CS 4037
Introduction to Cloud Computing
Lecture 26.1

Danyal Farhat

FAST School of Computing

NUCES Lahore

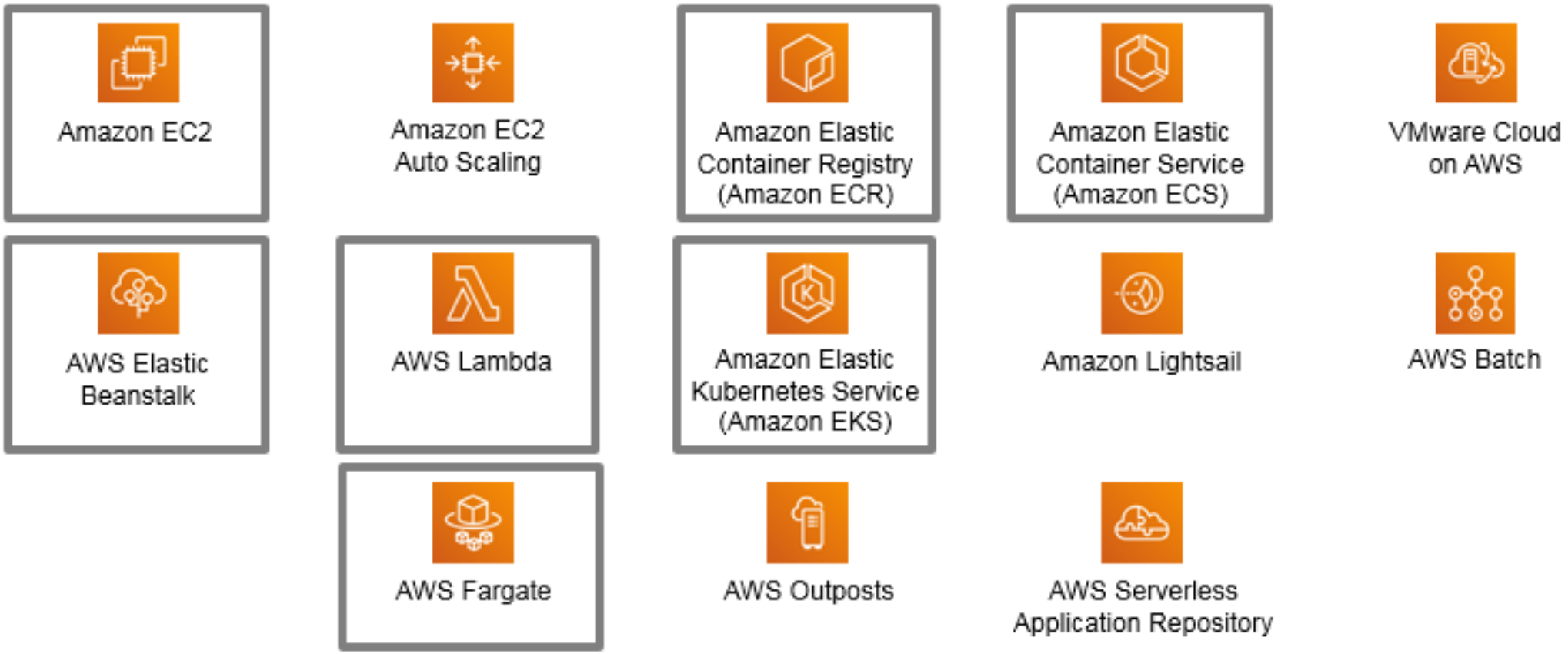
AWS Compute – Part 1

Lecture's Agenda

- **Compute Services Overview**
- Amazon EC2
- EC2 Cost Optimization
- Container Services
- Introduction to AWS Lambda
- Introduction to AWS Elastic Beanstalk

AWS Compute Services

- **AWS offers many compute services. This module will discuss the highlighted services.**



Categorizing Compute Services

Services	Key Concepts	Characteristics	Ease of Use
• Amazon EC2	• Infrastructure as a service (IaaS) • Instance-based • **Virtual machines**	• Provision virtual machines that you can manage as you choose	A familiar concept to many IT professionals.
• AWS Lambda	• **Serverless** computing • Function-based • Low-cost	• Write and deploy code that runs on a schedule or that can be triggered by events • Use when possible (architect for the cloud)	A relatively new concept for many IT staff members, but easy to use after you learn how.
• Amazon ECS • Amazon EKS • AWS Fargate • Amazon ECR	• **Container-based** computing • Run multiple workloads on single OS • Instance-based	• Spin up and run jobs more quickly	AWS Fargate reduces administrative overhead, but you can use options that give you more control.
• AWS Elastic Beanstalk	• Platform as a service (PaaS) • For **web applications**	• Focus on your code (building your application) • Can easily tie into other services—databases, Domain Name System (DNS), etc.	Fast and easy to get started.

Choosing the Optimal Compute Service

- **The optimal compute service depends on customer's use case:**
  - **What is your application design?**
  - **What are your usage patterns?**
  - **Which configuration settings will you want to manage?**

- **Selecting the wrong compute solution for an architecture can lead to performance issues or cost increments**
- **A good starting place: Understand the available compute options**

Lecture's Agenda

- Compute Services Overview
- **Amazon EC2**
- EC2 Cost Optimization
- Container Services
- Introduction to AWS Lambda
- Introduction to AWS Elastic Beanstalk

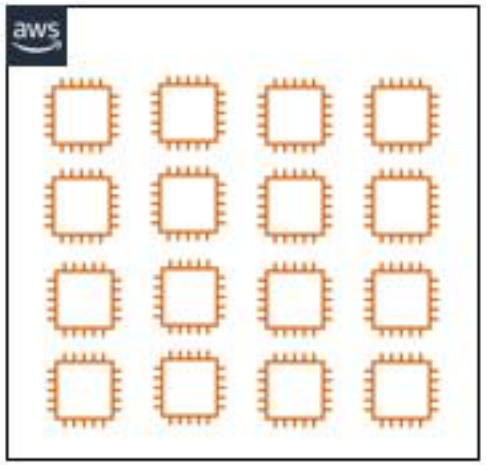
Amazon Elastic Compute Cloud (Amazon EC2)

