UNIT-2

Mr. Tamal Dey
Dept. of MCA,PESU

Index

- Setting up AWS account
- Amazon Compute (EC2-Ubuntu and Windows)
- Amazon Storage (S3)
- Networking in AWS (VPC & NAT)
- Amazon Database (RDS)
- Management & Governance(Cloud Watch)

Setting up AWS account

Mr. Tamal Dey
Dept. of MCA,PESU

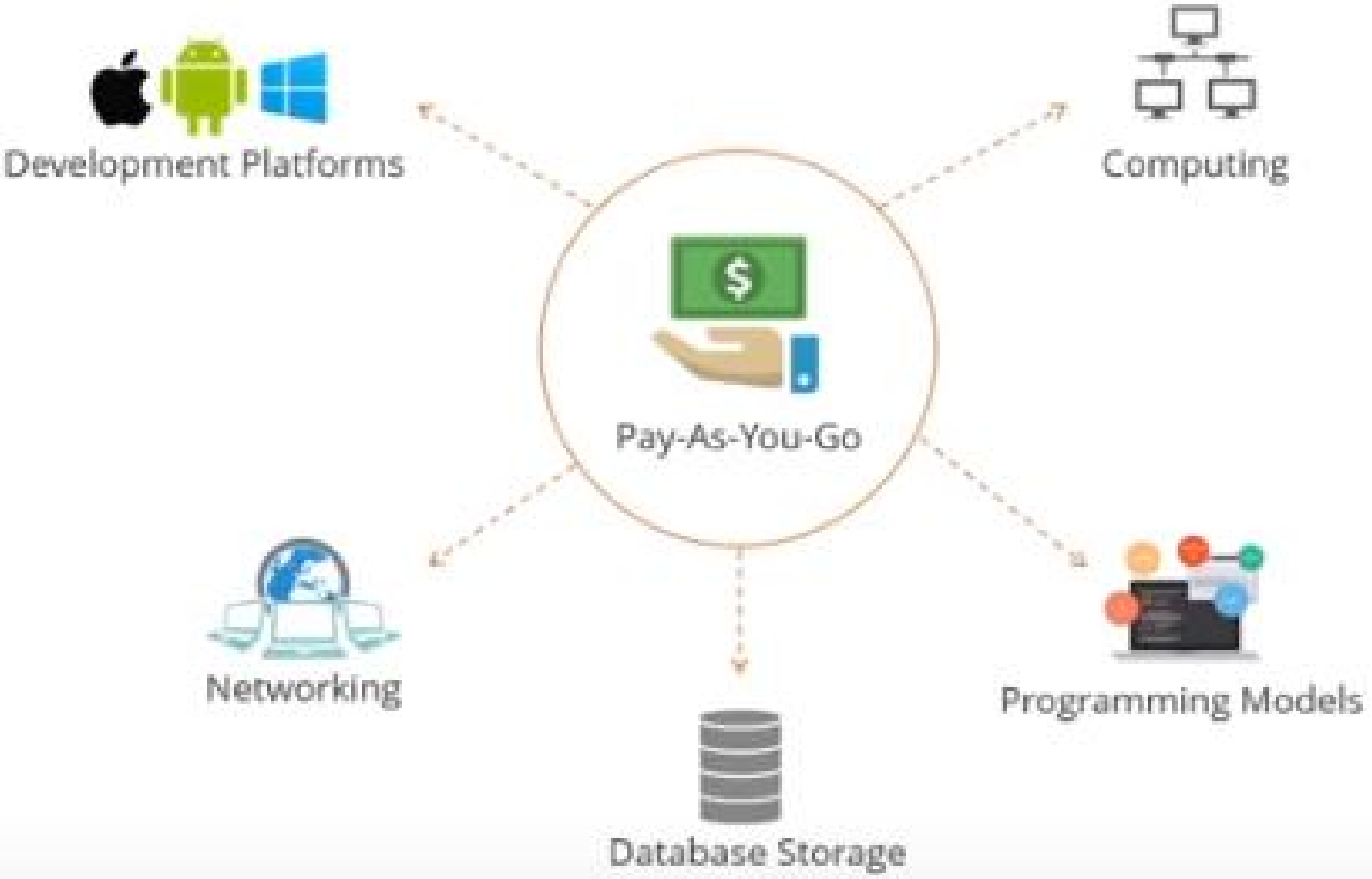
What is AWS

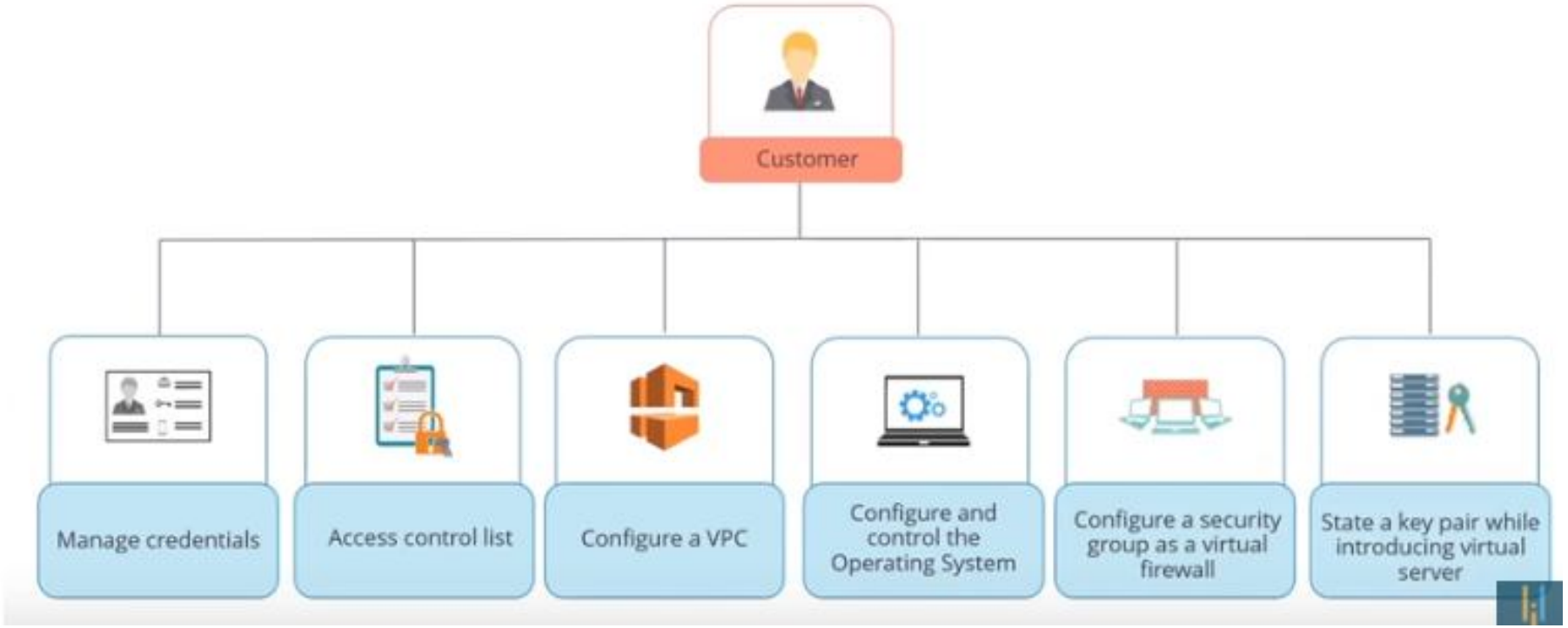
- It is global cloud platform which allows to host applications and services on the internet.
- Used by around 80% of fortune 500 companies to host the infrastructure
- They provide the IaaS, PaaS (java,php,ruby), SaaS (email)
- Hosting provider

Why it is such a hit?

- The billing - Per hour billing, Micro billing
- Easy signup process
- Simple billing dashboard
- Services are stable
- Trusted vendor

Why AWS?


Development Platforms
Computing
Pay-As-You-Go
Networking
Programming Models
Database Storage


Customer
Manage credentials
Access control list
Configure a VPC
Configure and control the Operating System
Configure a security group as a virtual firewall
State a key pair while introducing virtual server

Service overview

- EC2 – bare service, a machine can be launched
- VPC- virtual private cloud
- S3- simple storage service – upload and share files
- RDS- run and manage databases on the cloud – Mysql, Oracle, Postgresql
- Route 53 – DNS service
- Auto scaling – capacity to scale on the fly
- ELB- Elastic load balancing – scale up in multiple traffic

How much it costs

- Per hour billing for almost everything
- Region specific pricing –Virginia-cheapest
- Term specific pricing – year pricing is cheap (has discounts)
- Spot resources

How big is it?

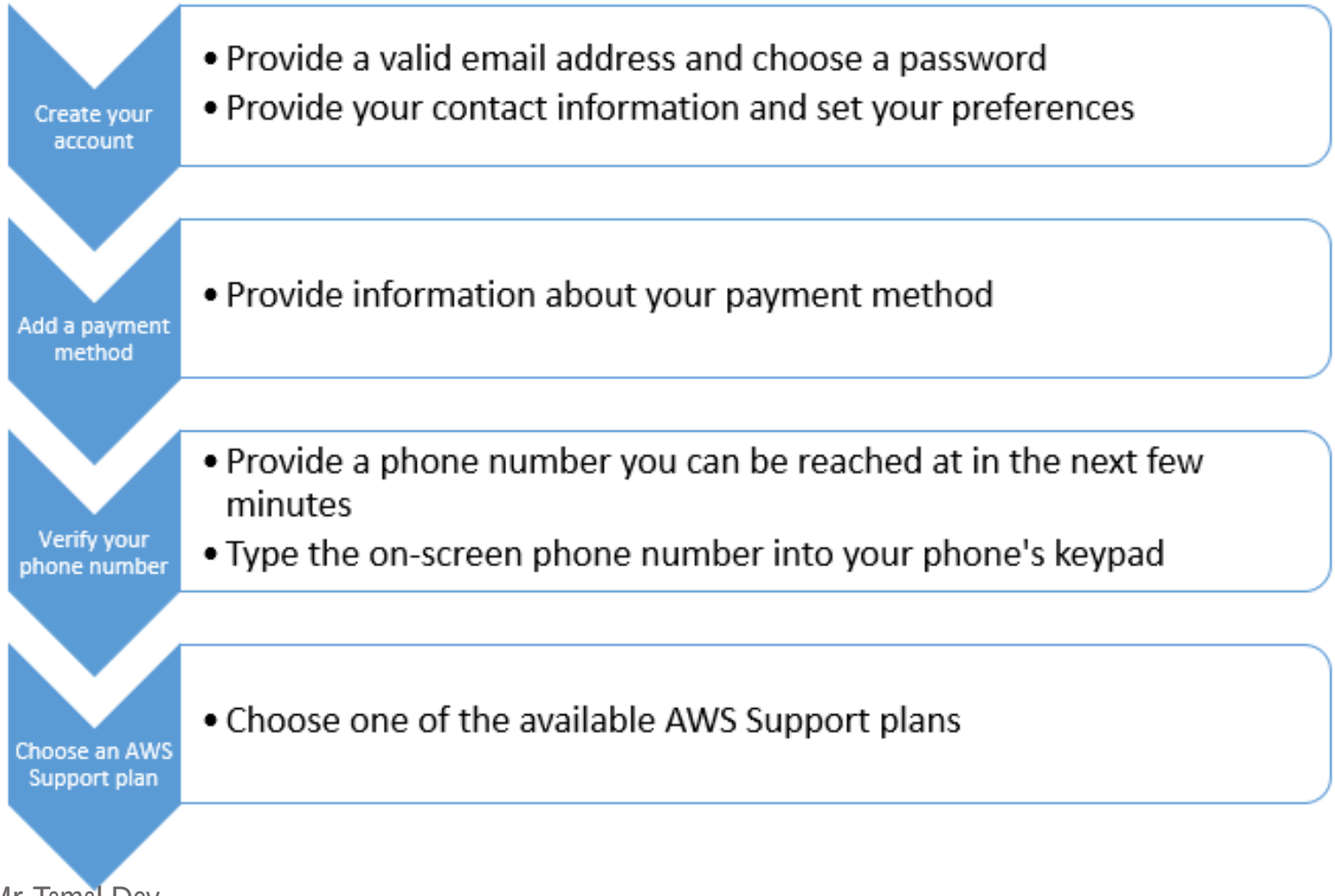
- 18 regions
- Global footprint – 1 million active customers in 190 countries, and steadily increasing
- Massive data centres (3000 – 5000)
- Multiple availability zones per region

AWS Free-Tier

Advantages

- 64 services currently
- Launching new services in all domains
- Focus on machine learning
- Focus on SAAS products
- Reduction in costs

Setting up AWS account



1. Go to the Amazon Web Services home page.
2. Choose Sign Up.
   Note: If you've signed in to AWS recently, it might say Sign In to the Console.
3. Type the requested account information, and then choose Continue.
   Note: If Create a new AWS account isn't visible, first choose Sign in to a different account, and then choose Create a new AWS account. When creating a new account, be sure that you enter your account information correctly, especially your email address. If you enter your email address incorrectly, you might not be able to access your account or change your password in the future.
4. Choose Personal or Professional.
   Note: These two account types are identical in functionality.
5. Type the requested company or personal information.
6. Read the AWS Customer Agreement, and then check the box.
7. Choose Create Account and Continue.
8. Note: After you receive an email to confirm that your account is created, you can sign in to your new account using the email address and password you supplied. However, you must continue with the activation process before you can use AWS services.

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

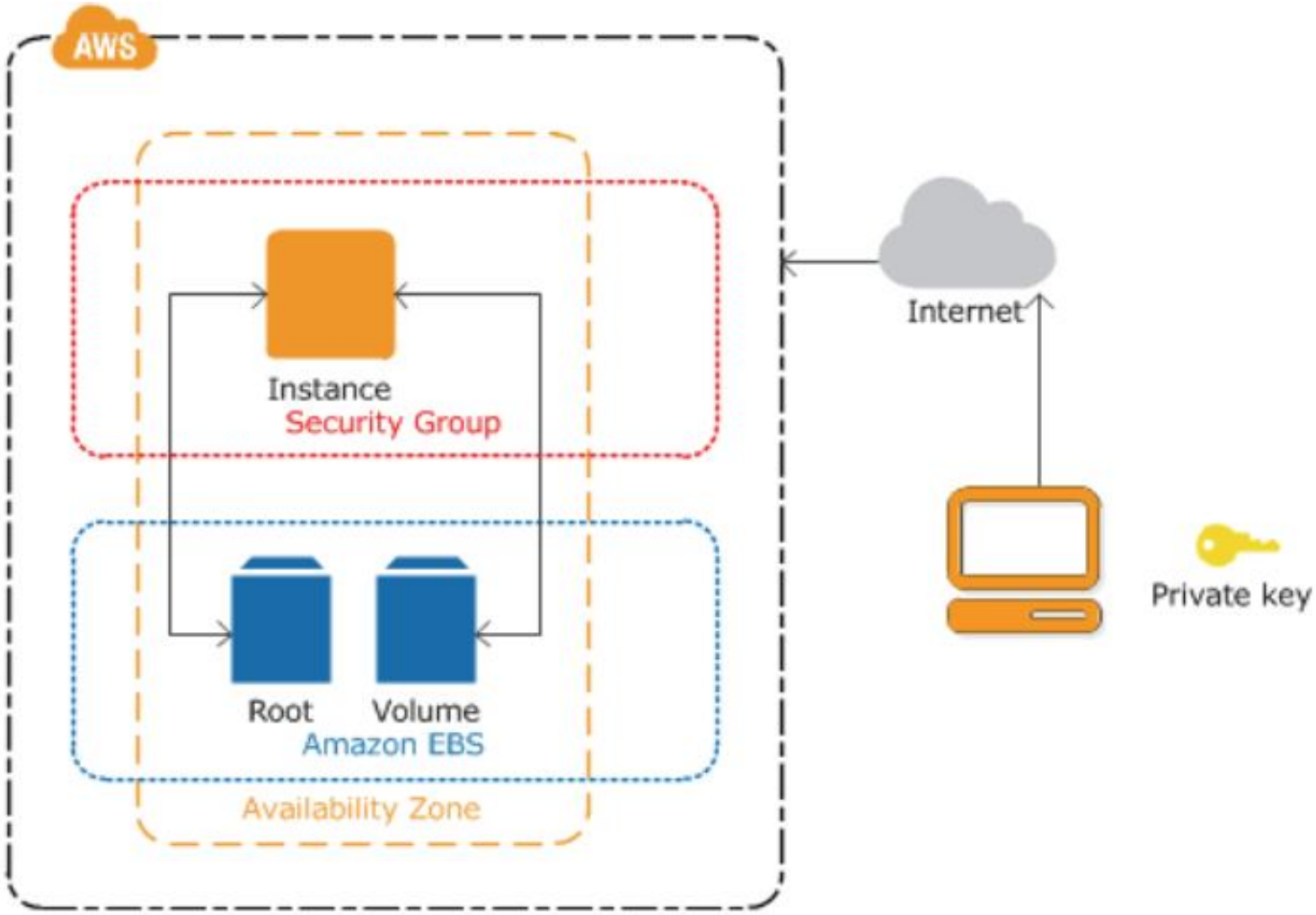
- Region is a specific geographic area
- Region Identifiers ends in numbers
- Each region has multiple, isolated locations known as Availability Zones (AZ)
- AZ identifiers ends in alphabets
- The host is physically located in an Availability Zone (AZ)
- Some services are region specific and some are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

Amazon Elastic Compute Cloud (EC2)

