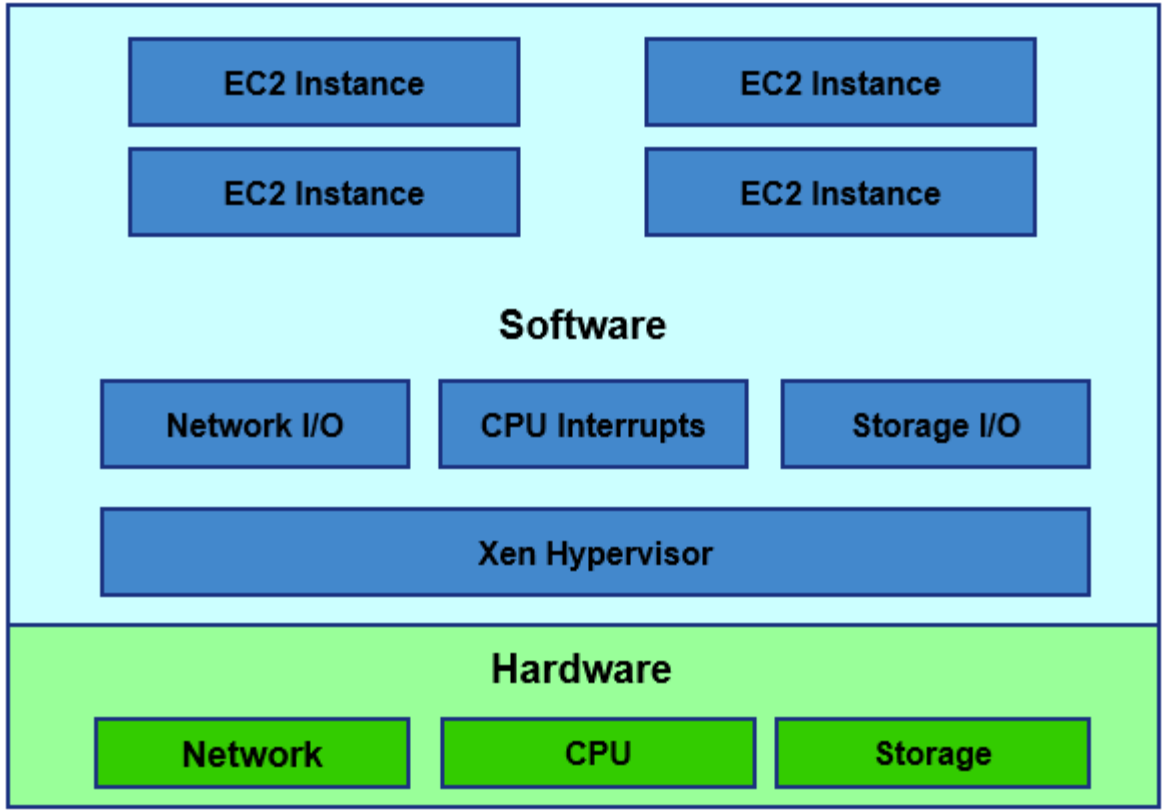
AWS

- What is EC2
- EC2 – Instance Access
- EC2 –Demo (Windows, Linux)
- How to SSH into your EC2 Instance, Security Groups, Private vs Public IP vs EIP
- AMI - Amazon Machine Image
- Instance Lifecycle
- Tag Amazon Resources
- EC2 – Limits, Root/Boot Volume
- EC2 – Charges
- Instance Families
- Monitoring
- Stopping an EC2 instance
- Instance Termination
- Termination Protection
- IAM Roles
- Bastion Hosts
- Instance Purchasing Options
- Instance User Data
- Enhanced networking on Linux
- Placement groups

What is EC2

- Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides secure, resizable compute capacity in the AWS cloud
- Virtual server in Amazon Web services terminology
- Virtual Machine



EC2 – Instance Access

- To access an instance you need a **key and key pair name**
  - When you launch a new EC2 instance, you can create a public/private key pair
  - **You can download the private key only once**

    Save it in a safe place so you won't loose it
  - The **public key is saved by AWS** to match it to the key pair name, and private key when you try to login to the EC2 instance
  - If you launch your instance without a key pair, you will not be able to access it (via RDP or SSH)

EC2 instance Demo....

