

Amazon Compute

By Tamal Dey
CA, PESU

What is EC2?

- Elastic Compute Cloud – EC2
- It is web service that provide secure , resizable compute capacity in the cloud
- It is designed to make web-scale cloud computing easier for developers
- EC2 is a simple web service interface makes it easier to provision and configure capacity by using APIs
- You will have the complete control on the computing resources and they will be running on amazons infrastructure.

- EC2 reduces the time required to obtain and boot new server instances in minutes.
- You can scale up and down capacity as your requirements change
- Pay-as-you-go
- It provides the users to build the failure-resilient applications and isolate them from **common failure** scenarios :
 - "Beginner mistakes" on the part of service providers.
 - Security flaws that hackers eventually expose.
 - Poor processes within the cloud.

EC2 instance

- Several properties has to be configured in the EC2 instance
 - AMI – Amazon Machine Image
 - The instance type/hardware profile – General purpose, compute optimised, memory optimized, accelerated compute, storage optimized
 - Security groups
 - Storage
 - Key pairs

Quiz

- The application will run on Linux. What service will you use to create your first server instance in AWS to run Linux?
 - Amazon S3
 - Amazon Cognito
 - Amazon EC2

ANS: Amazon EC2

So where is my code really running?

- EC2 instance is a VM that gives us resources such as VCPU and RAM
- This VM runs on a physical server within the AWS facility
- In AWS, we call this as the host machine
- A hypervisor mediates access between your code and the underlying server and provides isolation from other workloads that may be running on that machine



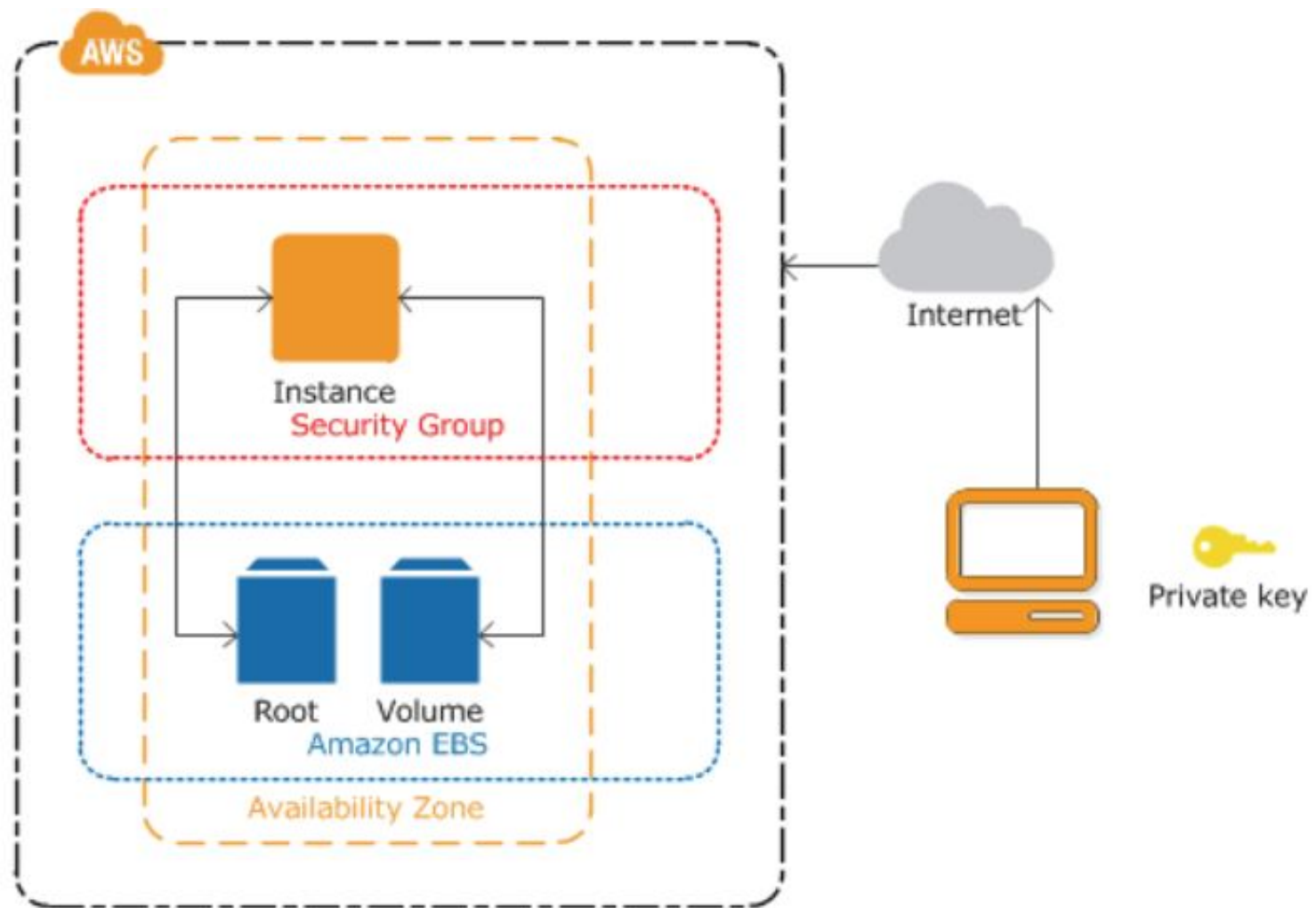
- Region is a specific geographic area
- Region Identifiers ends in **numbers**
- Each **region** has multiple, isolated locations known as **Availability Zones (AZ)**
- AZ identifiers ends in alphabets
- **The host is physically located in an Availability Zone (AZ)**
- Some services are region specific and some are AZ specific
- EC2 is AZ specific i.e, **Resources aren't replicated across regions unless you do so specifically.**