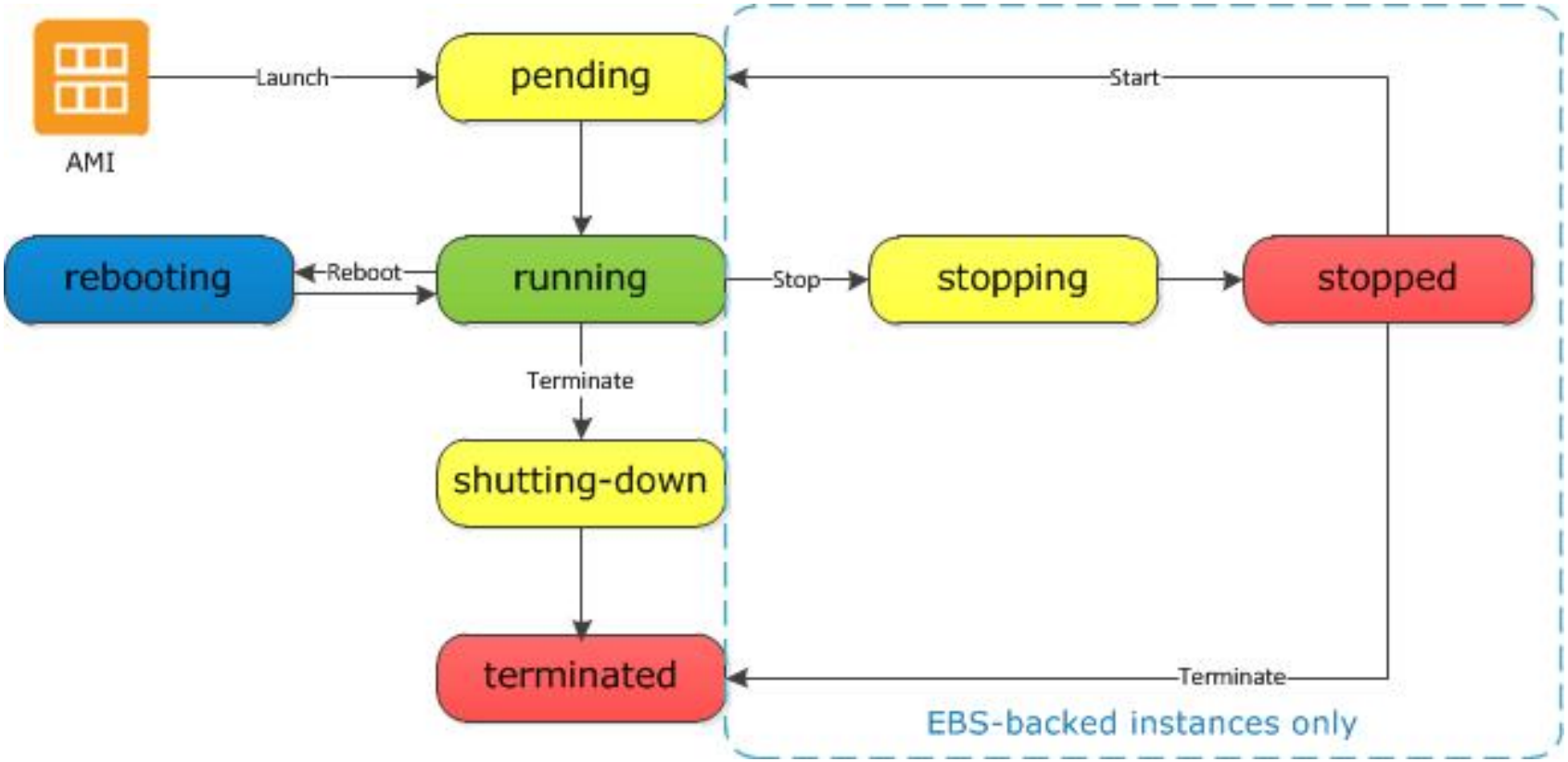
- Amazon Machine Images (**AMIs**) are the basic building blocks of Amazon EC2
- An AMI is a template that contains a software configuration (operating system, application server and applications) that can run on Amazon's computing environment
- AMIs can be used to launch an ***instance***, which is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


AWS
Instance
Security Group
Root
Volume
Amazon EBS
Availability Zone
Internet
Private key

Instance Lifecycle



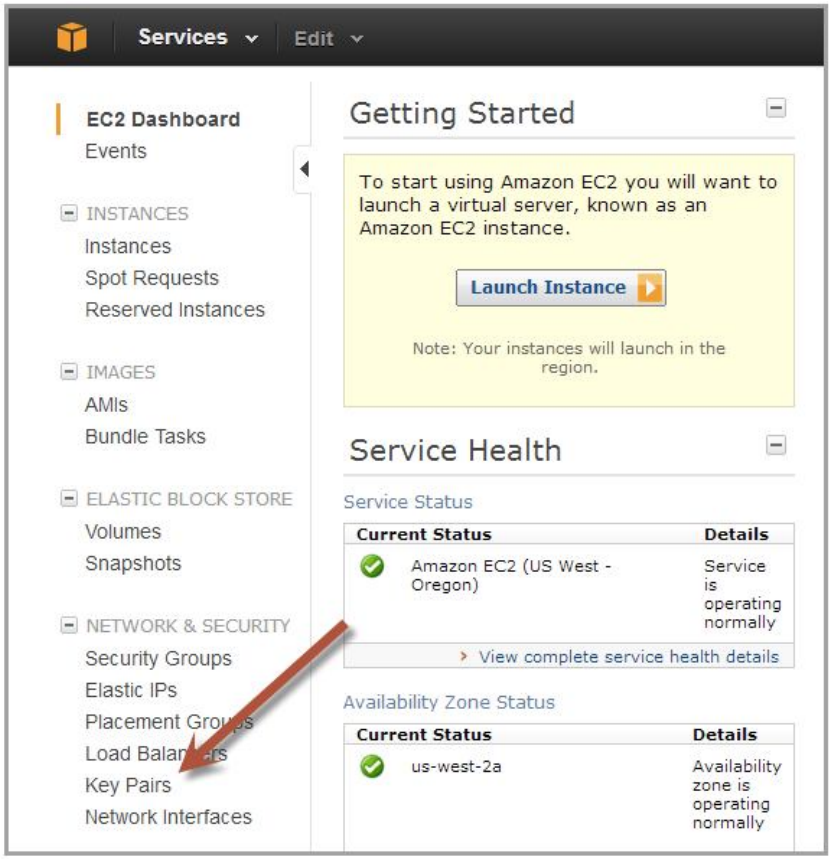
- Instance Usage Billing Info.
- https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-instance-lifecycle.html

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



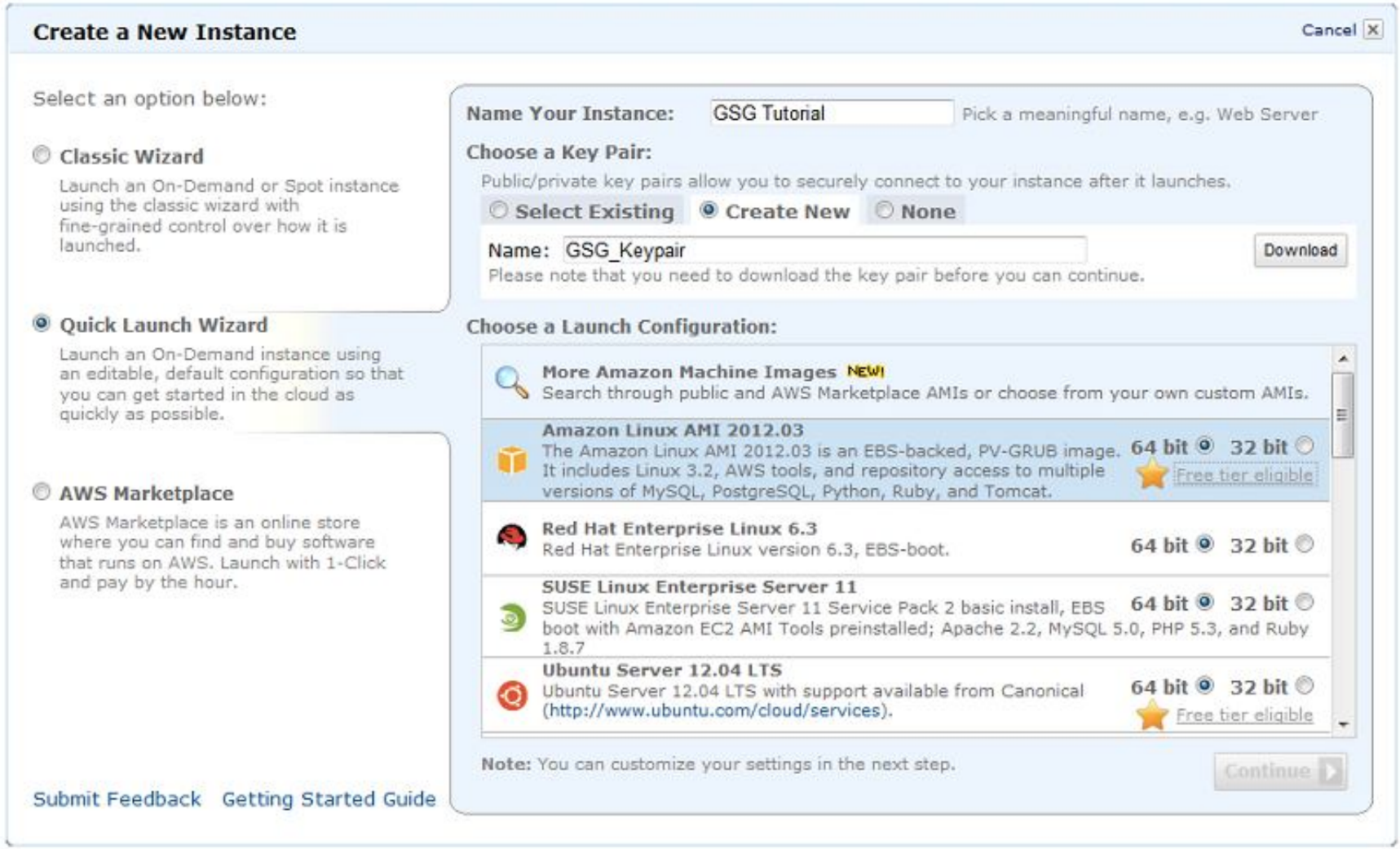
Generating a key pair with EC2 (cont.)

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**



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**

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

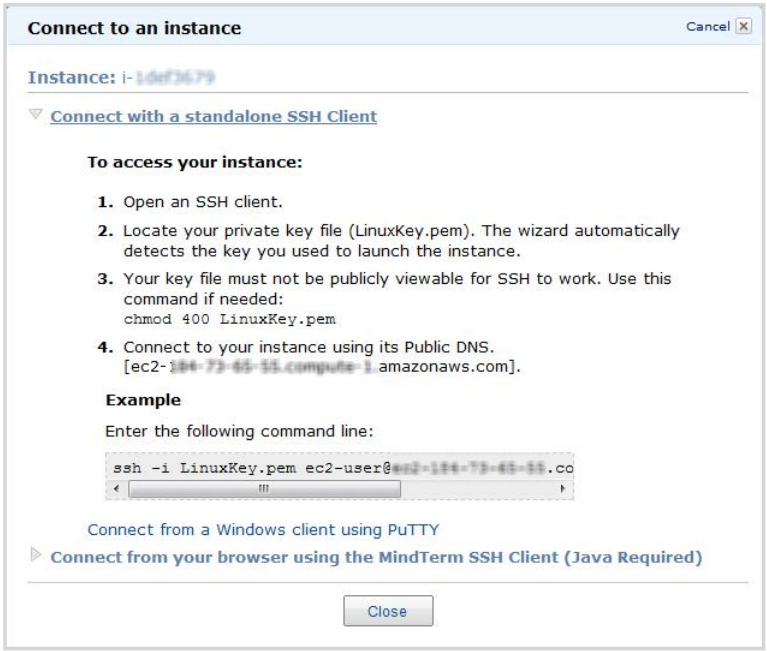
Prerequisites:

- 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

```
ssh -i <your key a name>.pem ec2-user@ec2-184-72-204-112.compute-1.amazonaws.com
```

