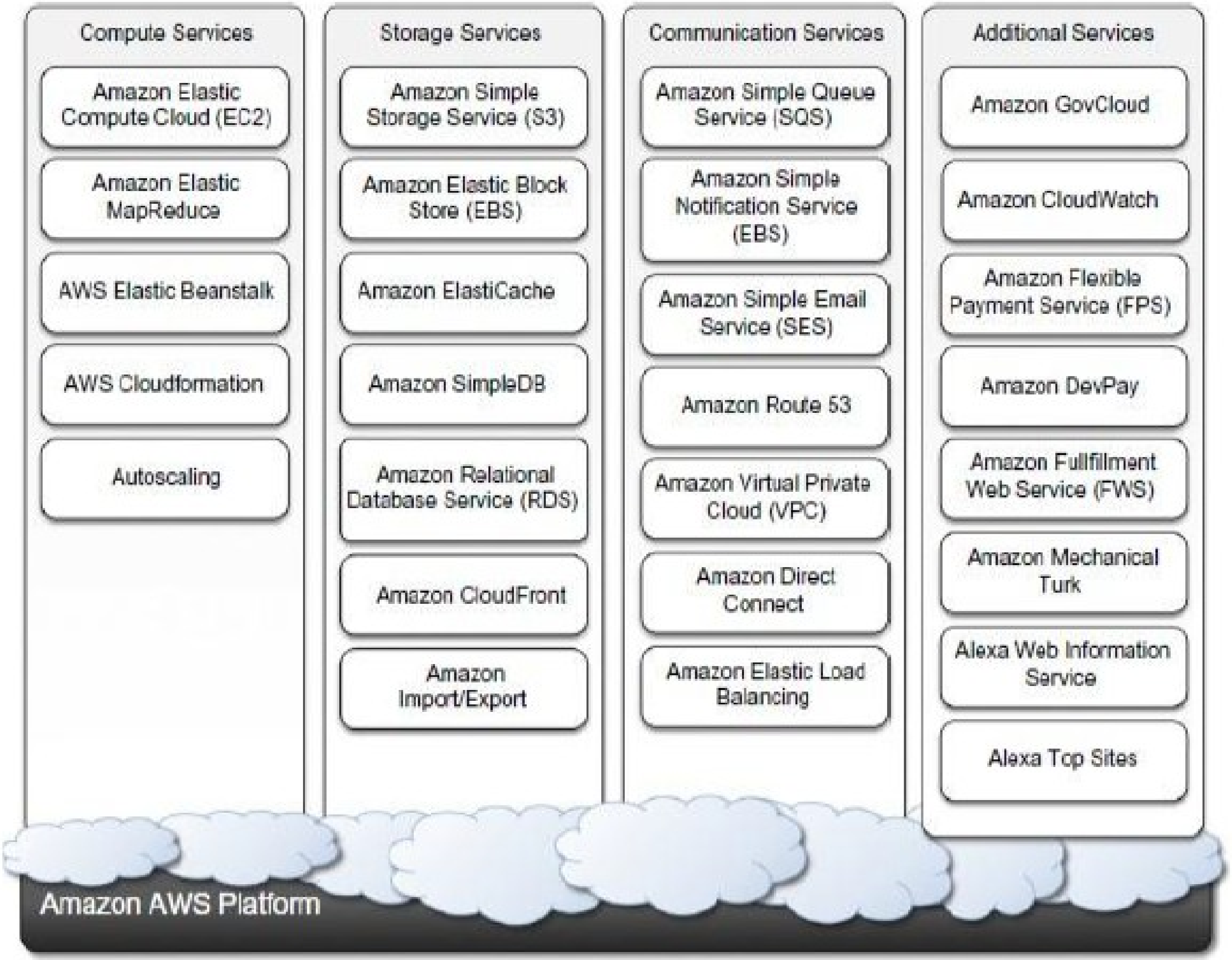
AWS Compute & Storage Services

Module 2



Compute Services

- Amazon Elastic Compute Cloud (EC2)
- Amazon Elastic MapReduce
- AWS Elastic Beanstalk
- AWS Cloudformation
- Autoscaling

Storage Services

- Amazon Simple Storage Service (S3)
- Amazon Elastic Block Store (EBS)
- Amazon ElastiCache
- Amazon SimpleDB
- Amazon Relational Database Service (RDS)
- Amazon CloudFront
- Amazon Import/Export

Communication Services

- Amazon Simple Queue Service (SQS)
- Amazon Simple Notification Service (EBS)
- Amazon Simple Email Service (SES)
- Amazon Route 53
- Amazon Virtual Private Cloud (VPC)
- Amazon Direct Connect
- Amazon Elastic Load Balancing

Additional Services

- Amazon GovCloud
- Amazon CloudWatch
- Amazon Flexible Payment Service (FPS)
- Amazon DevPay
- Amazon Fullfillment Web Service (FWS)
- Amazon Mechanical Turk
- Alexa Web Information Service
- Alexa Top Sites

Amazon AWS Platform

Figure: Amazon Web Services ecosystem

Introduction to AWS Compute

AWS compute is an Infrastructure As A Service(IAAS). Put simply, AWS compute is the means to provision and manage infrastructure(virtual machines/containers) for your use case.

AWS provides many flexible computing services so as to meet the requirements of business organizations like Amazon Elastic Compute Cloud (EC2), Amazon Elastic Container Service (ECS), Amazon Elastic Container Service for Kubernetes (EKS), Amazon Lightsail, AWS Lambda and many more.

This infrastructure as a service can be considered as the processing power required by your applications, to host applications or run computation-intensive tasks.

Introduction to AWS Compute

In AWS, with the use of these computing services, users can dynamically provision the number of resources they are using and then pay only for the computing resources they have used for.

This leads to the reduction of the upfront capital investment required.

These compute resources are closely related to regular server components like CPU and RAM. However, for regular server components, you need to manage and buy the infrastructure, provide for backups and emergency recovery, and ensure enough server capacity to handle traffic-intensive times. **With AWS compute all this headache is handed over to the AWS team.**

Amazon EC2

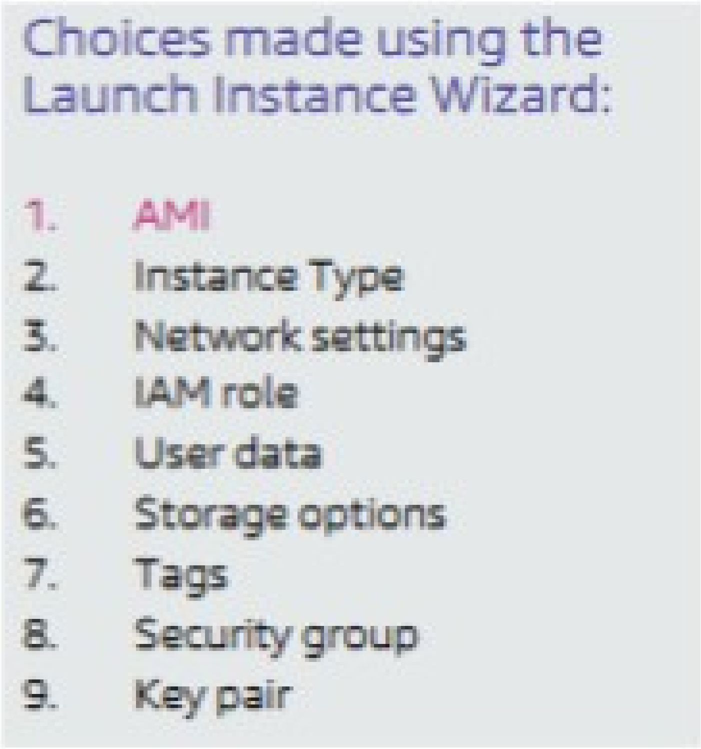
Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides secure, resizable compute capacity in the cloud. It is designed to make web-scale computing easier for developers.

Amazon EC2 reduces the time required to obtain and boot new server instances (called Amazon EC2 instances) to minutes, allowing you to quickly scale capacity, both up and down, as your computing requirements change.

