AWS Academy Cloud Foundations

Introduction to AWS Networking and Compute

aws academy

Topics

Networking

- Networking basics
- Amazon VPC
- VPC networking
- VPC security

Compute

- Compute services overview
- Amazon EC2

Networking

Amazon VPC Networking

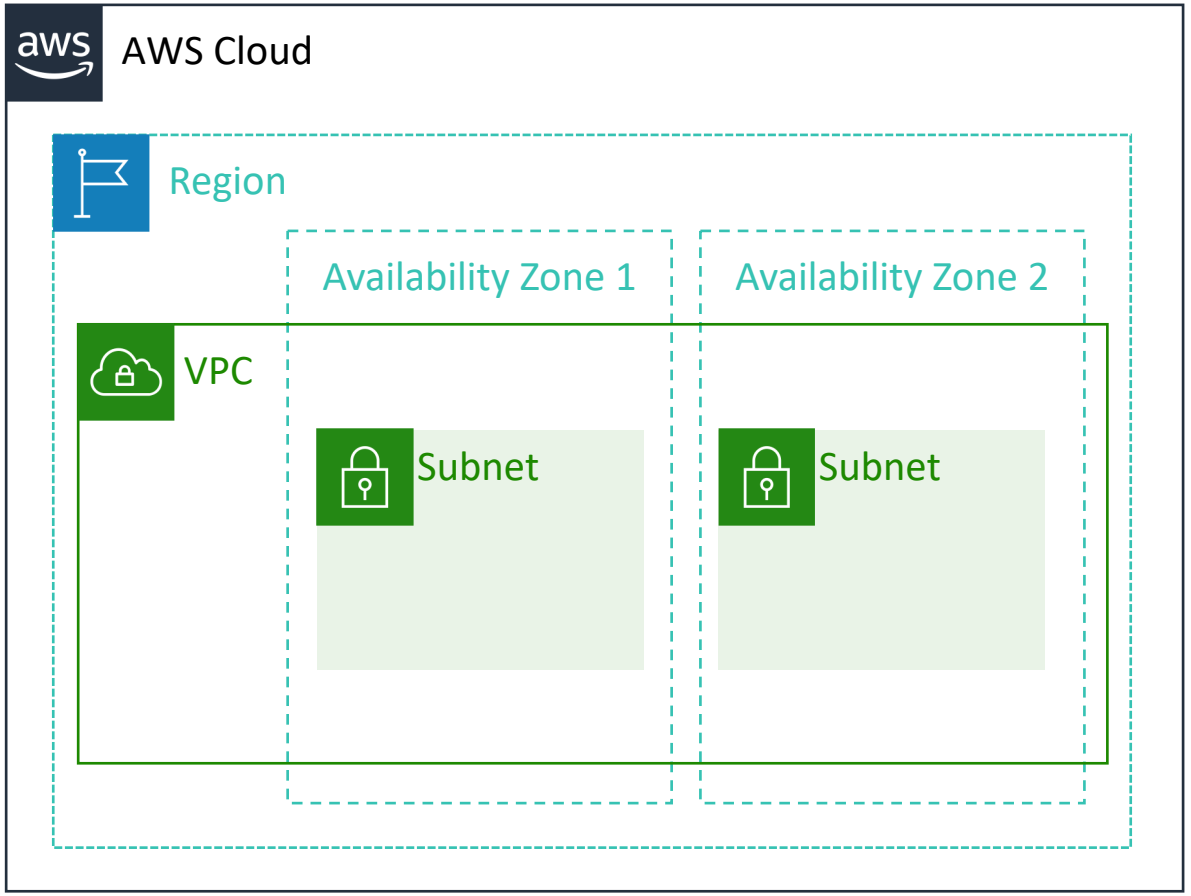
Amazon VPC

Amazon VPC

- Enables you to provision a **logically isolated** section of the AWS Cloud where you can launch AWS resources in a virtual network that you define
- Gives you **control over your virtual networking resources**, including:
  - Selection of IP address range
  - Creation of subnets
  - Configuration of route tables and network gateways
- Enables you to **customize the network configuration** for your VPC.
- Enables you to use **multiple layers of security**.

VPCs and subnets

- VPCs:
  - **Logically isolated** from other VPCs
  - **Dedicated** to your AWS account
  - Belong to a single **AWS Region** and can span multiple Availability Zones
- Subnets:
  - **Range of IP addresses** that divide a VPC
  - Belong to a single **Availability Zone**
  - Classified as **public** or **private**



IP addressing

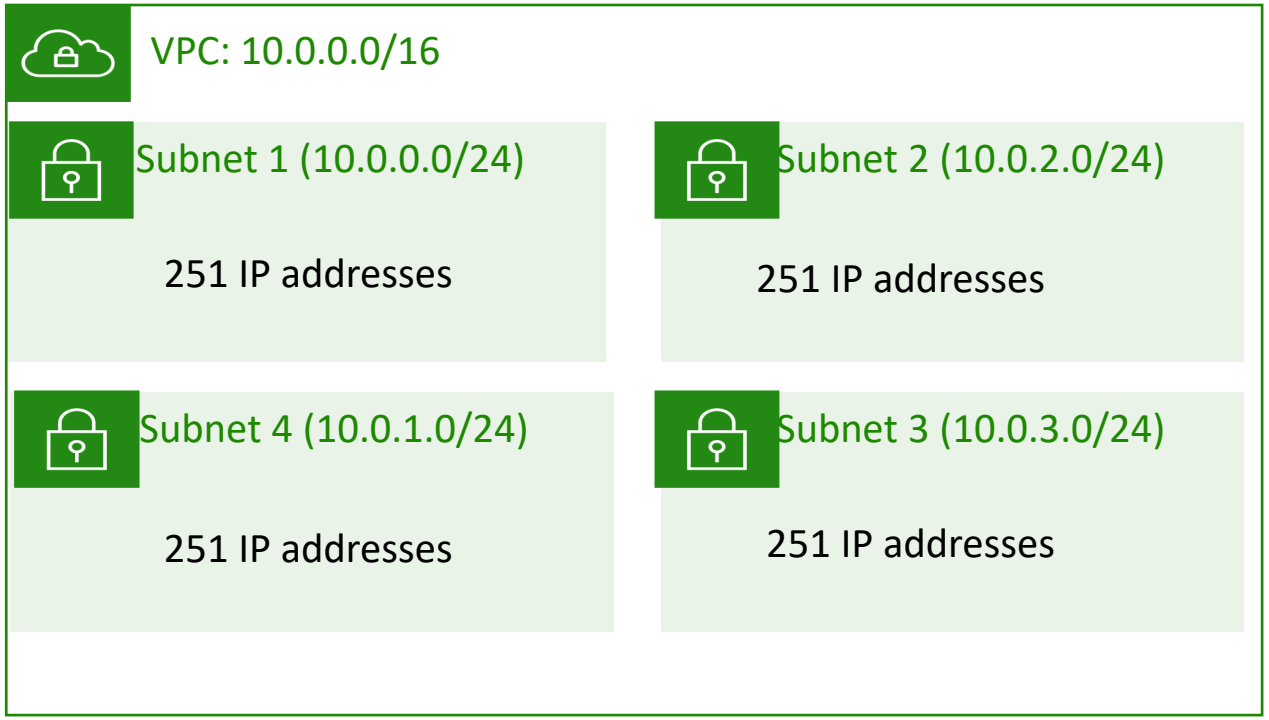
- When you create a VPC, you assign it to an IPv4 **CIDR block** (range of **private** IPv4 addresses).
- You **cannot change the address range** after you create the VPC.
- The **largest** IPv4 CIDR block size is **/16**.
- The **smallest** IPv4 CIDR block size is **/28**.
- IPv6 is also supported (with a different block size limit).
- CIDR blocks of subnets **cannot overlap**.

VPC

x.x.x.x/16 or 65,536 addresses (max)
to
x.x.x.x/28 or 16 addresses (min)

Reserved IP addresses

Example: A VPC with an IPv4 CIDR block of 10.0.0.0/16 has 65,536 total IP addresses. The VPC has four equal-sized subnets. Only 251 IP addresses are available for use by each subnet.



IP Addresses for CIDR block 10.0.0.0/24	Reserved for
10.0.0.0	Network address
10.0.0.1	Internal communication
10.0.0.2	Domain Name System (DNS) resolution
10.0.0.3	Future use
10.0.0.255	Network broadcast address