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:~
```

Example: scp -i "WebSec.pem" hello.txt ec2-user@10.0.2.82:~/.

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

Amazon s3

Tamal Dey,
Dept. of CA
PESU

What's in it for you?

- What is Cloud storage?
- Types of storage
- Before Amazon S3
- What is S3?
- Benefits of S3
- Objects and Buckets
- How does Amazon S3 work
- Features of S3

What is cloud storage?

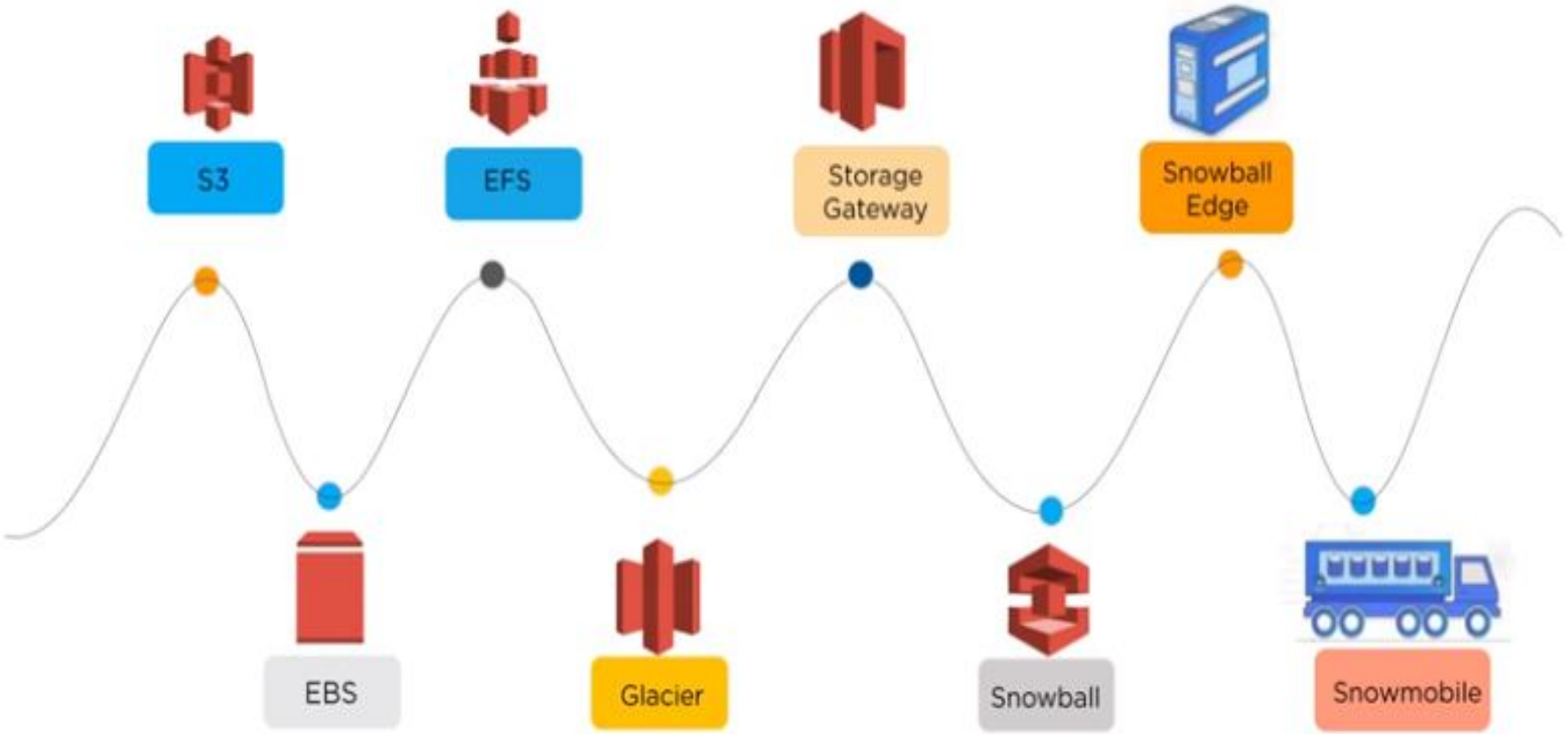
Cloud storage provides a web service where your data can be stored, accessed and easily backed up by users over the internet

Benefits

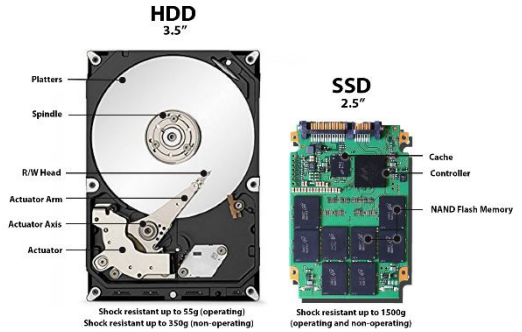
Cloud storage is

- Reliable
- Scalable
- Secure

Types of Storage in AWS



Types of Storage in AWS



- **S3** (Simple Storage Service) – cloud storage
- **EBS** (Elastic Block Store)– similar C drive or E drive (SSD drives attached to instances)
- **EFS** (Elastic File System) - shared file systems (multiple systems)
- **Glacier** – archiving solution (low cost back up)[store infrequently used data, or "cold data."]
- **Storage gateway** – safely moving data from on-premises to cloud, hybrid cloud storage
- **Snowball** – data import and export system (h/w given to premise for data storage securely)
- **Snowmobile** – massive data centre on mobile. You can transfer up to 100PB per Snowmobile

Before Amazon S3

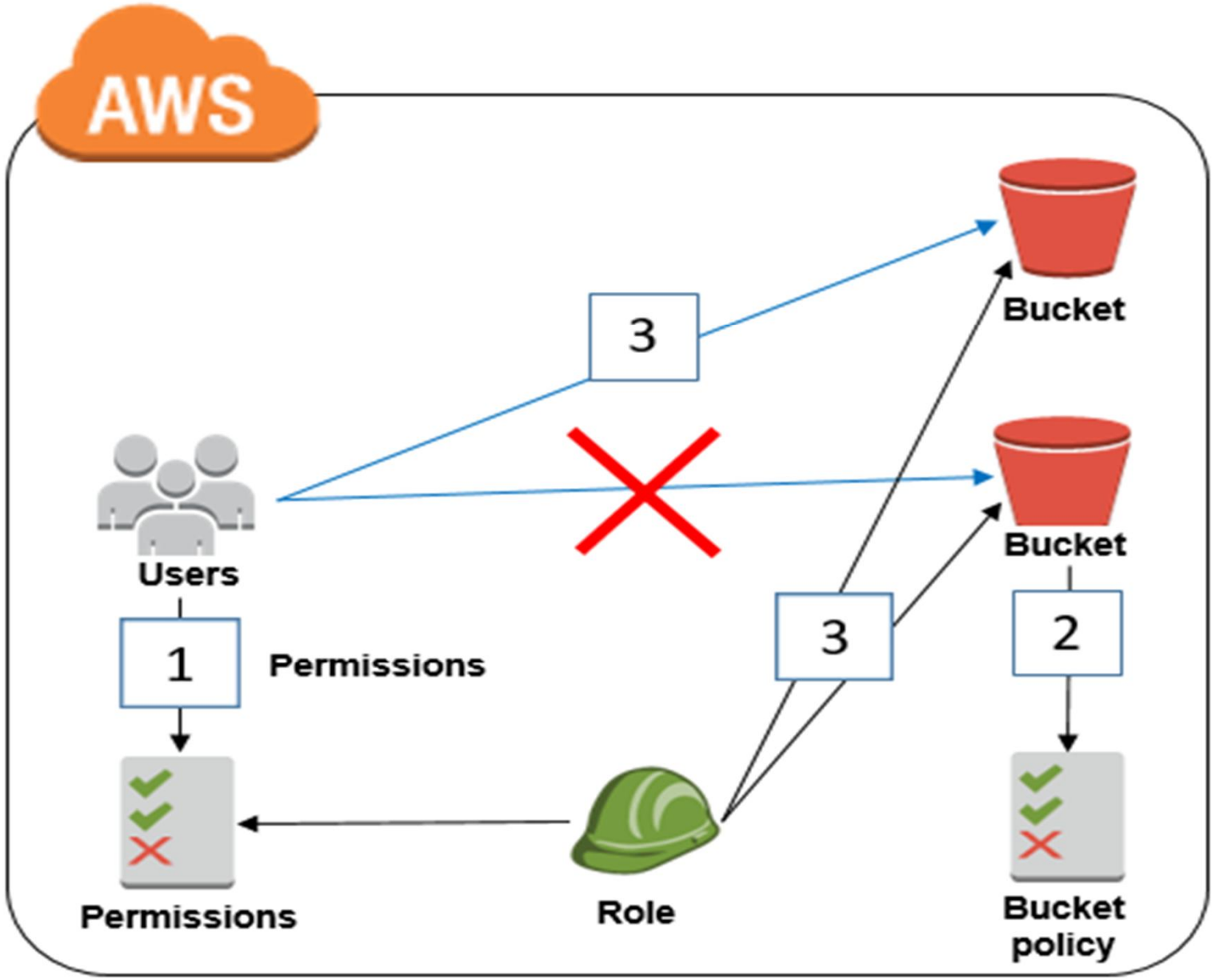
MAINTAINING YOUR OWN REPOSITORY IS EXPENSIVE AND TIME CONSUMING

Factors that make a repository expensive and time consuming are:

- To purchase hardware and software components
- Hiring a team of experts for maintenance
- Lack of scalability based on your requirements
- Data security requirements

What is S3?

Amazon S3 (Simple Storage Service) provides object storage which is built for storing and recovering any amount of information or data from anywhere over the internet


AWS
Bucket
3
Bucket
Users
1
Permissions
3
2
Permissions
Role
Bucket
policy

Know about S3

- S3- simple storage service - It provides **object storage** service
- Amazon S3 provides storage through web service interface
- It is designed for developers where web-scale computing can be easier for them
- You cannot install anything on S3
- It can store files upto 5 TB in size
- durability(99.999999999%)
- 99.99% availability, expected loss of 0.000000001% of objects
- S3 is cheap
- S3 is a regional service (any region you can opt)
- Lot of security provision

	Approx. Bytes	Actual Bytes	Approx. Bits	Typical file/media
1B	1	1	8	Text email, SMS
1KB	1000B = 10^3	1024B = 2^{10}	8 x 10^3	Word document
1MB	1000KB = 10^6	1024KB = 2^{20}	8 x 10^6	Digital photo
1GB	1000MB = 10^9	1024MB = 2^{30}	8 x 10^9	DVD
1TB	1000GB = 10^{12}	1024GB = 2^{40}	8 x 10^{12}	Hard disk
1PB	1000TB = 10^{15}	1024TB = 2^{50}	8 x 10^{15}	Cloud?

Highlights of S3

What is object and bucket?

An object consists of data, key(assigned name) and metadata

A bucket stores objects

When data is added to the bucket, Amazon S3 creates a unique version ID and allocates it to the object

For Example:

Object: folder/Penguins.jpg → Key(name)
Bucket: simplilearn → Version ID
Link Address: https://s3.amazonaws.com/simplilearn/folder/Penguins.jpg

How does it work?

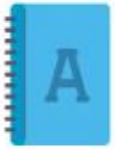
- ✓ When files are uploaded to the bucket, the user will specify the type of S3 storage class to be used for those specific objects
- ✓ Later, users can define features to the bucket like bucket policy, lifecycle policies, versioning control etc.

Storage Classes

- **Standard**
  - For frequently accessed data. Stores object data redundantly across multiple geographically separated Availability Zones
- **Standard-IA**
  - For infrequently accessed data. Stores object data redundantly across **multiple** geographically separated Availability Zones. Minimum **30-day** retention period and minimum **128 KB** object size.
- **One Zone-IA**
  - For infrequently accessed data. Stores object data in **only one** Availability Zone at a lower price than Standard-IA. Minimum **30-day** retention period and minimum **128 KB** object size
- **Glacier**
  - low-cost cloud storage service to move infrequently accessed data
- **Reduced redundancy**
  - For **frequently** accessed data. Stores noncritical, reproducible data at lower levels of redundancy than Standard.
- To define the storage classes – Go to objects – click properties

Storage class in Amazon S3 with a "School" use case

Amazon S3 Standard for frequent data access	Suitable for a use case where the latency should be low Example: Frequently accessed data will be the data of students' attendance, which should be retrieved quickly
Amazon S3 Standard for infrequent data access	Can be used where the data is long lived and less frequently accessed Example: Students' academic record will not be needed on a daily basis, but if they have any requirement, their details should be retrieved quickly
Amazon Glacier	Can be used where the data has to be archived and high performance is not required Example: Ex-student's old record (like admission fee), will not be required on a daily basis and even if it is necessary, low latency is not needed



Students

Student's old record

Storage class in Amazon S3 with a “School” use case

One Zone-IA Storage Class	Can be used where the data is infrequently accessed and stored in a single region Example: Student’s report card is not used on a daily basis and stored in a single availability region (i.e., school)
Amazon S3 Standard Reduced Redundancy storage	Suitable for a use case where the data is non critical and reproduced quickly Example: Books in the library are non critical data and can be replaced if lost

Student’s report card

Library Books

Storage Class Summary

Amazon S3 Standard for frequent data access	Amazon S3 Standard for infrequent data access	Amazon Glacier
• For frequently accessed data • It is a default storage class • Can be used for cloud applications, dynamic websites, content distribution, gaming applications, and Big data analytics	• For infrequently accessed data • Demands rapid access • Suitable for backups, disaster recovery and lifelong storage of data	• Suitable for archiving data where data access is infrequent • Vault-lock feature provides a long term data storage • Provides the lowest cost availability