Amazon EC2 changes the economics of computing by allowing you to pay only for capacity that you actually use. Amazon EC2 provides developers and system administrators the tools to build failure resilient applications and isolate themselves from common failure scenarios

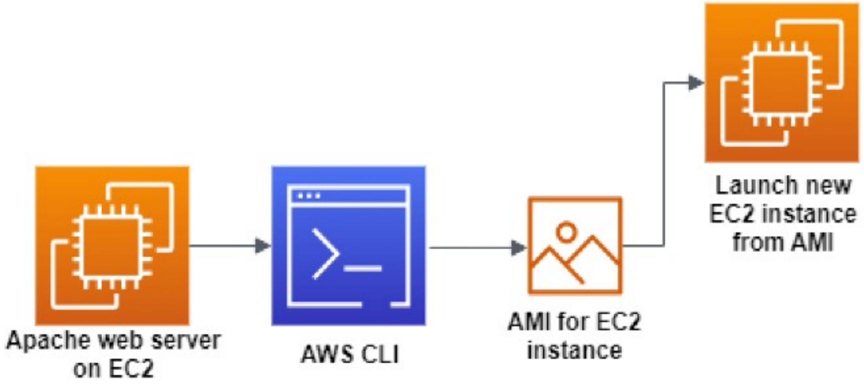
Launching an Amazon EC2 instance

[nine key decisions to make when you create an EC2 instance by using the AWS Management Console Launch Instance Wizard]

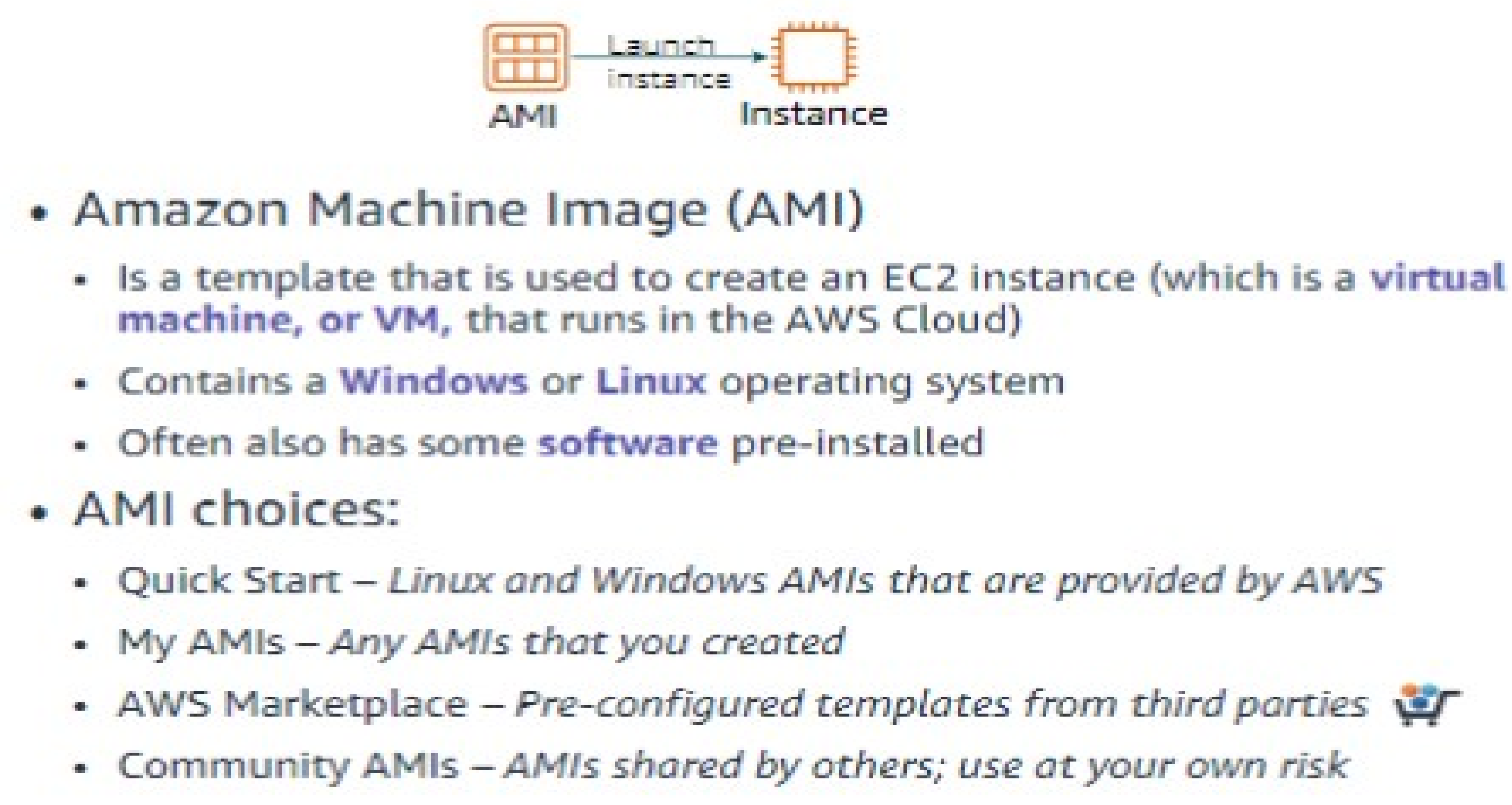


Choices made using the Launch Instance Wizard:

1. AMI
2. Instance Type
3. Network settings
4. IAM role
5. User data
6. Storage options
7. Tags
8. Security group
9. Key pair



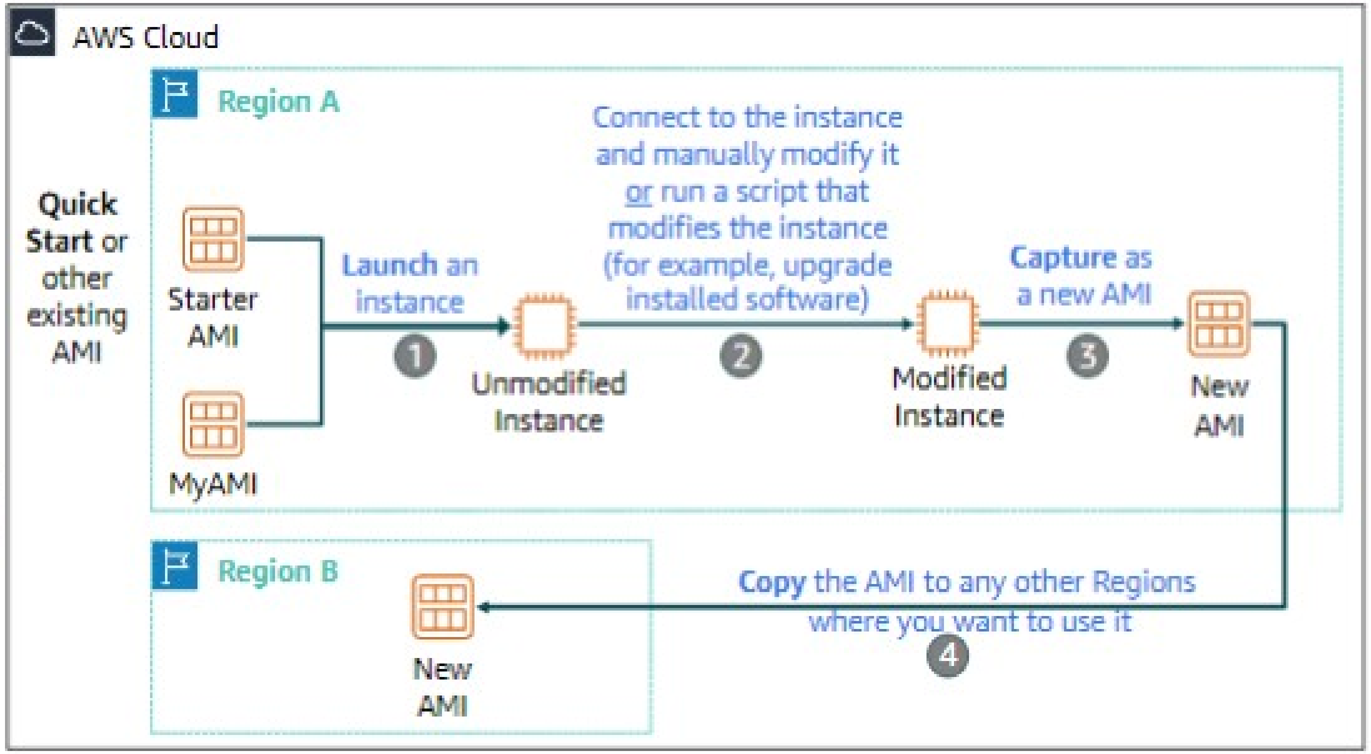
Select an AMI


Launch instance
AMI
Instance

- Amazon Machine Image (AMI)
  - Is a template that is used to create an EC2 instance (which is a **virtual machine, or VM,** that runs in the AWS Cloud)
  - Contains a **Windows** or **Linux** operating system
  - Often also has some **software** pre-installed
- 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



