# DAY- 3
# AWS EC2

# AWS Architecture and Design

1. Day 1 Overview of Cloud Computing
2. Day 2 Overview of AWS
3. **Day 3 Amazon EC2***
4. Day 4 Amazon EBS *
5. Day 5 Amazon CloudWatch *
6. Day 6 Amazon S3*
7. Day 7 Amazon Elastic Load Balancer *
8. Day 8 Amazon Auto Scaling *
9. Day 9 Amazon VPC *
10. Day 10 Amazon IAM *
11. Day 11 Amazon RDS
12. Day 12 Amazon Route 53 *
13. Day 13 Amazon DynamoDB* & Glacier
14. Day 14 Amazon Cloudfront* & Import Export & Amazon SES *
15. Day 15 Amazon ElasticBeanStalk & Amazon Cloudformation & Amazon OpsWorks
16. Day 16 AWS Economics & AWS Account Overview *
17. Day 17 AWS Architecture
18. Day 18 AWS Certification Preparation

[With Hands on Demo]

# AWS Elastic Compute Cloud

# AWS EC2

- What is EC2?
- What is AMI and types of AMI
- EC2 Pricing
- EC2 Key Terminology
  - Security Groups
  - Key Pairs
  - Elastic Ips
- EC2 Pricing
- Demo

# Amazon EC2

→ AWS EC2 gives you virtual machines in the cloud. These machines are also known as server instances. They are:

→ Launched in Amazon's data centers using APIs or available tools and utilities

→ You are allowed to use server instances at any time (on-demand), for as long as you need, and for any legal purpose

→ Instances are available in different sizes and configurations(scalable)

→ This allows AWS to provide different instance types that you can use to meet specific needs (resource pooling)

→ What makes Amazon EC2 different is that you use only the capacity that you need(pay-as-you go)

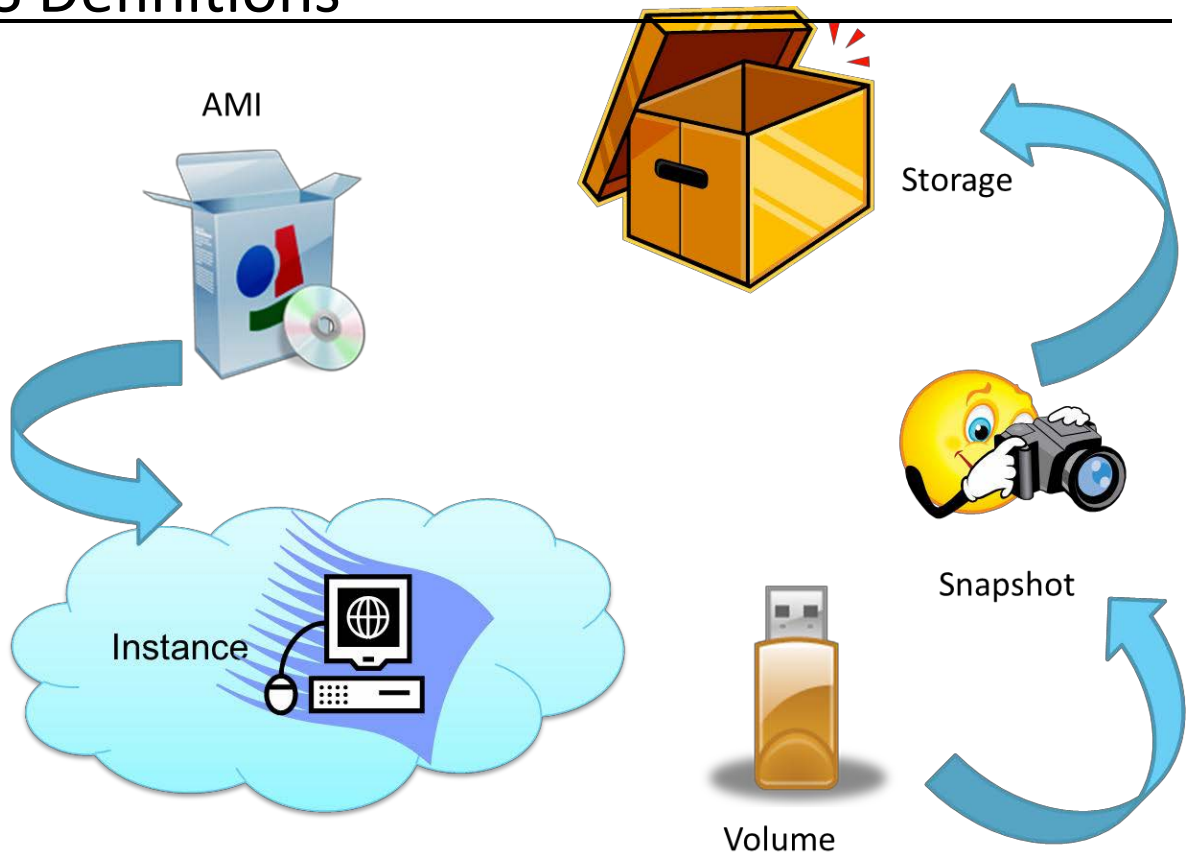
→ Saves your cost in compare to in house Data centres