Storage Classes Comparison in Amazon S3

Storage Class	Durability	Availability	SSL support	First byte latency	Lifecycle Management Policies
STANDARD	99.999999999%	99.99%	Yes	Milliseconds	Yes
STANDARD_IA	99.999999999%	99.99%	Yes	Milliseconds	Yes
ONEZONE_IA	99.999999999%	99.5%	Yes	Milliseconds	Yes
GLACIER	99.999999999%	99.99%	Yes	Minutes or Hours	Yes
RRS	99.99%	99.99%	Yes	Milliseconds	Yes

Life Cycle Management

... and after 60 days, it is moved to Glacier

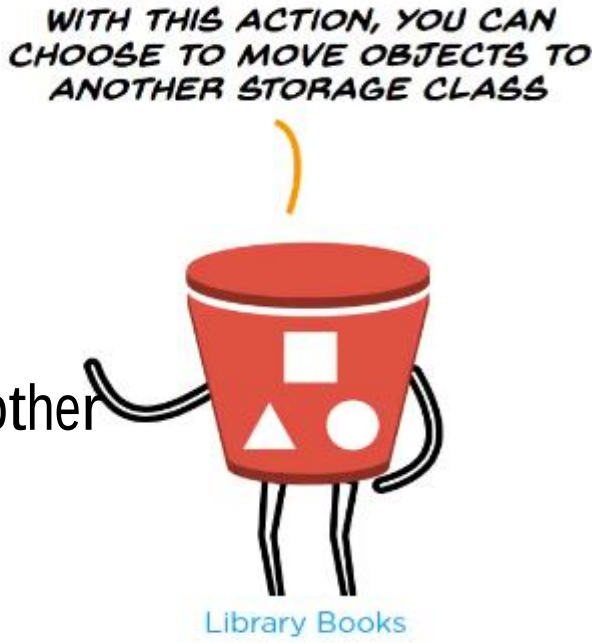
Transition actions

THIS LIFECYCLE MANAGEMENT HELPS YOU TO AUTOMATICALLY MIGRATE YOUR DATA TO LOWER COST STORAGE AS YOUR DATA AGES

You can configure S3 to move your data between various storage classes on a defined schedule

Life Cycle Management

In lifecycle management, Amazon S3 applies a set of rules that define actions to a group of objects



- **Transition actions**
  - Move data from one storage class and another storage class
- **Expiration actions**
  - Expiration date can be fixed for any object
- To do Life cycle management go to bucket and click to Management tab and add the transition action and expiration action.

Bucket Policy

- ✓ Bucket policy is an IAM policy where you can allow and deny permission to your Amazon S3 resources
- ✓ With bucket policy, you also define security rules that apply to more than one file within a bucket
- ✓ For example: If you do not want a user to access the "Simplilearn" bucket, then with the help of JSON script you can set permissions

- Create a bucket policy
  - Suppose you want to deny particular user
  - First generate the policy as a JSON file
  - Goto https://awspolicygen.s3.amazonaws.com/policygen.html
  - Or type "aws policy generator" in Google.
    - Effect – Deny
    - Principal – action (give * for all)
    - Service – amazon s3
    - Actions – click all
    - ARN – copy from aws s3 bucket properties
    - Add condition – specify the user = or <>
    - Add statement
    - Generate policy -> A JSON script will be generated.
    - Copy the JSON file and paste in the bucket policy.

Data Protection

- ✓ Amazon S3 provides IT teams a highly durable, protected and scalable infrastructure designed for object storage

S3 Data Protection Techniques

- ✓ Amazon S3 provides IT teams a highly durable, protected and scalable infrastructure designed for object storage
- ✓ Amazon S3 protects your data using 2 methods:
  - ❑ Data Encryption and
  - ❑ Versioning

Data encryption

Versioning

Data Encryption

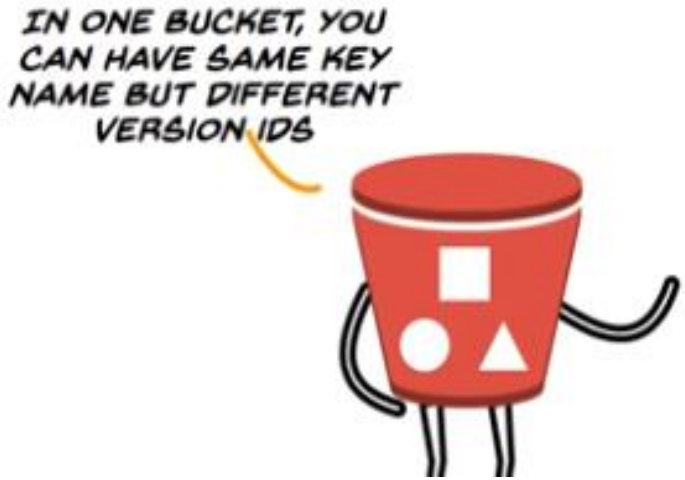
- ✓ It refers to protection of data while it's being transmitted and at rest
- ✓ Data Encryption can happen in two ways:

Client-Side Encryption - Data encryption at rest

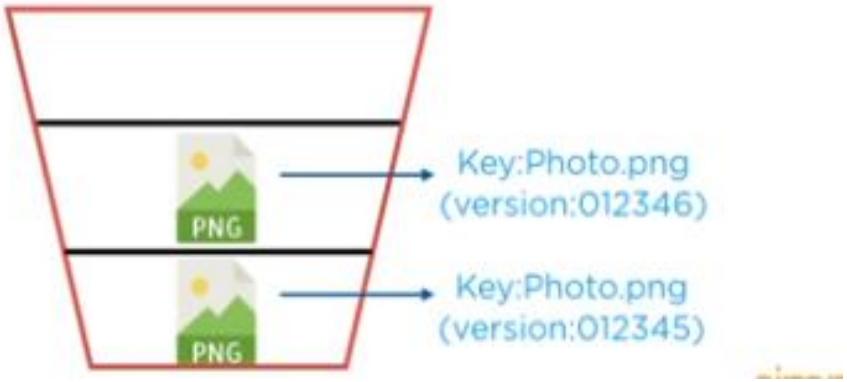
Server-Side Encryption - Data encryption in motion

Versioning

- ✓ It can be utilized to preserve, recover and restore early versions of every object you store in your Amazon S3 bucket
- ✓ Unintentional erase or overwriting of objects can be easily regained with versioning

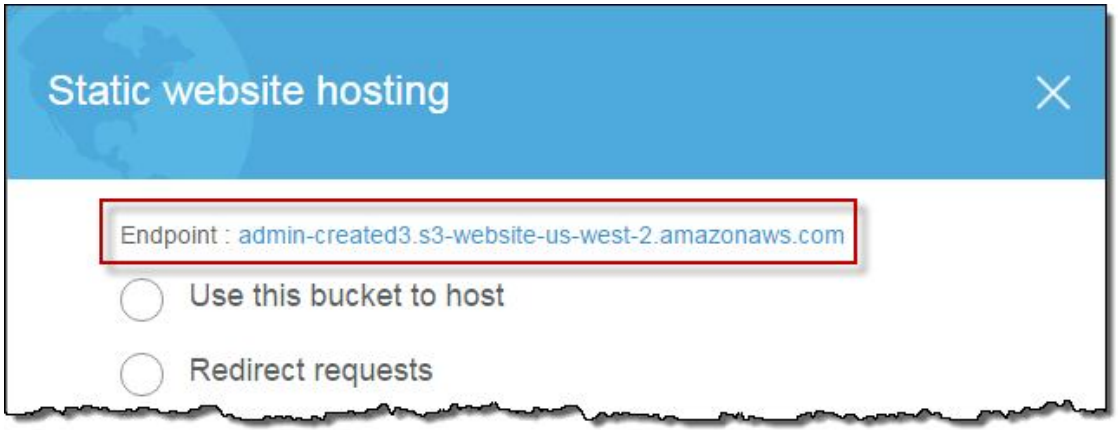


For Example:



Static Webpage Hosting

- Click on Bucket Properties and Enable Static Webpage Hosting
- Click on the Static Webpage Hosting menu and write two file names (index.html and error.html) and save
- Create and Edit index.html and error.html in your local machine and upload the files on the bucket with public access permission
- Click on the endpoint link on Static Webpage Hosting menu to get the page view



Cross-Region Replication

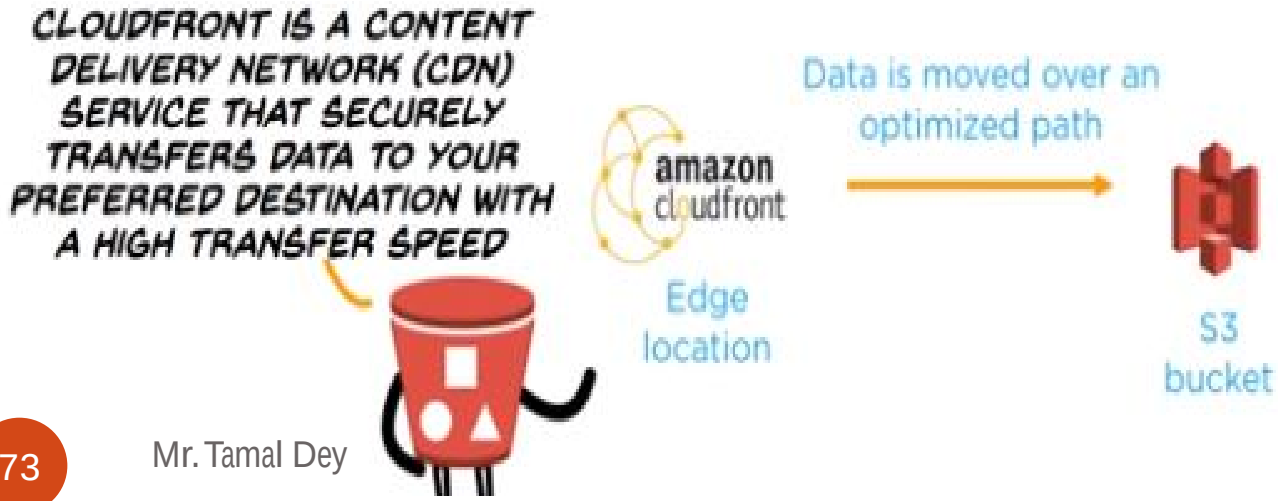
Cross-Region Replication provides automatic copying of every object uploaded to your buckets (source bucket and destination bucket) in different AWS regions

Note: Versioning must be turned on to enable CRR

- Before doing CRR both buckets should have versioning enabled.
- Create a destination bucket in a different region
- Under Bucket -> properties -> Replicate

Accelerated Transfer

- ✓ It enables fast, easy and secure transfers of files over long distances between your client and S3 bucket
- ✓ The edge locations around the world provided by Amazon CloudFront are taken advantage by transfer acceleration
- ✓ It works via carrying data over an optimized network bridge that keeps running between the AWS Edge Location (closest region to your clients) and your Amazon S3 bucket



- Cloud front helps to copy the bucket from one region to another region by serveral intermediate copies to the nearest region and not directly to the destination region.
- This is useful, when the source and destination regions are farther.
- Under bucket properties ->Enable accelerated transfer

Reading resources

- https://www.youtube.com/watch?v=XGcoeEyt2UM

Networking in AWS

Tamal Dey,
Dept. of CA, PESU

Networking and Content Delivery

- **Amazon VPC**
- Amazon CloudFront
- Amazon Route 53
- AWS PrivateLink
- AWS Direct Connect
- AWS Global Accelerator
- Amazon API Gateway
- AWS Transit Gateway
- AWS App Mesh
- AWS Cloud Map
- Elastic Load Balancing

Why VPC?

- Provision a logically isolated section of the Amazon Web Services (AWS) Cloud where you can launch AWS resources in a virtual network that you define.
- Creator has complete control over your virtual networking environment, including selection of your own IP address range, creation of subnets, and configuration of route tables and network gateways.
- You can use both IPv4 and IPv6 in your VPC for secure and easy access to resources and applications.
- You can easily customize the network configuration for your Amazon VPC.

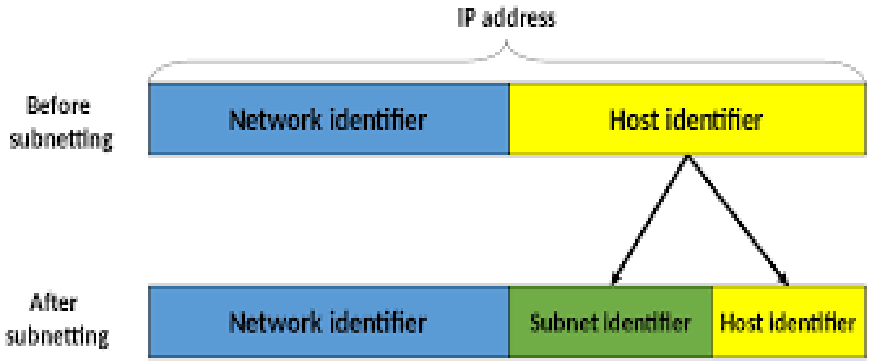
What Is Amazon VPC?

- Amazon Virtual Private Cloud (Amazon VPC) enables you to launch AWS resources into a virtual network that you've defined. This virtual network closely resembles a traditional network that you'd operate in your own data center, with the benefits of using the scalable infrastructure of AWS.