Photo by Taylor Vick on Unsplash

On-premises servers

Example uses of Amazon EC2 instances

- ✓ Application server
- ✓ Web server
- ✓ Database server
- ✓ Game server
- ✓ Mail server
- ✓ Media server
- ✓ Catalog server
- ✓ File server
- ✓ Computing server
- ✓ Proxy server



Amazon EC2 instances

Photo by panumas nikhomkhai from Pexels

Amazon EC2 Overview

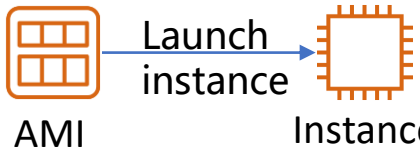
- **Provides virtual machines—referred to as EC2 instances**
- **Gives full control over the guest operating system**
  - **Windows, Linux, Ubuntu, RedHat, CentOS etc.**
- **Can launch instances of any size into an Availability Zone anywhere in the world**
- **Launch instances from Amazon Machine Images (AMIs).**
  - **Launch instances with a few clicks or a line of code, and they are ready in minutes**
- **You can control traffic to and from instances using SGs.**

Launching an EC2 Instance

- **9 key steps are involved in launching an EC2 instance.**
  - **Select an AMI**
  - **Select an Instance Type**
  - **Specify Network Settings**
  - **Attach IAM Role (optional)**
  - **Add User Data Script (optional)**
  - **Specify Storage**
  - **Add Tags**
  - **Security Group Settings**
  - **Identify or Create the Key Pair**

1. Select an AMI

- **"Amazon Machine Image (AMI) is a template used to create an EC2 instance"**
  - **Contains a Windows or Linux operating system**
  - **May also contain pre-installed software like SQL server, Visual Studio etc.**

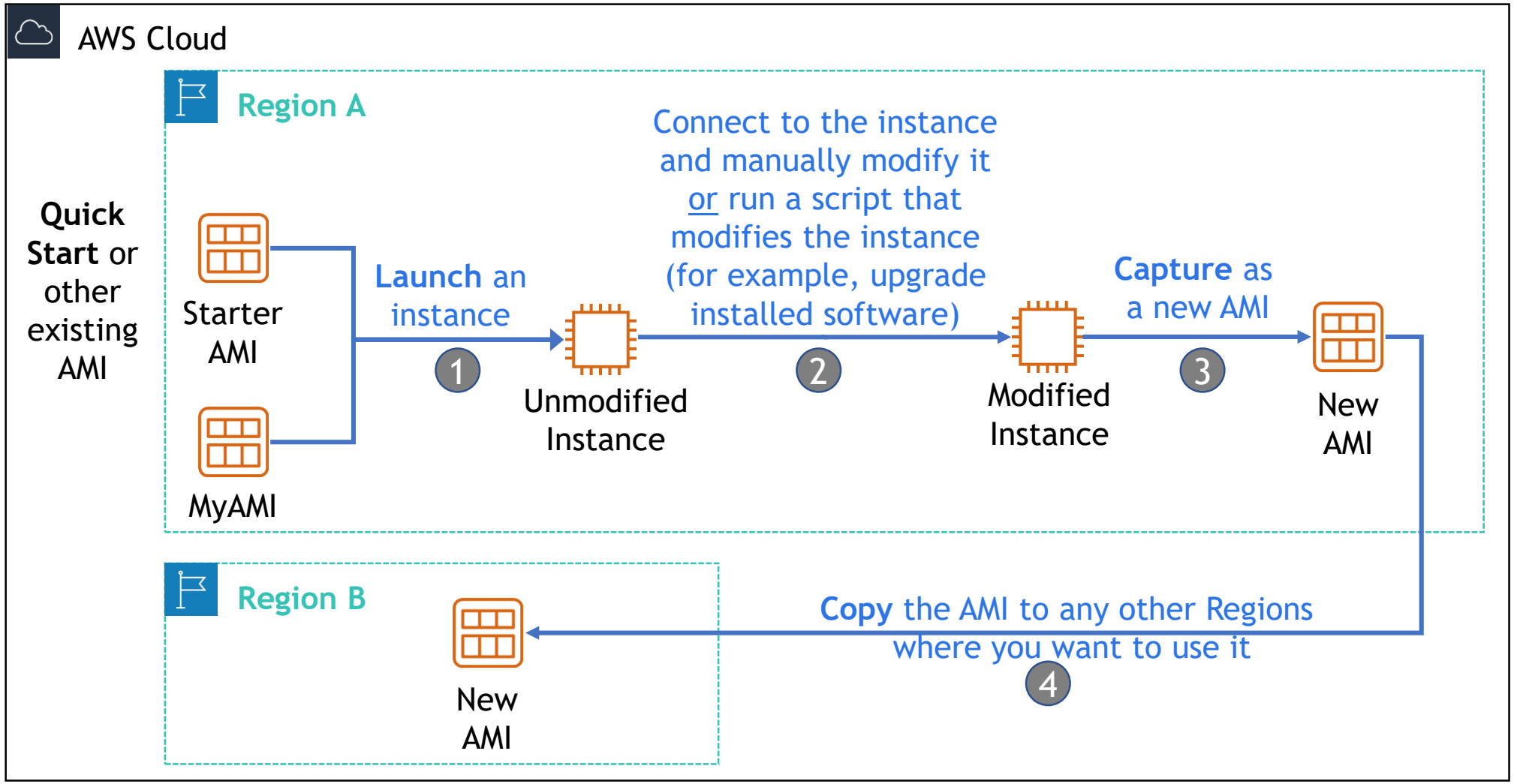


- **AMI choices:**
  - **Quick Start – Linux and Windows AMIs that are provided by AWS**
  - **My AMIs – Any AMIs that you created**
  - **AWS Marketplace – Pre-configured templates from third parties**
  - **Community AMIs – AMIs shared by others; use at your own risk**

Creating a New AMI: Example

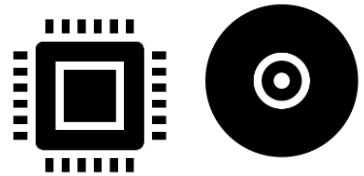
AMI details

(Optional) Import a virtual machine



2. Select an Instance Type

- **Instance type determines:**
  - **Memory (RAM)**
  - **Processing power (CPU)**
  - **Disk space and disk type (Storage)**
  - **Network performance**
- **Instance type categories:**
  - **General purpose**
  - **Compute optimized**
  - **Memory optimized**
  - **Storage optimized**
  - **Accelerated computing**
- **Instance types offer family, generation, and size.**