http://aws.amazon.com/ec2/

# EC2 Highlights

| | | |
|---|---|---|
| Elastic | Flexible | Secure |
| Cheaper | Ready to use AMIs | Reliable |

# AWS Definitions



# AMI Types

An Amazon EC2 instance can be launched either from an AMI backed by Amazon S3 or an AMI backed by Amazon EBS.

## Instance launched from EBS-backed AMI

→ The default root device is an Amazon EBS volume

→ Amazon creates an Amazon EBS volume from the associated AMI snapshot as root volume

→ EBS root device helps persist any changes to the local storage, it also helps in surviving instance failures and restarts

→ Since the EBS volume is mounted as the root device, you will also be billed for storage of all static data such as operating systems files, etc., external to your application or configuration (but very negligible cost).

# AMI Types (Contd.)

## Instance launched from Instance Store backed AMIs:

→ When an instance is launched from this type of AMI, it will always have an ephermal storage attached as root device.

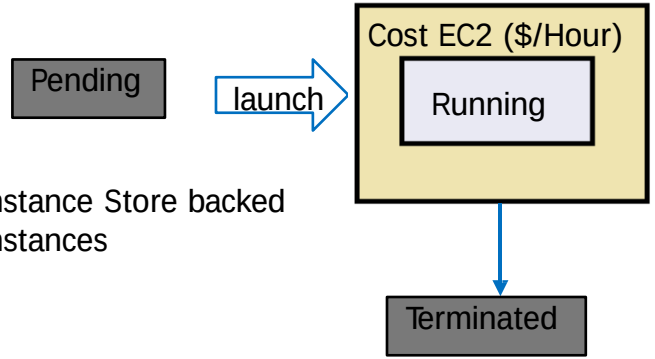
→ Since these instances use local storage as their root device the data is not persistent. If you want to make data persistent then you have to manually attach and mount an additional EBS volume for persisting your data.

→ At the same time it helps in save cost as you are charged only for additional EBS volume and not for ephermal storage.

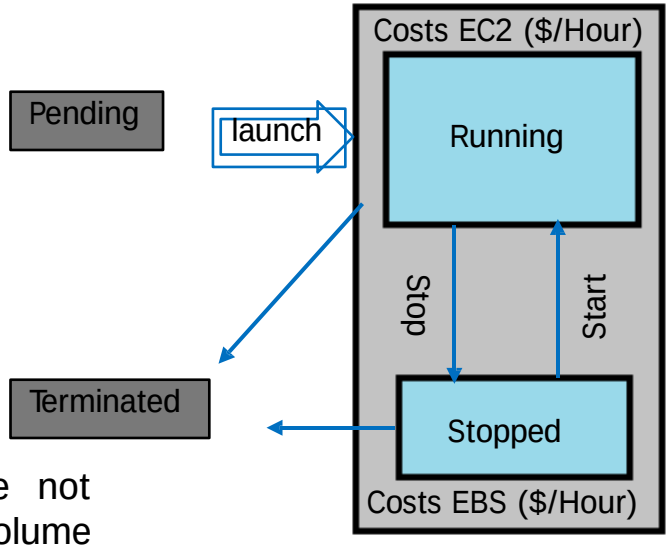
→ It also gives you direct control over what files to persist.