- **Benefits**
  - Define Custom Networks.
  - Assign static private IPv4 address to instances.
  - Define network interface and attach one or more network interface to the instances.
  - Define the routing between different subnets.
  - Define network security by allowing or denying the traffic.
  - Control the out bound traffic along with inbound traffic using ACL.

VPC Key Concepts

- A ***subnet*** is a range of IP addresses in your VPC. To launch AWS resources into a specified subnet. Use a public subnet for resources that must be connected to the internet, and a private subnet for resources that won't be connected to the internet.
- To protect the AWS resources in each subnet, you can use multiple layers of security, including **security groups** and **network access control lists** (**ACL**).
- A virtual private cloud (VPC) is a virtual network dedicated to your AWS account. It is logically isolated from other virtual networks in the AWS Cloud. You can launch your AWS resources, such as Amazon EC2 instances, into your VPC.
- VPC in the form of a **Classless Inter-Domain Routing** (CIDR) block; for example, ***10.0.0.0/16.*** This is the primary CIDR block for your VPC, a way to allow more flexible allocation of Internet Protocol (IP) addresses.

Subnet



Subnet ID illustration

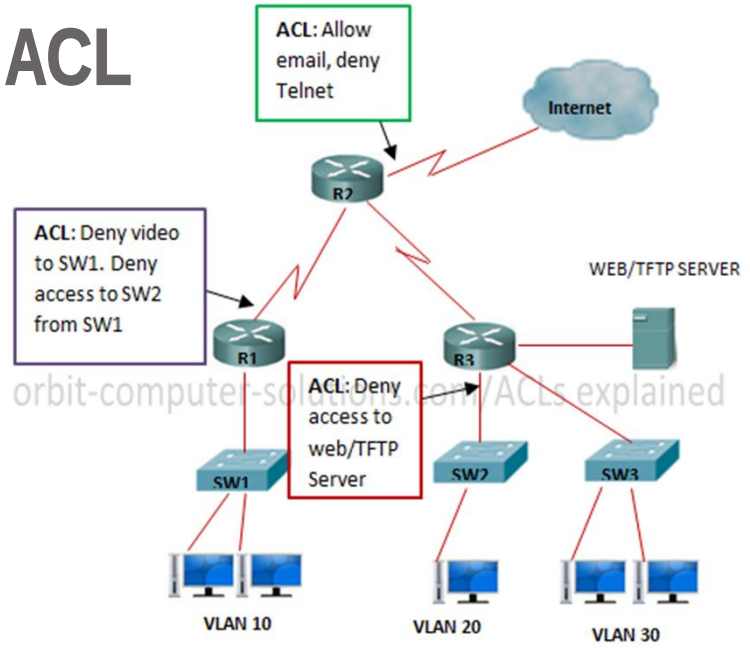
Network Prefix: 172.16.0.0, Subnet ID: 172.16.2.0, Host ID: 15

Network Prefix		Subnet ID		Host ID		
1010 1100	.	0001 0000	.	0000 0010	.	0000 1111
1111 1111	.	1111 1111	.	1111 1111	.	0000 0000
1010 1100	.	0001 0000	.	0000 0010	.	0000 0000
172	.	16	.	2	.	15

CIDR

IPv4 CIDR IP/CIDR	Δ to last IP addr	Mask	Hosts (*)	Class
a.b.c.d/32	+0.0.0.0	255.255.255.255	1	1/256 C
a.b.c.d/31	+0.0.0.1	255.255.255.254	2	1/128 C
a.b.c.d/30	+0.0.0.3	255.255.255.252	4	1/64 C
a.b.c.d/29	+0.0.0.7	255.255.255.248	8	1/32 C
a.b.c.d/28	+0.0.0.15	255.255.255.240	16	1/16 C

ACL

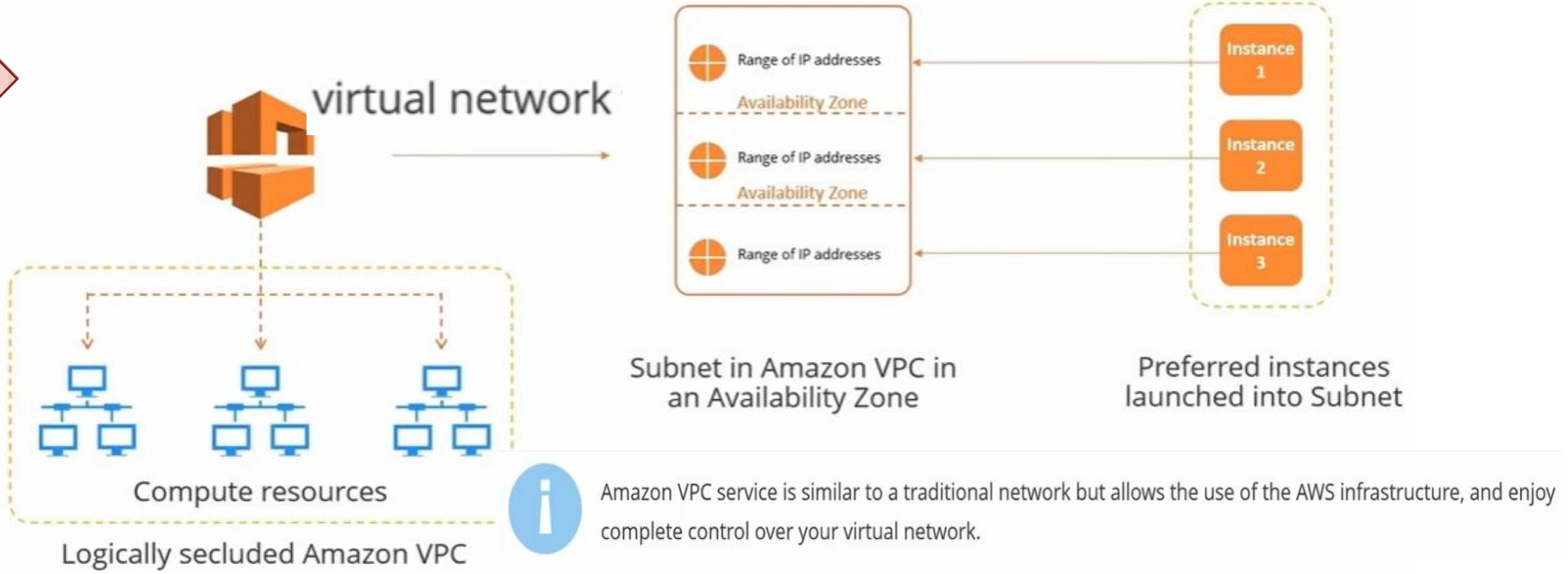


VPC Key Concepts

- **Security Groups**: Each rule is comprised of four fields: 'Type', 'Protocol', 'Port Range', and 'Source'. This applies for both 'Inbound' and 'Outbound' rules.
- The drop down list allows you to select common protocols like SSH, RDP, or HTTP
- A **network access control list** ***(ACL)*** is an optional layer of security for your VPC that acts as a firewall for controlling traffic in and out of one or more subnets.
- A **route table** contains a set of rules, called *routes*, that are used to determine where network traffic is directed.
- Each subnet in your VPC must be associated with a route table; the table controls the routing for the subnet. A subnet can only be associated with one route table at a time, but you can associate multiple subnets with the same route table.
- An **internet gateway** is a horizontally scaled, redundant, and highly available VPC component that allows communication between instances in your VPC and the **internet.**

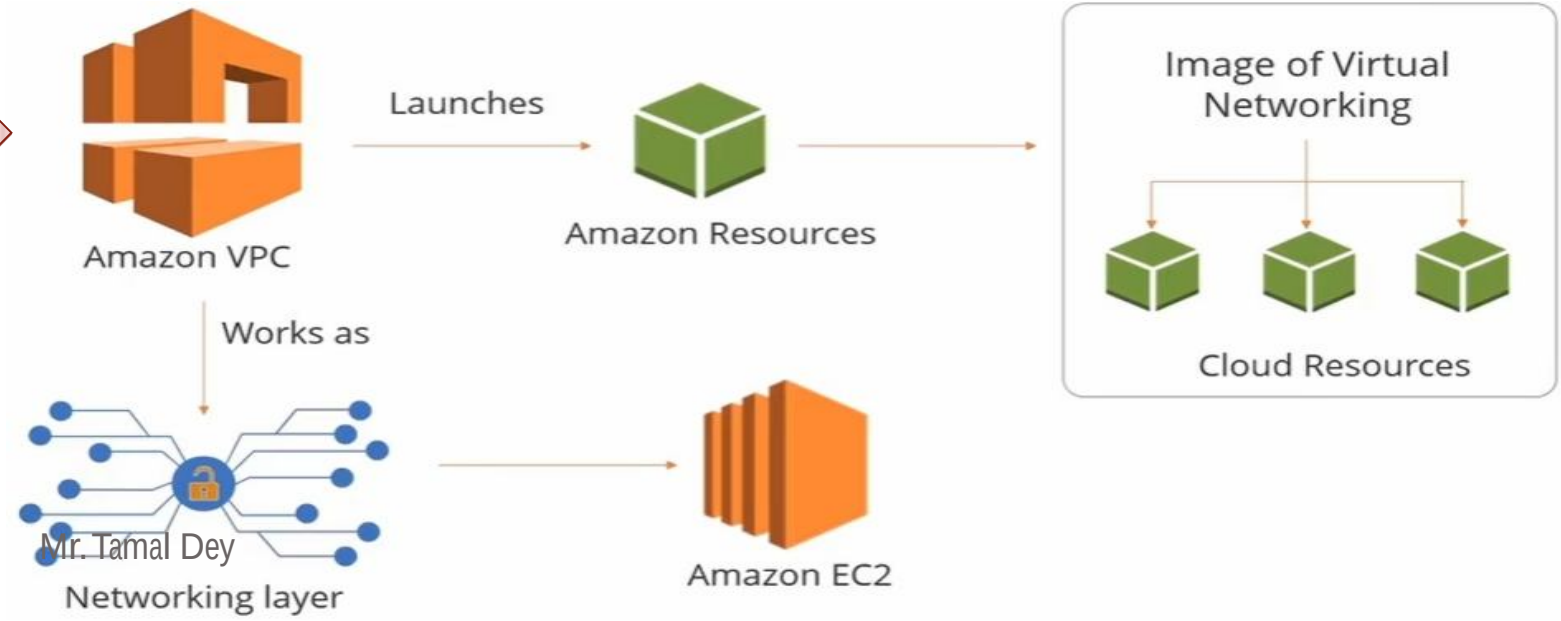
Amazon Virtual Private Cloud

1

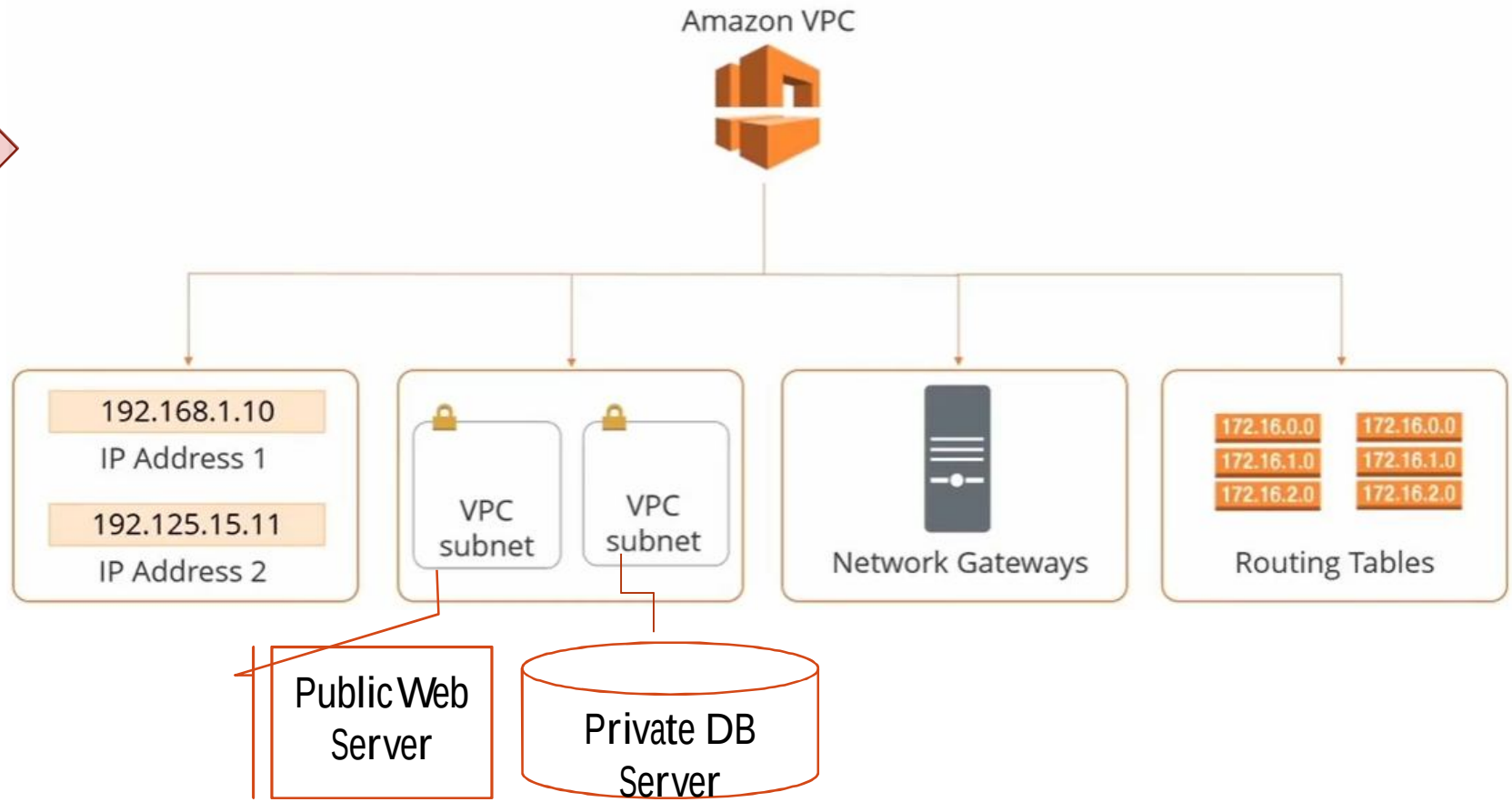


Amazon VPC service is similar to a traditional network but allows the use of the AWS infrastructure, and enjoy complete control over your virtual network.

2



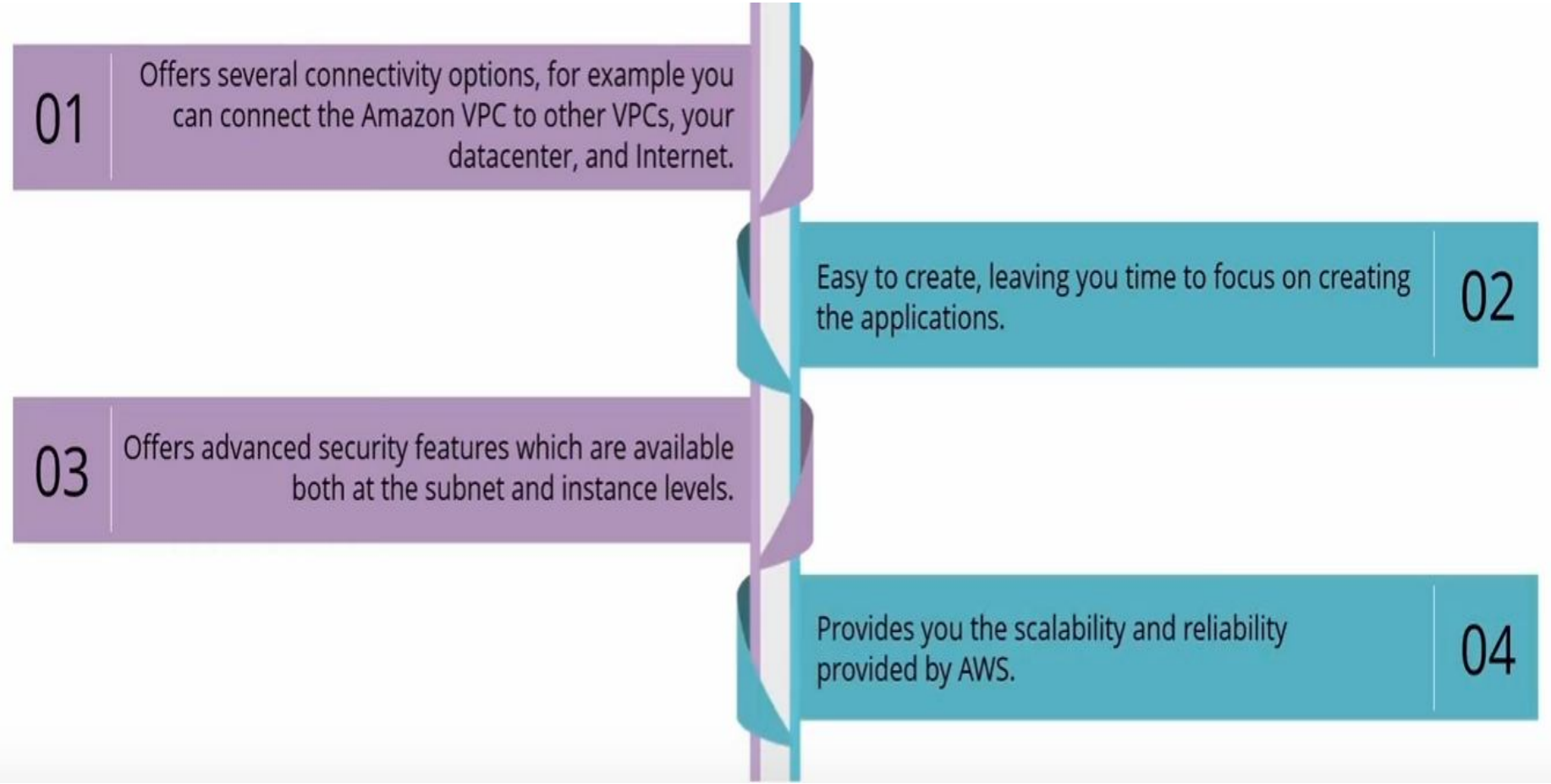
3


Amazon VPC
192.168.1.10
IP Address 1
192.125.15.11
IP Address 2
VPC subnet
VPC subnet
Network Gateways
172.16.0.0
172.16.1.0
172.16.2.0
172.16.0.0
172.16.1.0
172.16.2.0
Routing Tables
Public Web Server
Private DB Server

4

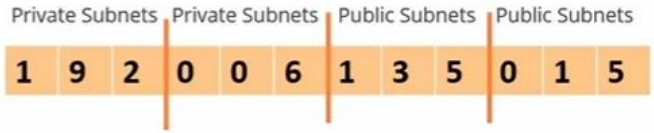

Amazon VPC
VPN Connection
Data Center
AWS
AWS Cloud

Benefits of AWS

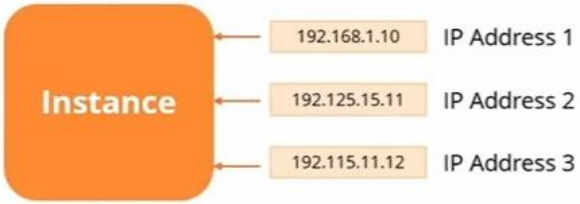


Benefits of Launching Instances in VPC

Run Instances on the hardware used by a single entity



Split the range of private IP addresses of VPC



Allocate multiple IP addresses to Instances

Allocate static private IP addresses to Instances

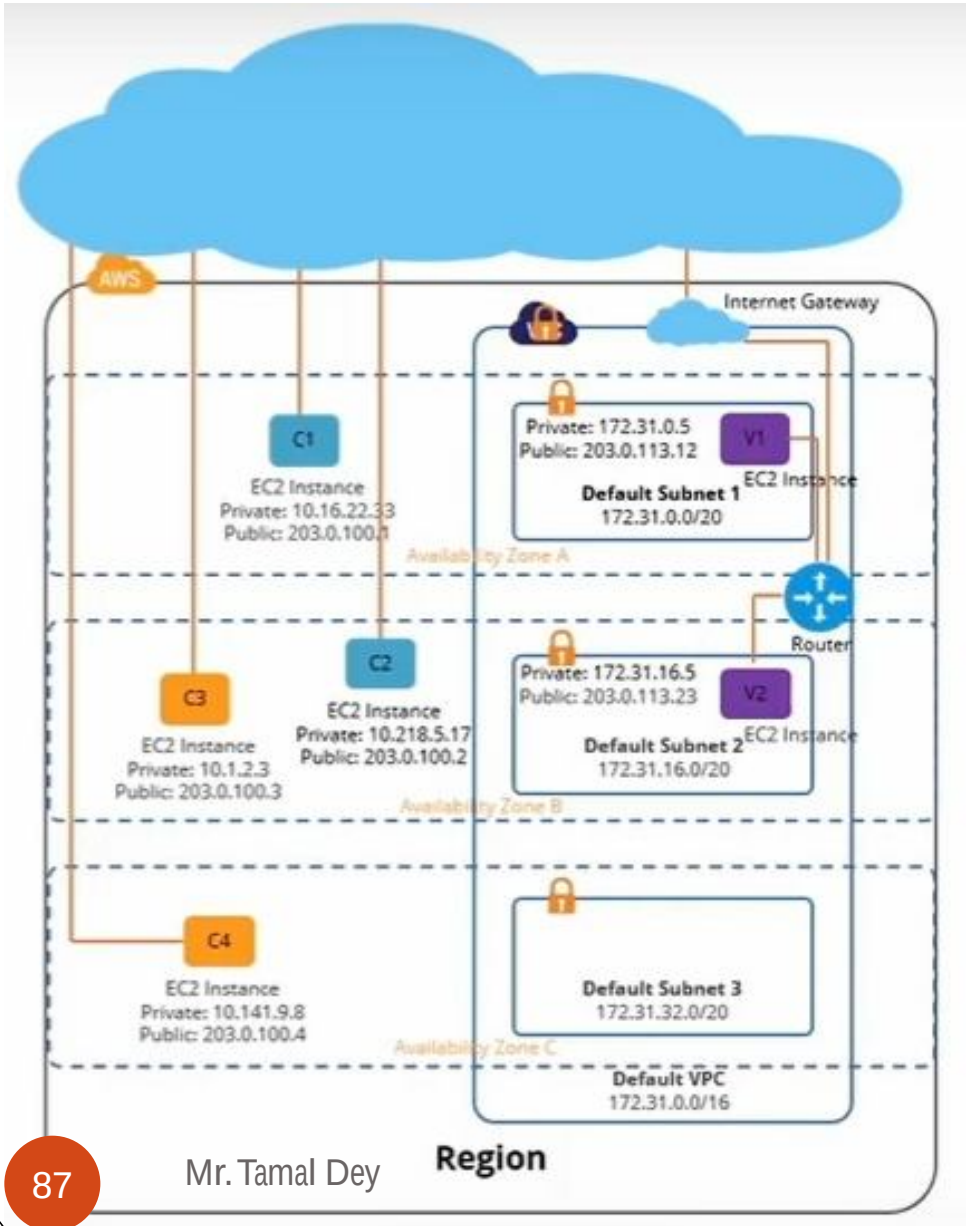
Define Network Interfaces

Control inbound traffic to Instances

Add an extra layer of Access Control to Instances

Change the membership of Security Group of Instances

Default and Non-Default VPC



The default VPC contains a Subnet in each availability zone.

It is ready to use, offering advanced features of the EC2-VPC platform.

Even with an AWS account, you can create and configure a VPC as per your requirements.

Additional subnets in a default VPC and a non-default VPC are termed as non-default Subnets.

SCENARIO

- **VPC with a Single Public Subnet**
- VPC with Public and Private Subnets
- VPC with Public and Private Subnets and Hardware VPN Access
- VPC with a Private Subnet Only and Hardware VPN Access

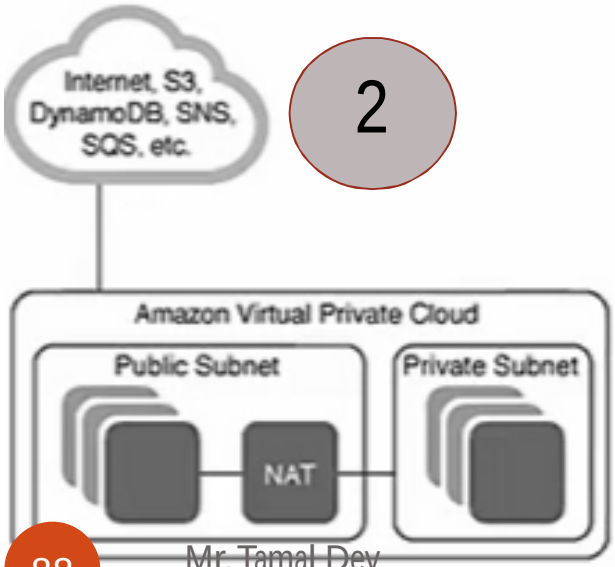
1

VPC with a Single Public Subnet: **Here i**nstances run in a private, isolated section of the AWS cloud with direct access to the Internet. Network access control lists and security groups can be used to provide strict control over inbound and outbound network traffic to your instances.

Creates:

A /16 network with a /24 subnet. Public subnet instances use Elastic IPs or Public IPs to access the Internet.

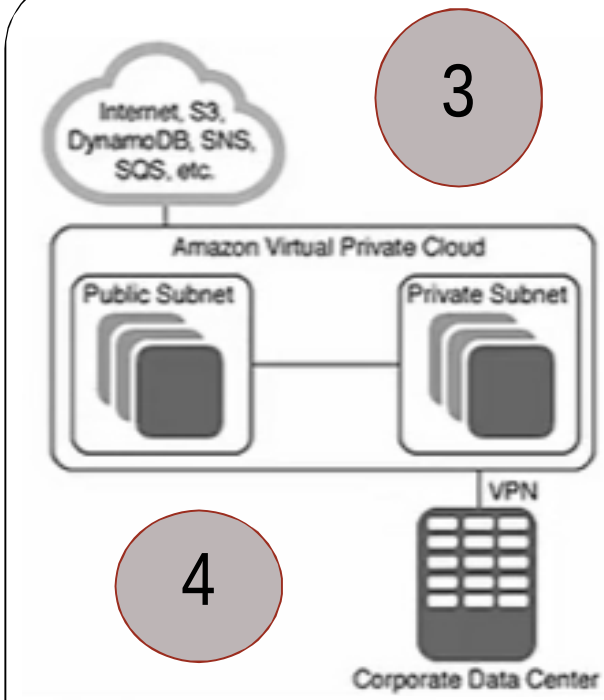
2



VPC with Public and Private Subnets: A public subnet, this configuration adds a private subnet whose instances are not addressable from the Internet. Instances in the private subnet can establish outbound connections to the Internet via the public subnet using Network Address Translation (NAT).

Creates:

A /16 network with two /24 subnets. Public subnet instances use Elastic IPs to access the Internet. Private subnet instances access the Internet via Network Address Translation (NAT). (Hourly charges for NAT devices apply.)

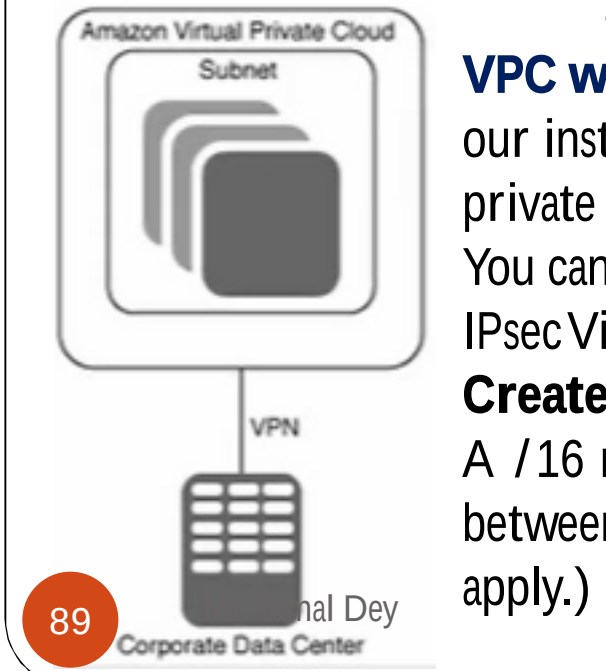


VPC with Public and Private Subnets and Hardware VPN Access

This configuration adds an IPsec Virtual Private Network (VPN) connection between your Amazon VPC and your data center - effectively extending your data center to the cloud while also providing direct access to the Internet for public subnet instances in your Amazon VPC.

Creates:

A /16 network with two /24 subnets. One subnet is directly connected to the Internet while the other subnet is connected to your corporate network via IPsec VPN tunnel. (VPN charges apply.)



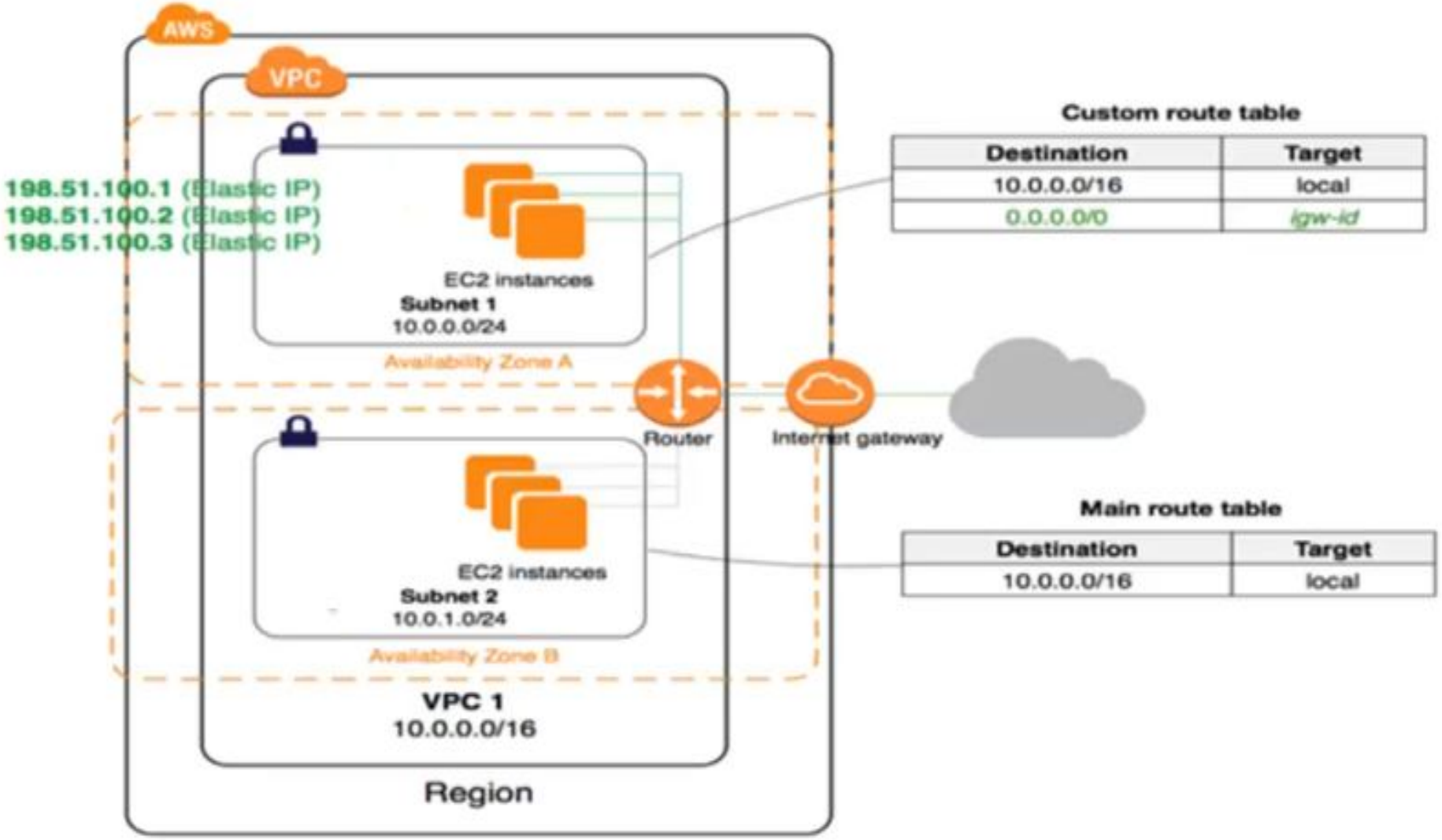
VPC with Private Subnets Only and Hardware VPN Access

our instances run in a private, isolated section of the AWS cloud with a private subnet whose instances are not addressable from the Internet. You can connect this private subnet to your corporate data center via an IPsec Virtual Private Network (VPN) tunnel.

Creates:

A /16 network with a /24 subnet and provisions an IPsec VPN tunnel between your Amazon VPC and your corporate network. (VPN charges apply.)

Implementation Scenario



Steps to Follow

1. Network->VPC
   A. Check/Note Default VPC, Subnet, Internet Gateway, Routing Table, Network ACLs, Security Options, DHCP Option Set
2. Create new VPC (MyVPC)
   A. IPV4 CIDR Block range (10.0.0.0/16) with default Tenancy
   B. Check all the fields of new VPC
   C. Check Route Table, Network ACLs, Security Groups
   D. No Subnet Created and Internet Gateway
3. Create 2 Subnet Name(10.0.1.0-AP S 1A & 10.0.2.0-AP S 1B) in New custom VPC (1p-s-1a and 1b)
   A. IPv4 CDIR Block- 10.0.1.0/24 and 10.0.2.0/24
   B. 1 Subnet in 1 Availability zone . Check new subnet details
4. Create Internet Gateway(MyInternetGateway)
   A. Attach to VPC [1 Gateway only for 1VPC]

5. Create Route Table(MyPublicRoute) from subnet ->Internet Gateway
   A. Edit (save) / Add new Route (0.0.0.0/0) and Target (MyInternetGateway)
   B. Subnet Association (Public -1A Subnet-> Internet Gateway)
   C. Go to Subnet and (Right side> auto assign public IPv4 address enable from Actions button- Auto-assign IPv4

Operating System	Format	Tool(s)
Debian	`.deb`	`apt`, `apt-cache`, `apt-get`, `dpkg`
Ubuntu	`.deb`	`apt`, `apt-cache`, `apt-get`, `dpkg`
CentOS	`.rpm`	`yum`
Fedora	`.rpm`	`dnf`
FreeBSD	Ports, `.txz`	`make`, `pkg`

6. Launce One EC2 instance in Public Subnet
   A. Amazon AMI (Free Tier) , change to new VPC and Public subnet)
   B. Advance Details [Add the **text** in next slide]

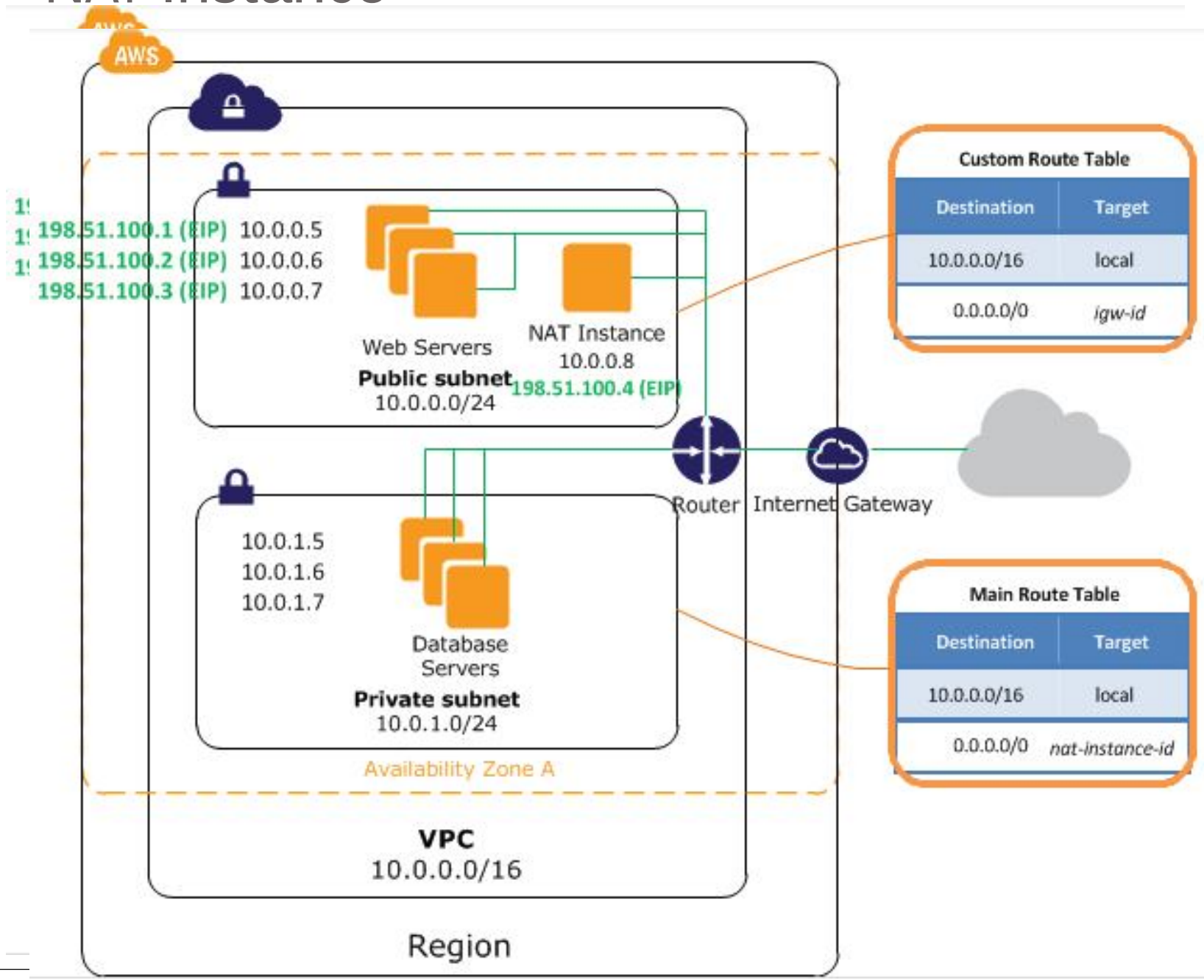
```
#!bin/bash
yum install httpd -y
yum update -y
service httpd start
chkconfig httpd on
echo "<html><h1>Hello World!</h1>
</html>" > /var/www/html/index.html
```

   C. Add default Storage and Add Tag (Name -> Webserver)
   D. Security group Name and Desc. (WebSec) and Add HTTP Protocol
   E. Download new private key pair (WebSec.pem) & Launch Instance
   F. Write auto created IPv4 address in browser (Result)

7. Launce One EC2 instance Private instance
   A. Create Amazon AMI (Free Tier) instance
   B. Choose Custom VPC and Private Subnet with no auto assign IP address
   C. Add storage and Add Tag (Name-> DBServer) [Private Access]
   D. Add Security group protocol for the following (with **Public-IP Address**)
      I. SSH – 10.0.1.0/24
      II. MySQL-10.0.1.0/24
      III. ICMP-IPv4- 10.0.1.0/24
      IV. ICMP-IPv4- 10.0.1.0/24
   E. Download the DB Key pair (WebSec.pem-Existing key pair)
   F. Run with Putty (Windows User) by using private key
      I. Ping DBServer IP Address

End of VPC Type 1

NAT Instance



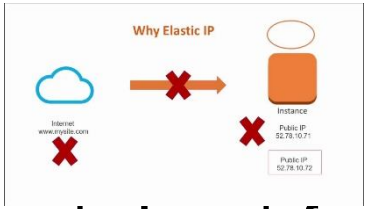
NAT Instances

- **Network Address Translation** (NAT) instance in a public subnet in your VPC to enable instances in the private subnet to initiate outbound IPv4 traffic to the Internet or other AWS services, but prevent the instances from receiving inbound traffic initiated by someone on the Internet.
- *NAT is not supported for IPv6 traffic*—use an egress-only Internet gateway.
- ***Note:*** Use a NAT gateway, which is a managed NAT service that provides better availability, higher bandwidth, and requires less administrative effort.
- For common use cases, we recommend that you use a *NAT gateway* rather than a *NAT instance*.

NAT Gateways

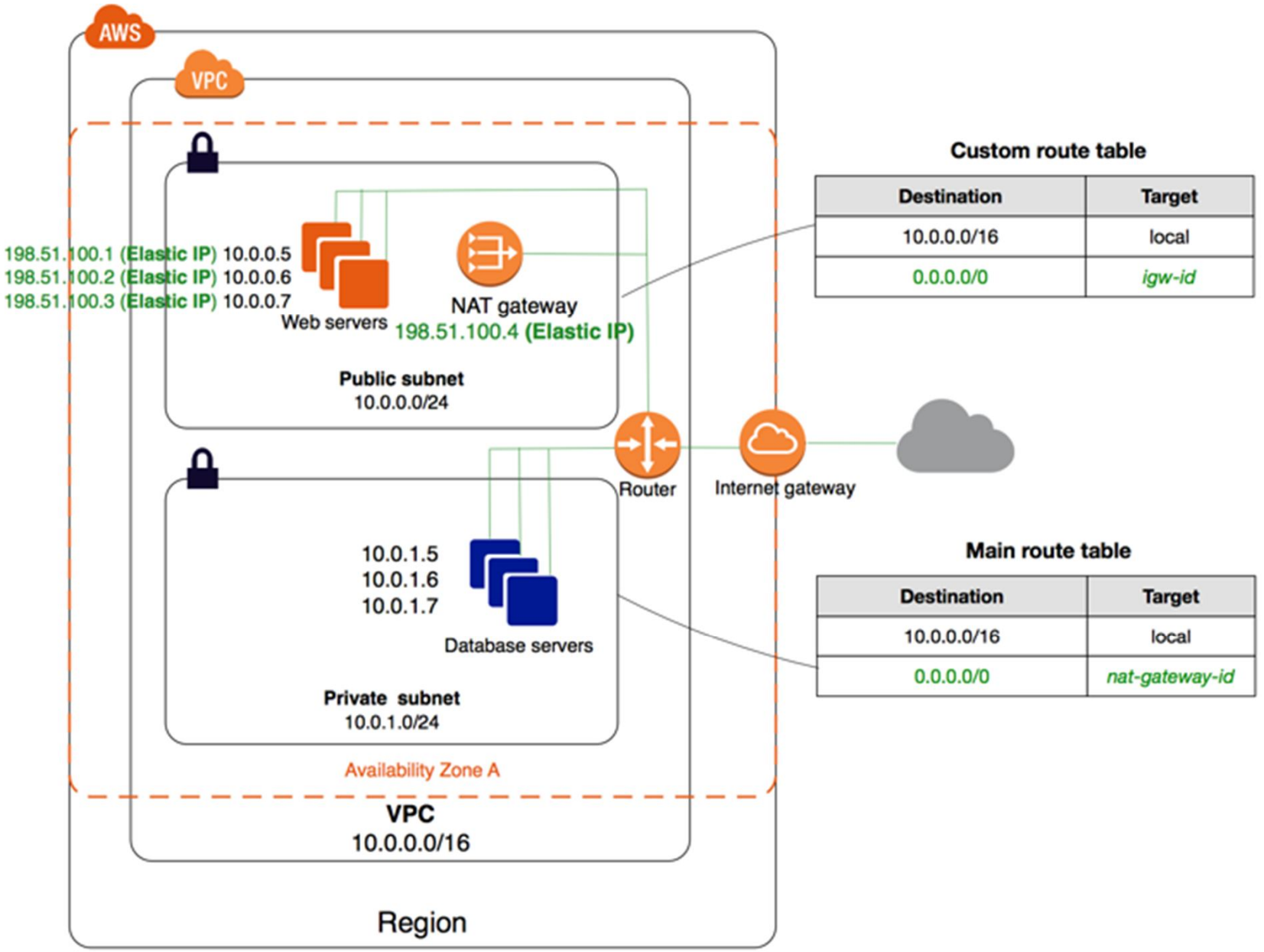
- Enable instances in a private subnet to connect to the internet or other AWS services, but prevent the internet from initiating a connection with those instances.
- You are charged for creating and using a NAT gateway in your account. NAT gateway hourly usage and data processing rates apply. Amazon EC2 charges for data transfer also apply.
  - https://aws.amazon.com/vpc/pricing/
- NAT gateways are not supported for IPv6 traffic
- Diff. NAT Instance vs. NAT Gateway
- https://docs.aws.amazon.com/vpc/latest/userguide/vpc-nat-comparison.html

Elastic IP



- An **Elastic IP** address is a static IPv4 address designed for dynamic cloud computing.
- An **Elastic IP** address is associated with your AWS account.
- With an **Elastic IP** address, you can mask the failure of an instance or software by rapidly remapping the address to another instance in your account.
- An Elastic IP address is a public IPv4 address, which is reachable from the internet.
- If your instance does not have a public IPv4 address, you can associate an Elastic IP address with your instance to enable communication with the internet;
  - for example, to connect to your instance from your local computer.

Lab 6-NAT Gateway



Steps for NAT Gateway

1. Go to EC2 Instance
   A. **Search** in Community Instance (Left Side menu)
      i. NAT (First Option-Free Tier Only)
      ii. Choose Custom VPC and Public Subnet
      iii. All other default Setting, Add a Tag - NATInstance
      iv. Choose existing Security Group created for Public Subnet (WebServer)
      v. Choose Existing Key Pair (WebSec.pem) for key generation
      vi. Launch the instance
      vii. Select your NAT instance after Launch- and Go to

   Action ->Networking-> Change Source/ Destination change (Bypassing the request in this Point)

B. Go to VPC and Select NAT Gateways-> Create One
   i. Choose **public subnet** and create New Elastic IP
   ii. Create NAT Gateway
   iii. Edit Rout Table -> Select default subnet route
   iv. Edit (save) / Add new Route (0.0.0.0/0) and Target (NAT Gateway)

2. To Check the working of the NAT Gateway
   A. Login to Public subnet (putty)
   B. Transfer WebSec.pem to public subnet (WinSCP)