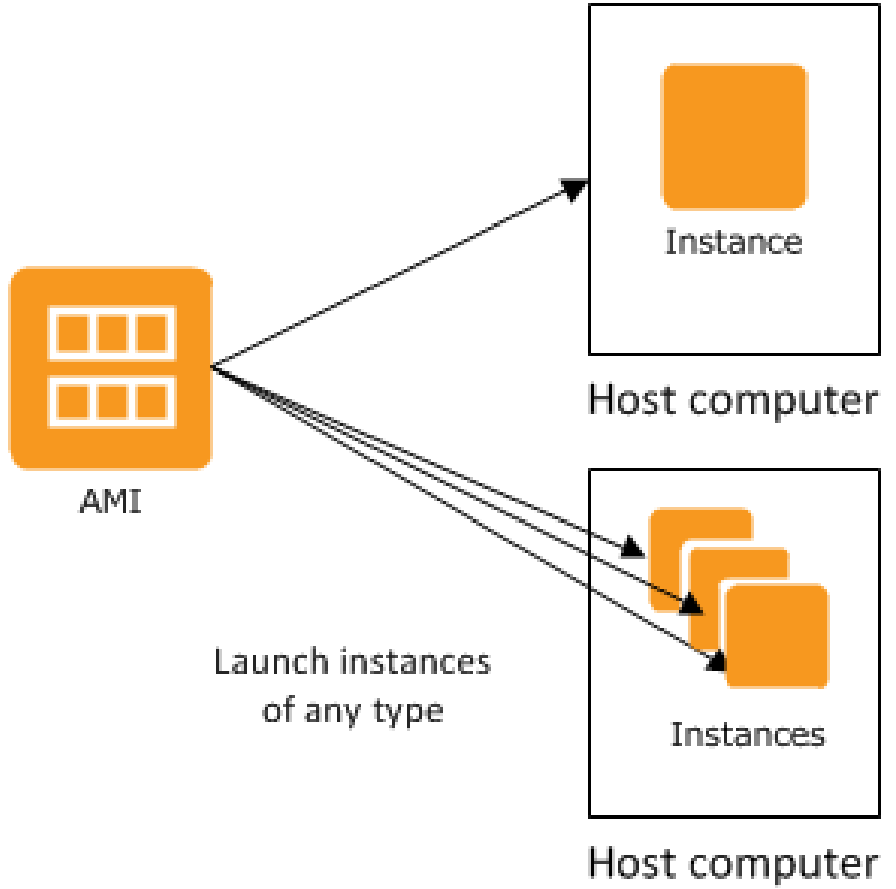
- AMI, EC2 Instance Families, Instance Access (public key/private key)
- Root/Boot Volume, Provide Public IP
- Stopping an EC2 instance
- Instance Termination, termination Protection
- IAM Roles
- Instance Purchasing Options (OnDemand)
- Instance User Data
- Monitoring

SSH into your EC2 Instance, Security Groups, Private vs Public IP vs EIP

- SSH into your EC2 Instance
- Download the private key
- Create Security Groups
- Creation of Public IP, private IP
- Creation of Elastic IP

