EC2 Fundamentals

Setting up billing alert

If you would like your IAM user to also be able to set up billing alerts then you would have to enable that setting under **Account** as the root user.

EC2

Most popular AWS service. Elastic compute cloud, infrastructure as a service (renting out virtual servers as a service).

What can you do with EC2?

- You can have your own virtual machine, EC2 instances
- You can store data on the virtual drivers that's connected to EC2 instances, called Elastic block storage
- Distribute load across machines using Elastic load balancer
- Finally, scale the service using auto-scaling group (Create or terminate EC2 instances based on demand)

EC2 sizing & configuration options

- Operating system: Linux, Windows or Mac OS
- How much CPU you want
- How much RAM
- How much storage space:
  - Network-attached (EBS and EFS)
  - Hardware (EC2 instance store)
- Network card: How fast is the internet, control what is the public IP address
- Firewall rules: What traffic can go in and out
- Bootstrap script (run at first launch)/EC2 User Data

EC2 User Data

Bootstrapping means launching commands when a machine starts. You can use EC2 User Data to write some bash scripts that will be ran when the EC2 instance is **FIRST BOOTED ONLY.** On restart of the same instance it will not run again.

Use it to automate boot task like installing updates, software, downloading common files from the internet, or anything you can think of.

It will run as the root user! So keep that in mind.

EC2 status

You can stop an instance to stop it from running. AWS will not be charging you if it is Stopped. If you stopped the instance EBS storage will be kept, meaning data on disk is kept intact until next start.

You can also terminate the instance and delete it from existence. Which will also delete the storage if you configured it so.

You can start an instance after it is stopped, OS boots and EC2 user data script is run.

Everytime you stop and start up an EC2 instance it will be given another public ipv4 address! Private ipv4 will always be kept the same.

EC2 instance types

There are variety of EC2 types that are optimized for certain type of work/different use cases. AWS also has a naming convention for the EC2 instance that they have.

m5.2xlarge

m: Tells the instance class
5: Tells the generation of the instance class (It is improved over time)
2xlarge: Tells the size of the instance class (how much cpu, memory, networking capability they have)

General purpose: Great for diversity of workloads like web servers or code repositories.

They balance between compute power, memory, and networking. t2.micro is a General purpose EC2 instance.

Compute optimized: Optimized for compute-intensive tasks that require high performance processors.

Great for batch processing workloads, media transcoding, high performance web servers, high performance computing, dedicated gaming servers.

Memory optimized: Fast performance for workloads that process large data sets in memory

If you use it for high performance, relational/non-relational databases, distributed web scale cache stores, applications performing real-time processing of big unstructured data.

Storage optimized: Great for storage-intensive tasks that require high, sequential read and wrtie access to large data sets on local storage

Use it for high frequency online transaction processing systems, relational and NoSQL databases, cache for in-memory databases, distribute file systems

Security groups

They are firewall on EC2 instance. They let you control what kind of traffic is allowed in or out of EC2 instances.

One EC2 instance can have multiple security groups, they aren't limited to only one! The rules will just add on to each other.

They regulate access to ports, authorized IP range, control inbound network and outbound network.

For example: You can specify for TCP over port 443 to allow it or disallow it for said EC2 instances.

Security groups only contain allow rules, and you can reference by IP or other by security group (reference each other).