Route tables and routes

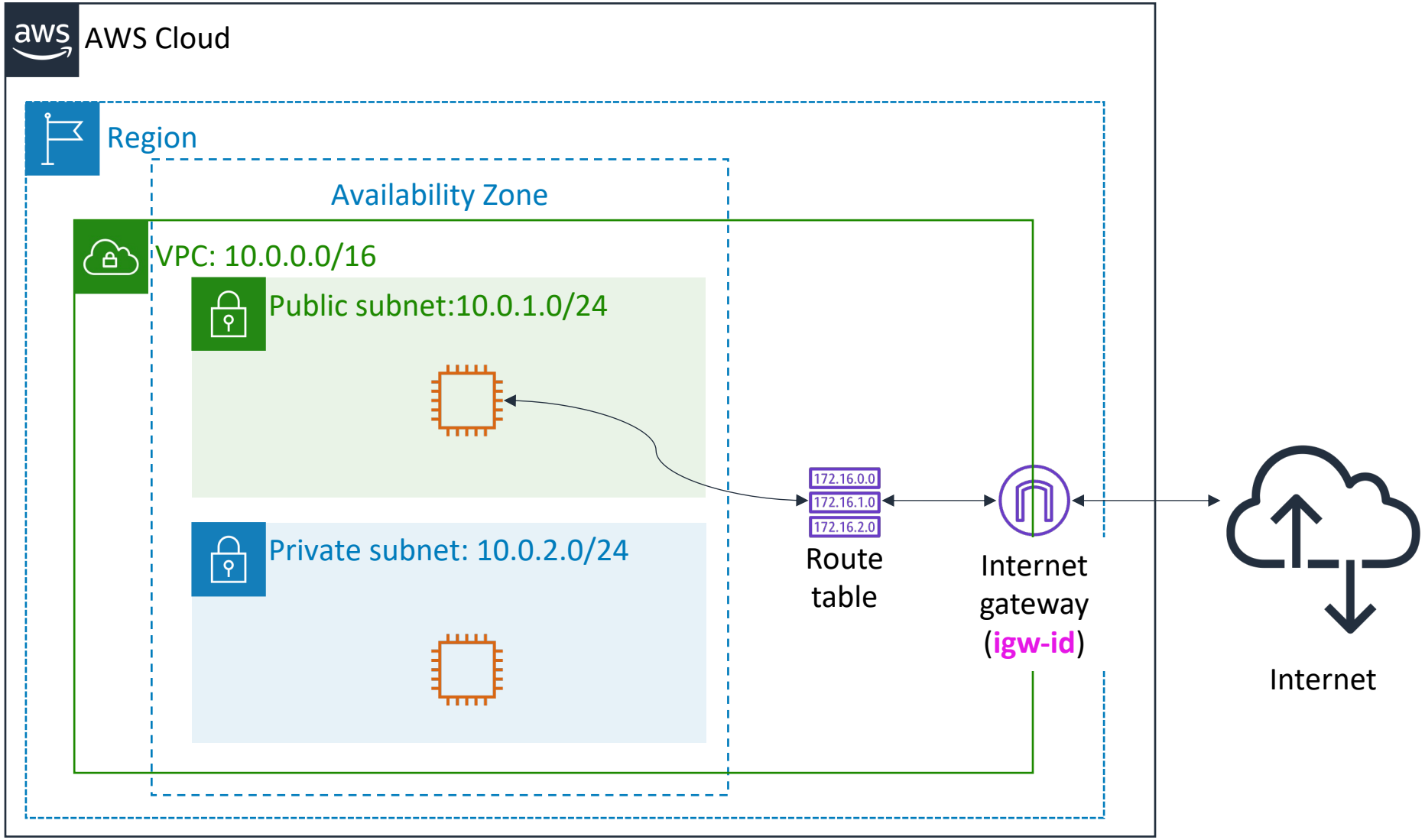
- A **route table** contains a set of rules (or routes) that **you can configure** to direct network traffic from your subnet.
- Each **route** specifies a destination and a target.
- By default, every route table contains a **local route** for communication within the VPC.
- Each **subnet must be associated with a route table** (at most one).