Amazon Elastic Compute Cloud (EC2)

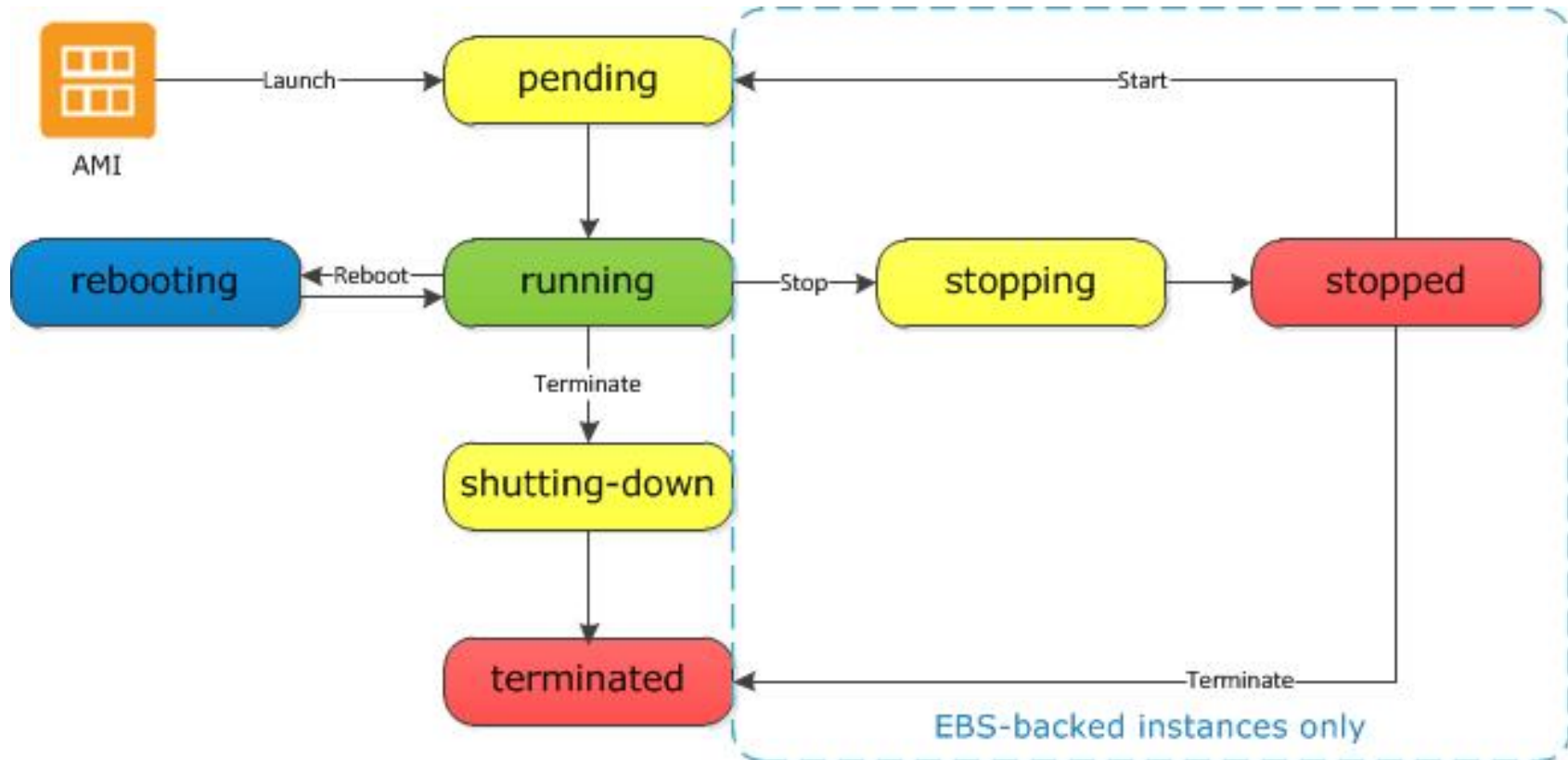
- Amazon Machine Images (**AMIs**) are the basic building blocks of Amazon EC2
- An AMI is a template that contains a software configuration (operating system, application server and applications) that can run on Amazon's computing environment
- AMIs can be used to launch an **instance**, which is a copy of the AMI running as a virtual server in the cloud.