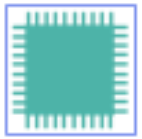
EC2 Instance Type: Naming and Sizes

Instance Type Naming

- **Example: t3.large**
  - **t is the family name**
  - **3 is the generation number**
  - **Large is the size**
  - **Network bandwidth is tied with instance type**

Instance Name	vCPU	Memory (GB)	Storage
t3.nano	2	0.5	EBS-Only
t3.micro	2	1	EBS-Only
t3.small	2	2	EBS-Only
t3.medium	2	4	EBS-Only
t3.large	2	8	EBS-Only
t3.xlarge	4	16	EBS-Only
t3.2xlarge	8	32	EBS-Only

Select Instance Type: Based on Use Case

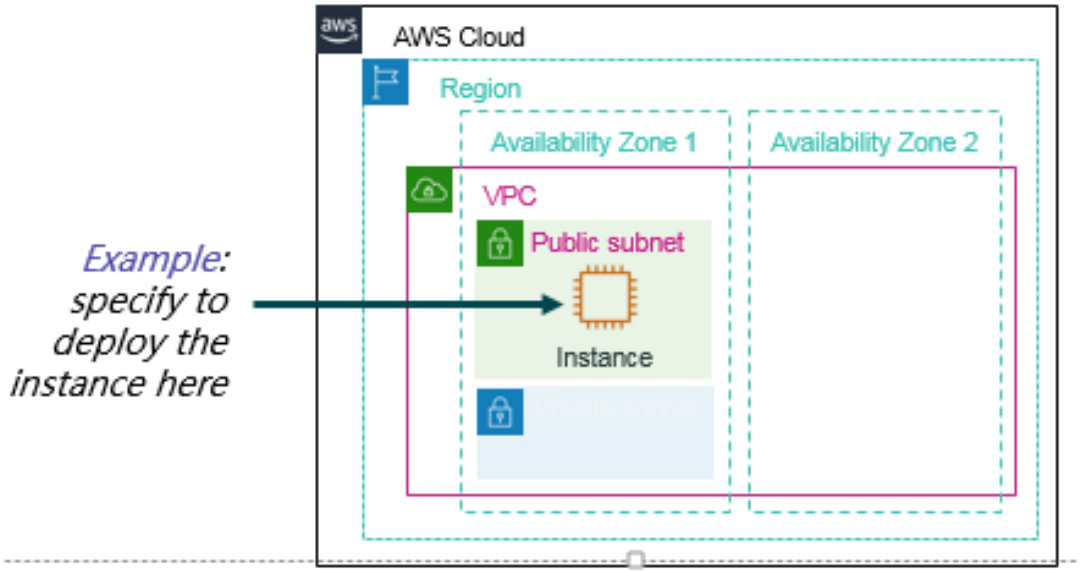
	 General Purpose	 Compute Optimized	 Memory Optimized	 Accelerated Computing	 Storage Optimized
Instance Types	a1, m4, m5, t2, t3	c4, c5	r4, r5, x1, z1	f1, g3, g4, p2, p3	d2, h1, i3
Use Case	Broad	High performance	In-memory databases	Machine learning	Distributed file systems

Instance Type: Networking Features

- **Network bandwidth (Gbps) varies by instance type**
- **To maximize networking performance of instance type:**
  - **In case of interdependent instances, launch them into the same cluster placement group**
  - **Enable enhanced networking**
- **Enhanced networking types are supported on most instance types**
- **Enhanced networking types:**
  - **Elastic Network Adapter (ENA): Supports network speeds of up to 100 Gbps**
  - **Intel 82599 Virtual Function interface: Supports network speeds of up to 10 Gbps**

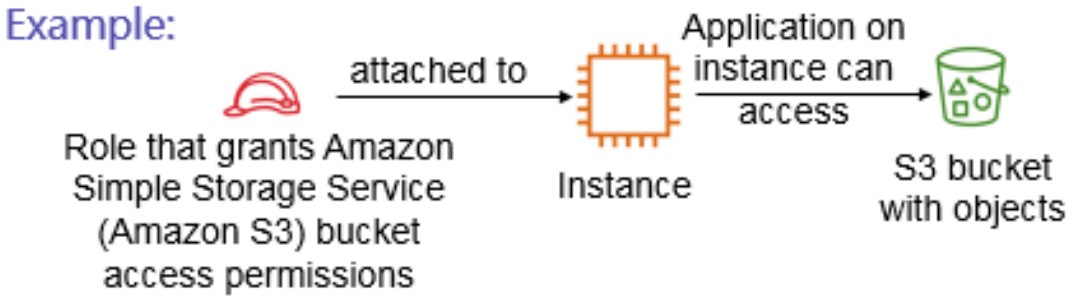
3. Specify Network Settings

- **Where should the instance be deployed?**
  - **Identify the VPC and optionally the subnet**
- **Should a public IP address be automatically assigned?**
  - **To make it internet-accessible**



4. Attach IAM Role (Optional)

- **Will software on the EC2 instance need to interact with other AWS services?**
  - **If yes, attach an appropriate IAM Role**
- **An AWS IAM role that is attached to an EC2 instance is kept in an instance profile**
- **You are not restricted to attaching a role only at instance launch**
  - **You can also attach a role to an instance that already exists**

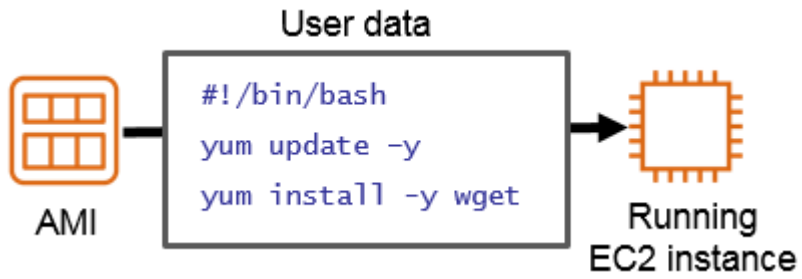


5. User Data Script (Optional)

- **User have the option to pass user data to the instance**
  - **By specifying a user data script at instance launch**
- **User data scripts are used to customize the runtime environment of the instance**
  - **Script runs the first time the instance starts**
  - **User data script might patch and update the instance's operating system, fetch and install software license keys, or install additional software**
- **Can be used strategically**
  - **For example, reduce the number of custom AMIs that you build and maintain**