   C. chmod 400 WebSec.pem
   D. ssh -i "WebSec.pem" ec2-user@Private Server-IP
   E. ping google (8.8.8.8)

Custom VPC Deletion

- No Running instances (Under Custom VPC)
- Go to NAT Gateway –> Action ->Delete First
- Go to VPC -> Action -> Delete Custom VPC

Default VPC Deletion and Create again

- No Running instances (under Default VPC)
- Go to VPC -> Action -> Delete Default VPC and Create same

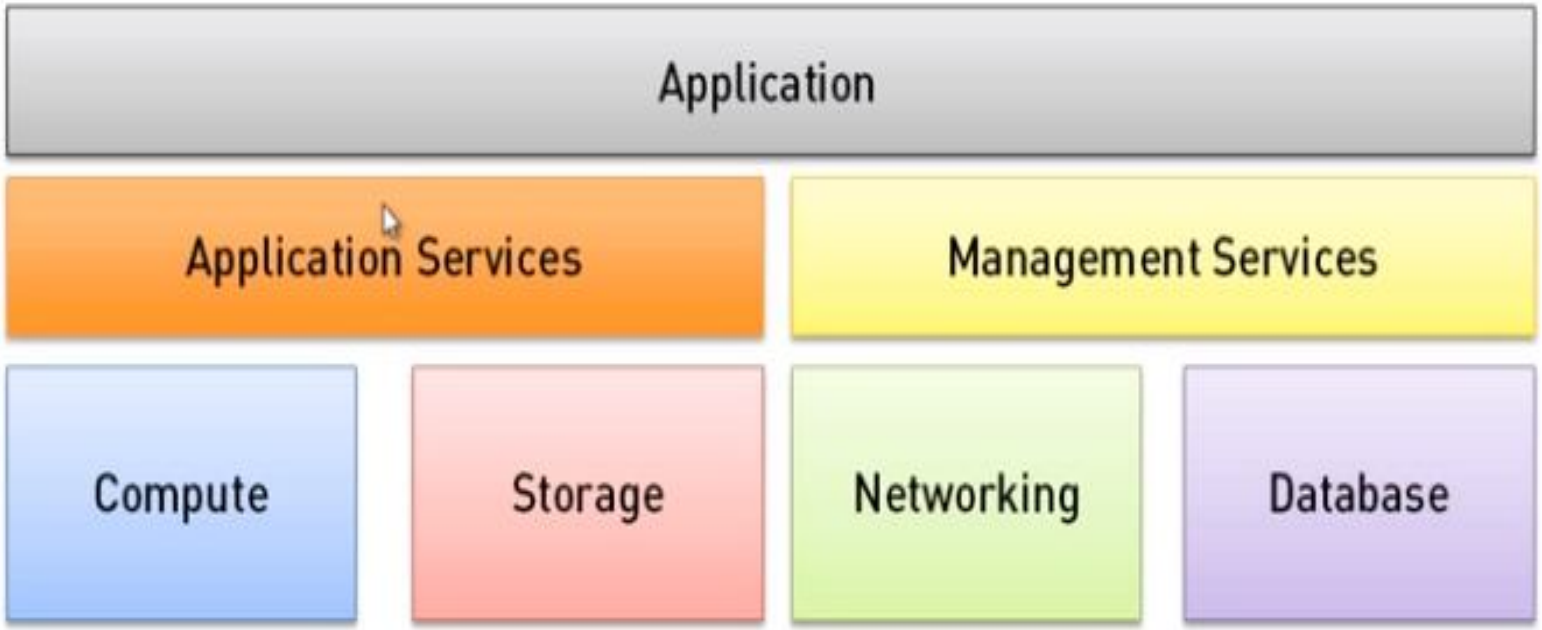
AWS-Relational Data Store

Tamal Dey
MCA,PESU

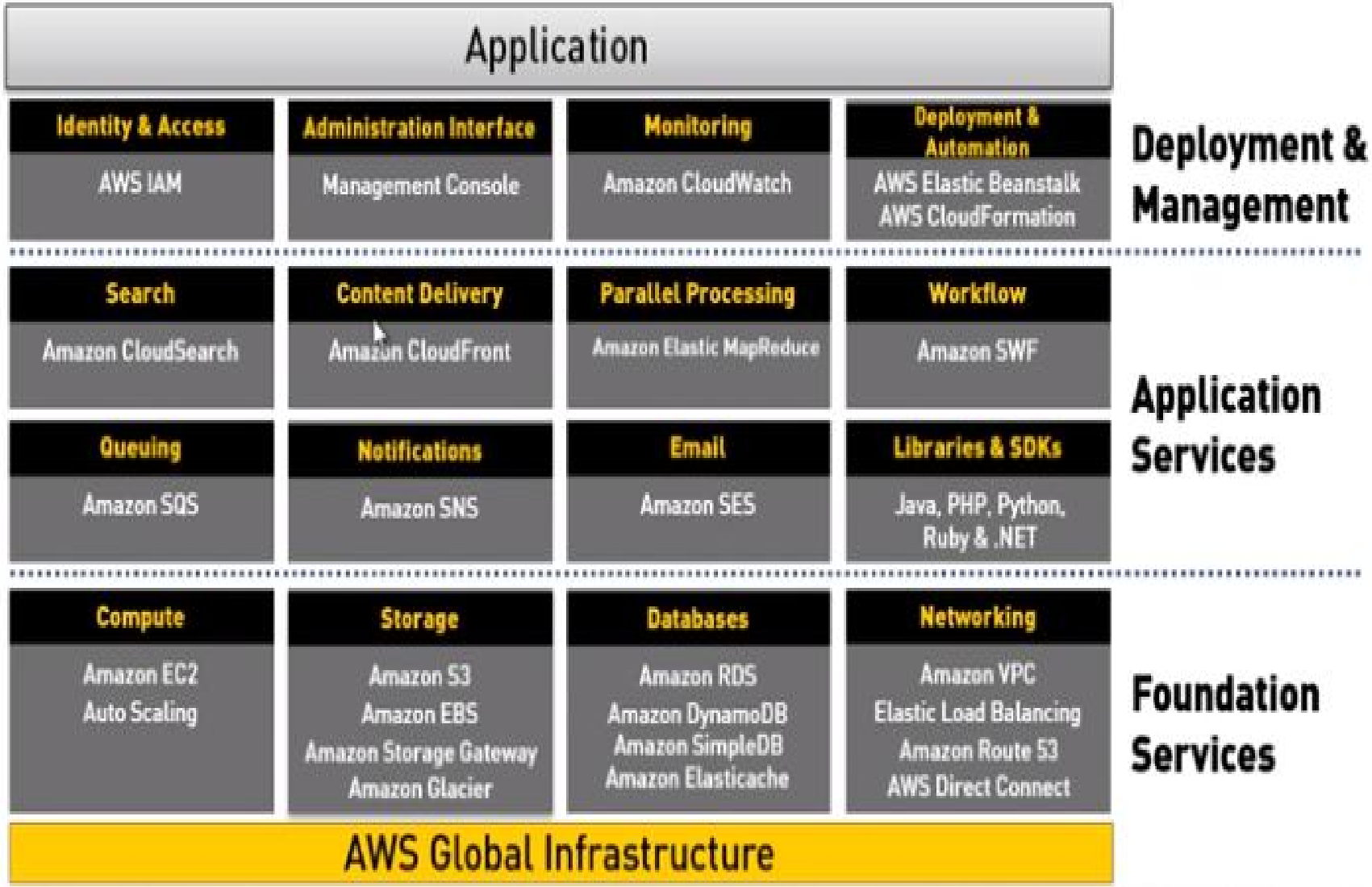
Agenda

- What powers the application?
- Dealing with databases on Cloud
- Key concepts of Amazon RDS
- Migrating the database to Amazon RDS

What Powers the application?



Big Picture of AWS



Quiz

- Can we Run a Database in EC2 instance?

YES

Then why Amazon Relational Database Service ????

Amazon will manage the DB

AWS-Database Service

RDS - Managed relational database in AWS cloud that you can launch in minutes with few clicks

Aurora - Fully managed MySQL compatible relational database with 5x performance and enterprise level features

DynamoDB - A managed NoSQL database offering extremely fast performance, elastic scalability and reliability

RedShift - Fast, fully managed, petabyte scale data warehouse at less than a tenth the cost of traditional solutions

Elasticache - Deploy, operate and scale in-memory cache in AWS cloud that supports Redis and memcached

Data Migration Service - Migrate your databases to cloud easily, inexpensively with zero downtime

RDS

- Setup, operate and scale relational databases
- Access to familiar databases such as MySQL, Oracle, MS SQL Server
- Support scale out for read-heavy database workloads on MySQL
- No upfront investment ; pay-as-you-go pricing
- **Facilities:** Manage patching, backup and recovery
- **Automatic backup**
  - Daily basis & Transaction Logs
  - Retention Period- Point in time recovery1-35 Days
- **DB Snapshots**
  - User initiated from AWS Mngt. Console or using API call
  - DB Snapshot Commnad

RDS Multi AZ

- **Production Version** (Parallel Environment)
  - , Disaster Recovery (**DR**), Continuation of Business (COB)
  - **RDS Read Replica** can be also given for primary DB Source
    - MySQL, MariaDB Postgres SQL
- ~~**DB Dev / Test**~~
- ~~**Free Tier**~~

RDS Encryption Resources

- Instances are encrypted by AES-256 algorithm Advanced Encryption Standard (**AES**)
  - Used by gamers

RDS – Benefits and features

- Easy to administer
- Available and durable
- Secure
- Higly scalable
- Fast
- Inexpensive

Use Cases

- Web and Mobile applications
- Mobile and Online Games
- E-Commerce Applications

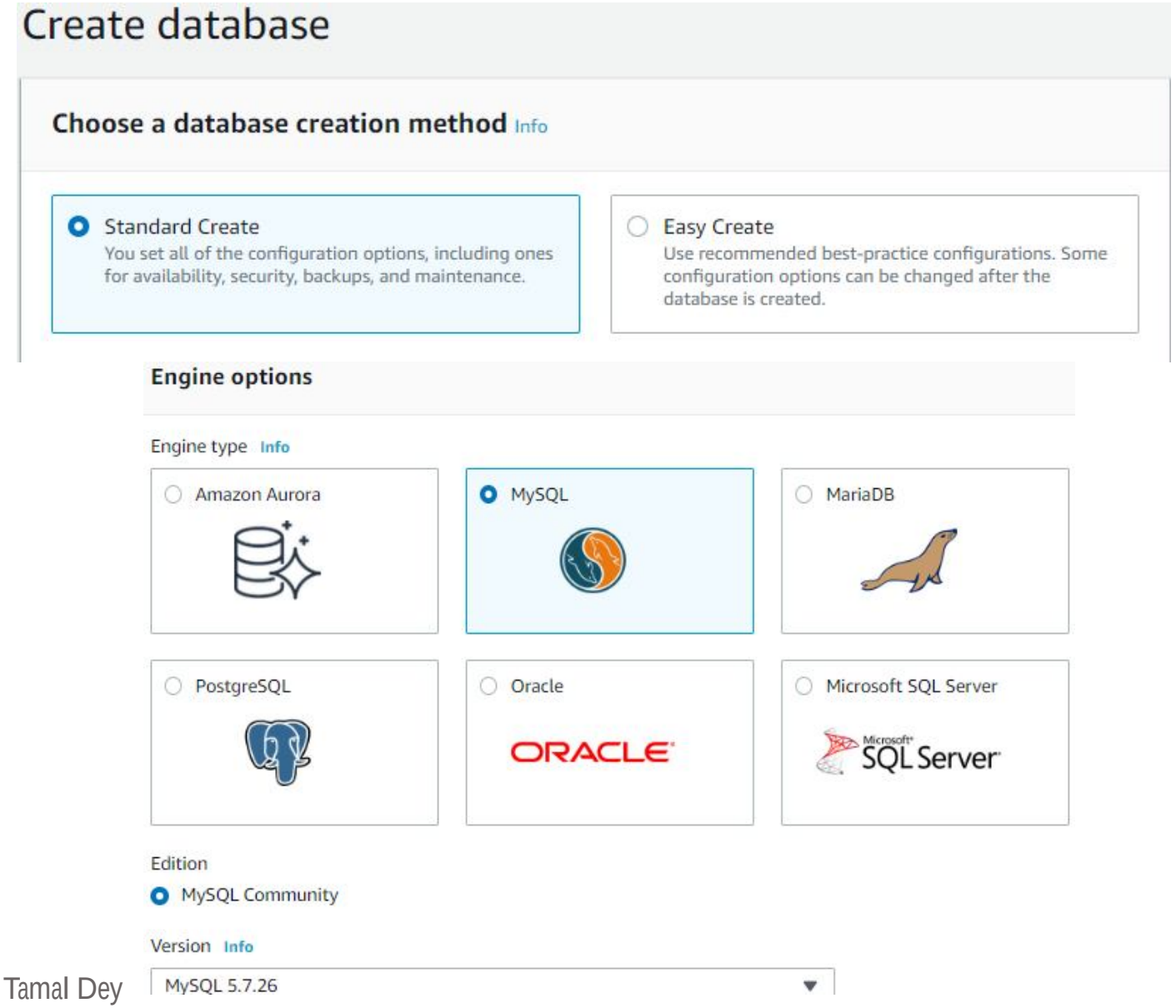
Getting started with Amazon-RDS

- Step 1 – Go to AWS console and Search RDS->Create Database
- Step 2 - Launch an Instance
- Step 3 – Authorize access
- Step 4 – Connect from the client

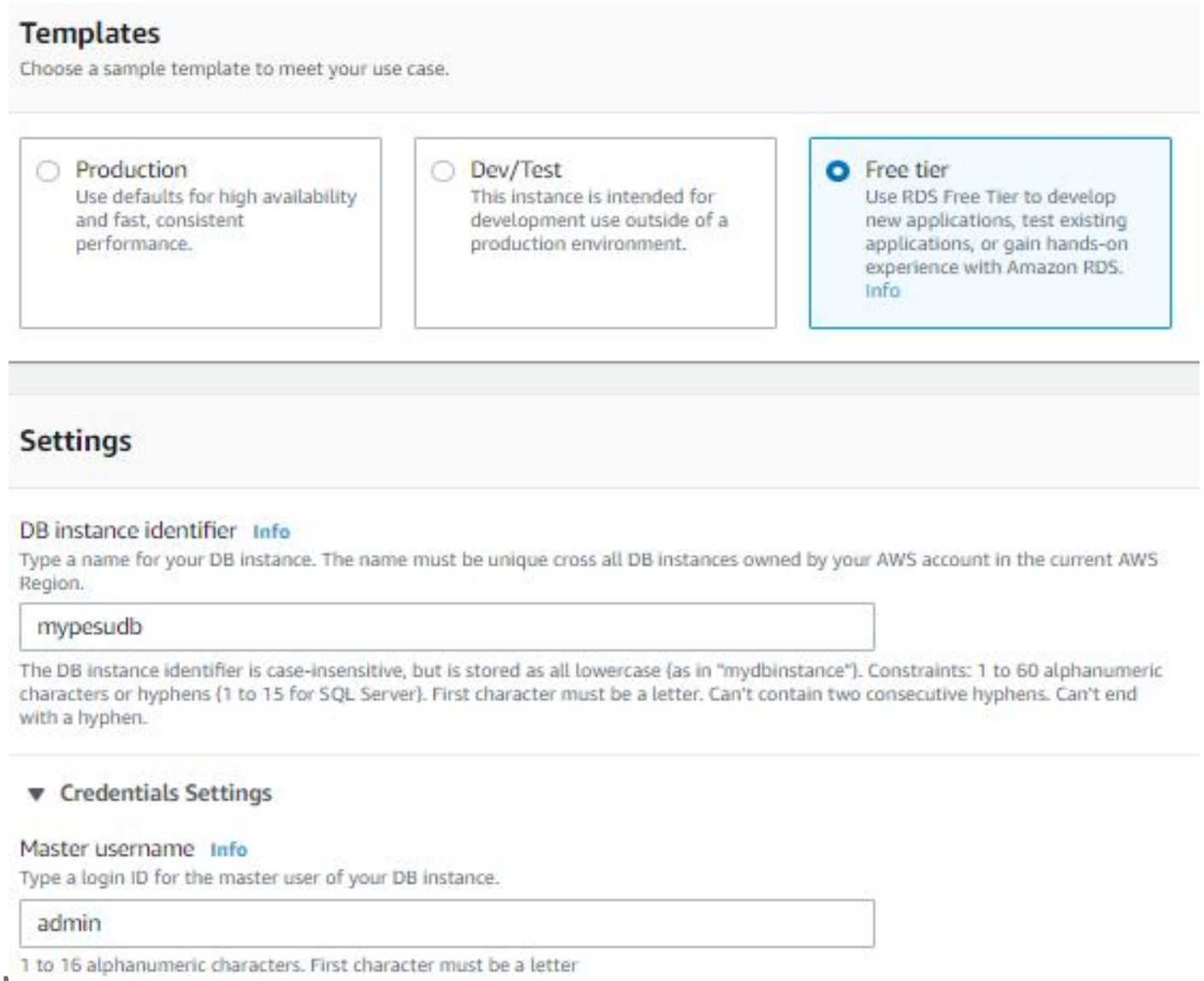
Getting started with Amazon-RDS

- Step 1 – Go to AWS console
- Step 2 - Launch an Instance
- Step 3 – Authorize access
- Step 4 – Connect from the client

Step 1: Select Engine



Step 2: Choose Use-Case



Step 3: Specify db details

- Db instance – **t2 micro**
- Multi-AZ – Yes / No as required
- Specify settings
  - Db instance identifier
  - Username and password

RDS Instance Types

Elastic Compute Unit (ECU)

Name	ECU	Cores per ECU	RAM
Micro	Up to 2	1	630MB
Small	1	1	1.7GB
Medium	2	1	3.75
Large	2	2	7.5GB
Extra Large	2	4	15GB
High-Memory Extra Large	3.25	2	17.1GB
High-Memory Double Extra Large	3.25	4	34GB
High-Memory Quadruple Extra Large	3.25	8	68GB

Connectivity

- Connectivity->Additional connectivity configuration
- Choose default VPC and Subnet Group
- Choose existing- **Default** Security Group (Refer Next Slide)
- Choose default availability zone and port number

- Note: **Publicly accessible (yes)**
  - Amazon EC2 instances and devices outside the VPC can connect to your database. Choose one or more VPC security groups that specify which EC2 instances and devices inside the VPC can connect to the database.

Connectivity

Virtual Private Cloud (VPC) Info

VPC that defines the virtual networking environment for this DB instance.

Default VPC (vpc-5e2e1536)

Only VPCs with a corresponding DB subnet group are listed.

After a database is created, you can't change the VPC selection.

▼ Additional connectivity configuration

Subnet group Info

DB subnet group that defines which subnets and IP ranges the DB instance can use in the VPC you selected.

default-vpc-5e2e1536

Publicly accessible Info

Yes

Amazon EC2 instances and devices outside the VPC can connect to your database. Choose one or more VPC security groups that specify which EC2 instances and devices inside the VPC can connect to the database.

Step 4: Configure advanced settings

- Choose retention period
- Choose public access required/not
- Choose sample db name

- Click launch the instance
- After launching wait until the endpoint is created. End point is like the DNS name in EC2.

Perform following Operations

- Take Snapshots
- Create Read Replica (Backup)
- Create Aura read replica
- Restore to point in time
- Delete the instances (Must)
- Migrate Snapshot
- Start /Stop /Reboot

Getting started with Amazon-RDS

- Step 1 – Go to AWS console
- Step 2 - Launch an Instance
- Step 3 – Authorize access
- Step 4 – Connect from the client

Authorize access

- Go to the **security group** and check for the inbound and outbound details
- In inbound -> add new rule if you want to connect through ec2 instance also (Protocol must be MySQL/Aurora) and CIDR/IP Address choose security group of your EC2 instance.
- Click "Save" to save your changes.
- In out bound check whether the all traffic is enabled

- In Amazon EC2-Linux-sudo yum update and sudo yum install mysql
- In Ubuntu EC2-sudo apt update and sudo apt install mysql-client

Connect from EC2-Linux/Ubuntu

- mysql -h <end point of DB instance> -u root/admin -p
- mysql -h mypesudb.cdbwue56chbe.ap-south-1.rds.amazonaws.com -u admin -p

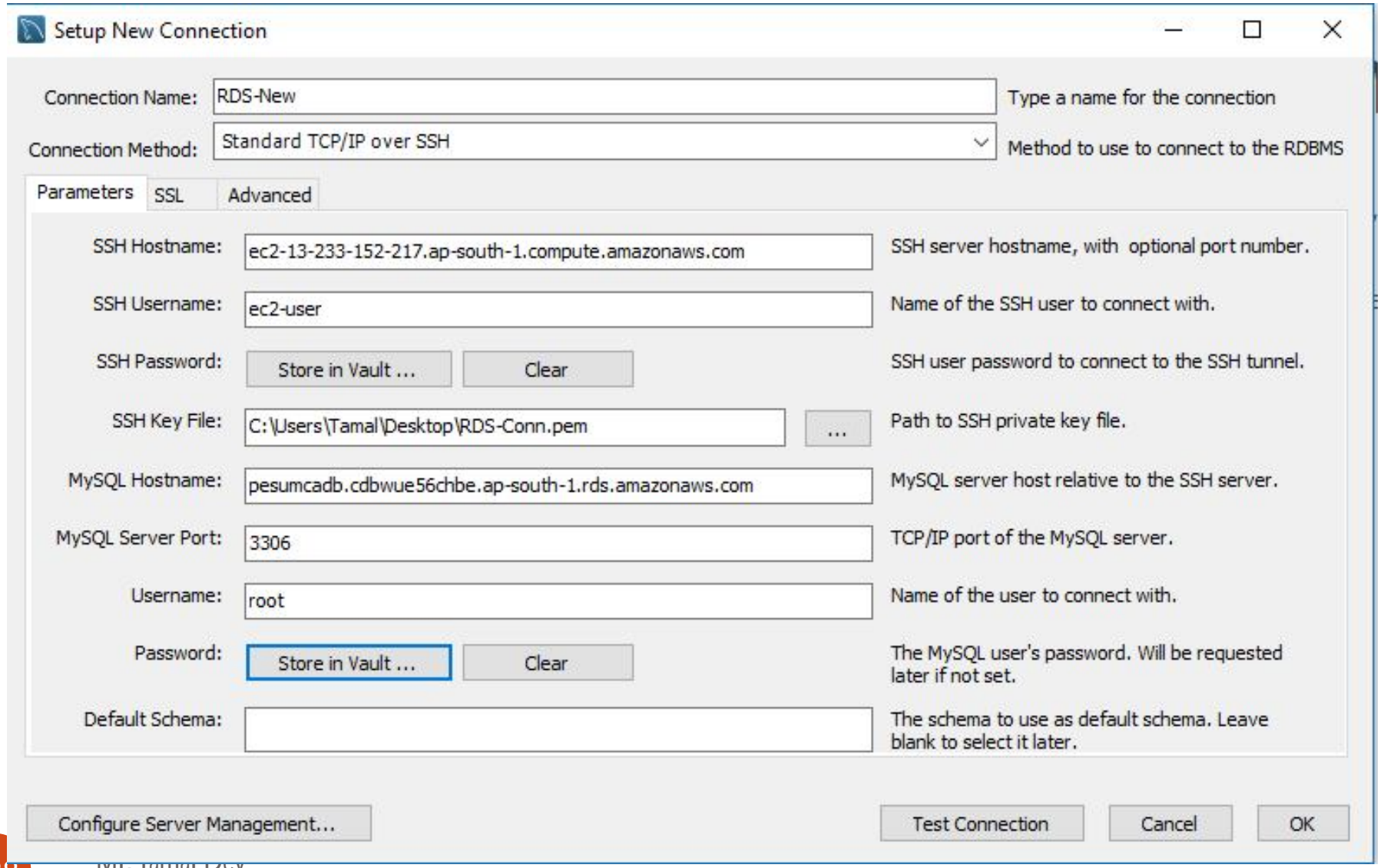
Getting started with Amazon-RDS