Getting Started with Amazon EC2

- Step 1: Sign up for Amazon EC2
- Step 2: Create a key pair
- Step 3: Launch an Amazon EC2 instance
- Step 4: Connect to the instance
- Step 5: Customize the instance
- Step 6: Terminate instance and delete the volume created



Instance Lifecycle



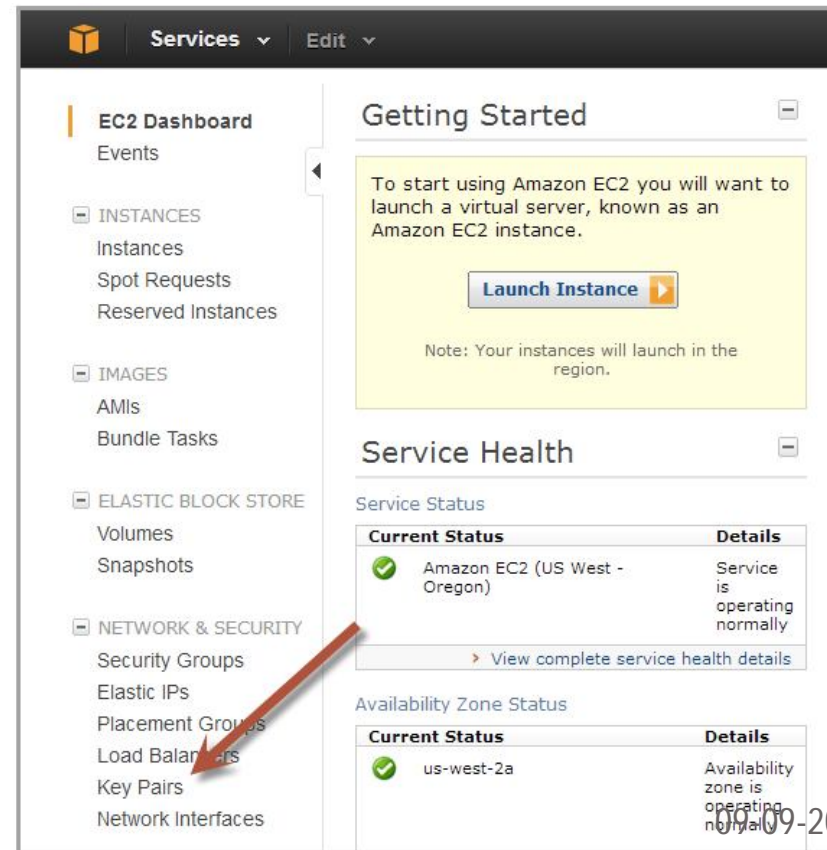
- Instance Usage Billing Info.
- <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-instance-lifecycle.html>