5. User Data Script (Optional) [Cont.]

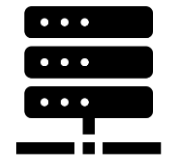
- **Linux Bash Shell Script**



- **#!/bin/bash**
  - **bash shell should be used**
- **yum update –y**
  - **update all softwares to the latest version**
- **yum install -y wget**
  - **Wget utility should be installed**
  - **wget is a common utility for downloading files from web**

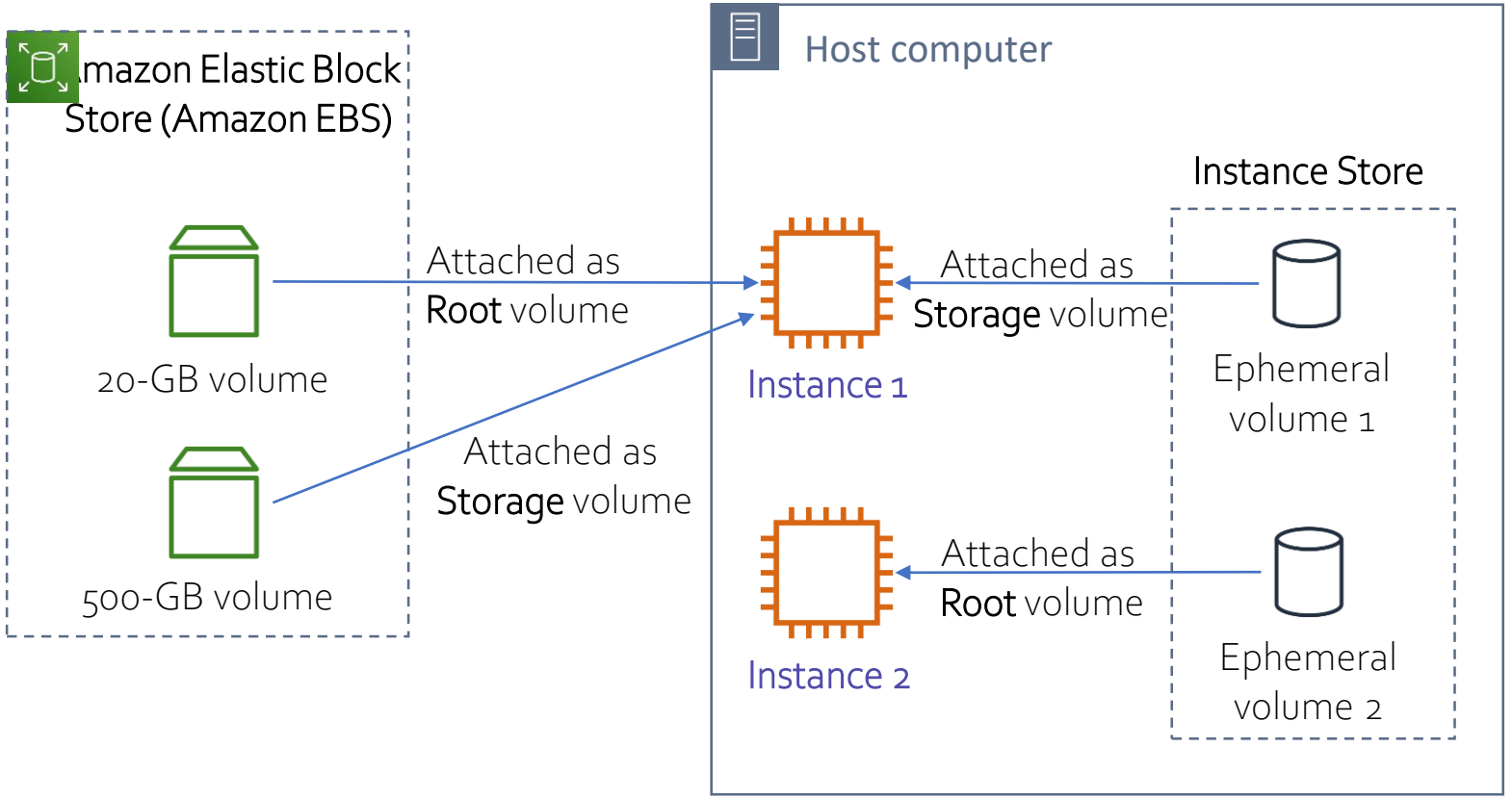
6. Specify Storage

- **Configure the root volume**
  - ➢ Where the guest operating system is installed
- **Attach additional storage volumes (optional)**
  - ➢ AMI might already include more than one volume
- **For each volume, specify:**
  - ➢ The size of the disk (in GB)
  - ➢ The volume type
    - ✓ Different types of solid state drives (SSDs) and hard disk drives (HDDs) are available
- **If the volume will be deleted when the instance is terminated, and If encryption should be used**



Example Storage Options

- **Instance 1** characteristics –
  - It has an **Amazon EBS** *root volume* type for the operating system.
  - What will happen if the instance is stopped and then started again?
- **Instance 2** characteristics –
  - It has an **Instance Store** *root volume* type for the operating system.
  - What will happen if the instance stops (because of user error or a system malfunction)?