AWS Cloud

Region A

Quick Start or other existing AMI

Starter AMI

MyAMI

Launch an instance

1

Unmodified Instance

Connect to the instance and manually modify it or run a script that modifies the instance (for example, upgrade installed software)

2

Modified Instance

Capture as a new AMI

3

New AMI

Region B

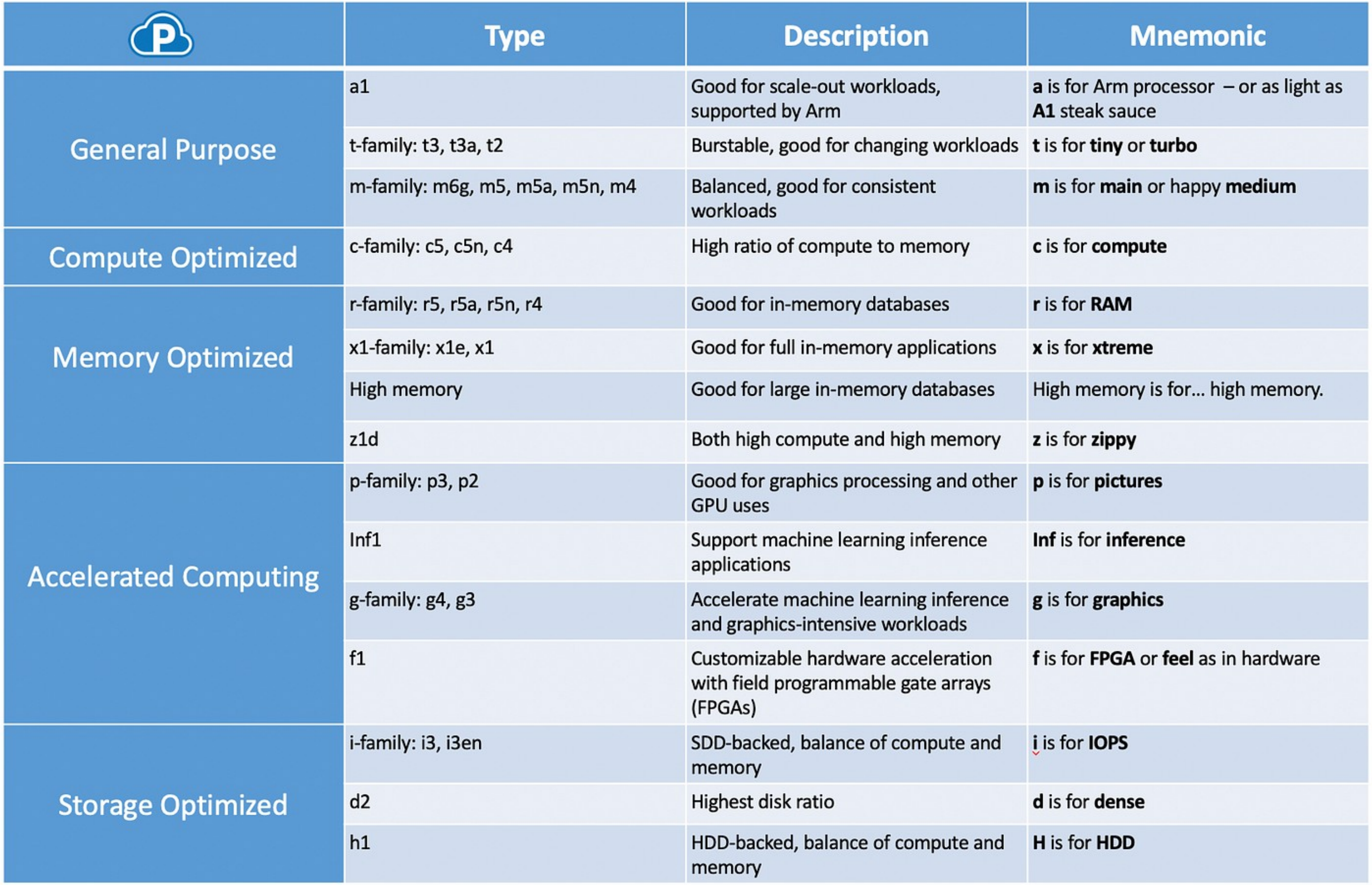
New AMI

Copy the AMI to any other Regions where you want to use it

4

(Optional) Import a virtual machine

Instance types

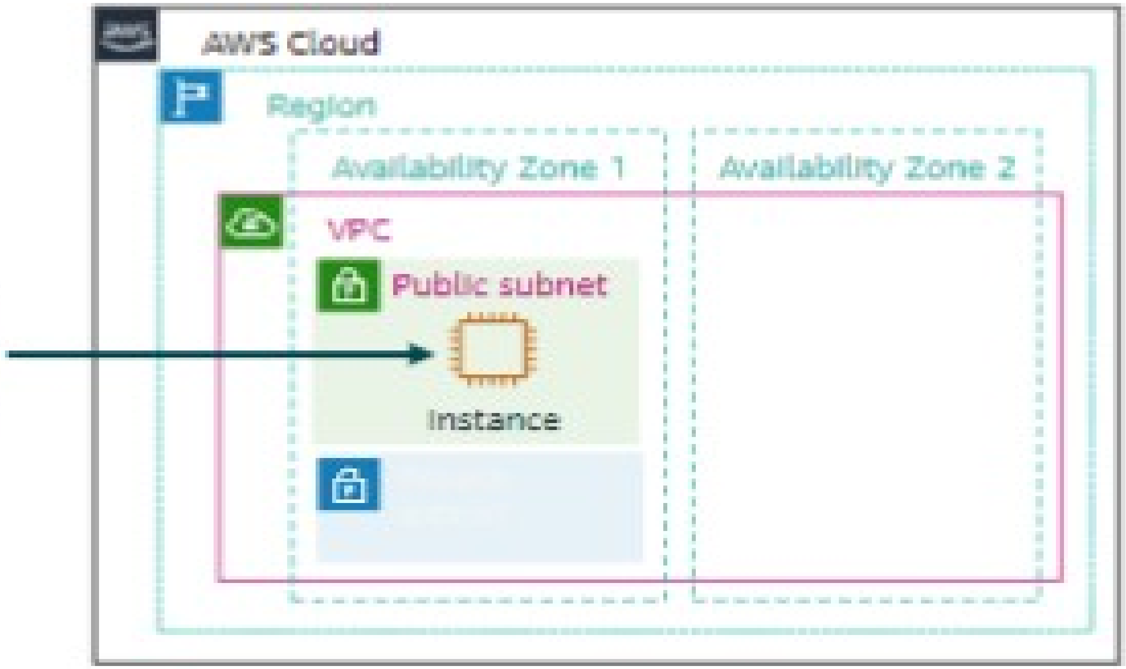


	Type	Description	Mnemonic
General Purpose	a1	Good for scale-out workloads, supported by Arm	**a** is for Arm processor – or as light as **A1** steak sauce
	t-family: t3, t3a, t2	Burstable, good for changing workloads	**t** is for **tiny** or **turbo**
	m-family: m6g, m5, m5a, m5n, m4	Balanced, good for consistent workloads	**m** is for **main** or happy **medium**
Compute Optimized	c-family: c5, c5n, c4	High ratio of compute to memory	**c** is for **compute**
Memory Optimized	r-family: r5, r5a, r5n, r4	Good for in-memory databases	**r** is for **RAM**
	x1-family: x1e, x1	Good for full in-memory applications	**x** is for **xtreme**
	High memory	Good for large in-memory databases	High memory is for… high memory.
	z1d	Both high compute and high memory	**z** is for **zippy**
Accelerated Computing	p-family: p3, p2	Good for graphics processing and other GPU uses	**p** is for **pictures**
	Inf1	Support machine learning inference applications	**Inf** is for **inference**
	g-family: g4, g3	Accelerate machine learning inference and graphics-intensive workloads	**g** is for **graphics**
	f1	Customizable hardware acceleration with field programmable gate arrays (FPGAs)	**f** is for **FPGA** or **feel** as in hardware
Storage Optimized	i-family: i3, i3en	SDD-backed, balance of compute and memory	**i** is for **IOPS**
	d2	Highest disk ratio	**d** is for **dense**
	h1	HDD-backed, balance of compute and memory	**H** is for **HDD**