Creating a key pair

- AWS uses **public-key cryptography** to encrypt and decrypt login information.
- AWS **only stores the public key**, and the **user stores the private key**.
- There are **two options** for creating a key pair:
 - Have Amazon EC2 generate it for you
 - Generate it yourself using a third-party tool such as OpenSSH, then import the public key to Amazon EC2

Generating a key pair with Amazon EC2

1. Open the Amazon EC2 console at <http://console.aws.amazon.com/ec2/>
2. On the navigation bar select region for the key pair
3. Click **Key Pairs** in the navigation pane to display the list of key pairs associated with the account



Generating a key pair with EC2 (cont.)

4. Click **Create Key Pair**
5. Enter a name for the key pair in the **Key Pair Name** field of the dialog box and click **Create**
6. The private key file, with .pem extension, will automatically be downloaded by the browser.

Launching an Amazon EC2 instance

1. Sign in to AWS Management Console and open the Amazon EC2 console at <http://console.aws.amazon.com/ec2/>
2. From the navigation bar select the region for the instance



3. From the Amazon EC2 console dashboard, click **Launch Instance**

Create a New Instance

Cancel

Select an option below:

☒ **Classic Wizard**

Launch an On-Demand or Spot instance using the classic wizard with fine-grained control over how it is launched.

☒ **Quick Launch Wizard**

Launch an On-Demand instance using an editable, default configuration so that you can get started in the cloud as quickly as possible.

☐ **AWS Marketplace**

AWS Marketplace is an online store where you can find and buy software that runs on AWS. Launch with 1-Click and pay by the hour.

[Submit Feedback](#) [Getting Started Guide](#)

Name Your Instance: Pick a meaningful name, e.g. Web Server.

Choose a Key Pair:

Public/private key pairs allow you to securely connect to your instance after it launches.

☐ Select Existing ☒ Create New ☐ None

Name:

Please note that you need to download the key pair before you can continue.

Choose a Launch Configuration:

More Amazon Machine Images NEW!

Search through public and AWS Marketplace AMIs or choose from your own custom AMIs.

Amazon Linux AMI 2012.03	The Amazon Linux AMI 2012.03 is an EBS-backed, PV-GRUB image. It includes Linux 3.2, AWS tools, and repository access to multiple versions of MySQL, PostgreSQL, Python, Ruby, and Tomcat.	64 bit <input checked="" type="radio"/> 32 bit <input type="radio"/>
Red Hat Enterprise Linux 6.3	Red Hat Enterprise Linux version 6.3, EBS-boot.	64 bit <input checked="" type="radio"/> 32 bit <input type="radio"/>
SUSE Linux Enterprise Server 11	SUSE Linux Enterprise Server 11 Service Pack 2 basic install, EBS boot with Amazon EC2 AMI Tools preinstalled; Apache 2.2, MySQL 5.0, PHP 5.3, and Ruby 1.8.7	64 bit <input checked="" type="radio"/> 32 bit <input type="radio"/>
Ubuntu Server 12.04 LTS	Ubuntu Server 12.04 LTS with support available from Canonical (http://www.ubuntu.com/cloud/services).	64 bit <input checked="" type="radio"/> 32 bit <input type="radio"/>

Note: You can customize your settings in the next step.