Main (Default) Route Table

Destination	Target
10.0.0.0/16	local

VPC CIDR block

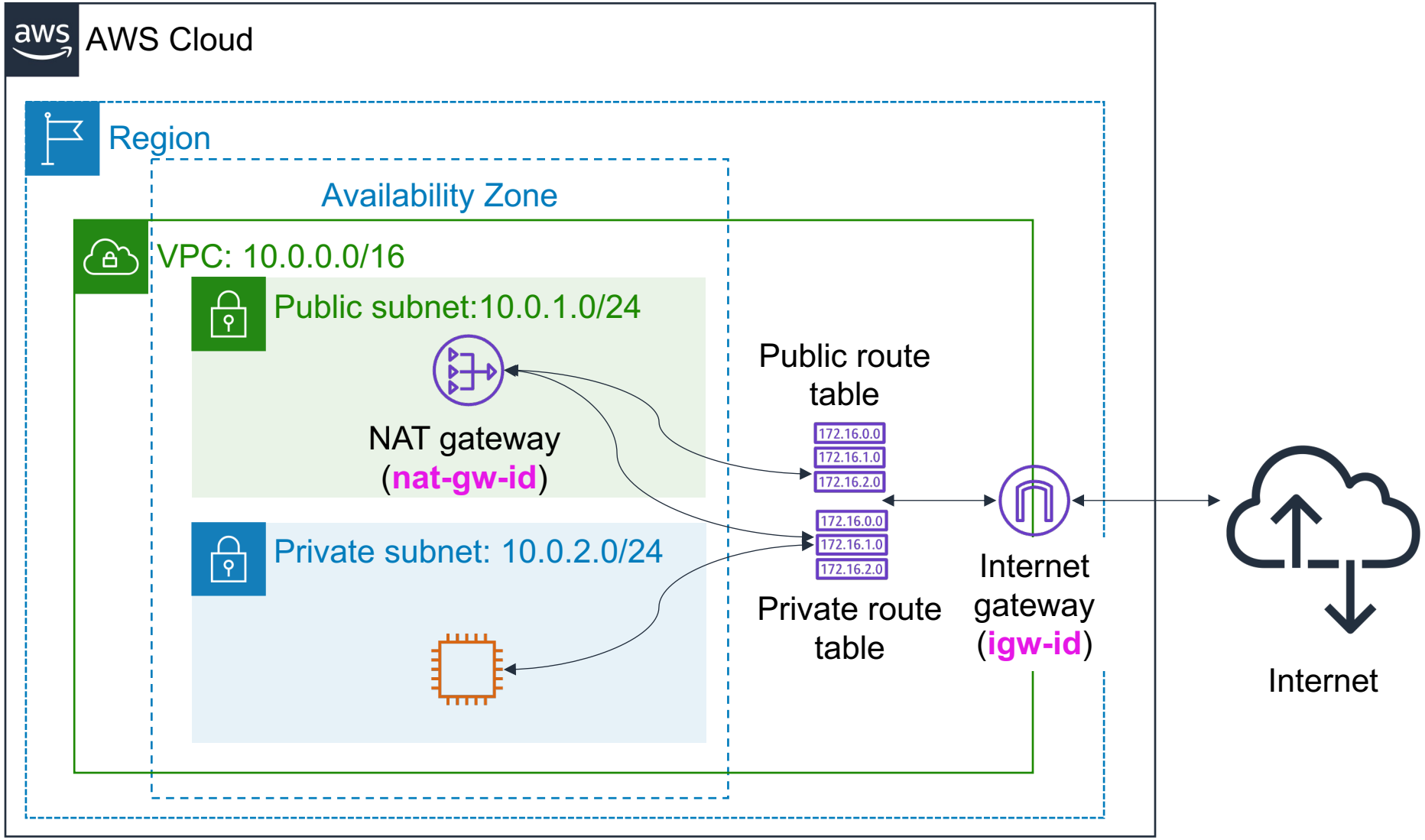
Internet gateway



Public Subnet Route Table

Destination	Target
10.0.0.0/16	local
0.0.0.0/0	igw-id

Network address translation (NAT) gateway



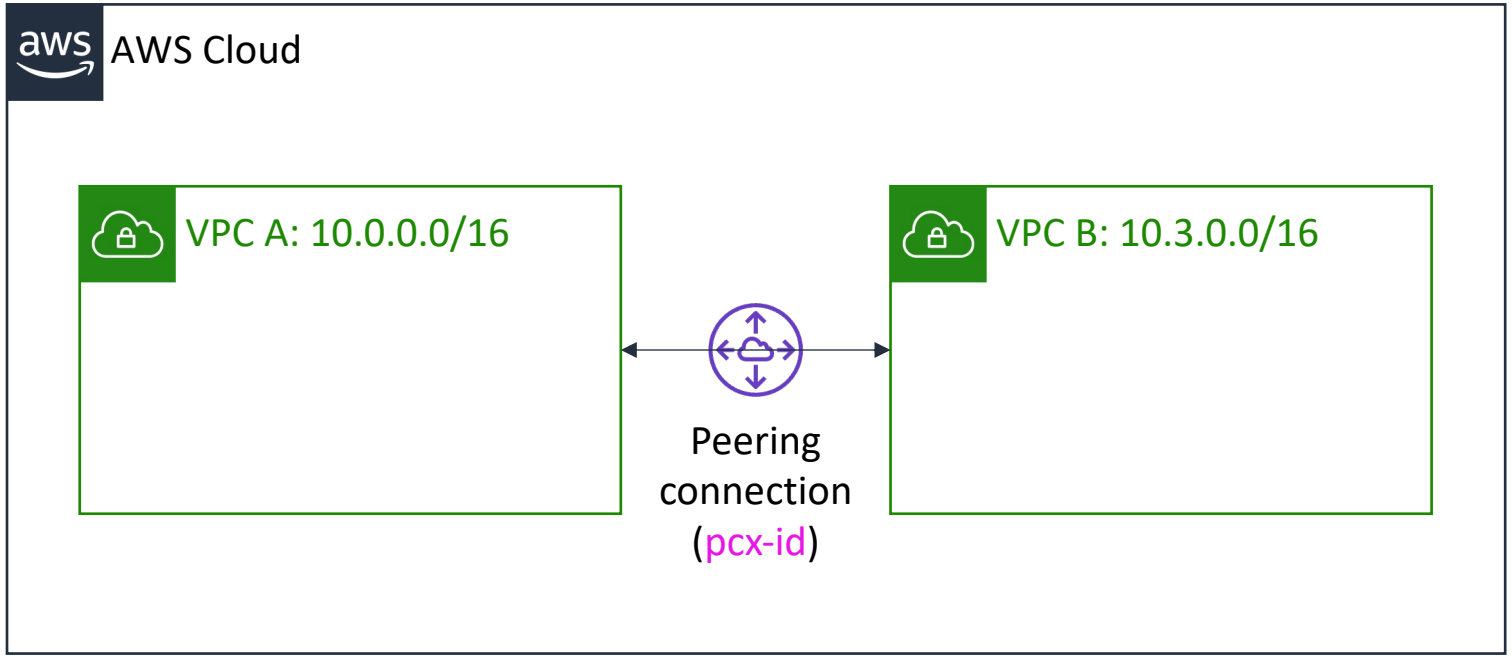
Public Subnet Route Table

Destination	Target
10.0.0.0/16	local
0.0.0.0/0	igw-id

Private Subnet Route Table

Destination	Target
10.0.0.0/16	local
0.0.0.0/0	nat-gw-id

VPC peering



Route Table for VPC A

Destination	Target
10.0.0.0/16	local
10.3.0.0/16	pcx-id

Route Table for VPC B

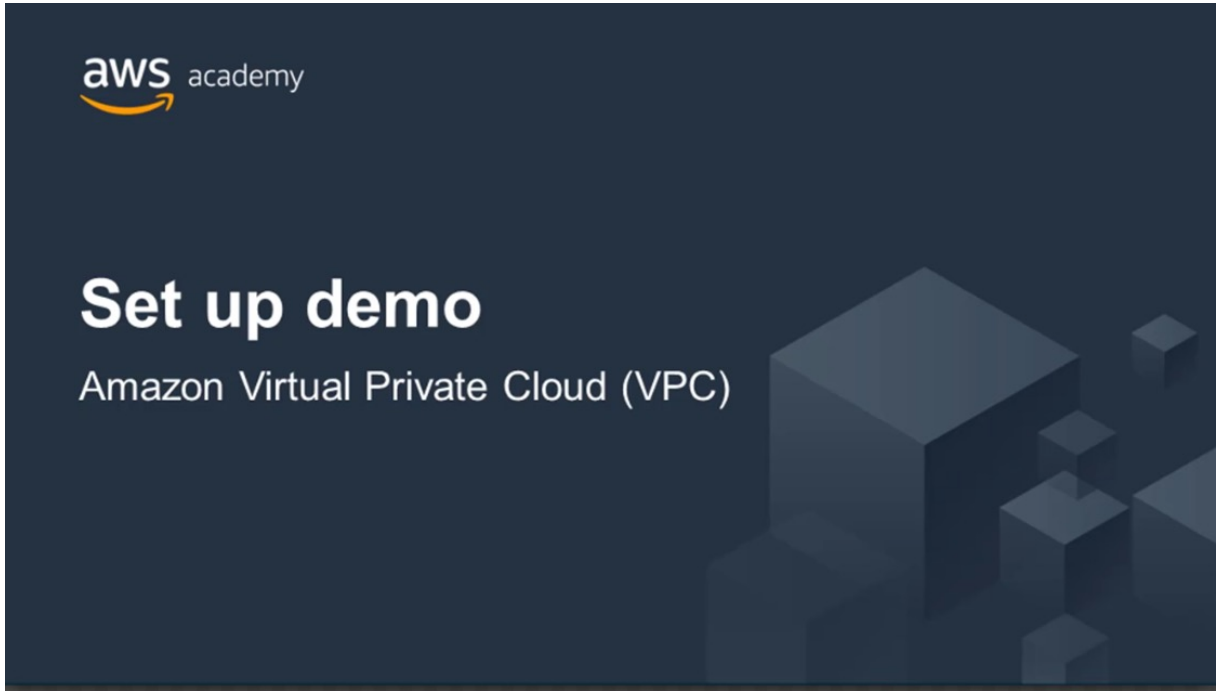
Destination	Target
10.3.0.0/16	local
10.0.0.0/16	pcx-id

You can connect VPCs in your own AWS account, between AWS accounts, or between AWS Regions.

Restrictions:

- IP spaces cannot overlap.
- Transitive peering is not supported.
- You can only have one peering resource between the same two VPCs.

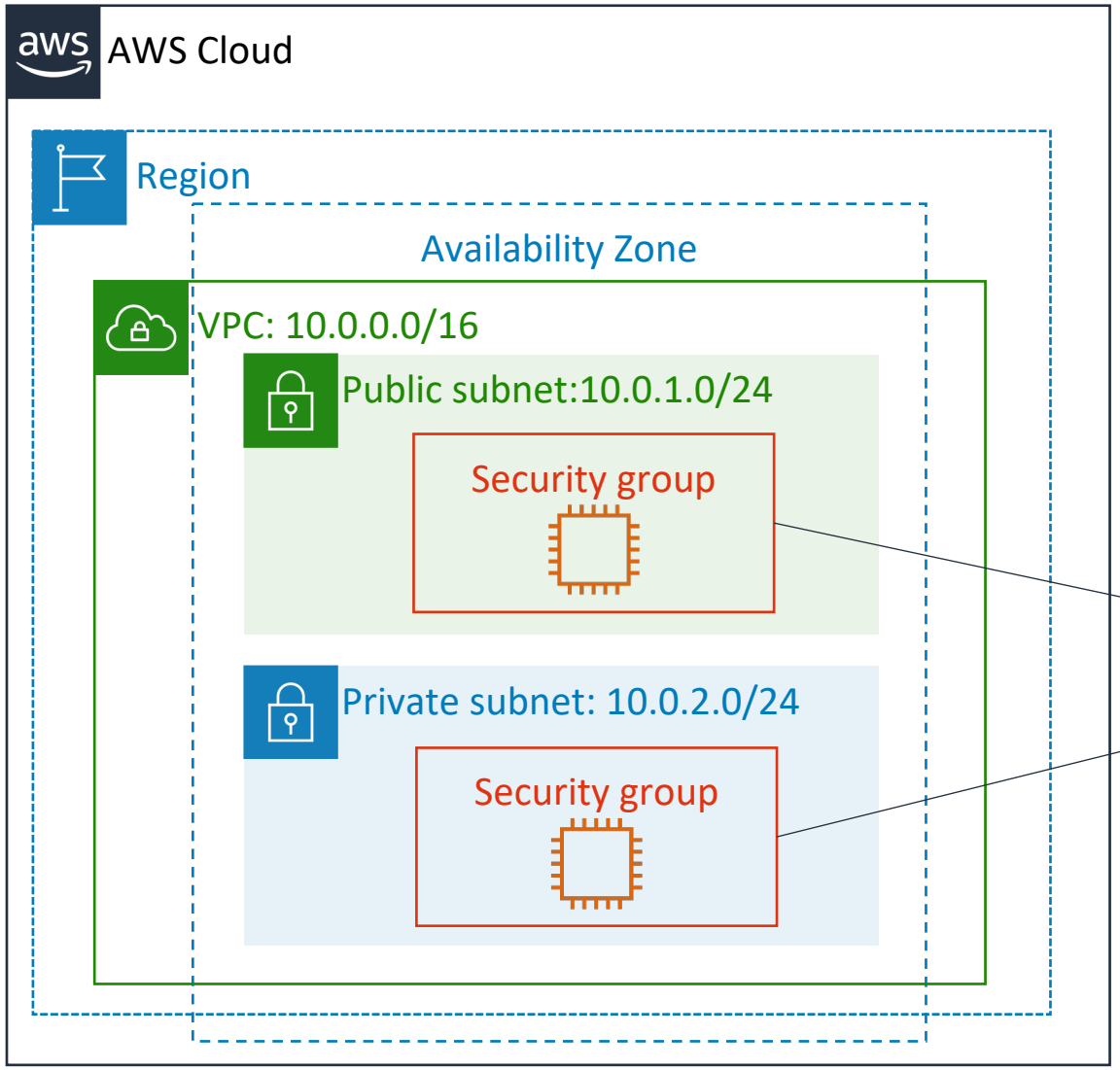
Recorded Amazon VPC demonstration



Networking

VPC security

Security groups



Security groups act at the **instance level**.

Security groups

- Security groups have **rules** that control inbound and outbound instance traffic.
- Default security groups **deny all inbound** traffic and **allow all outbound** traffic.
- Security groups are **stateful**.

Inbound			
Source	Protocol	Port Range	Description
sg-*xxxxxxxx*	All	All	Allow inbound traffic from network interfaces assigned to the same security group.

Outbound			
Destination	Protocol	Port Range	Description
0.0.0.0/0	All	All	Allow all outbound IPv4 traffic.
::/0	All	All	Allow all outbound IPv6 traffic.