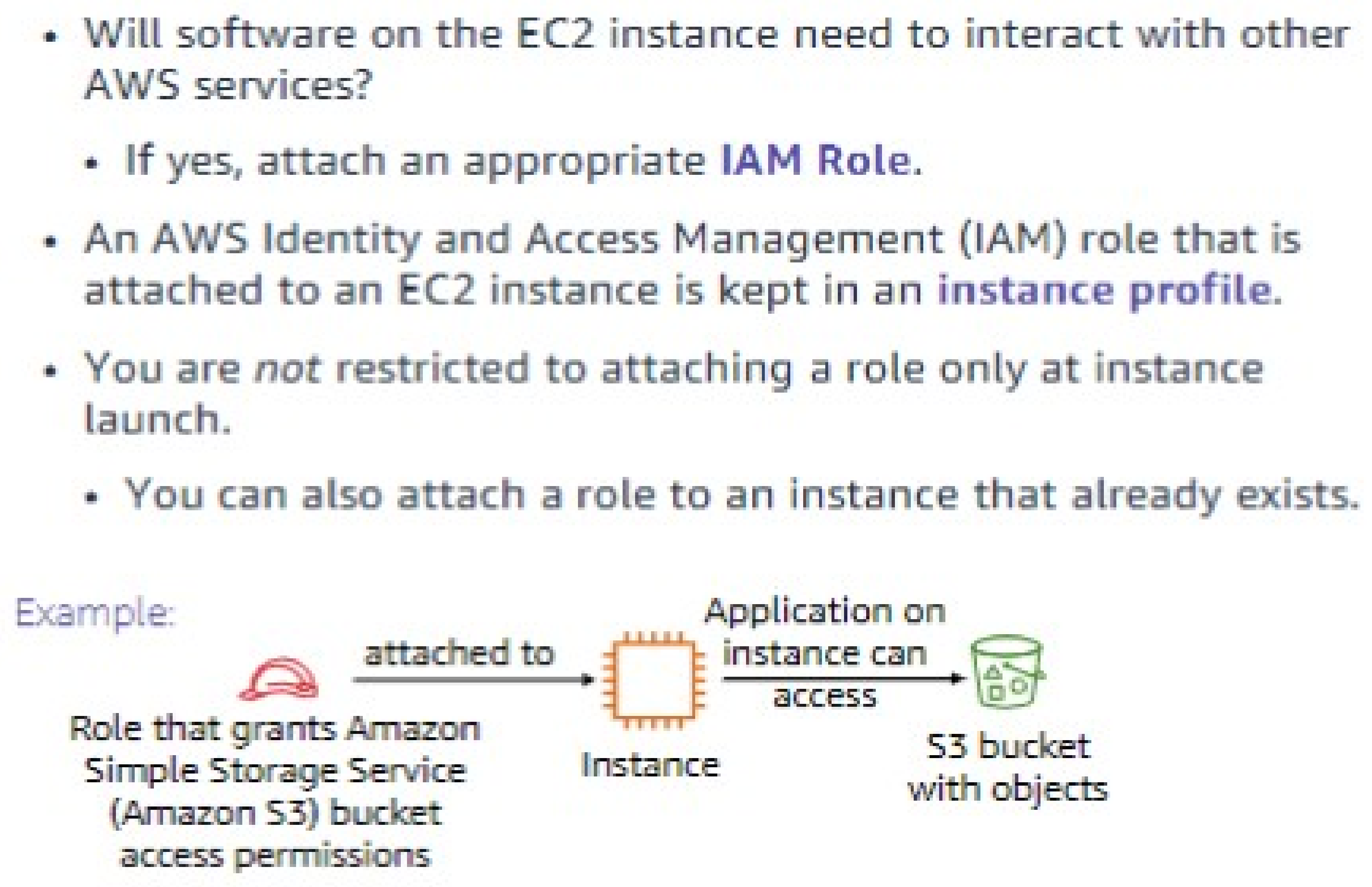
Specify Network Settings

- 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

Example: specify to deploy the instance here



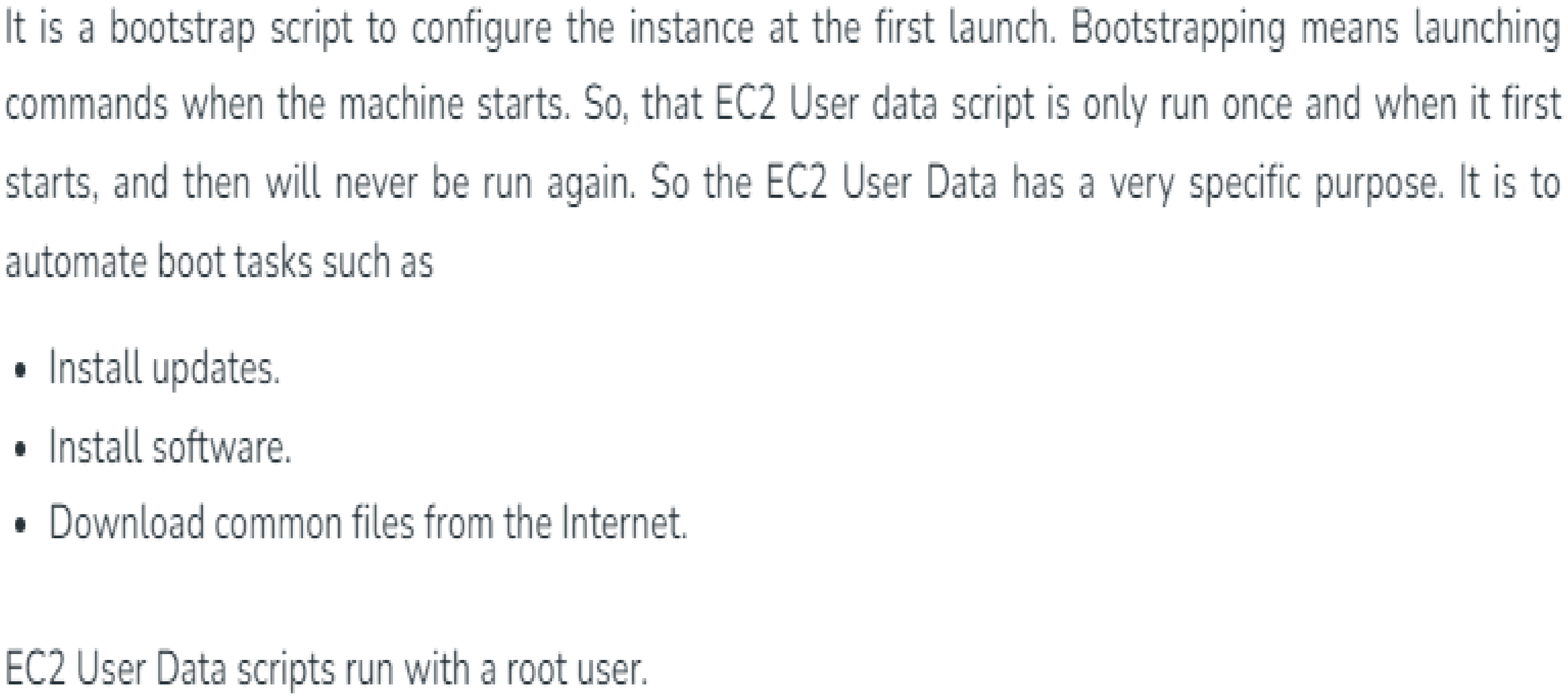
Attach IAM role (optional)



- Will software on the EC2 instance need to interact with other AWS services?
  - If yes, attach an appropriate **IAM Role**.
- An AWS Identity and Access Management (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.