Additional information

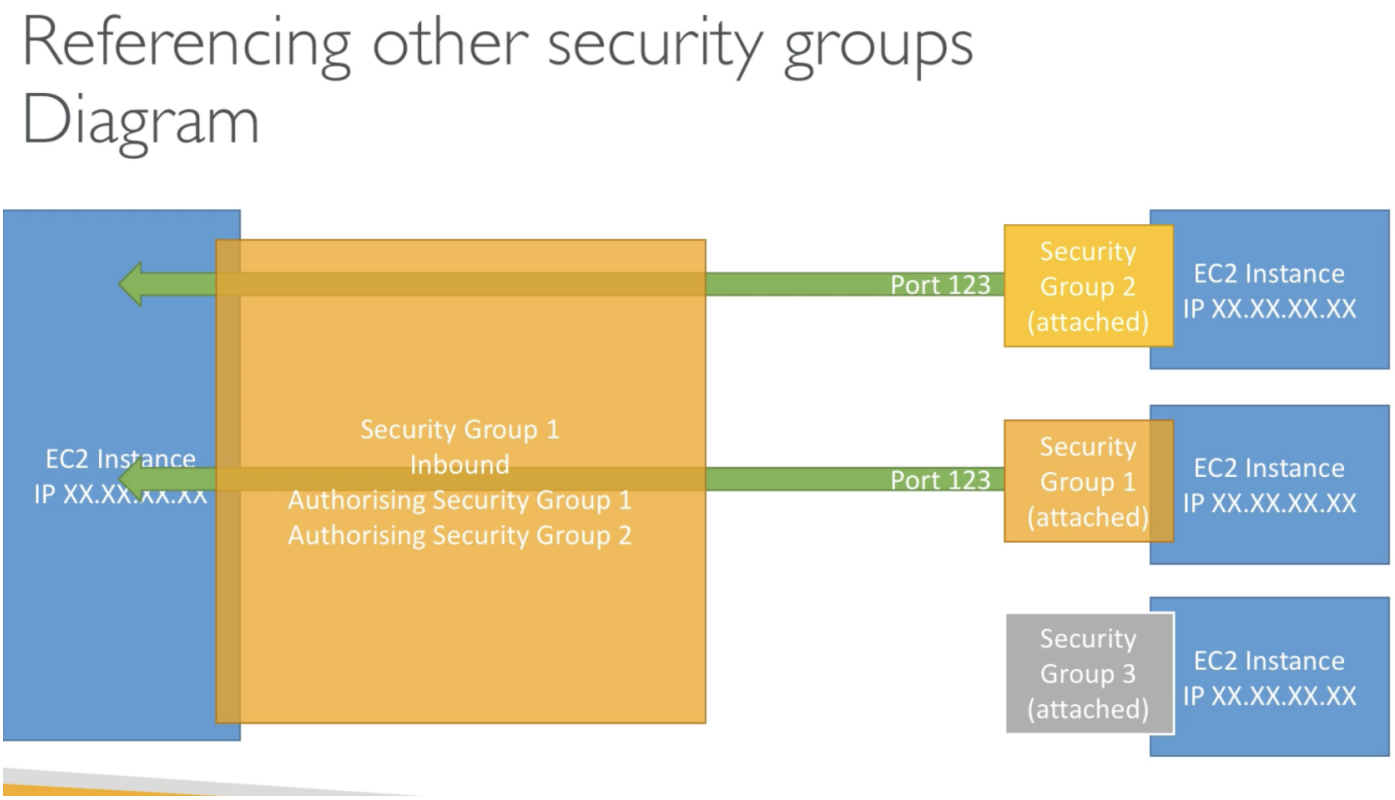
You can attach a security group to multiple instances, and they are locked down to a region / VPC combination. Meaning if you switch to another region or set up another VPC, then you will have to reconfigure the security group as they are not carried over.

Good to maintain one separate security group for SSH access.

Referencing other security groups

You can set up security groups to reference other security groups, what does it mean? If you have an EC2 instance you can set up its security group such that it authorize Security Group 1 and Security Group 2. So if there are other EC2 instance with either Security Group 1 or Security Group 2 attached to it, their traffic to the current EC2 instance will not be restricted, they can communicate directly with the current EC2 without having to set an explicit IP address or inbound/outbound rule.

If you have another EC2 instance say with Security Group 3 attached to it, then it will not be able to send any traffic to the current EC2 instance since it is not an authorized security group.



Classic port to know

port 22 is for SSH, let you log into EC2 instance

port 21 is for FTP, upload files into a file share

port 22 is also for SFTP, upload file using SSH

port 80 for HTTP, access unsecured websites

port 443 for HTTPS, access secured websites

port 3389 for RDP (Remote desktop protocol), log into a Window instances

SSH EC2 instance

SSH allow you to remotely log into a machine and interact with the machine using command line. The default user that EC2 instance created for us is `ec2-user`.

The EC2 instance doesn't use password for login, only private key are allowed to establish the ssh connection. To use the `.pem` private key file you would do something like so: `ssh ec2-user@<ipv4 address> -i key.pem`

The -i option uses the private key file for logging into the EC2 instance.

More on SSH HERE

After SSHing into EC2

If you are going to use `aws` command-line tool which is installed by default, DO NOT upload your AWS credentials by doing `aws configure`. This is because once you upload it, everybody who has access to the EC2 instance can inspect your AWS credentials! They can see your access key as well as your secret access key! So do not upload any AWS credentials into your EC2 instances.

The way that you should be doing is to use IAM roles! You attach IAM role to EC2 instances or any compatible resources to give it permissions to access certain AWS CLI commands.

To give an EC2 IAM role, you would do **Action -> Security -> Modify IAM role**, then you can give it the IAM role you want to give. For example, if you are letting EC2 instance to be able to read all of the IAM users, then you give it the IAM role which contain the read only access to IAM!

Now you can run `aws iam list-users` on EC2 instances without providing any credentials because it assumed the IAM role, giving it temporary credentials to be able to carry out that command!

EC2 purchasing options

EC2 On Demand

This is pay for what you use model. Linux or Window is billing per second, after the first minute. All other operating system is billing per hour.

This has the highest cost but it has no upfront payment and there is no long-term commitment

Recommended for short-term and uninterrupted workloads, where you can't predict how the application will behave.

EC2 Reserved Instances

Get lots of discount compared to on demand. You will be reserving a specific instance attribute (consist of the instance type, in which region/availability zone you are reserving it, tenancy are you going to be sharing it with other customer, OS) over a long period of time.

Reservation can be done for 1 year or 3 years, with 3 year offering the most discount.

You can pay no upfront, partial upfront, all upfront, of course paying all upfront netting you the most discount.