→ The biggest chalenge is once your instance is terminated, it also deletes the ephermal root device. So all your local instance storage will be lost. If you have additional EBS volume then your data and configuration will persist in that EBS volume.

# AMI Types (Contd.)




Instance Store backed Instances




EBS backed Instances

→When an Amazon EBS-backed instance is stopped, you're not charged for instance usage; however, you're still charged for volume storage.

→ AWS charges a full instance hour for every transition from a stopped state to a running state.

→ Even if you transition the instance multiple times within a single hour.

# AMI Types (Contd.)

| Characteristic | Amazon EBS | Instance (S3) Store |
|---|---|---|
| Lifecycle | You can start and Stop instance launched from EBS backed AMI. | These instances cannot be stopped. It will be always in either running or terminated state. |
| Data persistence | Data will persists in the EBS on instance failure or restart. Data can also be configured to persist when the instance is terminated, with deleteontermination flag. | Instance storage does not persist on instance shutdown or failure. It is possible to attach non-root devices using EBS for data persistence as needed. |
| Boot Time | Usually Less than 1 minute | Usually Less than 5 minutes. |
| Size Limit | 16 TB | Its Ephrmal Storage which can be in some TBs too. |
| Root Device | Amazon EBS volume | Instance storage (Ephermal Storage). |
| Upgrading | The kernel, instance type, and ramdisk can be changed while the instance is stopped. | Instance attributes are fixed for the life of an instance. |
| Charges | Instance usage, Amazon EBS volume usage, and AMI Storage. | Instance usage and AMI storage |
| AMI Creation | With one click or one single command | It's a long process and requires multi step process. |

# EC2 Instance Parameters

- When you are launching an instance you have to make decision for AMI & Instance

- Parameters for AMI Selection
  - Region
  - OS
  - Software
  - EBS Backed / Instance Store Backed

- Parameters for Instance Launch Criteria
  - Bit
  - Size (vCPU & Memory) & Family
  - IO Performance

# EC2 Instance Types & Size

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

# ECU vs vCPU

→ vCPUs are now the unit of measure on Amazon's EC2 instance description page.

→ A vCPU is a virtual Central Processing Unit (CPU). A multicore processor has two or more vCPUs

→ Each vCPU is seen as a single physical CPU core by the VM's operating system

→ vCPU is more standard measure of shared CPU power used in clouds such as VMware's vCloud

The amount of CPU that is allocated to a particular instance is expressed in terms of these EC2 Compute Units:

» ECU provides the relative measure of the integer processing power of an AWS EC2 instance

» It is simply a unit for CPU comparison between instances. It is not an actual CPU

» Two instances each with 2 vCPU will not necessarily be equivalent. One instance could reside on top of a 2012-based processor while the other could reside on top of a 2014-based processor

→ One EC2 Compute Unit provides the equivalent CPU capacity of a 1.0-1.2 GHz 2007 Opteron or 2007 Xeon processor

http://whatis.techtarget.com/definition/virtual-CPU-vCPU

# AWS Key Things

→ Key Pair

→ Security Group

→ Launch Instance

→ IP Address

# Amazon EC2 Key Pair

- → The public and private keys are known as a key pair
- → '.pem' stands for “Privacy Enhanced Mail Security Certificate”
- → AWS only stores the public key, and you store the private key
- → Anyone who possesses your private key can decrypt your login information
- → Amazon Key-Pair enables you to connect to instances through SSH
- → The key pair ensures that only you have access to the instance