Example:
attached to
Role that grants Amazon Simple Storage Service (Amazon S3) bucket access permissions
Instance
Application on instance can access
S3 bucket with objects

User data script (optional)

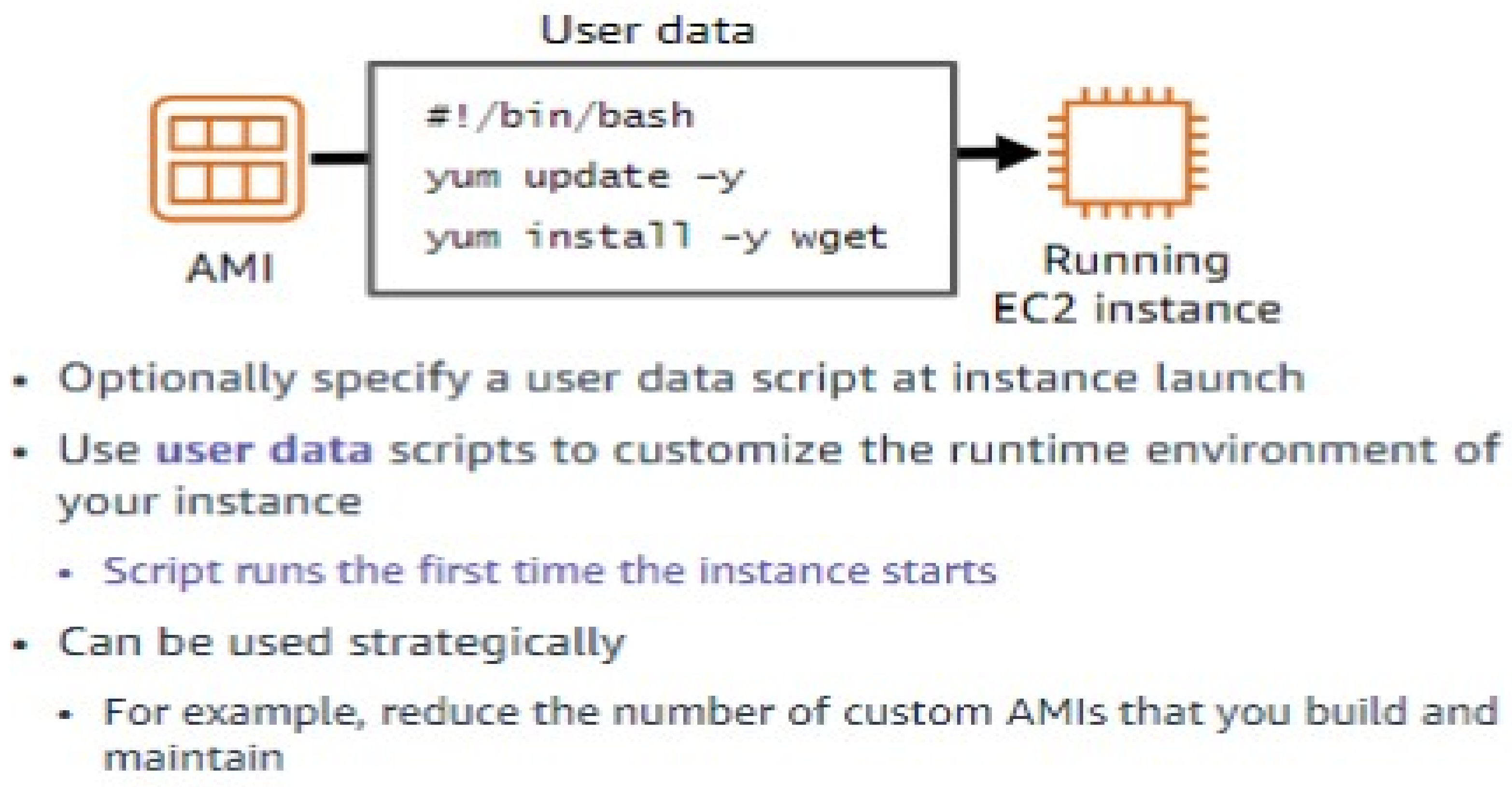


It is a bootstrap script to configure the instance at the first launch. Bootstrapping means launching commands when the machine starts. So, that EC2 User data script is only run once and when it first starts, and then will never be run again. So the EC2 User Data has a very specific purpose. It is to automate boot tasks such as

- Install updates.
- Install software.
- Download common files from the Internet.

EC2 User Data scripts run with a root user.

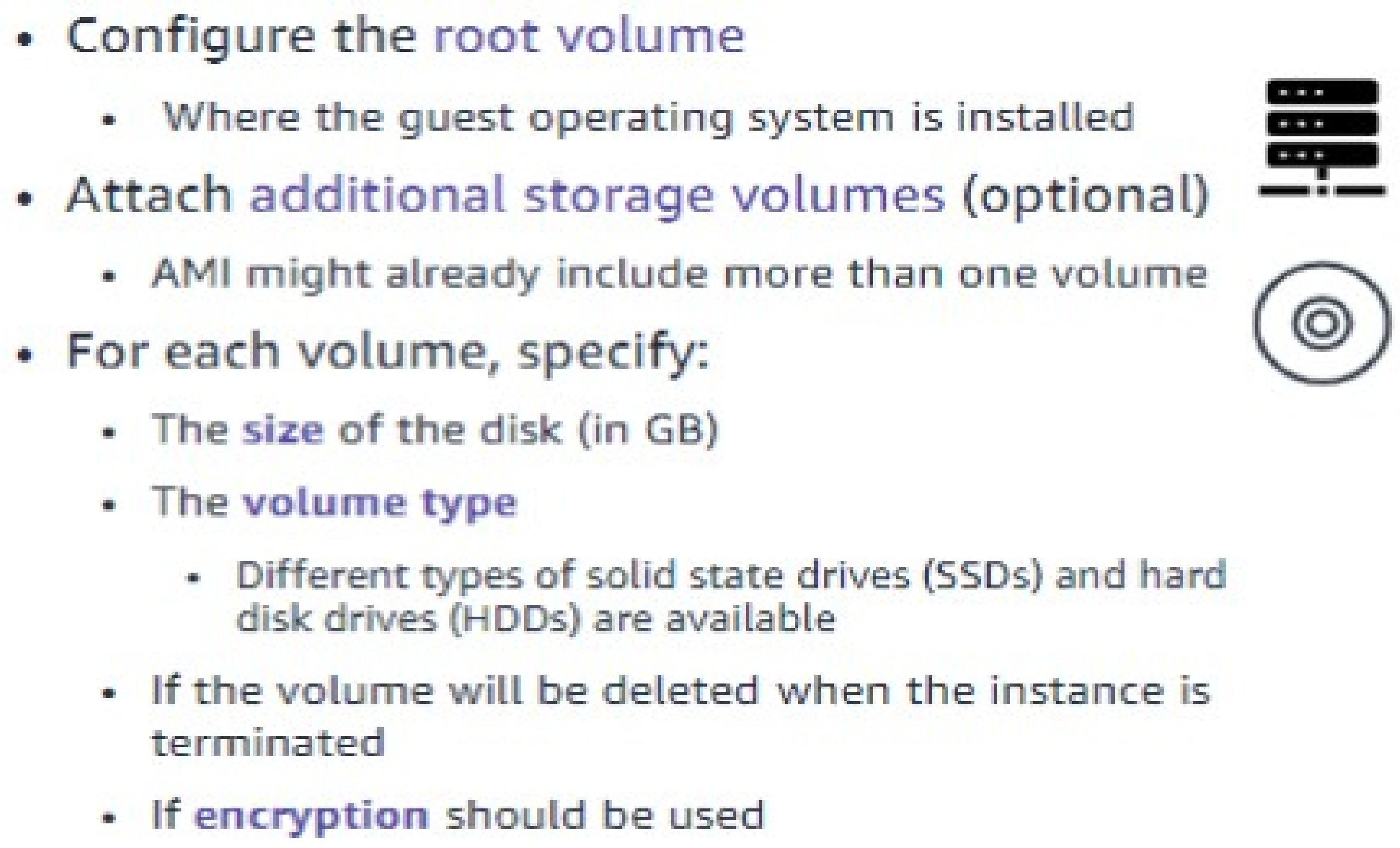
Example



```
#!/bin/bash
yum update -y
yum install -y wget
```

- Optionally specify a user data script at instance launch
- Use **user data** scripts to customize the runtime environment of your instance
  - Script runs the first time the instance starts
- Can be used strategically
  - For example, reduce the number of custom AMIs that you build and maintain