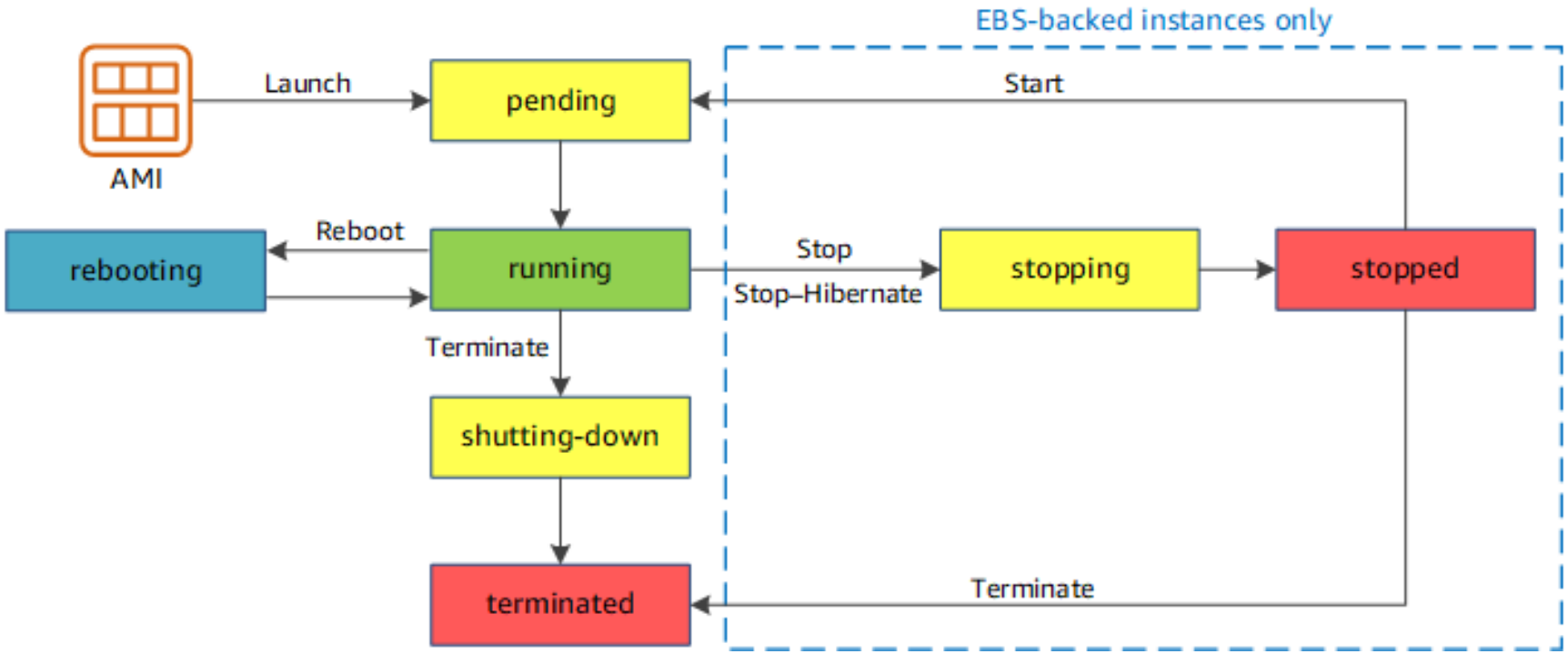
AMI - Amazon Machine Image

- An AMI is a template that contains a software configuration (for example, an operating system, an application server, and applications). From an AMI, you launch an instance, which is a copy of the AMI running as a virtual server in the cloud.
- Different types of instances can be launched from a single **AMI**. An ***instance type*** essentially determines **the hardware of the host computer** used for your instance.
- Each instance type offers different compute and memory capabilities.