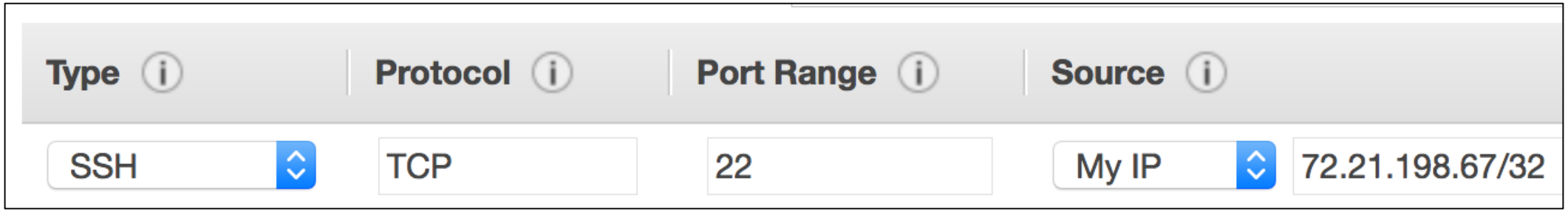


7. Add Tags

- **Tag is a label that user can assign to an AWS resource**
  - **Consists of a key and an optional value. Case sensitive tags**
- **Tagging is how user attach metadata to an EC2 instance**
- **Potential benefits of tagging: filtering, automation, cost allocation, and access control.**
- **Example:**

8. Security Group Settings

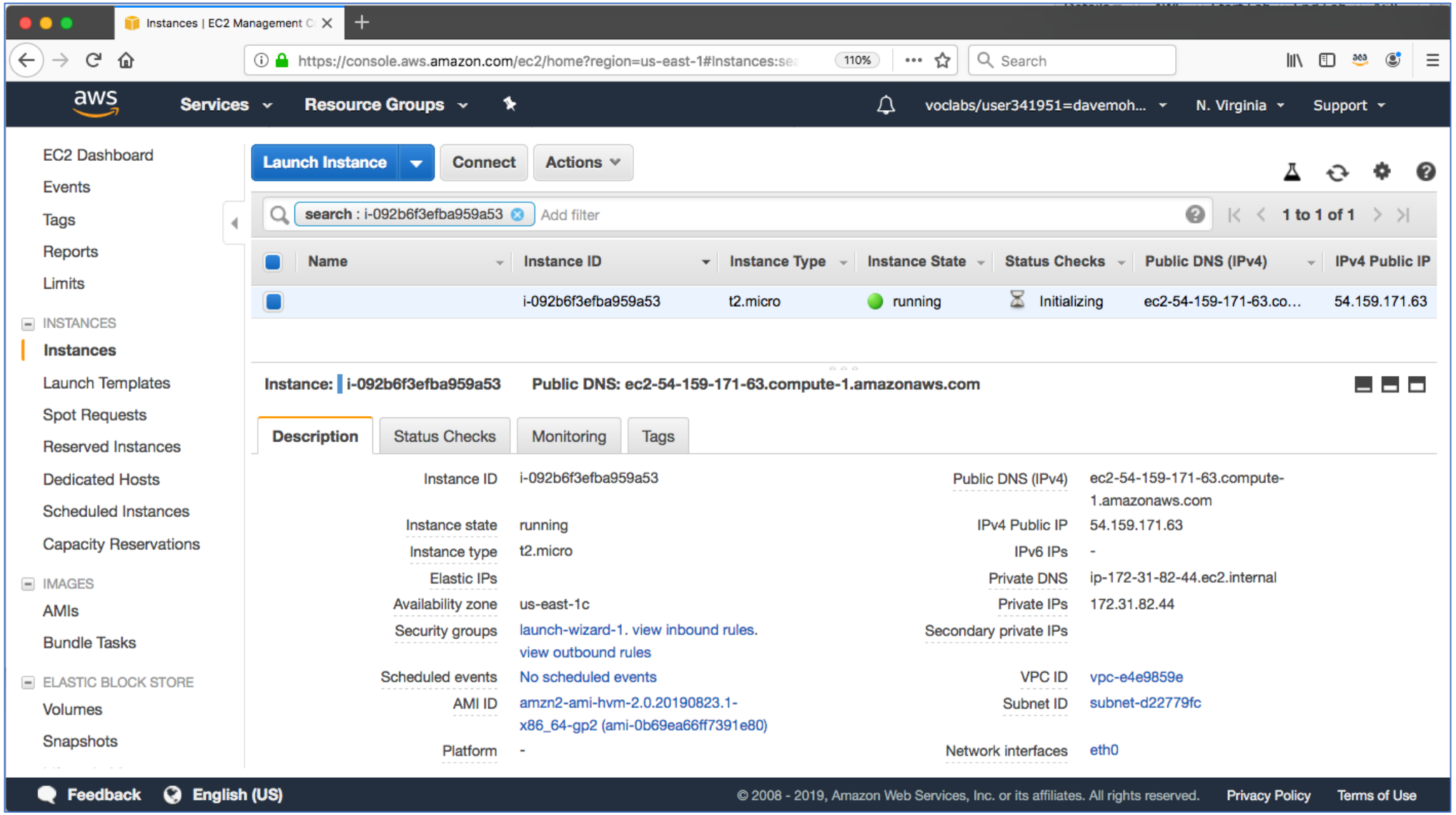
- **A security group is a set of firewall rules that control traffic to the instance**
  - **It exists outside of the instance's guest OS**
- **Create rules that specify the source and which ports that network communications can use**



9. Identify or Create the Key pair

- **At instance launch, customer specify an existing key pair or create a new key pair**
- **A key pair consists of:**
  - **A public key that AWS stores**
  - **A private key file that user store**
- **It enables secure connections to the instance**
- **For Windows AMIs**
  - **Use private key to obtain administrator password that user need to log in to the instance.**
- **For Linux AMIs**
  - **Use the private key to use SSH to securely connect to the instance.**

Console View of a Running EC2 Instance



Another Option: Launch an EC2 instance with AWS CLI

- **Can also be created programmatically**
  - **Using AWS CLI or any of the available SDK**

