In the **Pending:Wait** state, no lifecycle policies take effect.

The cooldown period is the number of seconds after a scaling activity completes before another can start.

### Auto Scaling Group Termination Policy



## **Placement Groups**

Placement groups:

* Are a logical grouping of instances in a single AZ
* Cannot span multiple AZs
* Are good for grid computing or any time you need **high throughput and very low network latency**between your instances.
* Must have a unique name in your AWS account
* May only contain certain types of instances:
  + Compute optimised
  + GPU
  + Memory optimised
  + Storage optimised
* Work best when instance types within are homogenous - i.e. all compute optimised or all memory optimised
* Cannot be merged, however it is possible to move existing instances into placement groups

## **Elastic IP**

An elastic IP is a public IPv4 address which is reachable from the internet. If your instance does not have a public IPv4 address, youc an associate an Elastic IP address with the instance.

IPv6 is not currently supported for Elastic IP.

There is no charge for Elastic IP addresses assuming the following are true:

* The Elastic IP address is associated with an Amazon EC2 instance
* The instance associated with the Elastic IP address is running
* The instance has only one Elastic IP address attached to it.

Otherwise, there is an hourly charge.