- Step 1 – Go to AWS console
- Step 2 - Launch an Instance
- Step 3 – Authorize access
- Step 4 – Connect from the client

Connect to the RDS

- Download and install mysql workbench (relevant for your operating system)
- Open mysql work bench
- Click on the + button to add a new mysql connection
  - Give a connection name of your choice
  - Connection method : **standard TCP / IP over SSH**
  - Fill in the values as described below which illustrated in next slide

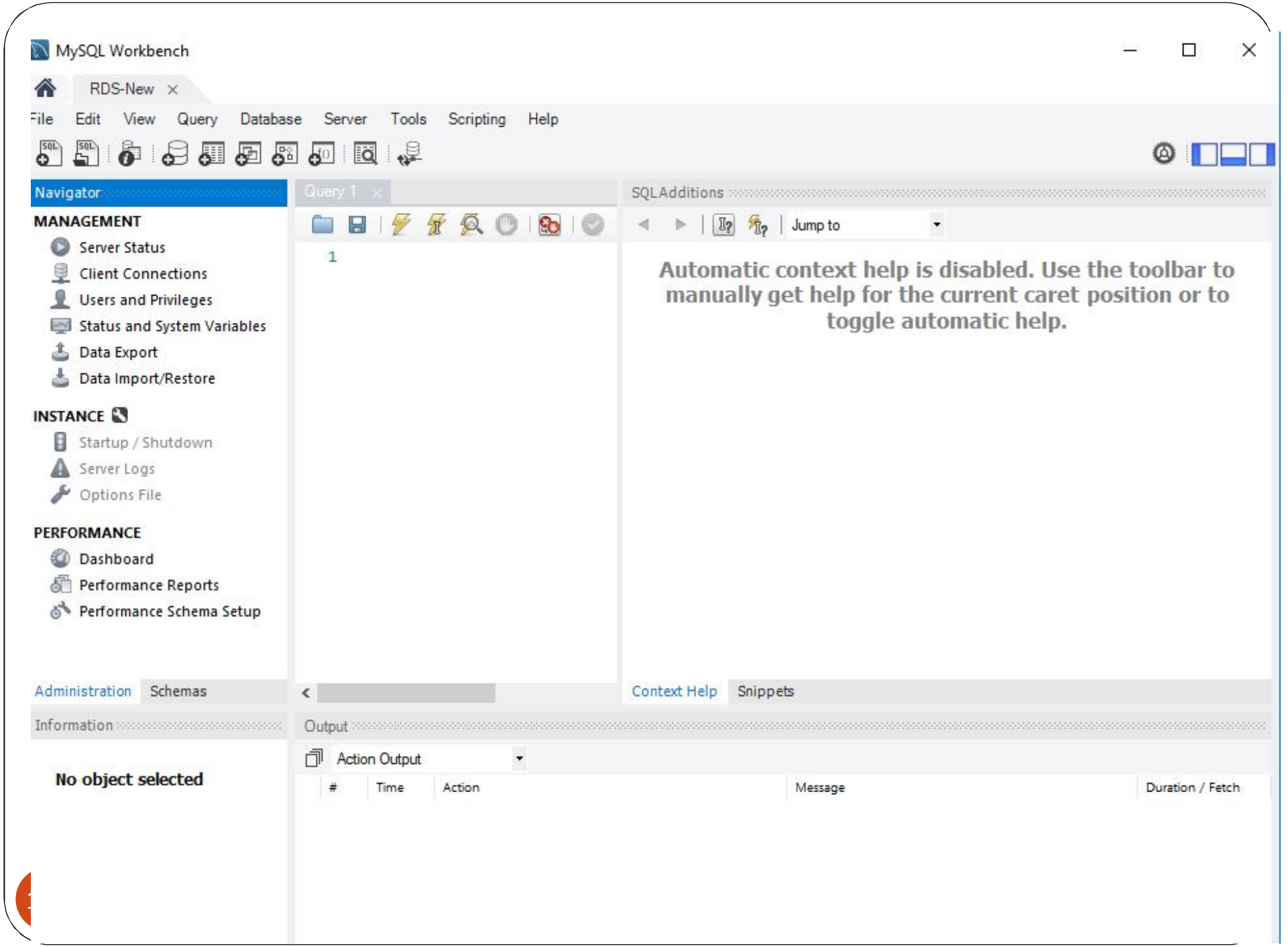
Choose TCP/IP over SSH



Explanation-Previous Slide

- **Connection Method** – Select Standard TCP/IP over SSH from the drop down list instead of Standard(TCP/IP). Then on the appearing two tabbed pane, fill the following values found under Parameters tab.
- **SSH Hostname** – Provide the Public DNS of the Amazon EC2 instace (refer Figure 3) which will be used as the intermediate server instance used to create the connection with the DB instance.
- **SSH Username** – Provide the user of the Amazon EC2 instance (refer Figure 3) which will be used as the intermediate server instance used to create the connection with the DB instance.
- **SSH Key File** – Provide the Private Key (xxx.pem) used to connect Amazon EC2 instace via SSH
- **MySQL Hostname** – Provide the Endpoint of the DB instance created in Amazon RDS(refer Figure 5).
- **Username** – Provide the Master Username of the DB instance created in Amazon RDS(refer Figure 5). Connection Name – Provide a name to identify your connection
- **Password** – Click on Store in Keyvault. button and type the password provided while creating the DB instance in Amazon RDS.

Check you connection. Wait for some time to get connected.


MySQL Workbench
RDS-New
File Edit View Query Database Server Tools Scripting Help
Navigator
MANAGEMENT
Server Status
Client Connections
Users and Privileges
Status and System Variables
Data Export
Data Import/Restore
INSTANCE
Startup / Shutdown
Server Logs
Options File
PERFORMANCE
Dashboard
Performance Reports
Performance Schema Setup
Administration
Schemas
Information
No object selected
Query 1
1
SQLAdditions
Jump to
Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.
Context Help
Snippets
Output
Action Output
Time Action Message Duration / Fetch

For Ubuntu users

- **sudo apt install mysql-workbench**
- Click on the + button to add a new mysql connection
  - Give a connection name of your choice
  - Connection method : standard TCP/IP over SSH
  - Hostname: EC2 instance name
  - Username: ubuntu password: <leave empty>
  - Browse for key file and upload it
  - Hostname: End point of RDS
  - Username: as what you have specified in the db instance
  - Password: as what you have specified in the db instance
  - Click ok.
  - Check you connection. Wait for some time to get connected.

Management & Governance

Amazon Cloud Watch

By Tamal Dey
Assistant Professor,
Dept. of MCA, PESU

AWS CloudWatch

Amazon CloudWatch is a monitoring service for AWS cloud resources and the applications you run on AWS.

What is CloudWatch?

- Amazon CloudWatch monitors operational and performance metrics for your AWS cloud resources and applications.
- You can create your own dashboards of which metric your want to measure
- A **metric** represents a time-ordered set of data points that are published to **CloudWatch**.
- A **metric** as a variable to monitor, and the data points as representing the values of that variable over time.
  - For example, the CPU usage of a particular EC2 instance is one **metric** provided by Amazon EC2.
- You can also set alarms and automate actions to manage your EC2, RDS and S3.

Monitoring

Amazon
CloudWatch

Basic monitoring
Is free
Polls every 5 minutes
10 metrics
5GB of data ingestion
5GB of data storage

Detailed monitoring
Is chargeable
Charged per instance per month
Polls every minute

Dashboards

Metrics

AWS CloudWatch allows you to record metrics for EBS, EC2, ELB, and S3.

Create Dashboard

- Provide a name to your dashboard
- Choose a widget type by which the reports should be available on the dashboard
- Now choose to which resource you want to monitor
  - Give instance -id in search box
- Now create a widget of any type by specifying the performance measure you want to watch
- You can add as many as widgets for the same resource

Alarms

Set alarms to warn based on resources usage, for example CPU utilization is too high.

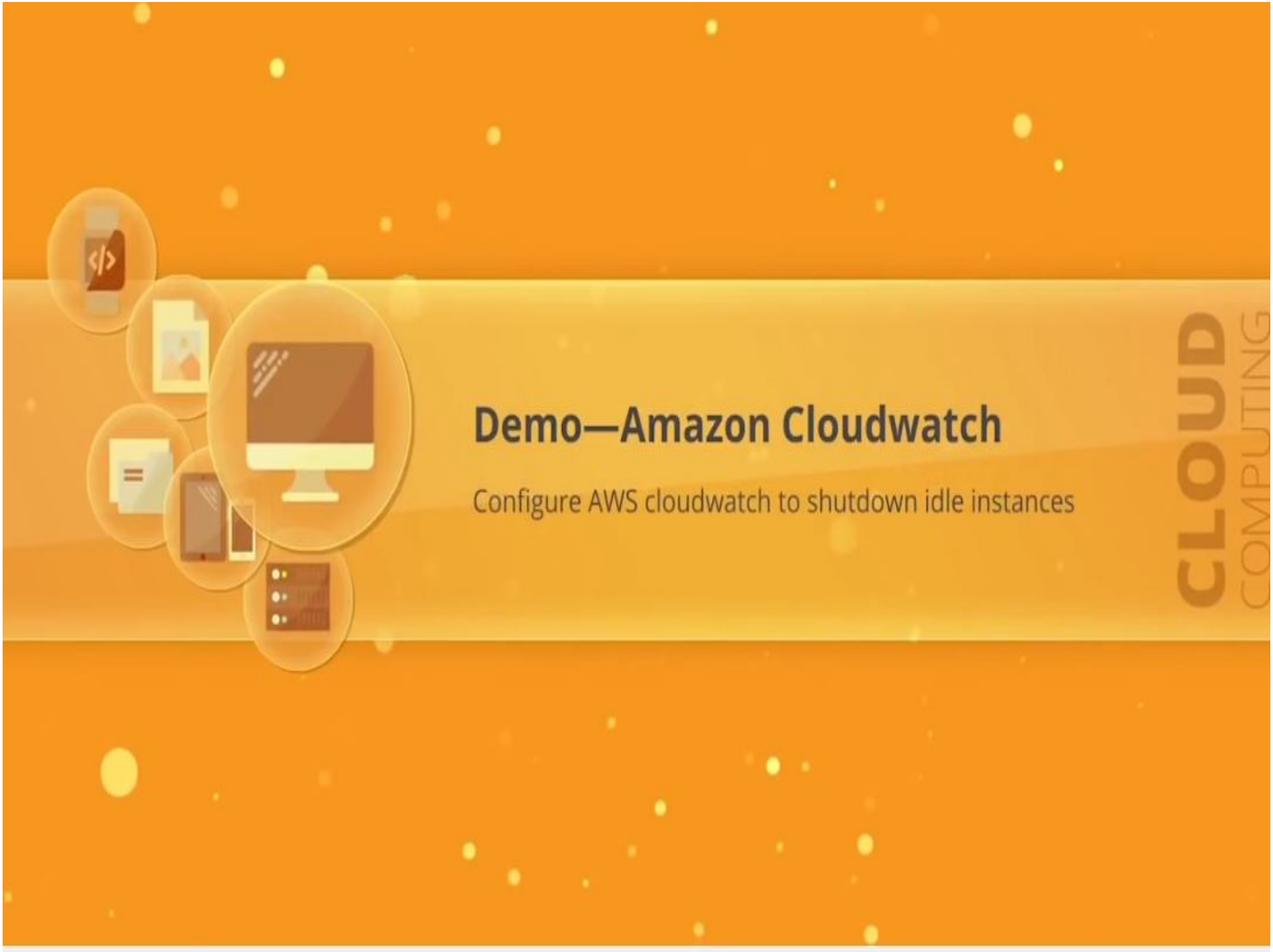
- Shutdown / Terminate / Add more instances to share the Load

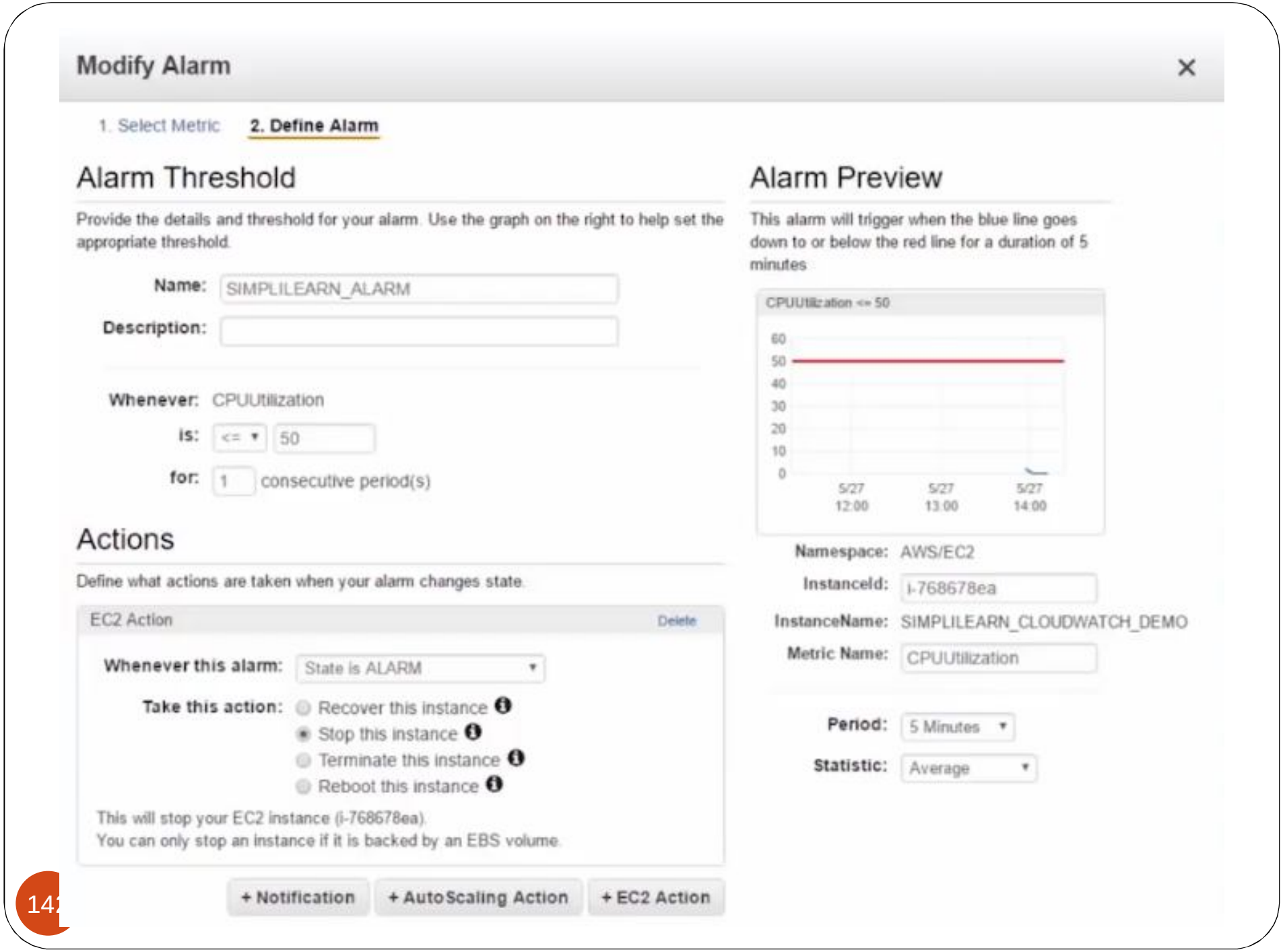
Create alarms

- You can create alarms and automate certain actions in the resources
- Give the alarm the name and the description
- Select the resource you want to trigger an alarm
- Set the threshold
- Specify the action.

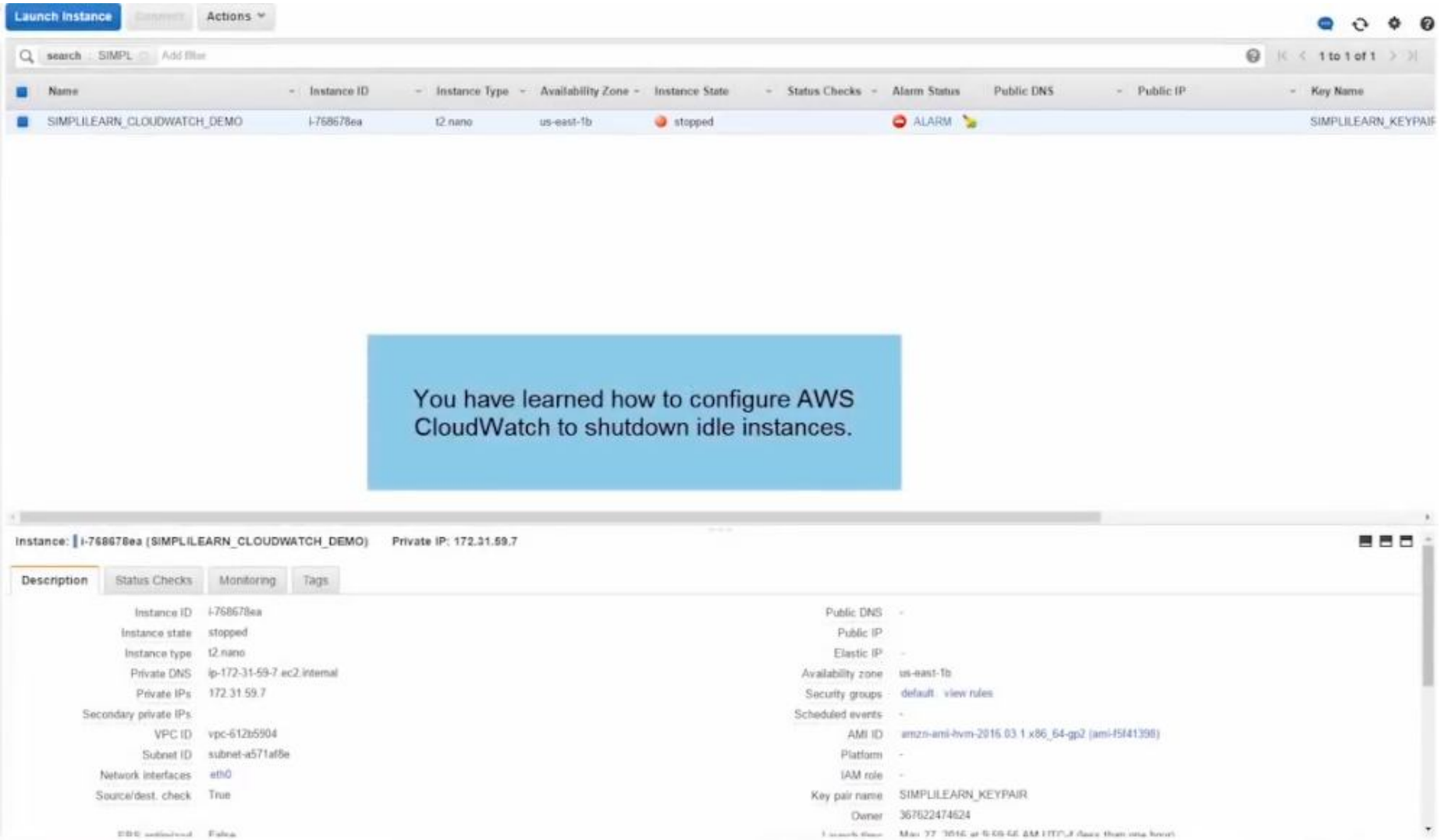
Logs

Install agents on EC2 instances to send monitoring data about the instance to CloudWatch.


Demo—Amazon Cloudwatch
Configure AWS cloudwatch to shutdown idle instances
CLOUD
COMPUTING


Modify Alarm
1. Select Metric
2. Define Alarm
Alarm Threshold
Provide the details and threshold for your alarm. Use the graph on the right to help set the appropriate threshold.
Name: SIMPLILEARN_ALARM
Description:
Whenever: CPUUtilization
is: <= 50
for: 1 consecutive period(s)
Actions
Define what actions are taken when your alarm changes state.
EC2 Action
Delete
Whenever this alarm: State is ALARM
Take this action:
Recover this instance
Stop this instance
Terminate this instance
Reboot this instance
This will stop your EC2 instance (i-768678ea).
You can only stop an instance if it is backed by an EBS volume.
+ Notification
+ AutoScaling Action
+ EC2 Action
Alarm Preview
This alarm will trigger when the blue line goes down to or below the red line for a duration of 5 minutes
CPUUtilization <= 50
60
50
40
30
20
10
0
5/27 12:00
5/27 13:00
5/27 14:00
Namespace: AWS/EC2
InstanceId: i-768678ea
InstanceName: SIMPLILEARN_CLOUDWATCH_DEMO
Metric Name: CPUUtilization
Period: 5 Minutes
Statistic: Average

Alarm Executed



Video Tutorial

- https://www.youtube.com/watch?v=_Tqce6pGb44