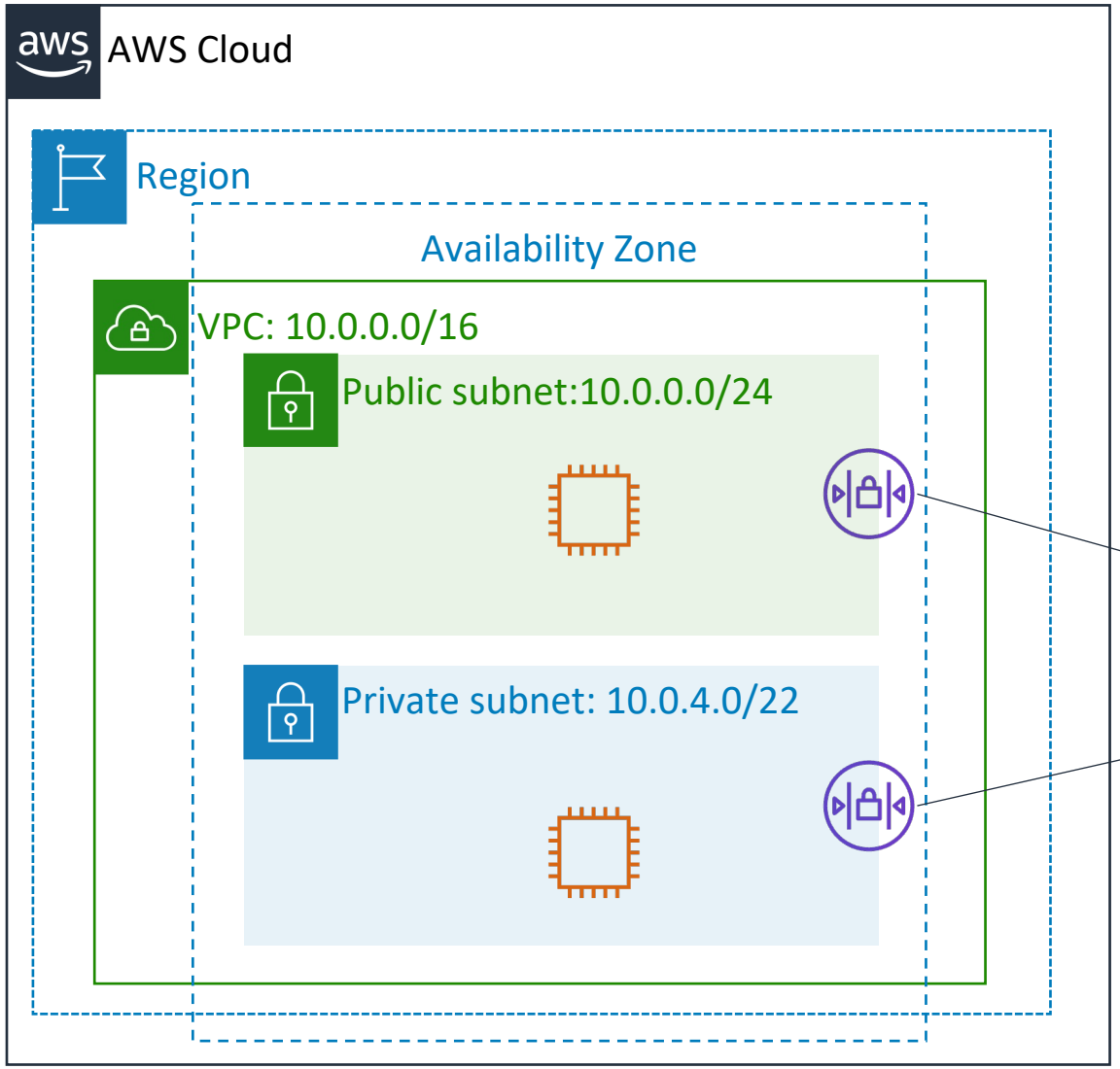
Custom security group examples

- You can **specify allow** rules, but not deny rules.
- **All rules are evaluated** before the decision to allow traffic.

Inbound			
Source	Protocol	Port Range	Description
0.0.0.0/0	TCP	80	Allow inbound HTTP access from all IPv4 addresses
0.0.0.0/0	TCP	443	Allow inbound HTTPS access from all IPv4 addresses
Your network's public IPv4 address range	TCP	22	Allow inbound SSH access to Linux instances from IPv4 IP addresses in your network (over the internet gateway)

Outbound			
Destination	Protocol	Port Range	Description
The ID of the security group for your Microsoft SQL Server database servers	TCP	1433	Allow outbound Microsoft SQL Server access to instances in the specified security group

Network access control lists (network ACLs)



Network ACLs act at the **subnet level**.

Network access control lists (network ACLs)

- A network ACL has **separate inbound and outbound rules**, and each rule can either **allow or deny traffic**.
- **Default** network ACLs **allow** all inbound and outbound IPv4 traffic.
- Network ACLs are **stateless**.

Inbound					
Rule	Type	Protocol	Port Range	Source	Allow/Deny
100	All IPv4 traffic	All	All	0.0.0.0/0	ALLOW
*	All IPv4 traffic	All	All	0.0.0.0/0	DENY

Outbound					
Rule	Type	Protocol	Port Range	Destination	Allow/Deny
100	All IPv4 traffic	All	All	0.0.0.0/0	ALLOW
*	All IPv4 traffic	All	All	0.0.0.0/0	DENY

Custom network ACLs examples

- **Custom** network ACLs **deny** all inbound and outbound traffic until you add rules.
- You can specify **both allow and deny** rules.
- Rules are evaluated in number order, starting with the **lowest number**.

Inbound					
Rule	Type	Protocol	Port Range	Source	Allow/Deny
100	HTTPS	TCP	443	0.0.0.0/0	ALLOW
120	SSH	TCP	22	192.0.2.0/24	ALLOW
*	All IPv4 traffic	All	All	0.0.0.0/0	DENY

Outbound					
Rule	Type	Protocol	Port Range	Destination	Allow/Deny
100	HTTPS	TCP	443	0.0.0.0/0	ALLOW
120	SSH	TCP	22	192.0.2.0/24	ALLOW
*	All IPv4 traffic	All	All	0.0.0.0/0	DENY

Security groups versus network ACLs

Attribute	Security Groups	Network ACLs
Scope	Instance level	Subnet level
Supported Rules	Allow rules only	Allow and deny rules
State	Stateful (return traffic is automatically allowed, regardless of rules)	Stateless (return traffic must be explicitly allowed by rules)
Order of Rules	All rules are evaluated before decision to allow traffic	Rules are evaluated in number order before decision to allow traffic

Lab 2:
Build Your VPC and Launch a Web Server
(~ 30 mins)

Lab 2: Scenario

In this lab, you use Amazon VPC to create your own VPC and add some components to produce a customized network. You create a security group for your VPC. You also create an EC2 instance and configure it to run a web server and to use the security group. You then launch the EC2 instance into the VPC.

Amazon VPC

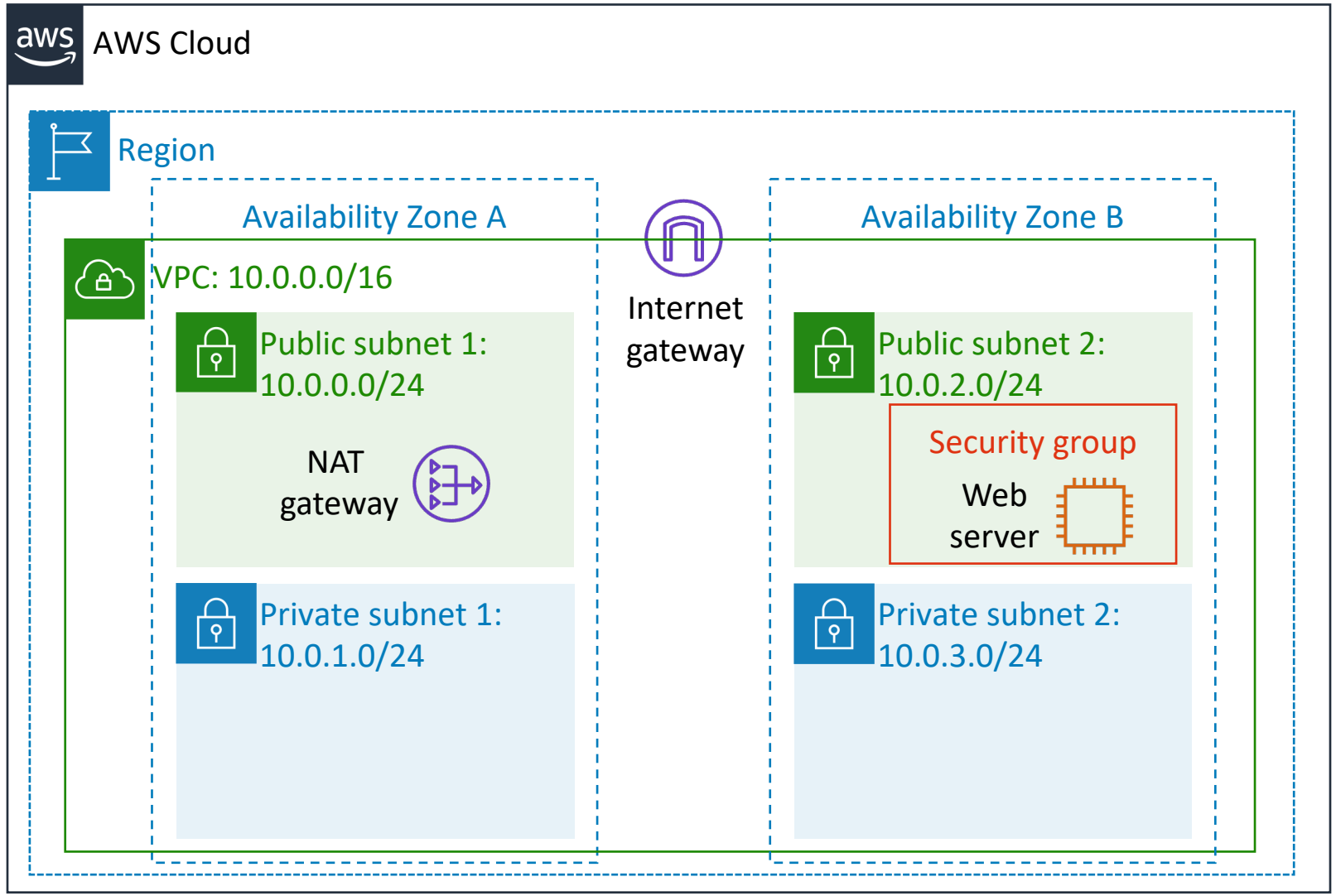
Amazon EC2

Lab 2: Tasks

- Create a VPC.
- Create additional subnets.
- Create a VPC security group.
- Launch a web server instance.