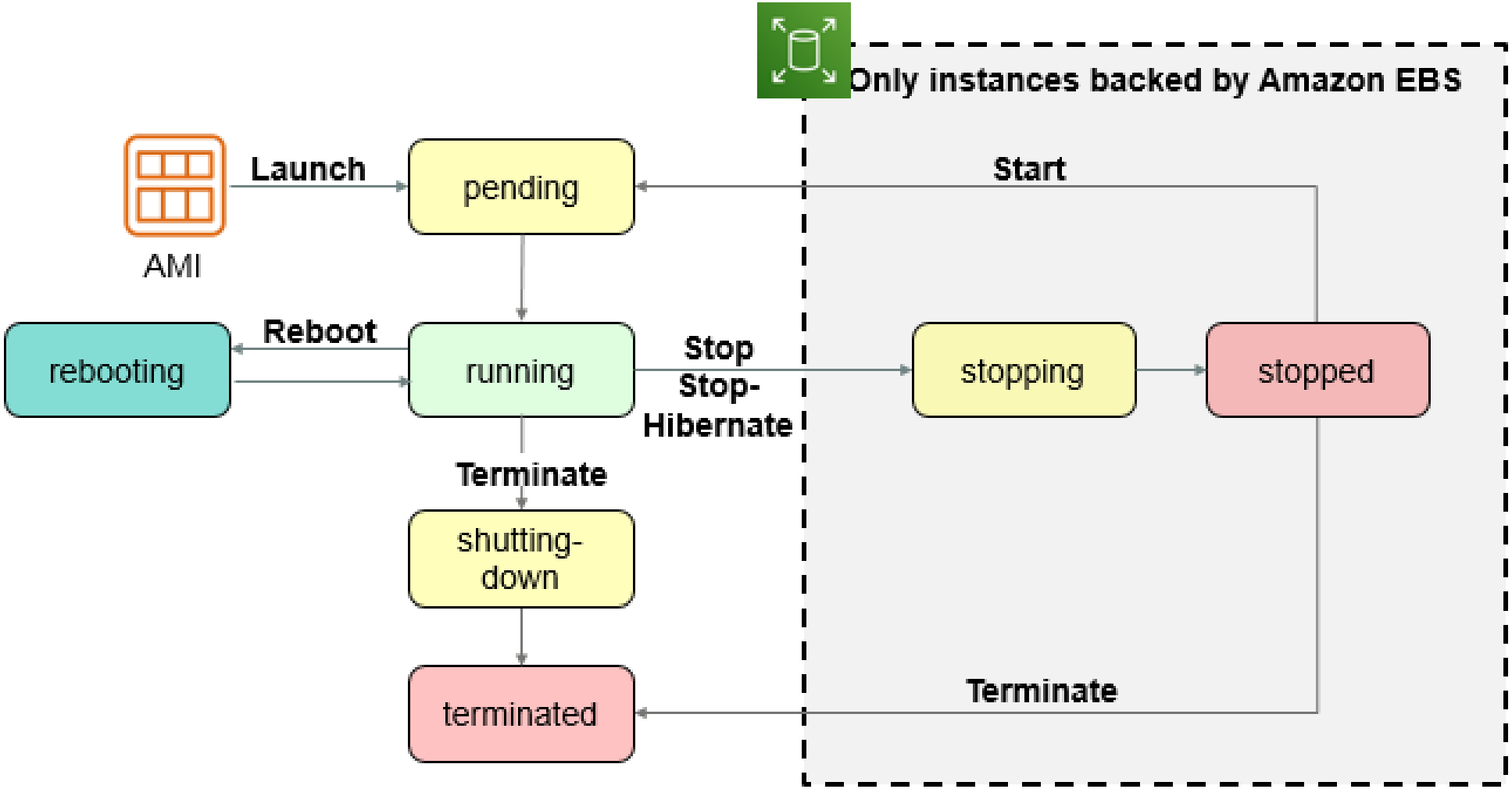
AWS Command Line Interface (AWS CLI)

- **Example shows how simple the command can be**
  - **This command assumes that the key pair and security group already exist**

Example command:

```
aws ec2 run-instances \
--image-id ami-1a2b3c4d \
--count 1 \
--instance-type c3.large \
--key-name MyKeyPair \
--security-groups MySecurityGroup \
--region us-east-1
```