4. On the **Create a New Instance** page, click **Quick Launch Wizard**
5. In **Name Your Instance**, enter a name for the instance
6. In **Choose a Key Pair**, choose an existing key pair, or create a new one
7. In **Choose a Launch Configuration**, a list of basic machine configurations are displayed, from which an instance can be launched
8. Click continue to view and customize the settings for the instance

9. Select a security group for the instance. A **Security Group** defines the firewall rules specifying the incoming network traffic delivered to the instance. Security groups can be defined on the Amazon EC2 console, in **Security Groups** under **Network and Security**

 **Security Group:** quicklaunch-1

Details **Inbound** Outbound

Create a new rule: Custom TCP rule

Port range:
(e.g., 80 or 49152-65535)

Source:
(e.g., 192.168.2.0/24, sg-47ad482e, or 1234567890/default)


 Add Rule

Apply Rule Changes

TCP Port (Service)	Source	Action
22 (SSH)	0.0.0.0/0	Delete

10. Review settings and click **Launch** to launch the instance
11. Close the confirmation page to return to EC2 console
12. Click **Instances** in the navigation pane to view the status of the instance. The status is **pending** while the instance is launching

After the instance is launched, its status changes to **running**

	Name	Instance	AMI ID	Root Device	Type	State	Public DNS
	GSG Tutorial	 i-e1ab569a	ami-aecd60c7	ebs	t1.micro	 pending	

	Name	Instance	AMI ID	Root Device	Type	State	Public DNS
	GSG Tutorial	 i-e1ab569a	ami-aecd60c7	ebs	t1.micro	 running	ec2-50-19-54-72.compute-1.amazonaws.com

Task to perform

1. Connect AmazonEC2 Linux instance from Windows machine
 - via (putty.exe and puttygen.exe)
2. Connect AmazonEC2 Linux instance from Linux/Ubuntu machine
 - via (ssh command)
3. Connect UbuntuEC2 instance from Linux/Ubuntu machine via
 - (ssh command)
4. Connect WindowsServerEC2 instance from windows machine via
 - (Remote Desktop connection- mstsc)
5. Connect WindowsServerEC2 instance from Linux machine via
 - (_____)
6. Copy a file from local windows machine to Linux Server
 - using WinSCP
7. Copy a file from local Linux machine to Linux Server
 - using SCP
8. Copy a file from local windows machine to Windows Server
 - using WinSCP

Connect to Your Linux Instance

Your local computer	Available connection methods
Linux or macOS X	SSH client EC2 Instance Connect
Windows	PuTTY SSH client

Connect to Your Windows Instance

Your local computer	Available connection methods
Linux or macOS X	SSH client EC2 Instance Connect
Windows	winSCP

Connecting to an Amazon EC2 instance

- There are several ways to connect to an EC2 instance once it's launched.
- **Remote Desktop Connection** is the standard way to connect to Windows instances. (⌨ + R)- Type mstsc
- An **SSH client** (standalone or web-based) is used to connect to Linux instances.