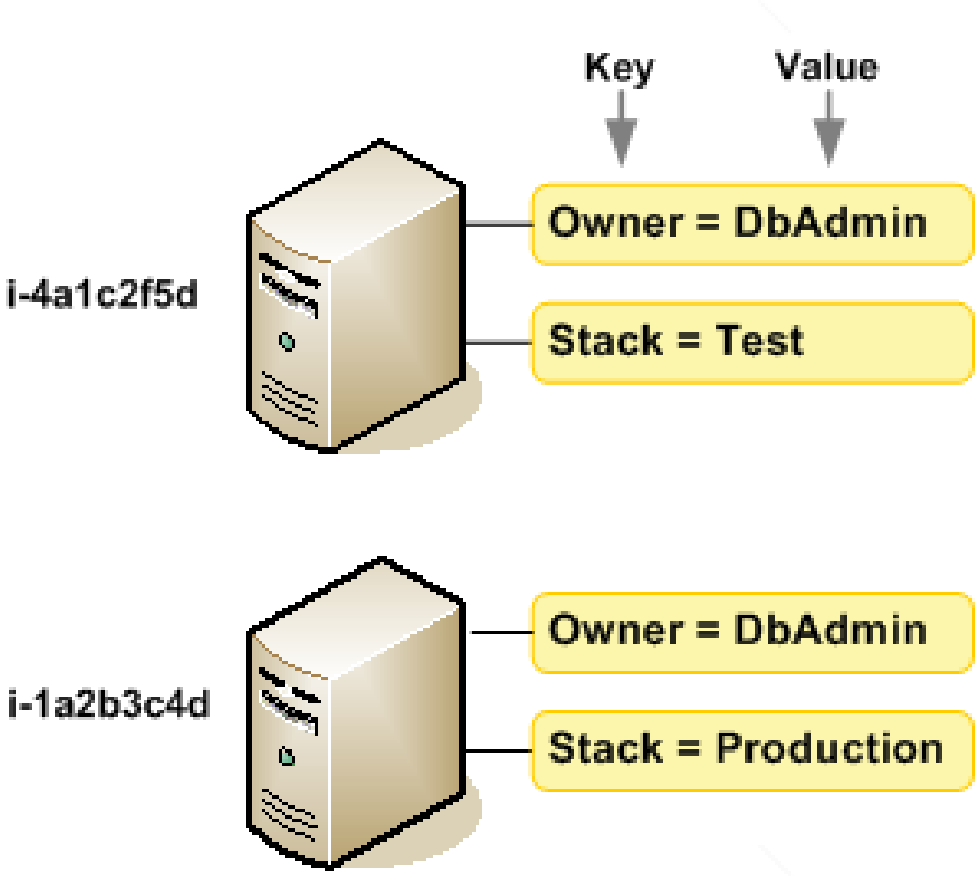
Instance Lifecycle

- An Amazon EC2 instance transitions through different states
- Differences between
  - Reboot
  - Stop
  - Terminate



Tag Amazon Resources

- A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value
- Tags enable you to categorize your AWS resources in different ways, for example, by purpose, owner, or environment



EC2 – Limits, Root/Boot Volume

- There is **20 EC2 instances soft limit per account.** You can submit a request to AWS to increase it
- Two types of Block store devices are supported:
  - Elastic Block Store (EBS)
    - Persistent
    - Network attached virtual drives
  - Instance-store
    - Basically, the virtual hard drive on the host allocated to this EC2 instance
- EC2 instance root/boot volumes can be EBS or Instance Store volumes
- EBS-Backed EC2 instance
  - It has an EBS root volume
- Instance-store backed EC2 instance
  - It has an Instance-store root volume

EC2 Instance Families

- Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instance types comprise varying combinations of CPU, memory, storage, and networking.
- General Purpose
  - Balanced memory and CPU
  - Suitable for most applications
  - Ex. M3, M4, T2
- Compute Optimized
  - More CPU than memory
  - Compute & HPC intensive use
  - Ex. C2, C4

- Memory Optimized
  - More RAM/memory
  - Memory intensive apps, DB, and caching
  - Ex. R3, R4
- GPU compute instances
  - Graphics Optimized
  - High performance and parallel computing
  - Ex. G2
- Storage Optimized
  - Very high, low latency, I/O
  - I/O intensive apps, data warehousing, Hadoop
  - Ex. I2, D2

https://aws.amazon.com/ec2/instance-types/

EC2 - Charges

- You are charged on EC2 service in hourly based or per second based pricing
- A reboot of an EC2 instance is considered as the instance is still running
- If the instance is stopped, you are not charged if it remains stopped
- You are also charged for data transfer in/out of EC2 instance (if sent to outside the AWS region)
- EBS Storage charges