Specify storage



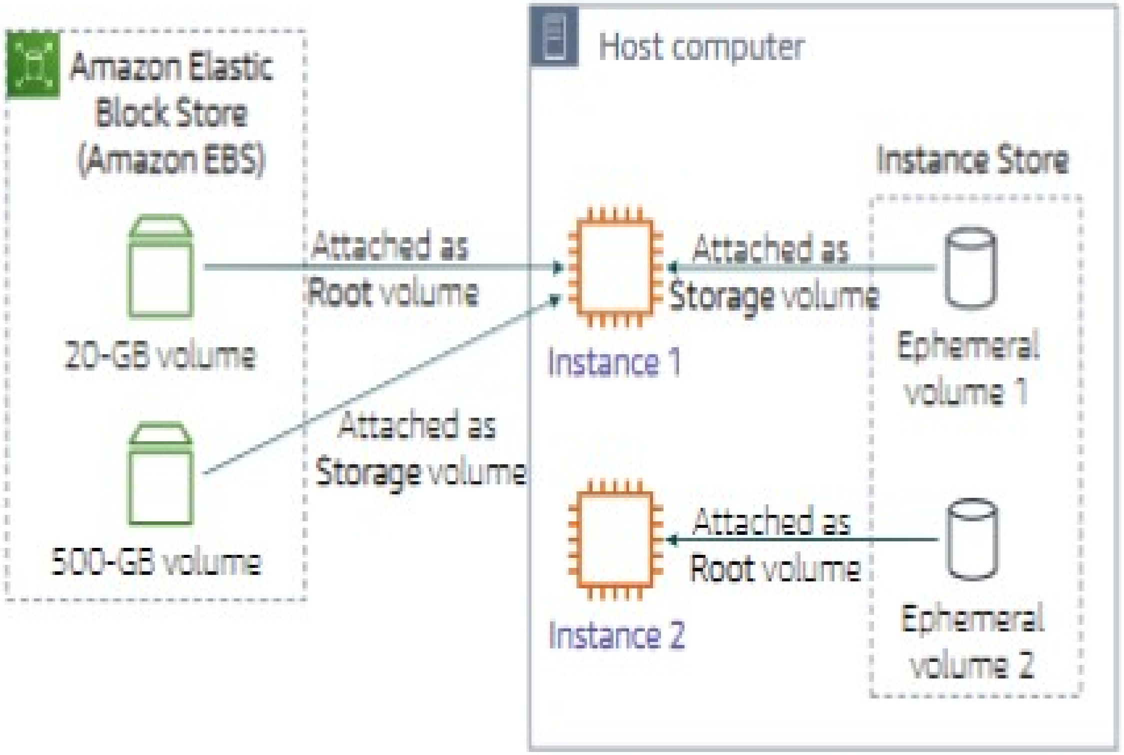
- Configure the root volume
  - Where the guest operating system is installed
- Attach additional storage volumes (optional)
  - AMI might already include more than one volume
- For each volume, specify:
  - The **size** of the disk (in GB)
  - The **volume type**
    - Different types of solid state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

Amazon EC2 storage options

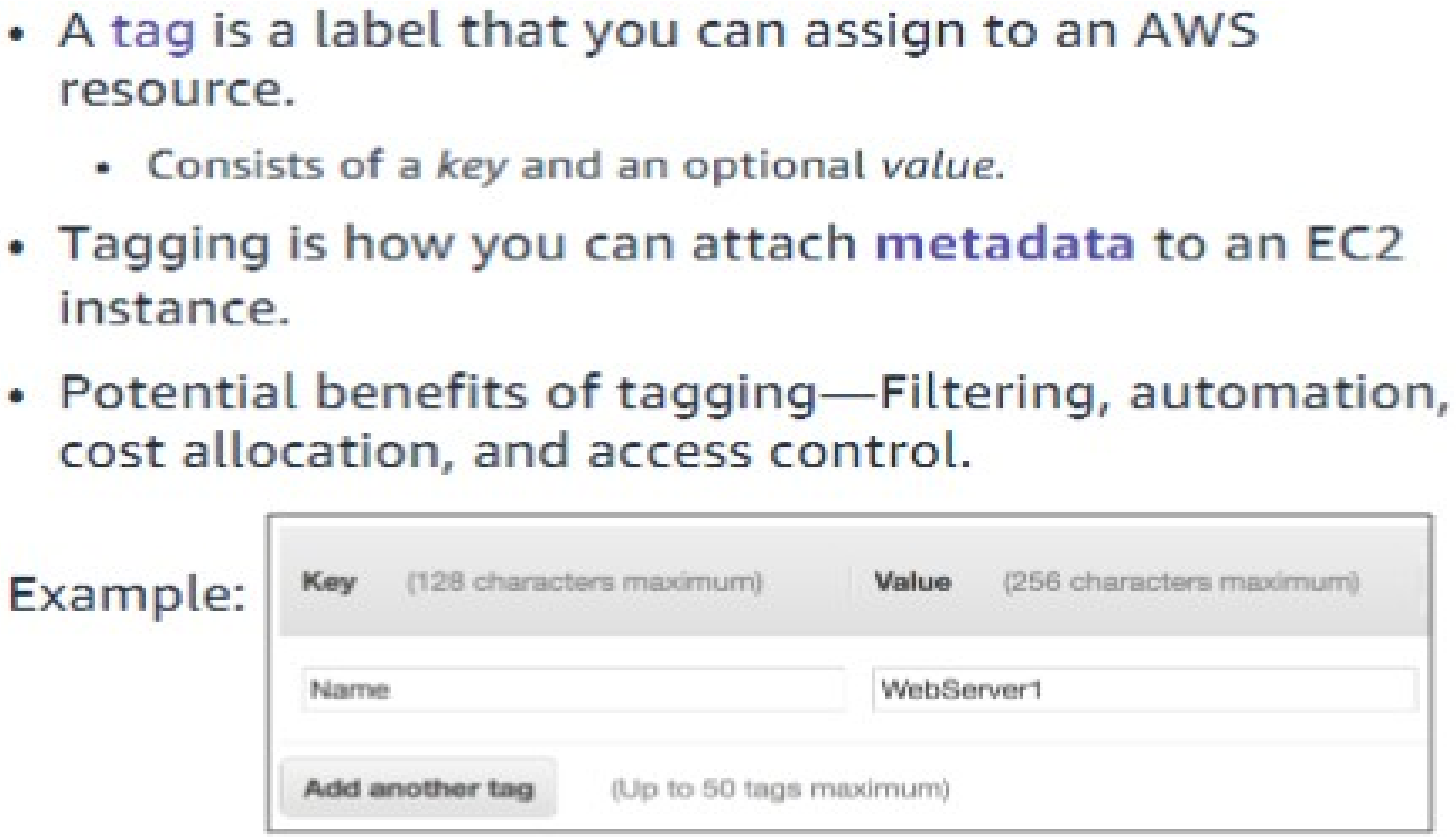
- **Amazon Elastic Block Store (Amazon EBS) –**
  - Durable, block-level storage volumes.
  - You can stop the instance and start it again, and the data will still be there.
- **Amazon EC2 Instance Store –**
  - Ephemeral storage is provided on disks that are attached to the host computer where the EC2 instance is running.
  - If the instance stops, data stored here is deleted.
- Other options for storage (not for the root volume) –
  - Mount an **Amazon Elastic File System (Amazon EFS)** file system.
  - Connect to **Amazon Simple Storage Service (Amazon S3)**.

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)?



Add tags



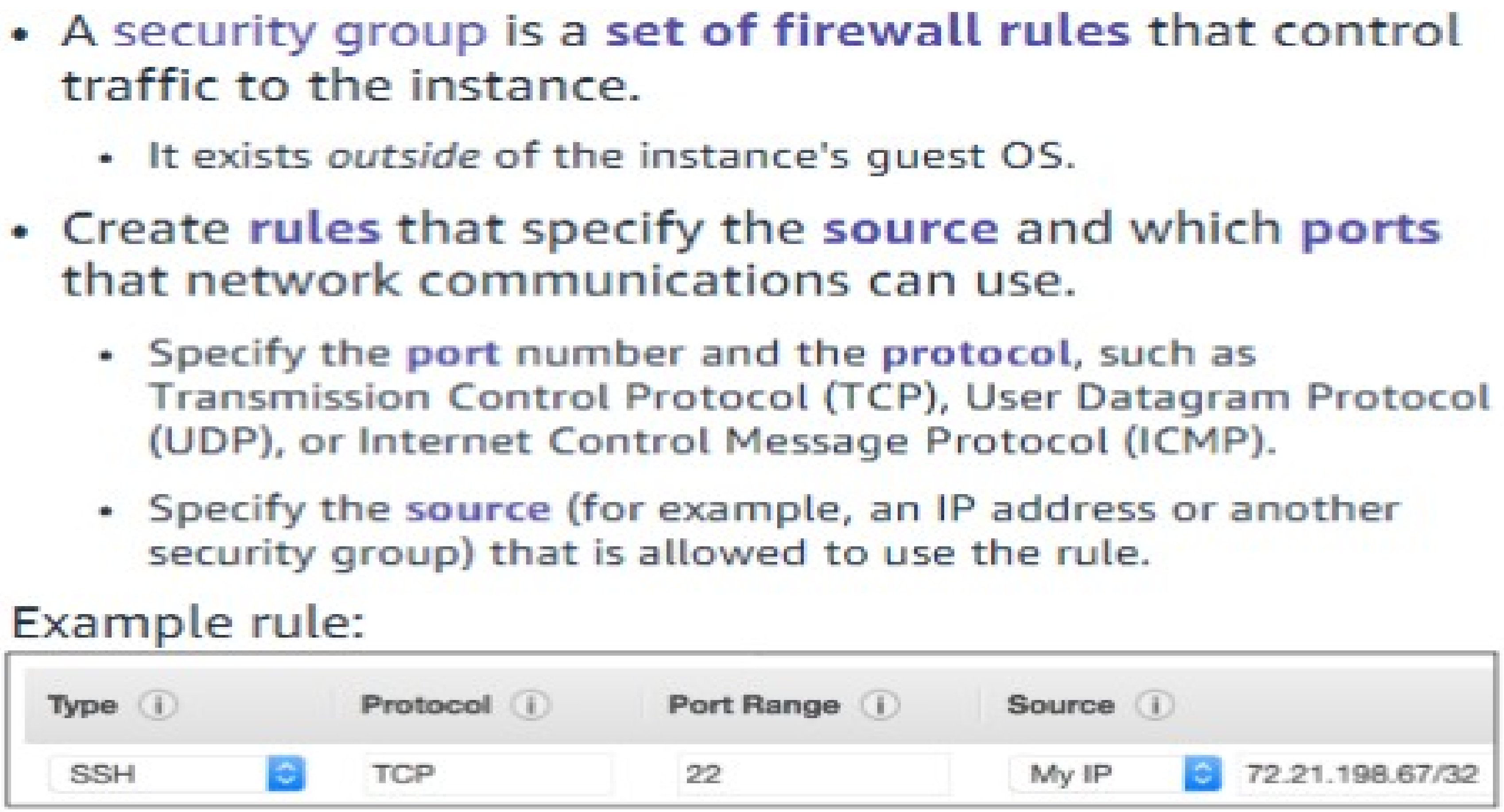
- A tag is a label that you can assign to an AWS resource.
  - Consists of a *key* and an optional *value*.
- Tagging is how you can attach **metadata** to an EC2 instance.
- Potential benefits of tagging—Filtering, automation, cost allocation, and access control.