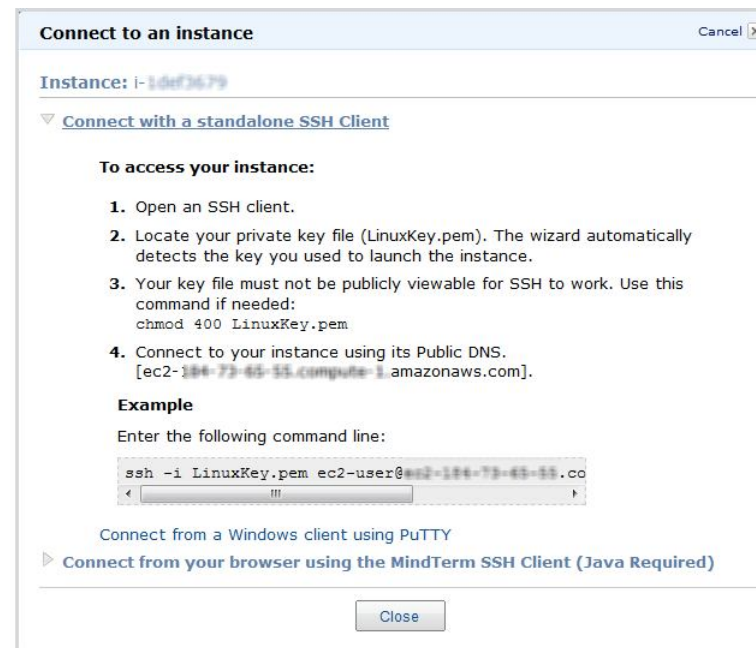
Connecting to Linux/UNIX Instances from Linux/UNIX with SSH

Prerequisites:

- Most Linux/UNIX computers include an SSH client by default, if not it can be downloaded from openssh.org
- Enable SSH traffic on the instance (using security groups)
- Get the path the private key used when launching the instance
 1. In a command line shell, change directory to the path of the private key file
 2. Use the **chmod** command to make sure the private key file isn't publicly viewable

```
chmod 400 My_Keypair.pem
```

3. Right click on the instance to connect to on the AWS console, and click **Connect**.
4. Click **Connect using a standalone SSH client**.
5. Enter the example command provided in the Amazon EC2 console at the command line shell



Mr. Tamal Dey

Transferring files to Linux/UNIX instances from Linux/UNIX with SCP

Prerequisites:

- Enable SSH traffic on the instance
- Install an SCP client (included by default mostly)
- Get the ID of the Amazon EC2 instance, public DNS of the instance, and the path to the private key

If the key file is `My_Keypair.pem`, the file to transfer is `samplefile.txt`, and the instance's DNS name is `ec2-184-72-204-112.compute-1.amazonaws.com`, the command below copies the file to the `ec2-user` home

```
scp -i My_Keypair.pem samplefile.txt ec2-user@ec2-184-72-204-112.compute-1.amazonaws.com:~
```

Terminating Instances

- If the instance launched is not in the free usage tier, as soon as the instance starts to boot, the user is billed for each hour the instance keeps running.
- A terminated instance cannot be restarted.
- To terminate an instance:
 1. Open the Amazon EC2 console
 2. In the navigation pane, click **Instances**
 3. Right-click the instance, then click **Terminate**
 4. Click **Yes, Terminate** when prompted for confirmation

Creating a Windows instance

- Sign in to the AWS Management Console and open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
- In the navigation pane, under **Instances**, choose **Instances**.
- Browse to and choose your Windows Server instance in the list.
- Choose **Connect**.
- Choose **Get Password**.
- Choose **Browse**. Browse to and choose the Amazon EC2 instance key pair file associated with the Windows Server Amazon EC2 instance, and then choose **Open**.
- Choose **Decrypt Password**. Make a note of the password that is displayed. You need it in step 10.
- Choose **Download Remote Desktop File**, and then open the file.

- If you are prompted to connect even though the publisher of the remote connection can't be identified, proceed.
- Type the password you noted in step 7, and then proceed. (If your (Remote Desktop Protocol) **RDP** connection client application prompts you for a user name, type **Administrator**.)
- If you are prompted to connect even though the identity of the remote computer cannot be verified, proceed.
- After you are connected, the desktop of the Amazon EC2 instance running Windows Server is displayed.
- You can now sign out of the running Amazon EC2 instance.

Converting the Ubuntu instance into web server

- Connect to the ubuntu instance
- `sudo apt-get install apache2`
- Copy the public DNS and paste it in the new browser window
- You will get the apache homepage. Now the ec2 instance acts as a web server.

Reading resources

- <https://www.youtube.com/watch?v=IZMkgOMYYIg>