EC2 Monitoring

- EC2 service can send its metric data to AWS CloudWatch **every 5 minutes** (enabled by default)
  - This is **free** of charge
  - It is called basic monitoring
- You can choose to enable detailed monitoring where the EC2 service will send its metric data to AWS CloudWatch **every 1 minute**
  - **Chargeable**
  - It is called detailed monitoring
- You can set CloudWatch alarm actions on EC2 instance(s) to :
  - Stop, Restart, Terminate, or Recover your EC2 instance
    - You can use Stop or Terminate actions to save cost
    - You can use the reboot and recover to move your **EC2 instance to another host**

Stopping an EC2 instance

- When you stop an **EBS backed instance**, any data in any Instance-store volumes is lost
  - Even though the instance can be re-started, all instance store data will be gone
- When you stop an EBS-Backed EC2 instance
  - Instance performs a shutdown
  - State changes from **running** -> Stopping -> Stopped
  - EBS volumes remain attached to the instance
  - Any data cached in RAM or Instance Store volumes is lost
  - Most probably, when restarted again, it will restart on a **new physical host**
  - Instance retains **its private IPv4 address**, any IPv6 address
  - **Instance releases its public IPv4 address back to AWS pool**
  - Instance retains its Elastic IP address

    You will start to be **charged** for un-used Elastic IP

EC2 – Instance Termination

- By default, EBS root device volumes (created automatically when the instance is launched) **are deleted automatically when the EC2 instance is terminated**
- Any additional (non boot/root) volumes attached to the instance (those you attach to the instance during launch or later), by default, persist after the instance is terminated
- You can modify both behaviors by modifying the "**DeleteOnTermination**" attribute of any EBS volume during instance launch or while running

EC2 Termination Protection

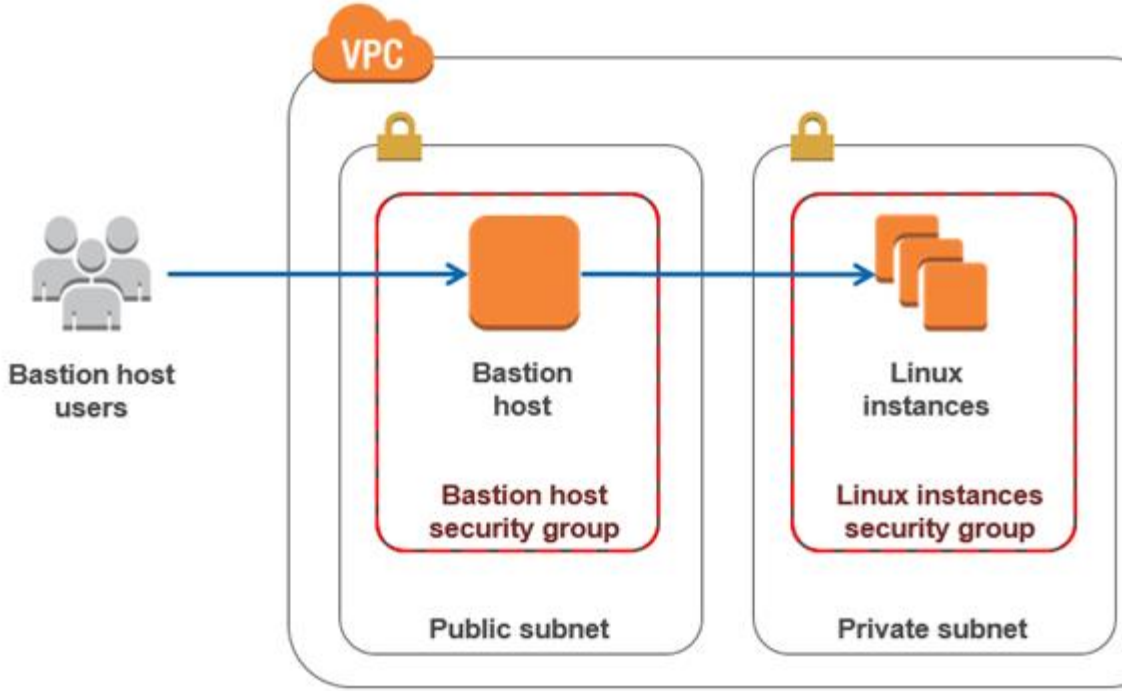
- This is a feature you can enable such that an EC2 instance is protected against **accidental termination** through API, Console, or CLI
- This can be enabled for Instance-store backed and EBS-Backed Instances
- CloudWatch can ONLY terminate EC2 instances if they do not have the termination protection enabled