Security group

Lab 2: Final product



Public Route Table

Destination	Target
10.0.0.0/16	Local
0.0.0.0/0	Internet gateway

Private Route Table

Destination	Target
10.0.0.0/16	Local
0.0.0.0/0	NAT gateway

Additional resources

- Amazon VPC overview page
- Amazon Virtual Private Cloud Connectivity Options whitepaper
- One to Many: Evolving VPC Design AWS Architecture blog post
- Amazon VPC User Guide
- Amazon CloudFront overview page

Compute

Compute services overview

AWS compute services

Amazon EC2

Amazon EC2 Auto Scaling

Amazon Elastic Container Registry (Amazon ECR)

Amazon Elastic Container Service (Amazon ECS)

VMware Cloud on AWS

AWS Elastic Beanstalk

AWS Lambda

Amazon Elastic Kubernetes Service (Amazon EKS)

Amazon Lightsail

AWS Batch

AWS Fargate

AWS Outposts

AWS Serverless Application Repository

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 • 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 or services that you use will depend on your use case
- Some aspects to consider –
  - 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 lower performance efficiency
  - A good starting place—Understand the available compute options

Compute

Amazon EC2

Amazon Elastic Compute Cloud (Amazon EC2)

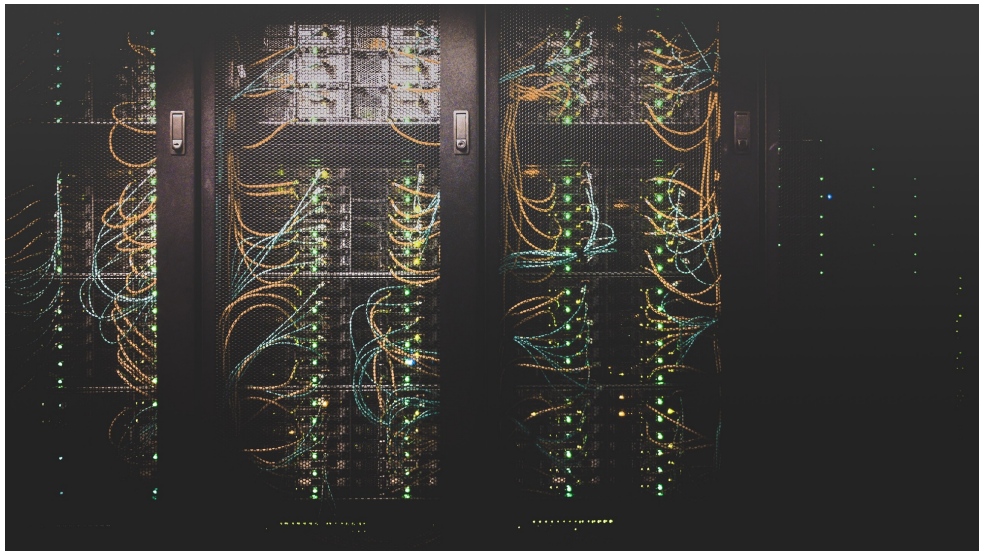
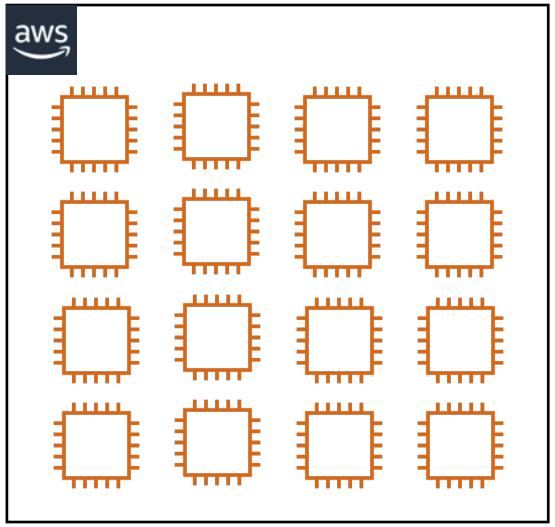


Photo by Taylor Vick on Unsplash

On-premises servers

Example uses of Amazon EC2 instances

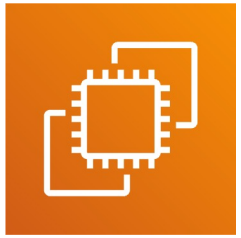
- ✓ Application server
- ✓ Web server
- ✓ Database server
- ✓ Game server
- ✓ Mail server
- ✓ Media server
- ✓ Catalog server
- ✓ File server
- ✓ Computing server
- ✓ Proxy server



Amazon EC2 instances

Photo by panumas nikhomkhai from Pexels

Amazon EC2 overview



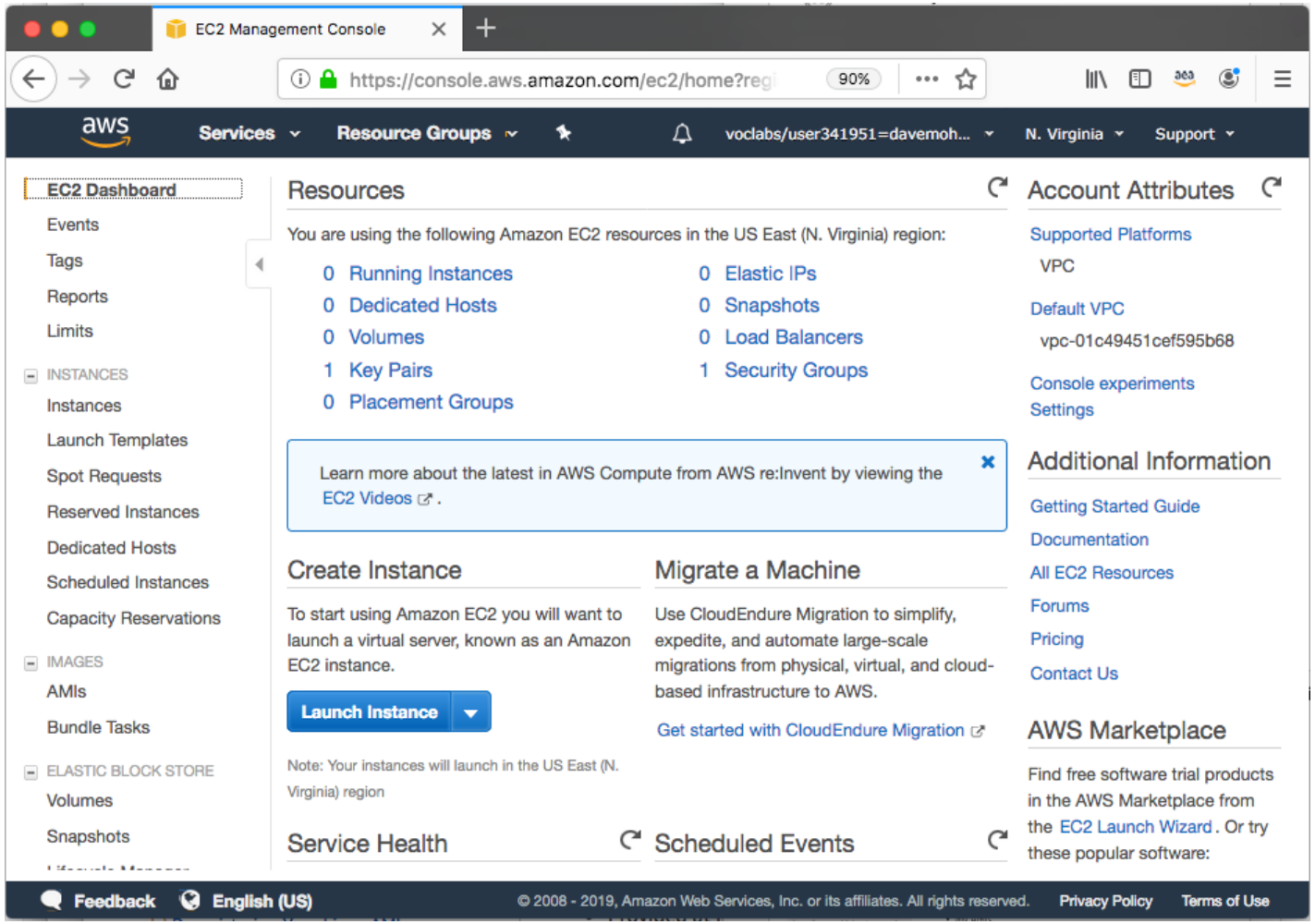
Amazon EC2

- **Amazon Elastic Compute Cloud (Amazon EC2)**
  - Provides virtual machines—referred to as EC2 instances—in the cloud.
  - Gives you *full control* over the guest operating system (Windows or Linux) on each instance.
- You 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.

Launching an Amazon EC2 instance

This section of the module walks through **nine key decisions** to make when you create an EC2 instance by using the AWS Management Console **Launch Instance Wizard.**

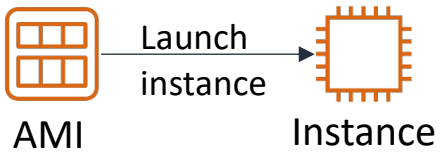
- Along the way, essential Amazon EC2 concepts will be explored.