# Security Group

→ A security group acts as a firewall that controls the traffic allowed to reach one or more instances

→ When you launch an instance, you assign it one or more security groups

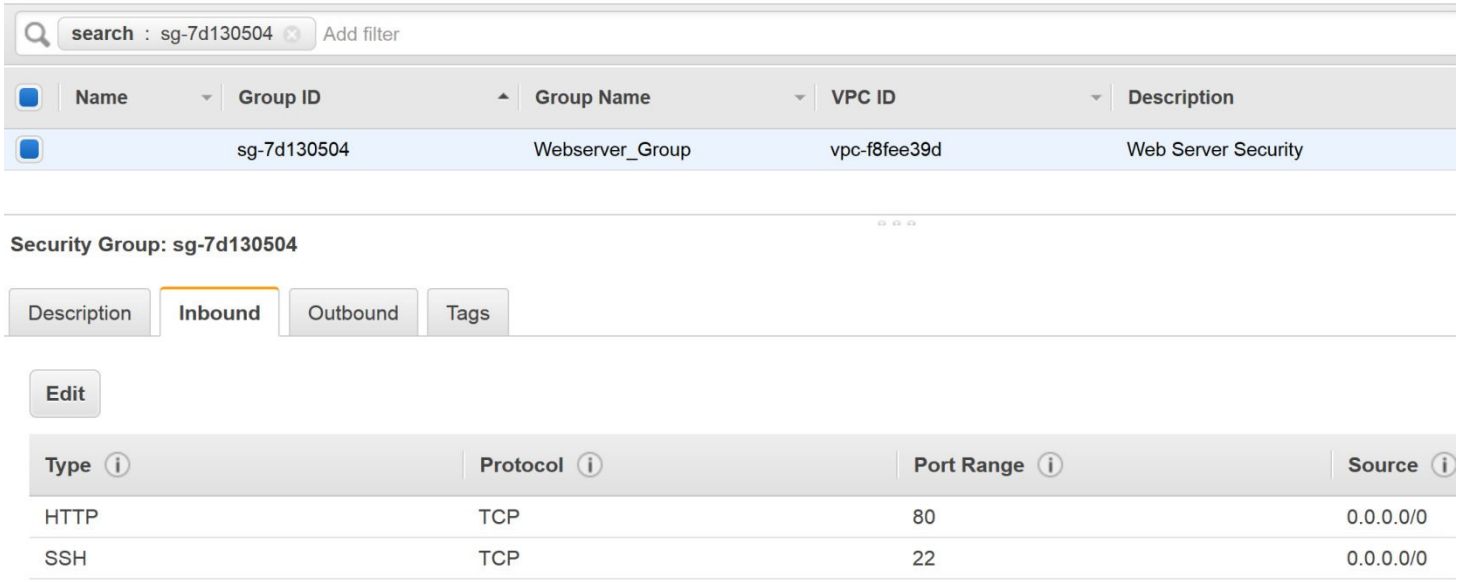
→ You can add rules to each security group that control traffic for the instance

→ The new rules are automatically applied to all instances to which the security group is assigned