Recommended for applications like databases.

You can also buy or sell it in the reserved instance marketplace if you do not need the EC2 instance after but still have the reservation.

Convertible reserved instance

Another type of reserved instance that allows you to change the EC2 instance type, instance family, OS, scope and tenancy.

EC2 Savings Plans

Get a discount based on long-term usage. You will be committed to a certain type of usage ($10/hour for 1 or 3 years). Any usage above the EC2 saving plan will be billed on-demand.

Saving plan you will be locked to a specific instance family and AWS region like M5 EC2 family in us-east-1. But you do get to switch between instance size m5.xlarge to m5.2xlarge, and the OS you can freely change as well as the tenancy.

EC2 Spot Instance

Give you the most discount compared to on-demand. These are instances that you can lose at any point of time if the max price you are willing to pay is less than the current spot price. Like an auction, if you can pay the highest bid price, then you get to use it, otherwise, you lose it.

This is the most cost-efficient instance in AWS.

Recommended for workloads that are resilient to failure. Batch jobs, data analysis, image processing, any distributed workload, workloads with a flexible start and end time.

NOT RECOMMENDED FOR CRITICAL JOBS OR HOSTING DATABASE.

EC2 Dedicated Host

A physical server with EC2 instance capacity fully dedicated to your use, this is the most expensive option.

Recommended for compliance requirements and use your existing server-bound software licenses. Gives you compliance because you have access to the actual lower level hardware server themselves, so companies can have better control to be compliant to government laws.

This is basically giving you your own server.

For dedicated host, you can buy it on-demand or can also do reserved.

EC2 Dedicated Instance

Instances run on hardware that's dedicated to you, but you may share hardware with other instances in the SAME account.

You have no control over where is instance placed.

Characteristic	Dedicated Instances	Dedicated Hosts
Enables the use of dedicated physical servers	X	X
Per instance billing (subject to a $2 per region fee)	X	
Per host billing		X
Visibility of sockets, cores, host ID		X
Affinity between a host and instance		X
Targeted instance placement		X
Automatic instance placement	X	X
Add capacity using an allocation request		X

With both you get a dedicated server to host your EC2 instances. however, with dedicated host the server will be the same, since it is literally renting you the server, meanwhile dedicated instances the instance might be deployed to another dedicated server, it doesn't have to be the same one.

Dedicated host you are paying per host, while dedicated instance you are still paying per instances.

EC2 Capacity Reservations

You can reserve on-demand instance capacity in a specific availability zone for any duration.

You are guaranteed that those instances will be available to you when you need it.

There is no time commitment meaning you can create/cancel the reservation anytime, which means there is no billing discount.

You have to combine it with regional reserved instance and saving plan to actually get billing discounts.

While the reservation is in effect, you will be paying the on-demand price whether you run instances or not.

Recommended for short-term, uninterrupted workloads that needs to be in a specific availability zone.

More on spot instances

You define a max spot price that you are willing to pay, and as long as the current spot price is less than your maximum spot price then you will be keeping that spot price.

The current spot price will change hourly based on offer and capacity, and if the current spot price is greater than what you are willing to pay you can choose to stop or terminate your instance with a 2 minutes grace period. Stopping will allow you to resume the instance with its state after you have get the spot again, and terminate will allow you to start off fresh with a new instance once you regain the spot.

The other strategy is spot block, you can block a spot instance for 1-6 hour without interruptions. It won't be claimed if you block it, but it is no longer supported.

How to terminate spot instance

You will be first create request (maximum price you willing to pay, desired number of instances, launch specification, whether it is a one-time or persistent, and specify the date range for this request).

If it is one-time, as soon as you get a spot instance then the request is fulfilled and will go away.

If it is persistent, after you get a spot instance your request will still stay, and if your spot instance gets stopped or interrupted then the spot request will still remain until it can claim another spot instance. The spot request will be persistent for the specified range.

You can only cancel spot instance request that are **open, active, or disabled**.

Cancelling spot request will not terminate any spot instances! You must first cancel spot request then you terminate the associated spot instances.

Spot fleet

A way to get a set of spot instances + optional on-demand instances

The spot fleet will try to meet the target capacity with price constraints.

How it works is that you will define a set of pools, a pool consist of # of instance, the type, the OS, and the AZ. You define multiple of the pools so that the fleet can choose depending on the strategy.

Then spot fleet will stop launching the instances when it reaches the max # you defined for that pool or reached max cost.

Strategy that the spot fleet will use is:

- lowestPrice: You pick the pool with the lowest price (cost optimized, for short workload)
- diversified: You launch instances from all the pools you have defined (good for availability, long workloads)
- capacityOptimized: You pick the pool with the largest number of instances

Ultimately, spot fleet let you automatically request for spot instances with lowest price after you define the pool. Since it will be picking the instances with the lowest price.

Revision #13
Created 2023-02-08 19:15:52 UTC by Tamarine
Updated 2023-07-27 01:55:04 UTC by Tamarine