EC2 – IAM Roles

- In General, for an AWS services to have permission to read or write to another service, an IAM role is required to be attached to the first AWS Service with rights/permissions on the second AWS service
- Drawing on that, for an EC2 instance to have access to other AWS services (example S3) you need to configure an IAM Role, which will have an IAM policy attached, under the EC2 instance.
    - Applications on the EC2 instance will get this role permission from the EC2 instance's metadata
- You can add an IAM to an EC2 instance during or after it is launched

EC2 – Bastion Hosts

- For inbound, secure, connectivity to your VPC to manage and administer public and/or private EC2 instances, you can use a bastion host or a jump box.
  - The Bastion host is an EC2 instance, whose interfaces will have a security group allowing
    - **inbound SSH access for Linux EC2 instances**
    - **inbound RDP access for windows instances**
  - Bastion hosts can have auto-assigned public IP addresses or Elastic IP addresses
  - Using Security groups you can further limit **which IP CIDRs can access** the Bastion Host.



EC2 – Instance Purchasing Options

Reserved Instances

– 1- or 3-years commitments, high savings, can be zonal (per AZ) or Regional scoped.

Scheduled instances

– Upfront purchase instance capacity for a recurring schedule

Spot Instances

– Request AWS unused EC2 instances, highest savings, availability not guaranteed when you need it

Dedicated hosts

– Pay for a fully dedicated physical host

Dedicated Instances

– Pay by the hour for instances that run on single-tenant hardware

On-Demand

– Pay by the second for instances that you launch

EC2 – Instance Meta Data

Instance Meta Data:

– This is instance data that you can use to configure or manage the instance

• Examples are IPv4 address, IPv6 address, DNS Hostnames, AMI-ID, Instance-ID,

Instance-Type, Local-hostname, Public Keys, Security groups…

– Meta data can be only viewed from within the instance itself

• i.e you have to logon to the instance

– Meta data is not protected by encryption (cryptography), anyone that has access to the instance can view this data

To view an EC2 Instance's Meta Data (from the EC2 instance console): GET http://169.254.169.254/latest/meta-data/ OR

curl http://169.254.169.254/latest/meta-data/

EC2 – Instance User Data

Instance user data:

– Is data supplied by the user at instance launch in the form of a script to be executed during the instance boot

– User data is limited to 16KB

– User data can only be viewed from within the instance itself (logon to it)

– You can change user data

To do so, you need to stop the instance first (EBS backed)

– Instance -> actions -> Instance-settings -> View/Change user data

– User data is not protected by encryption, do not include passwords or sensitive data in your user data (scripts)

– You are not charged for requests to read user data or metadata

Enhanced networking on Linux

- Enhanced networking uses **single root I/O virtualization** (SR-IOV) to provide high-performance networking capabilities on supported instance types.
- SR-IOV is a method of device virtualization that provides higher I/O performance and lower CPU utilization when compared to traditional virtualized network interfaces.
- Enhanced networking provides
  - higher bandwidth
  - higher packet per second (PPS) performance
  - consistently lower inter-instance latencies.
- There is **no additional charge** for using enhanced networking.

Placement groups

- When you launch a new EC2 instance, the EC2 service attempts to place the instance in such a way that all of your instances are spread out across underlying hardware to minimize correlated failures.
- You can use placement groups to influence the placement of a group of interdependent instances to meet the needs of your workload. Depending on the type of workload, you can create a placement group using one of the following placement strategies:
  - Cluster - Inside one AZ
  - Partition - spread logical partitions
  - Spread - distinct hardware