Table Of Contents

EC2

- It mainly consists of the following capabilities:
  - Renting virtual machines in the cloud (EC2)
  - Storing data on virtual drives (EBS)
  - Distributing load across multiple machines (ELB)
  - Scaling the services using an auto-scaling group (ASG)

Introduction to Security Groups (SG)

- Security Groups are the fundamental of networking security in AWS
- They control how traffic is allowed into or out of EC2 machines
- Basically they are firewalls

Security Groups Deep Dive

- Security groups regulate:
  - Access to ports
  - Authorized IP ranges - IPv4 and IPv6
  - Control of inbound and outbound network traffic
- Security groups can be attached to multiple instances
- They are locked down to a region/VPC combination
- They do live outside of the EC2 instances - if traffic is blocked the EC2 instance wont be able to see it
- *It is good to maintain one separate security group for SSH access*
- If the request for the application times out, it is most likely a security group issue
- If for the request the response is a “connection refused” error, then it means that it is an application error and the traffic went through the security group
- By default all inbound traffic is **blocked** and all outbound traffic is **authorized**
- A security group can allow traffic from another security group. A security group can reference another security group, meaning that it is no need to reference the IP of the instance to which the security group is attached

Elastic IP

- When an EC2 instance is stopped and restarted, it may change its public IP address
- In case there is a need for a fixed IP for the instance, Elastic IP is the solution
- An Elastic IP is a public IP the user owns as long as the IP is not deleted by the owner
- With Elastic IP address, we can mask the failure of an instance by rapidly remapping the address to another instance
- AWS provides a limited number of 5 Elastic IPs (soft limit)
- Overall it is recommended to avoid using Elastic IP, because:
  - They often reflect pool arhcitectural decisions
  - Instead, us e a random public IP and register a DNS name to it

EC2 User Data

- It is possible to bootstrap (run commands for setup) an EC2 instance using EC2 User data script
- The user data script is only run once at the first start of the instance
- EC2 user data is used to automate boot tasks such as:
  - Installing update
  - Installing software
  - Downloading common files from the internet

  - Any other start-up task
- THe EC2 user data scripts run with root user privileges

EC2 Instance Launch Types

- On Demand Instances: short workload, predictable pricing
- Reserved: known amount of time (minimum 1 year). Types of reserved instances:
  - Reserved Instances: recommended long workloads
  - Convertible Reserved Instances: recommended for long workloads with flexible instance types
  - Scheduled Reserved Instances: instances reserved for a longer period used at a certain schedule
- Spot Instances: for short workloads, they are cheap, but there is a risk of losing the instance while running
- Dedicated Instances: no other customer will share the underlying hardware
- Dedicated Hosts: book an entire physical server, can control the placement of the instance

EC2 On Demand

- Pay for what we use, billing is done per second after the first minute
- Hast the higher cost but it does not require upfront payment
- Recommended for short-term and uninterrupted workloads, when we can't predict how the application will behave

EC2 Reserved Instances

- Up to 75% discount compared to On-demand
- Pay upfront for a given time, implies long term commitment
- Reserved period can be 1 or 3 years
- We can reserve a specific instance type
- Recommended for steady state usage applications (example: database)
- **Convertible Reserved Instances**:
  - The instance type can be changed
  - Up to 54% discount
- **Scheduled Reserved Instances**:
  - The instance can be launched within a time window
  - It is recommended when is required for an instance to run at certain times of the day/week/month

EC2 Spot Instances

- We can get up to 90% discount compared to on-demand instances
- It is recommended for workloads which are resilient to failure since the instance can be stopped by the AWS if our max price is less then the current spot price
- Not recommended for critical jobs or databases
- Great combination: reserved instances for baseline performance + on-demand and spot instances for peak times

EC2 Dedicated Hosts

- Physical dedicated EC2 server
- Provides full control of the EC2 instance placement
- It provides visibility to the underlying sockets/physical cores of the hardware
- It requires a 3 year period reservation
- Useful for software that have complicated licensing models or for companies that have strong regulatory compliance needs

EC2 Dedicated Instances

- Instances running on hardware that is dedicated to a single account
- Instances may share hardware with other instances from the same account
- No control over instance placement
- Gives per instance billing

EC2 Spot Instances - Deep Dive

- With a spot instance we can get a discount up to 90%
- We define a max spot price and get the instance if the current spot price < max spot price
- The hourly spot price varies based on offer and capacity
- If the current spot price goes over the selected max spot price we can choose to stop or terminate the instance within the next 2 minutes
- Spot Block: block a spot instance during a specified time frame (1 to 6 hours) without interruptions. In rare situations an instance may be reclaimed
- Spot request - with a spot request we define:
  - Maximum price
  - Desired number of instances
  - Launch specifications
  - Request type:
    - One time request: as soon as the spot request is fulfilled the instances will be launched an the request will go away
    - Persistence request: we want the desired number of instances to be valid as long as the spot request is active. In case the spot instances are reclaimed, the spot request will try to restart the instances as soon as the price goes down
- Cancel a spot instance: we can cancel spot instance requests if it is in open, active or disabled state (not failed, canceled, closed)
- Canceling a spot request does not terminate the launched instances. If we want to terminate a spot instance for good, first we have to cancel the spot request and the we can terminate the associated instances, otherwise the spot request may relaunch them

Spot Fleet

- Spot Fleet is a set of spot instances and optional on-demand instances
- The spot fleet will try to meet the target capacity with price constraints
- AWS will launch instances from a launch pool, meaning we have to define the instance type, OS, AZ for a launch pool
- We can have multiple launch pools from within the best one is chosen
- If a spot a fleet reaches capacity or max cost, no more new instances are launched
- Strategies to allocate spot instances in a spot fleet:
  - **lowerPrice**: the instances will be launched from the pool with the lowest price
  - **diversified**: launched instances will be distributed from all the defined pools
  - **capacityOptimized**: launch with the optimal capacity based on the number of instances

EC2 Instance Types

- R: applications that needs a lot of RAM - in-memory cache
- C: applications that need good CPU - compute/database
- M: applications that are balanced - general / web app
- I: applications that need good local I/O - databases
- G: applications that need GPU - video rendering / ML