Example:

Key (128 characters maximum)	**Value** (256 characters maximum)
Name	WebServer1

Add another tag (Up to 50 tags maximum)

Security group settings



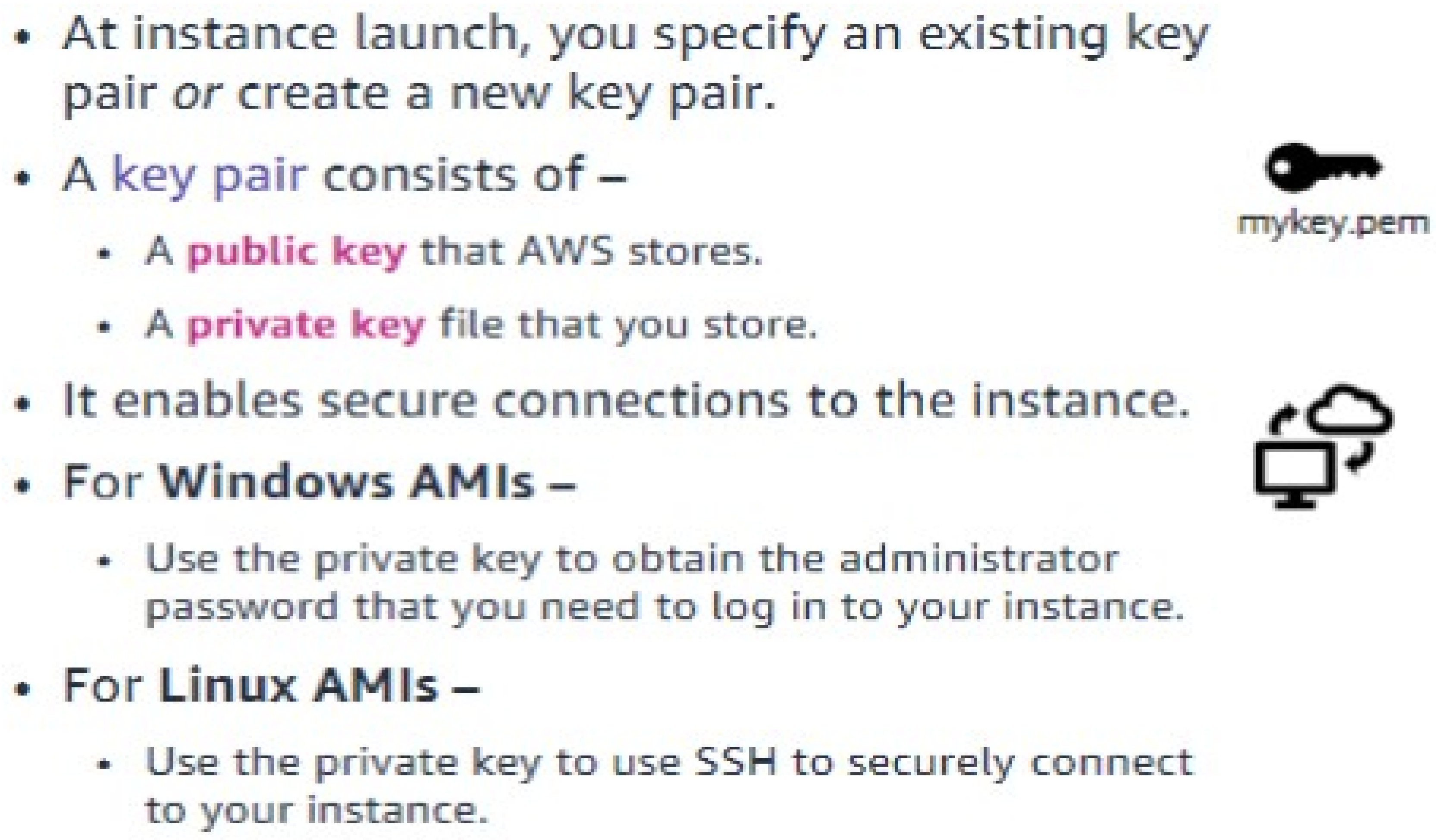
- A **security group** is a **set of firewall rules** that control traffic to the instance.
  - It exists *outside* of the instance's guest OS.
- Create **rules** that specify the **source** and which **ports** that network communications can use.
  - Specify the **port** number and the **protocol**, such as Transmission Control Protocol (TCP), User Datagram Protocol (UDP), or Internet Control Message Protocol (ICMP).
  - Specify the **source** (for example, an IP address or another security group) that is allowed to use the rule.

Example rule:

Type	Protocol	Port Range	Source	
SSH	TCP	22	My IP	72.21.198.67/32

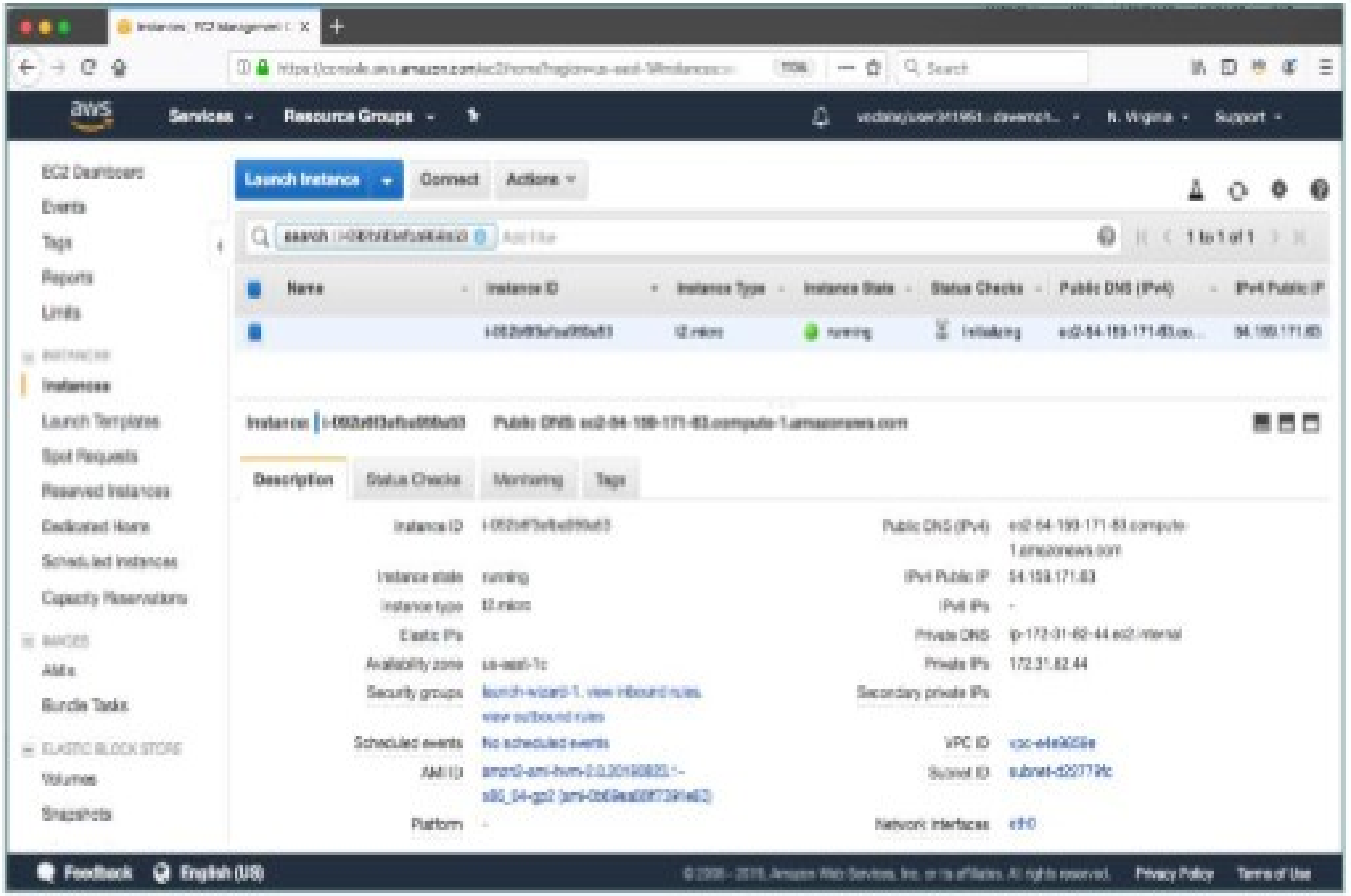
- When you define a rule, you can specify the allowable source of the network communication
(inbound rules) or destination (outbound rules).
- The source can be an IP address, an IP address range, another security group, a gateway VPC endpoint, or anywhere (which means that all sources will be allowed).
- By default, a security group includes an outbound rule that allows all outbound traffic.
- You can remove the rule and add outbound rules that only allow specific outbound traffic.
- If your security group has **no outbound rules**, no outbound traffic that originates from your instance is allowed