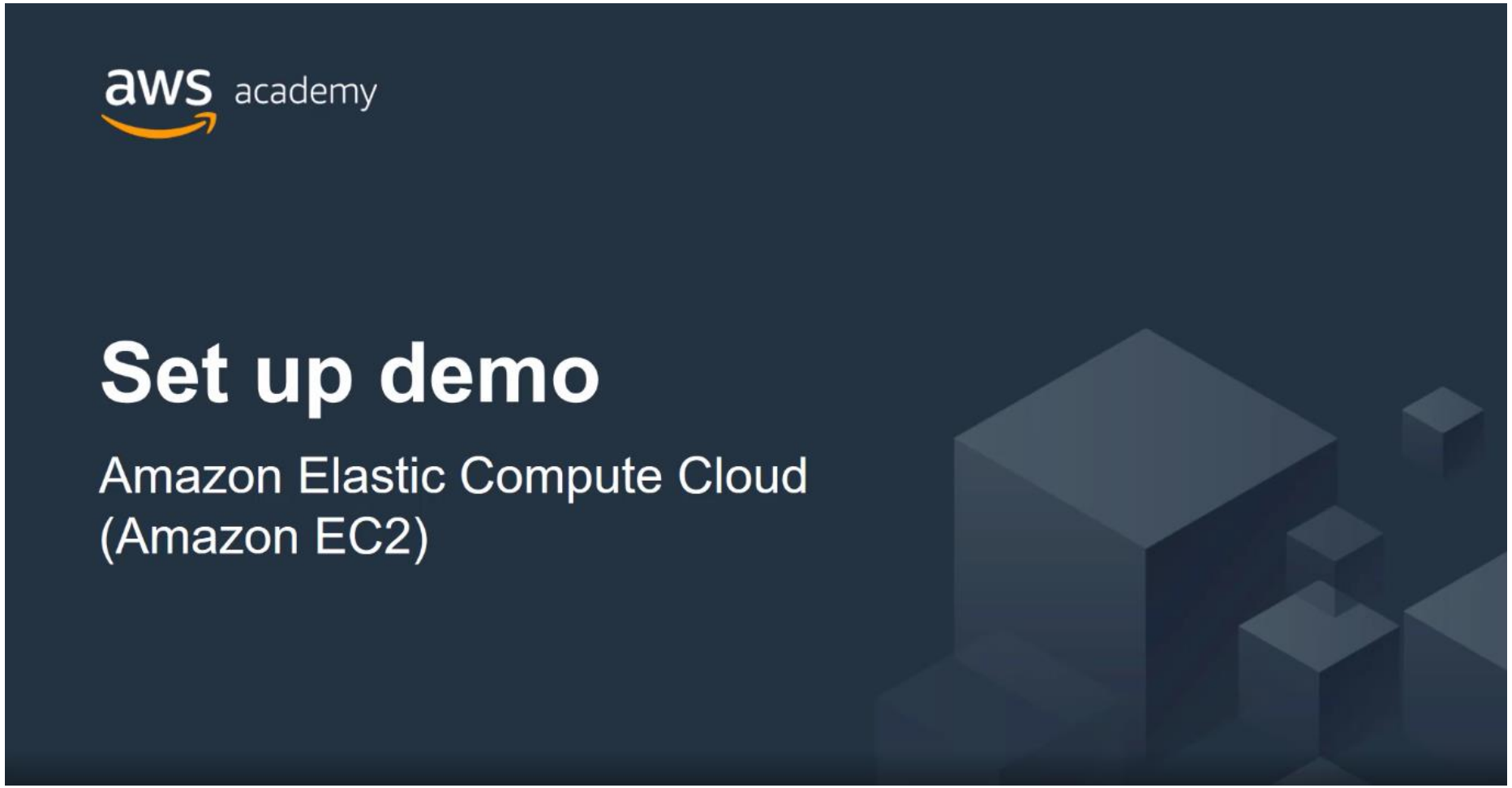
EC2 Instance Lifecycle



EC2 Instance Metadata

- **Instance metadata is data about the instance**
- **While you are connected to the instance, you can view it**
  - **In a browser: http://169.254.169.254/latest/meta-data/**
  - **In a terminal window: curl http://169.254.169.254/latest/meta-data/**
- **Example retrievable values**
  - **Public IP address, private IP address, public hostname, instance ID, security groups, Region, Availability Zone**
  - **Any user data specified at instance launch can also be accessed at: http://169.254.169.254/latest/user-data/**
- **It can be used to configure or manage a running instance**
  - **For example, author a configuration script that reads the metadata and uses it to configure applications or OS settings**

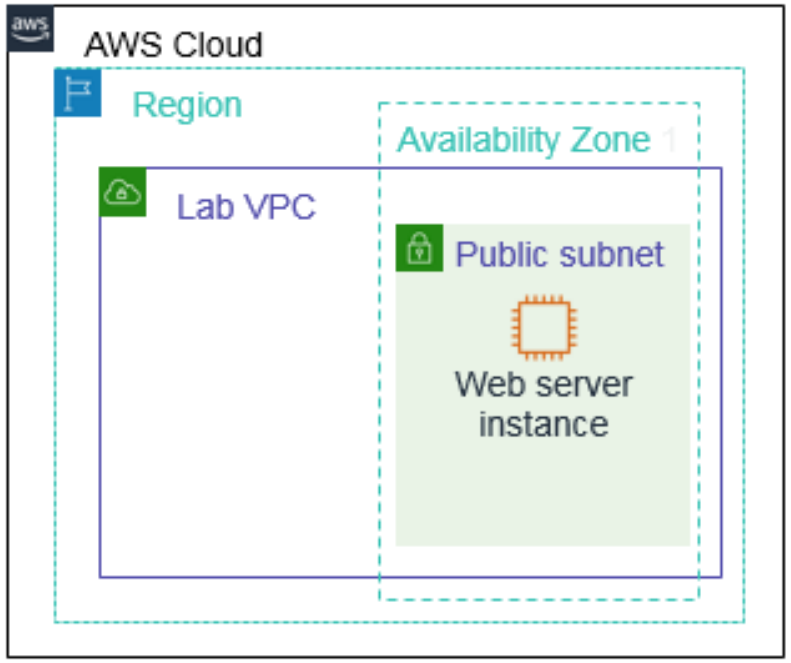
EC2 Demo


aws academy
Set up demo
Amazon Elastic Compute Cloud
(Amazon EC2)

Lab 3: Introduction to Amazon EC2

Lab Scenario:

In this lab, you will launch and configure your first virtual machine that runs on Amazon EC2.



Lab 3: Introduction to Amazon EC2 (Cont.)

Lab Tasks:

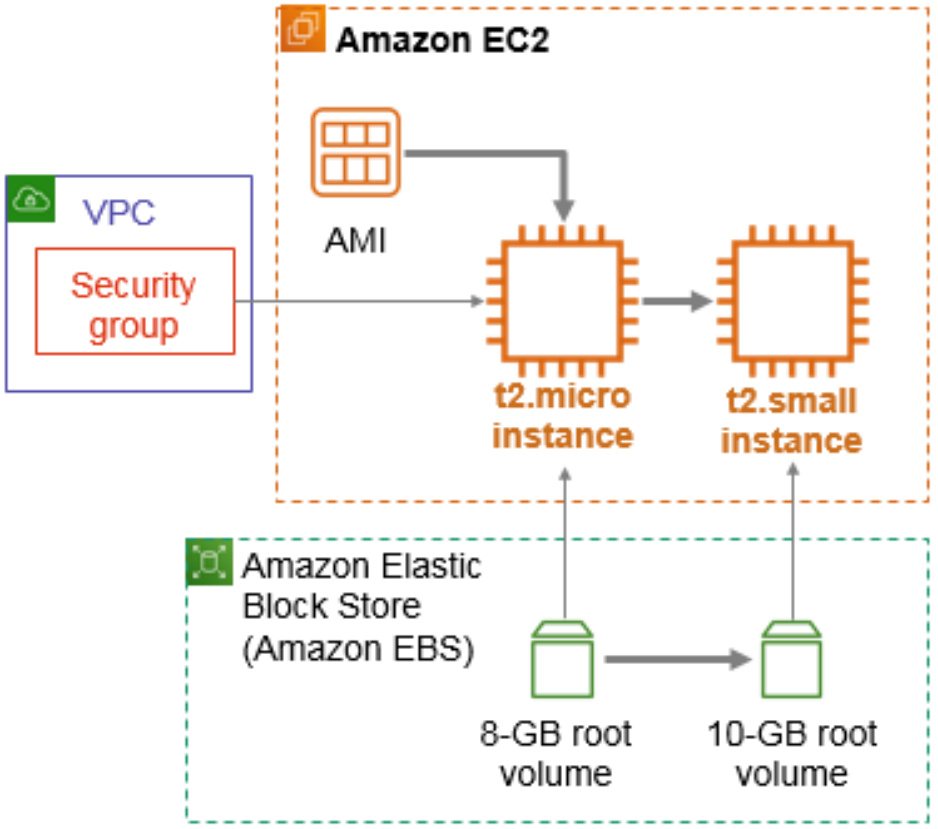
- **Task 1 – Launch Amazon EC2 Instance**
- **Task 2 – Monitor the Instance**
- **Task 3 – Update Security Group and Access the Web Server**
- **Task 4 – Resize the Instance: Instance Type**
- **Task 5 – Explore EC2 Limits**
- **Task 6 – Test Termination Protection**

Lab 3: Introduction to Amazon EC2 (Cont.)