1. Select an AMI

Choices made using the Launch Instance Wizard:

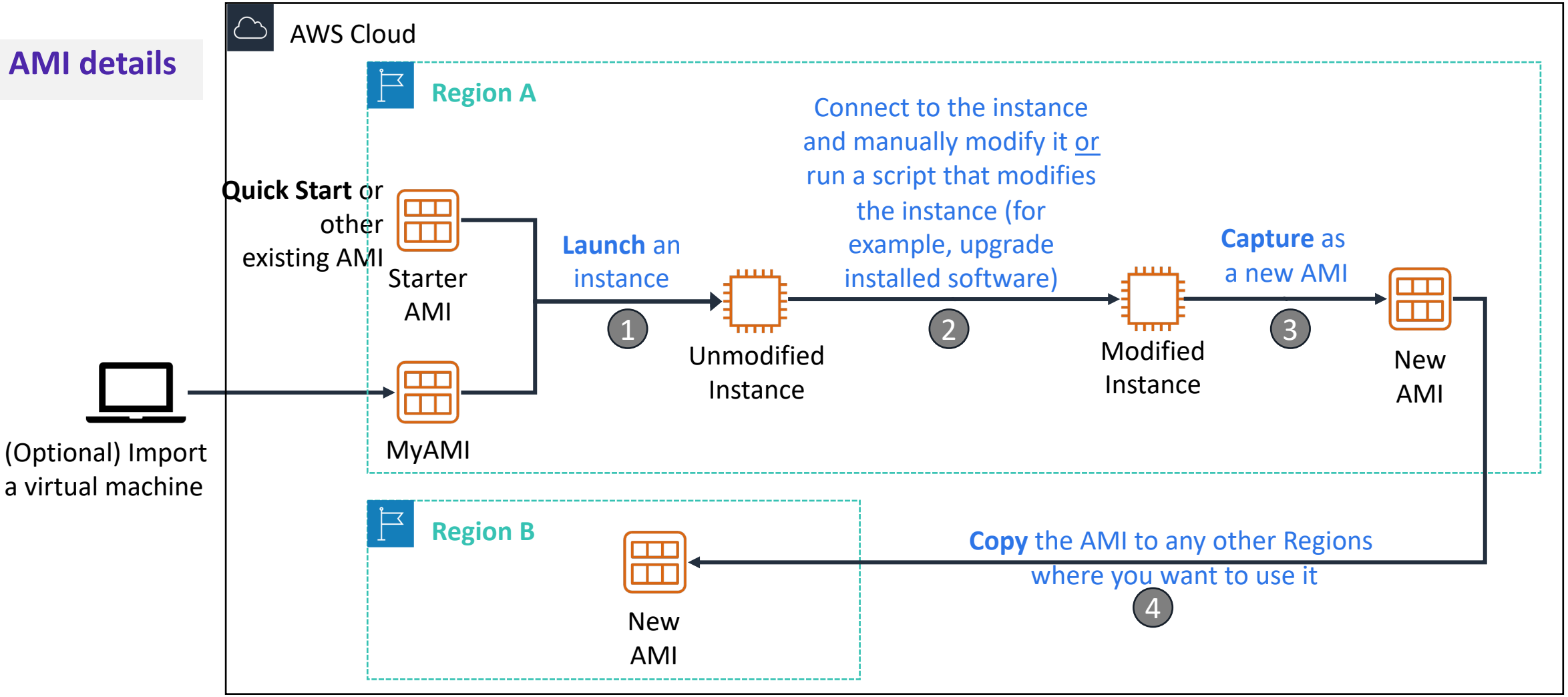
1. **AMI**
2. **Instance Type**
3. **Network settings**
4. **IAM role**
5. **User data**
6. **Storage options**
7. **Tags**
8. **Security group**
9. **Key pair**



- 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

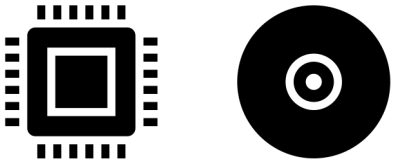


2. Select an instance type

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**

- Consider your use case
  - How will the EC2 instance you create be used?
- The **instance type** that you choose 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

aws academy

Instance type details

Instance type naming

- Example: **t3.large**
  - T is the family name
  - 3 is the generation number
  - Large is the size

Example instance sizes

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

aws academy

Instance type details

	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 types: Networking features

- The network bandwidth (Gbps) varies by instance type.
  - See Amazon EC2 Instance Types to compare.
- To maximize networking and bandwidth performance of your instance type:
  - If you have interdependent instances, launch them into a **cluster placement group**.
  - Enable enhanced networking.
- Enhanced networking types are supported on most instance types.
  - See the Networking and Storage Features documentation for details.
- 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.

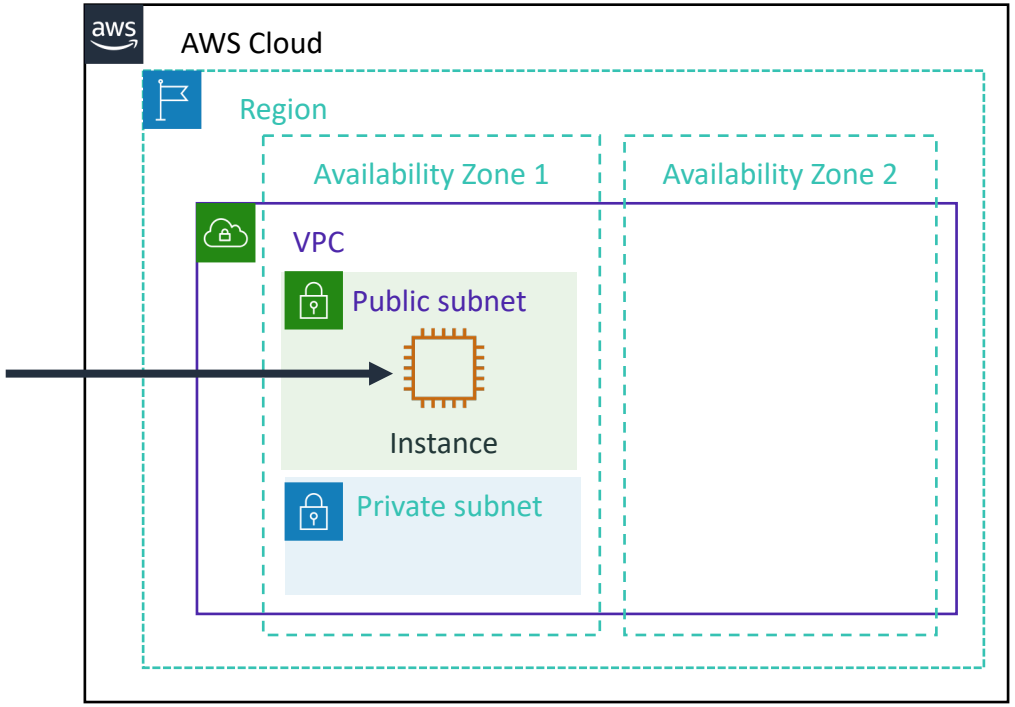
3. Specify network settings

Choices made by 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**

- 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



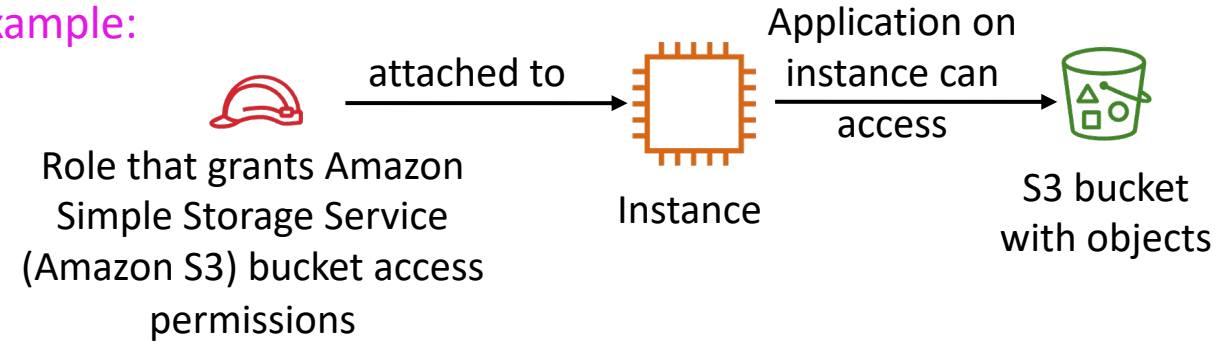
4. Attach IAM role (optional)

Choices made by 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**

- 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.

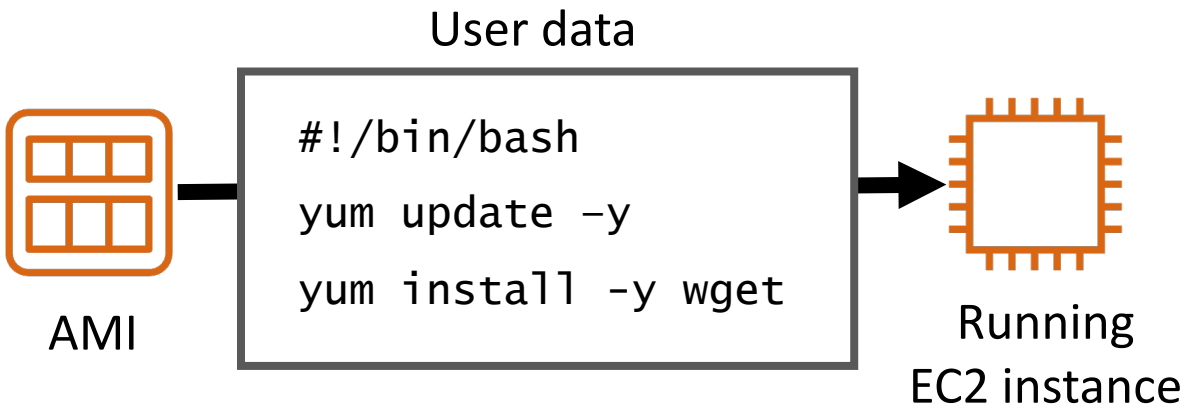
Example:



5. User data script (optional)

Choices made by 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**



- 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

6. Specify storage

Choices made by 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**

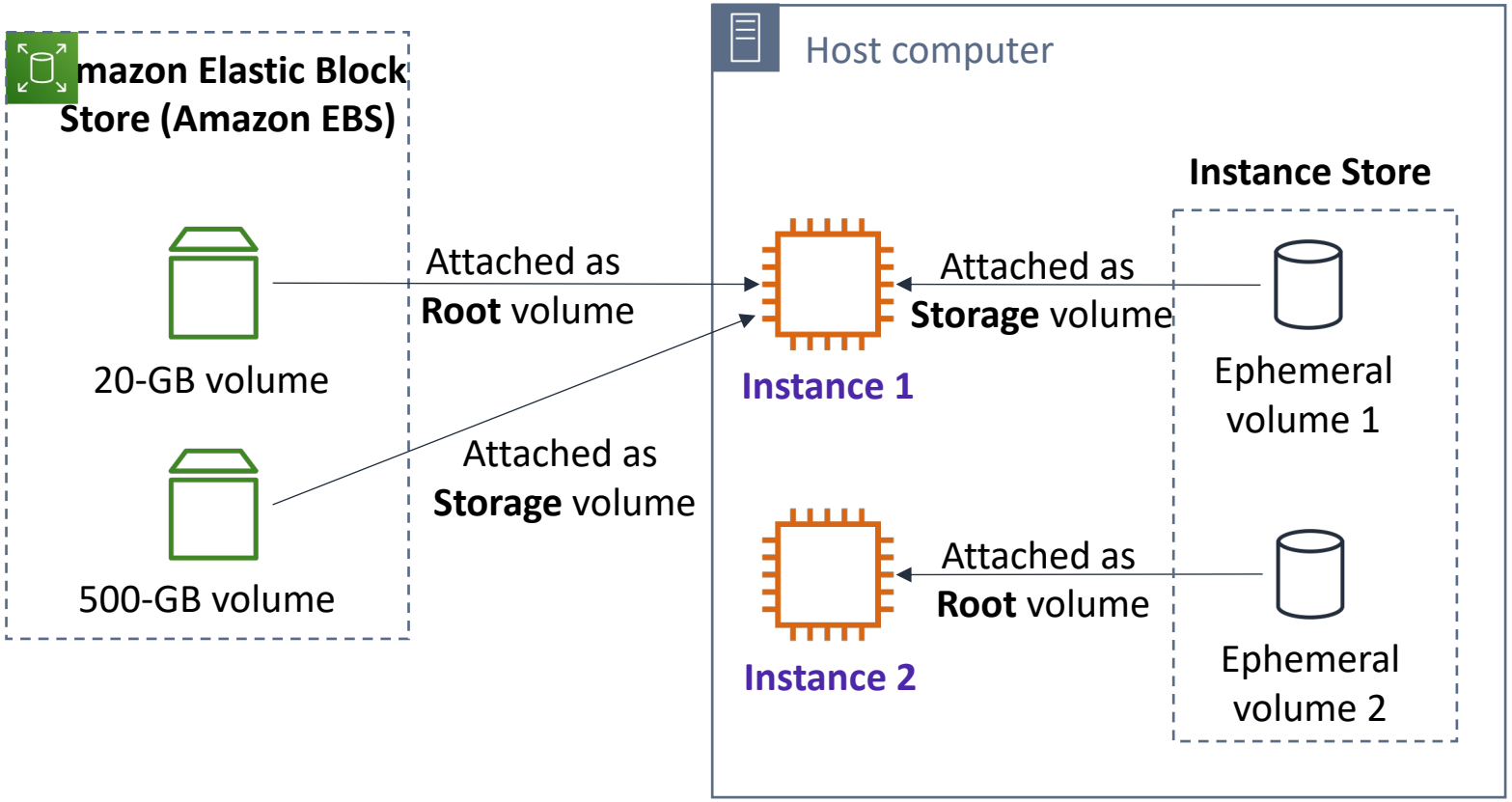
- Configure the root volume
  - Where the guest operating system is installed
- Attach additional storage volumes (optional)
  - AMI might already include more than one volume
- For each volume, specify:
  - The **size** of the disk (in GB)
  - The **volume type**
    - Different types of solid state 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)?



7. Add tags

Choices made by 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**

- 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:

8. Security group settings

Choices made by 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**

- 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.

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

9. Identify or create the key pair

Choices made by 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**

- 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.

mykey.pem

Amazon EC2 console view of a running EC2 instance

aws academy

Instances | EC2 Management C

https://console.aws.amazon.com/ec2/home?region=us-east-1#Instances:se

Search

aws Services Resource Groups voclabs/user341951=davemoh... N. Virginia Support

EC2 Dashboard
Events
Tags
Reports
Limits

INSTANCES
Instances
Launch Templates
Spot Requests
Reserved Instances
Dedicated Hosts
Scheduled Instances
Capacity Reservations

IMAGES
AMIs
Bundle Tasks

ELASTIC BLOCK STORE
Volumes
Snapshots

Launch Instance | Connect | Actions

search : i-092b6f3efba959a53 | Add filter | 1 to 1 of 1

Name	Instance ID	Instance Type	Instance State	Status Checks	Public DNS (IPv4)	IPv4 Public IP
	i-092b6f3efba959a53	t2.micro	running	Initializing	ec2-54-159-171-63.co...	54.159.171.63

Instance: i-092b6f3efba959a53 Public DNS: ec2-54-159-171-63.compute-1.amazonaws.com

Description | Status Checks | Monitoring | Tags

Instance ID	i-092b6f3efba959a53	Public DNS (IPv4)	ec2-54-159-171-63.compute-1.amazonaws.com
Instance state	running	IPv4 Public IP	54.159.171.63
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-e4e9859e
AMI ID	amzn2-ami-hvm-2.0.20190823.1-x86_64-gp2 (ami-0b69ea66ff7391e80)	Subnet ID	subnet-d22779fc
Platform	-	Network interfaces	eth0

Feedback English (US) © 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

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

- EC2 instances can also be created programmatically.

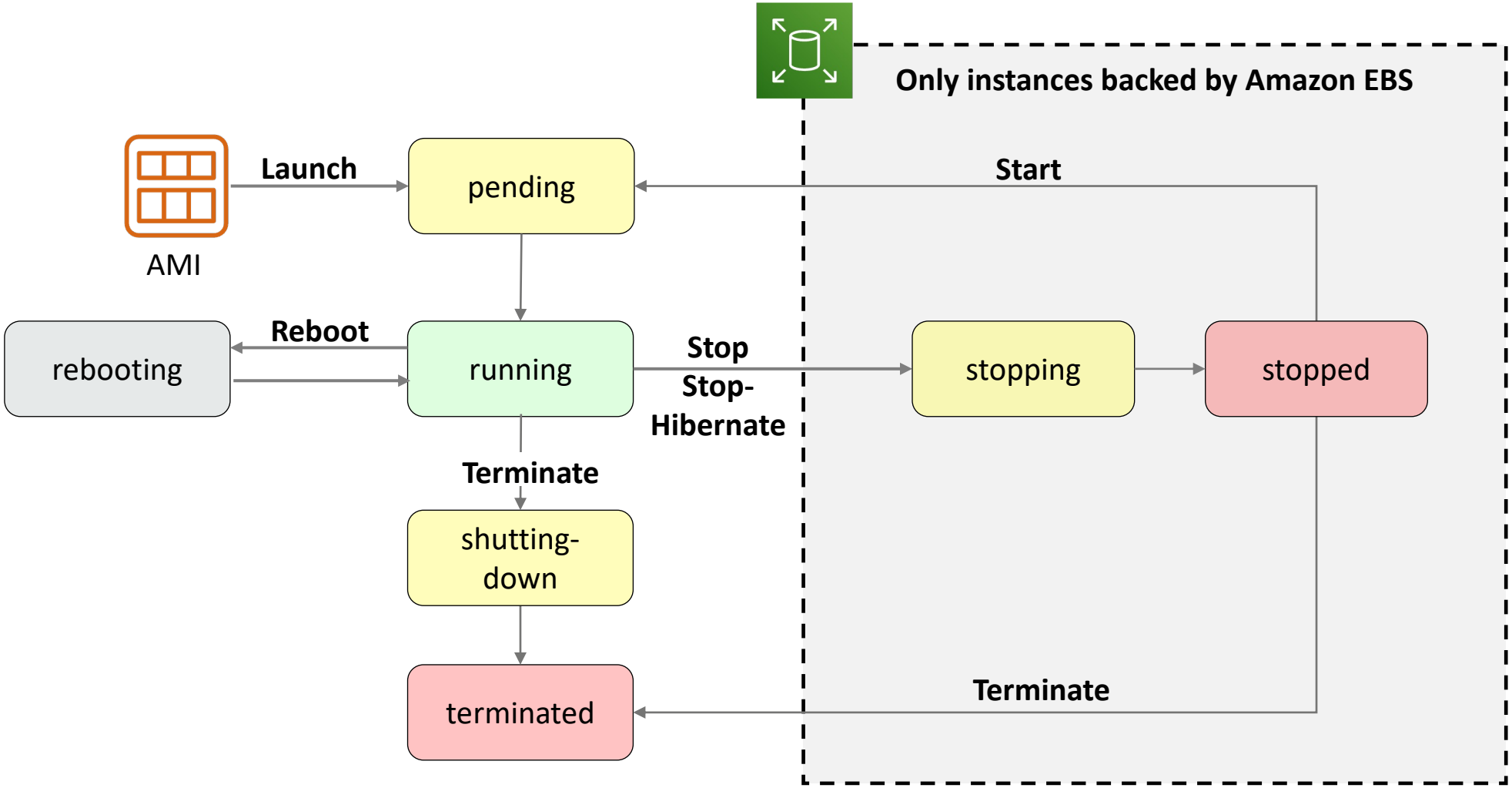
AWS Command Line Interface (AWS CLI)