Identify or create the key pair



- At instance launch, you 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 you store.
- It enables secure connections to the instance.
- For **Windows AMIs** –
  - Use the private key to obtain the administrator password that you need to log in to your instance.
- For **Linux AMIs** –
  - Use the private key to use SSH to securely connect to your instance.

Amazon EC2 console view of a running EC2 instance

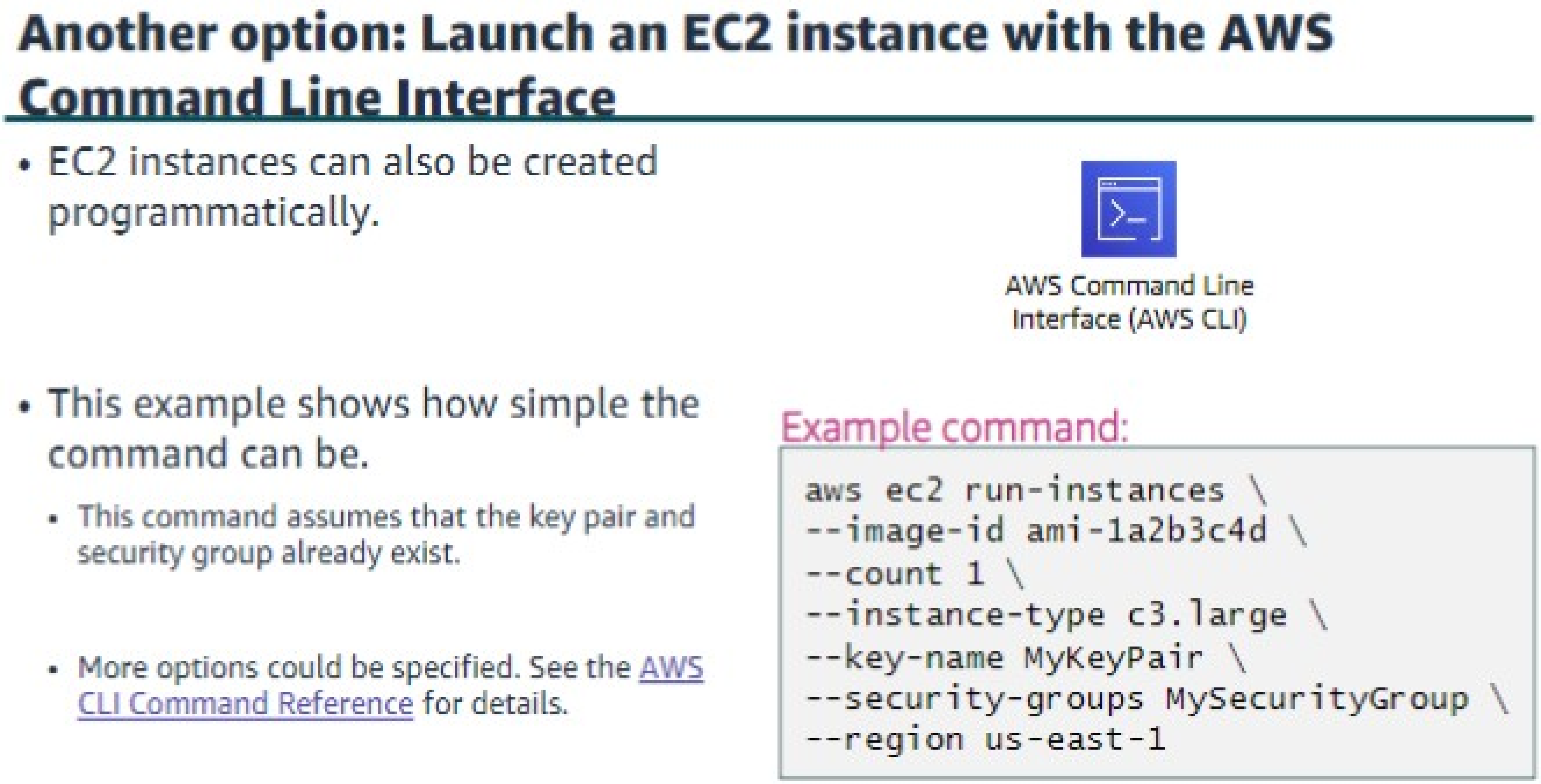


Name	Instance ID	Instance Type	Instance State	Status Checks	Public DNS (IPv4)	IPv4 Public IP
	[illegible]	t2.micro	running	Initializing	ec2-54-159-171-83.co...	54.159.171.83

Instance: [illegible] Public DNS: ec2-54-159-171-83.compute-1.amazonaws.com

Instance ID	[illegible]	Public DNS (IPv4)	ec2-54-159-171-83.compute-1.amazonaws.com
Instance state	running	IPv4 Public IP	54.159.171.83
Instance type	t2.micro	IPv6 IPs	-
Elastic IPs		Private DNS	ip-172-31-82-44.ec2.internal
Availability zone	us-east-1c	Private IPs	172.31.82.44
Security groups	launch-wizard-1. view inbound rules. view outbound rules	Secondary private IPs	
Scheduled events	No scheduled events	VPC ID	vpc-[illegible]
AMI ID	amzn2-ami-hvm-2.0.[illegible]-x86_64-gp2 (ami-[illegible])	Subnet ID	subnet-[illegible]
Platform	-	Network interfaces	eth0

© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.



Another option: Launch an EC2 instance with the AWS Command Line Interface

- EC2 instances can also be created programmatically.

AWS Command Line Interface (AWS CLI)

- This example shows how simple the command can be.
  - This command assumes that the key pair and security group already exist.
  - More options could be specified. See the AWS CLI Command Reference for details.

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
```

AWS Pricing Models

Free Tier

- Free
- Opportunity to try new services
- Suitable for trials and testing
- East to Set Up
- Impractical for production grade use

On-Demand

- No Commitment
- No Upfront Costs
- Highly Flexible
- East to Set Up
- Suitable for projects with variable load and traffic
- Most Expensive Option

Spot Instance

- No Commitment
- No Upfront Costs
- Limited Flexible
- Can be Terminated with little notice
- Suitable for Fault Tolerant Apps
- Cheap Option

Reserved Instance

- 1 or 3 year Commitment
- Upfront Cost Option
- Limited Flexible
- Suitable for Predictable Apps
- Cheaper than On-Demand

Savings Instance

- 1 or 3 year Commitment
- Upfront Cost Option
- Flexible
- Predictable Costs
- Easy to Work with
- Cheaper than On-Demand