Final Product:

By the end of the lab, you will have:

- **Launched an instance that is configured as a web server**
- **Viewed the instance system log**
- **Reconfigured a security group**
- **Modified the instance type and root volume size**



Lecture's Agenda

- Compute Services Overview
- Amazon EC2
- **EC2 Cost Optimization**
- Container Services
- Introduction to AWS Lambda
- Introduction to AWS Elastic Beanstalk

Amazon EC2 Pricing Models

On-Demand Instances:

- **Pay by the hour**
- **No long-term commitments**
- **Eligible for the AWS Free Tier**

Dedicated Hosts:

- **A physical server with EC2 instance capacity fully dedicated to customer use**

Amazon EC2 Pricing Models (Cont.)

Dedicated Instances:

- **Instances that run in a VPC on hardware that is dedicated to a single customer**

Reserved Instances:

- **Full, partial, or no upfront payment for instance the customer reserve**
- **Discount on hourly charge for that instance**
- **1-year or 3-year term**

Amazon EC2 Pricing Models (Cont.)

Spot Instances:

- **Instances run as long as they are available and customer's bid is above the Spot Instance price**
- **They can be interrupted by AWS with a 2-minute notification**
  - **Interruption options include terminated, stopped or hibernated.**
- **Prices can be significantly less expensive compared to On-Demand Instances**
  - **Good choice when user have flexibility in when your applications can run**
  - **More like grid computing resources; mainly used for batch processing**

Amazon EC2 Pricing Models (Cont.)

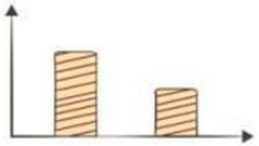
Scheduled Reserved Instances:

- **Purchase a capacity reservation that is always available on a recurring schedule customer specify**
  - **BISE Lahore Result Dates**
- **1-year term**

Per Second Billing:

- **Available for On-Demand Instances, Reserved Instances, and Spot Instances that run Amazon Linux or Ubuntu**

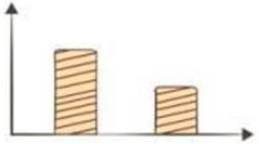
Amazon EC2 Pricing Models: Benefits

On-Demand Instances	Spot Instances	Reserved Instances	Dedicated Hosts
• Low cost and flexibility	• Large scale, dynamic workload	• Predictability ensures compute capacity is available when needed	• Save money on licensing costs • Help meet compliance and regulatory requirements

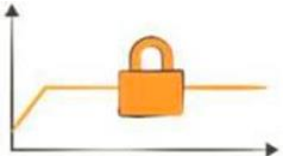
Amazon EC2 Pricing Models: Use Cases

Spiky Workloads



Time-Insensitive Workloads

Steady-State Workloads



Highly Sensitive Workloads

On-Demand Instances	Spot Instances	Reserved Instances	Dedicated Hosts
• Short-term, spiky, or unpredictable workloads • Application development or testing	• Applications with flexible start start and end times • Applications only feasible at very low compute prices • Users with urgent computing computing needs for large amounts of additional capacity capacity	• Steady state or predictable usage workloads • Applications that require reserved capacity, including disaster recovery • Users able to make upfront payments to reduce total computing costs even further	• Bring your own license (BYOL)

Four Pillars of Cost Optimization

Pillar 1 - Right size:

- **AWS offers approximately 60 instance types and sizes**
- **Provision instances to match the need**
  - **CPU, memory, storage, and network throughput**
  - **Select appropriate instance types for customer's use**
- **Use Amazon CloudWatch metrics**
  - **How idle are instances? When?**
  - **Downsize instances**
- **Best practice: Right size, then reserve**

Four Pillars of Cost Optimization (Cont.)

Pillar 2 - Increase Elasticity:

- **Stop or hibernate Amazon EBS-backed instances that are not actively in use**
  - **Example: non-production development or test instances**

- **Use automatic scaling to match needs based on usage**
  - **Automated and time-based elasticity**

Four Pillars of Cost Optimization (Cont.)

Pillar 3 - Optimal Pricing Model:

- **Leverage the right pricing model for the use case**
  - **Consider the usage patterns**
- **Optimize and combine purchase types**
- **Examples:**
  - **Use On-Demand Instance and Spot Instances for variable workloads**
  - **Use Reserved Instances for predictable workloads**
- **Consider serverless solutions (AWS Lambda)**

Four Pillars of Cost Optimization (Cont.)

Pillar 4 - Optimize Storage Choices:

- **Reduce costs while maintaining storage performance and availability**
- **Resize EBS volumes**
- **Change EBS volume types**
  - **Can you meet performance requirements with less expensive storage?**
  - **Example: Amazon EBS Throughput Optimized HDD (st1) storage typically costs half as much as the default General Purpose SSD (gp2) storage option**

Four Pillars of Cost Optimization (Cont.)

Pillar 4 - Optimize Storage Choices:

- **Delete EBS snapshots that are no longer needed**

- **Identify the most appropriate destination for specific types of data**
  - **Does the application need the instance to reside on Amazon EBS?**
  - **Amazon S3 storage options with lifecycle policies can reduce costs**

Measure, Monitor, and Improve

- **Cost optimization is an ongoing process**

- **Recommendations**
  - **Define and enforce cost allocation tagging**
  - **Define metrics, set targets, and review regularly**
    - **AWS Trusted Advisor, AWS Cost Explorer**
  - **Encourage teams to architect for cost**
  - **Assign the responsibility of optimization to an individual or to a team**

Additional Resources

- **Amazon EC2 Documentation**
  - **https://docs.aws.amazon.com/ec2/**
- **Amazon EC2 User Guide**
  - **https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts.html**
- **Windows User Data Scripts documentation**
  - **https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/user-data.html**
- **Amazon EC2 Pricing**
  - **https://aws.amazon.com/ec2/pricing/**

Questions?