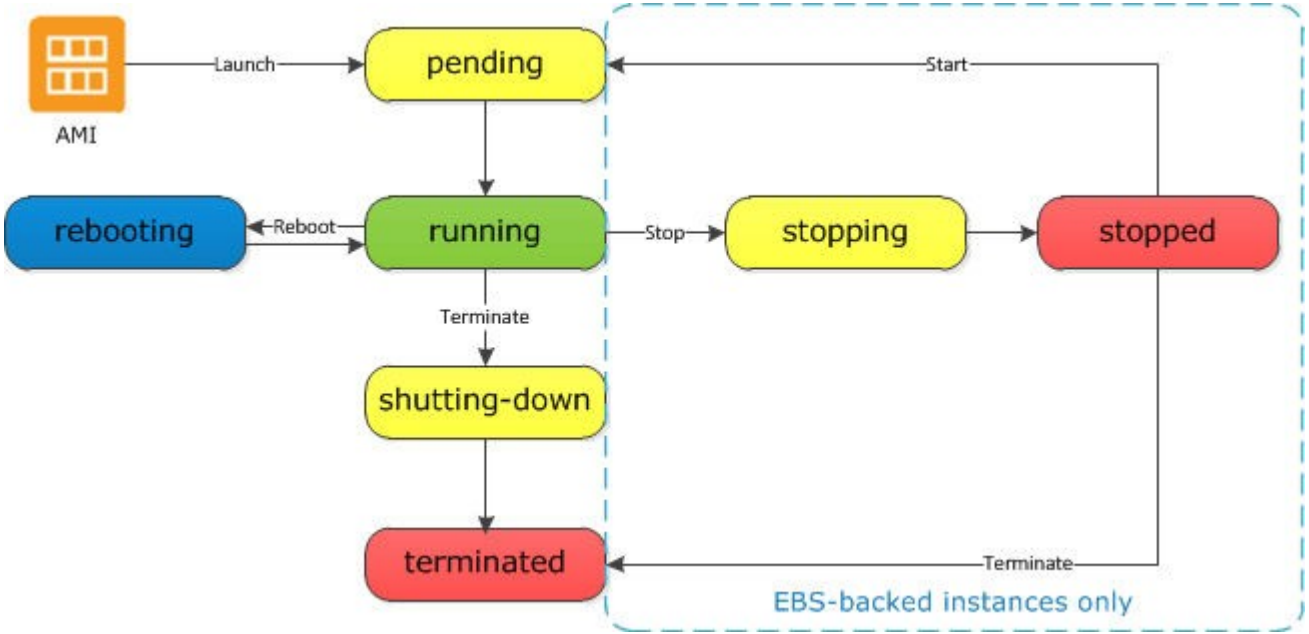
→ Control Inbound and Outbound traffic

→ The default Security Group does NOT allow any incoming traffic

# Security Group



# EC2 Instance Life Cycle



# EC2 Instance Addressing

Every EC2 instance is assigned:

→ Private IP address

→ Private DNS name (*.internal)

→ Public DNS name (*.amazonaws.com)

→ Public IP address

# EC2 Instance Addressing

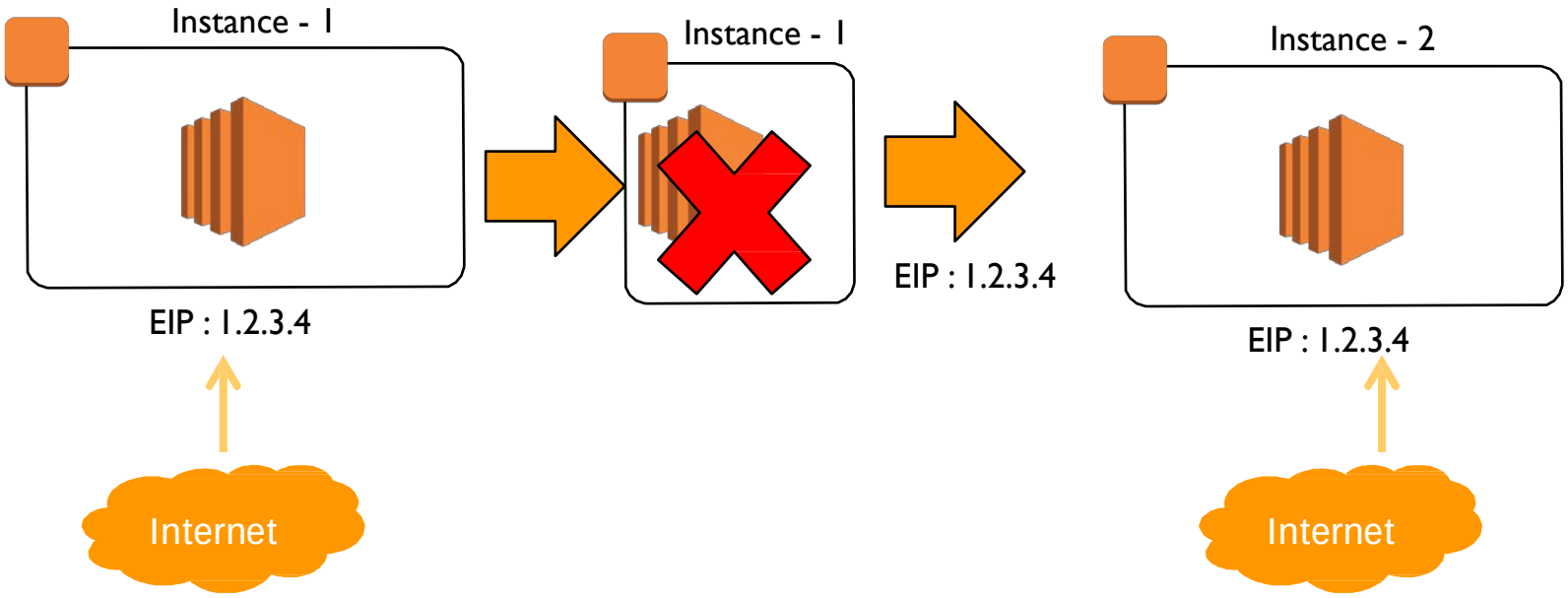
## Elastic IP Addresses:

→ Static IP belonging to your AWS account
→ It is not bound to a particular instance.
→ Available as long as you want
→ Achieve High Availability
→ By default 5 Elastic Ips per region

## IP Address Charging:

→ $0.005 per non-attached Elastic IP address per complete hour
→ $0.00 per Elastic IP address remap – first 100 remaps / month

# Elastic IP Addresses (Contd.)




http://docs.aws.amazon.com/AWSEC2/2011-05-15/UserGuide/using-instance-addressing.html

# Instances Pricing Models

# EC2 Instance Pricing Models

## Pay as You Go with On-demand Pricing:

→ Advantage of On-Demand

→ No long-term commitments

→ All Operation Expense and no Capital Expense

# EC2 Instance Pricing Models

## Reserved Instances Pricing:

→ Reserved Instances allows to reserve AWS capacity for a significantly discounted rate (upto 75%)

→ Three payment options like
  → All Upfront
  → Partial Upfront
  → No Upfront.

→ Will cost you even if you are not using

http://aws.amazon.com/ec2/purchasing-options/reserved-instances/

# Amazon EC2 Reserved Instances

→ Amazon Elastic Compute Cloud (Amazon EC2) Reserved Instances enables you to reserve capacity for your EC2 instances

→ Lowers your average instance cost

→ With Reserved Instances, you need to decide a few terms such as
  → Instance Size
  → Instance Region and Zone
  → Payment Term (1/3 Years)
  → Payment Option (All, Partial, No Upfront)

# Amazon EC2 Reserved Instances (Contd.)

| | | |
|---|---|---|
| Cheaper | Flexible | Reliable |
| Scalable | Market Place | Easy to User |

# EC2 Instance Types (Contd.)

## Spot Instances:

→ Spot Instances allow customers to bid on unused Amazon EC2 capacity and run those instances for as long
as their bid exceeds the current Spot Price

→ The Spot Price changes periodically based on supply and demand

→ Customers whose bids meet or exceed it, gain access to the available Spot Instances

→ If you have flexibility in when your applications can run, Spot Instances can significantly lower your Amazon EC2 costs

# Account Identifiers

## Account Identifiers

AWS assigns two unique IDs to each AWS account:

### 1. AWS Account ID

- A unique ID of which identifies your AWS account
- You can share AWS resources with other AWS account IDs

### 2. Canonical User ID

- Used with AWS S3
- an Amazon S3 bucket policy for cross-account access

# AWS Free Usage Tier

http://aws.amazon.com/free/

# Demo

In the next video we will do hands on with AWS EC2

# Thank You

Email us – support@intellipaat.com

Visit us - https://intellipaat.com