- 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
```

Amazon EC2 instance lifecycle



Consider using an Elastic IP address

- **Rebooting** an instance will *not* change any IP addresses or DNS hostnames.
- When an instance is **stopped** and then **started** again –
  - The *public* IPv4 address and *external* DNS hostname will change.
  - The *private* IPv4 address and internal DNS hostname do *not* change.
- If you require a persistent public IP address –
  - Associate an **Elastic IP address** with the instance.
- Elastic IP address characteristics –
  - Can be associated with instances in the Region as needed.
  - Remains allocated to your account until you choose to release it.

Elastic IP Address

EC2 instance metadata

- **Instance metadata** is data about your 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.

Amazon CloudWatch for monitoring

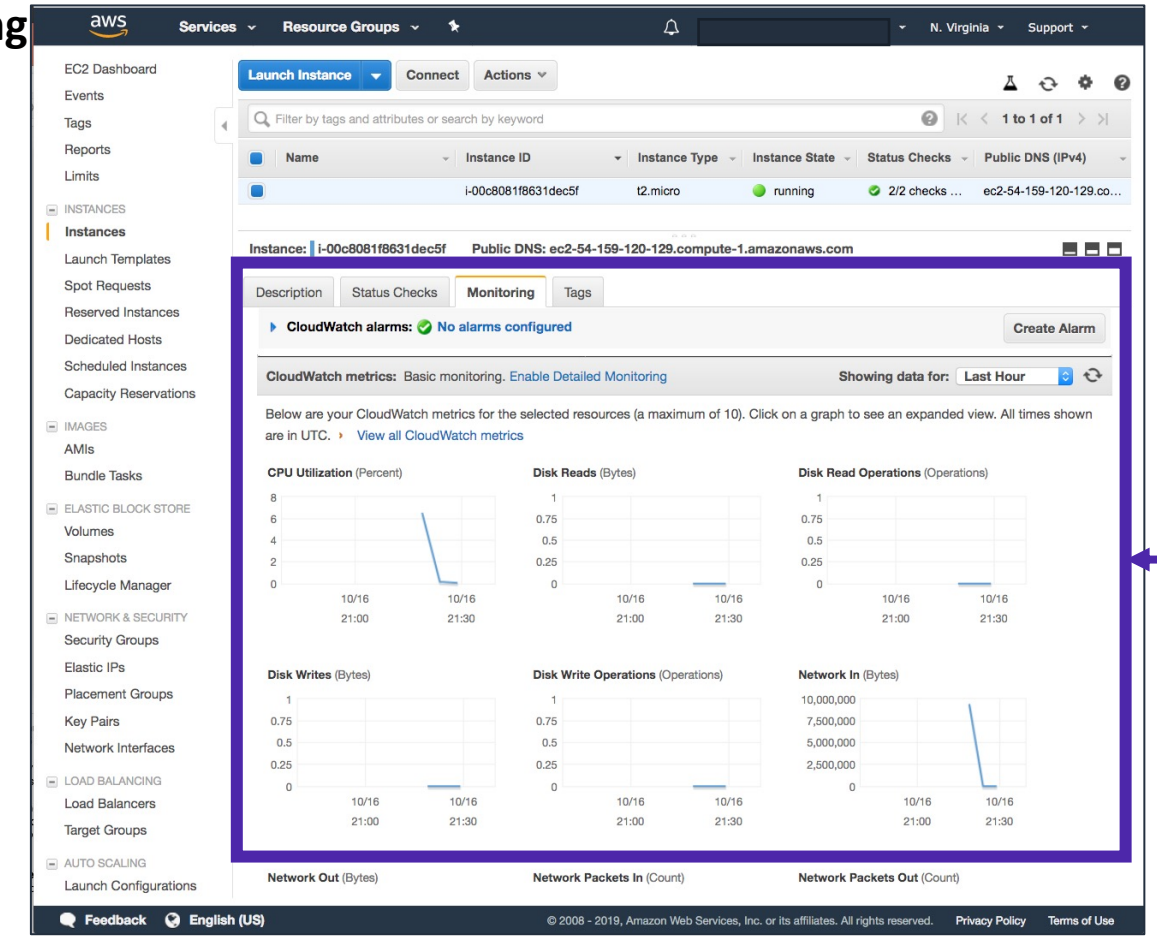
- Use **Amazon CloudWatch** to monitor EC2 instances
  - Provides near-real-time metrics
  - Provides charts in the Amazon EC2 console **Monitoring** tab that you can view
  - Maintains 15 months of historical data

- **Basic monitoring**
  - Default, no additional cost
  - Metric data sent to CloudWatch every 5 minutes

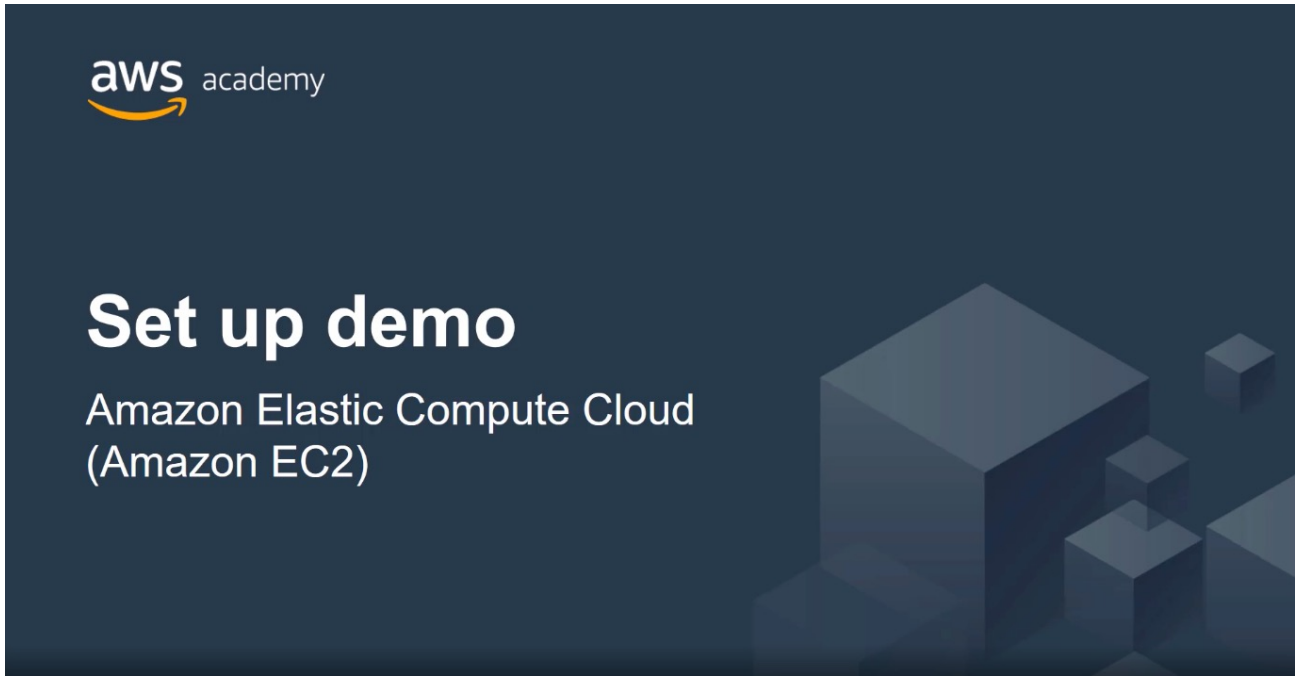
- **Detailed monitoring**
  - Fixed monthly rate for seven pre-selected metrics
  - Metric data delivered every 1 minute

Amazon CloudWatch

Instance with CloudWatch



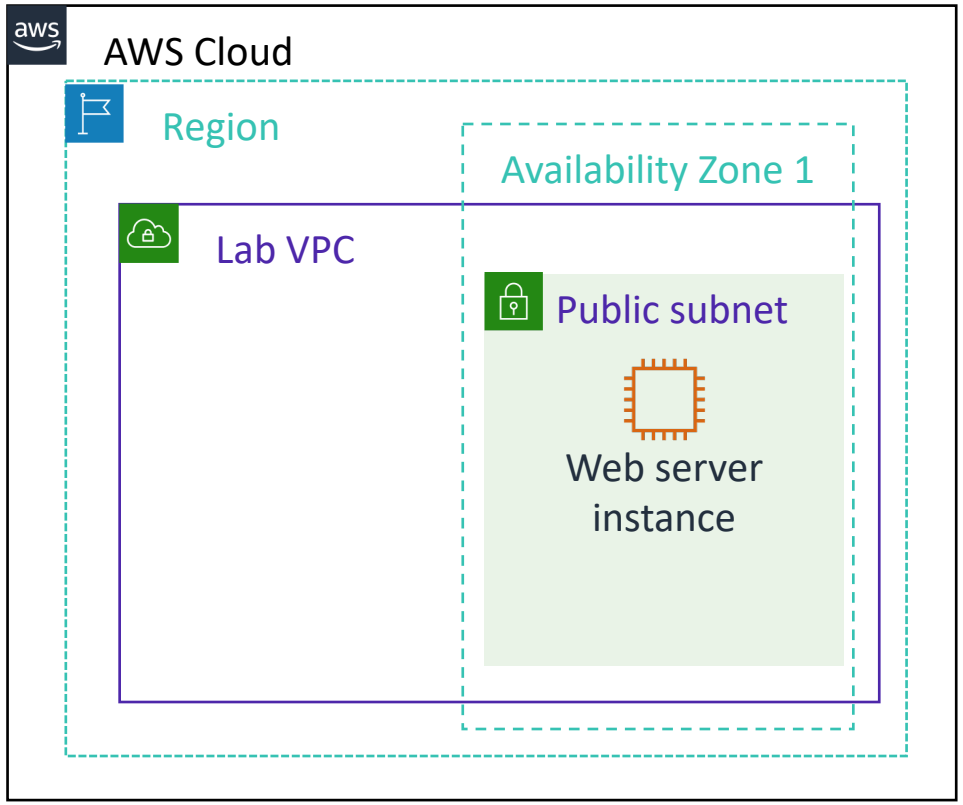
Recorded Amazon EC2 demonstration



Lab 3: Introduction to Amazon EC2 (~ 35 mins)

Lab 3 scenario

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



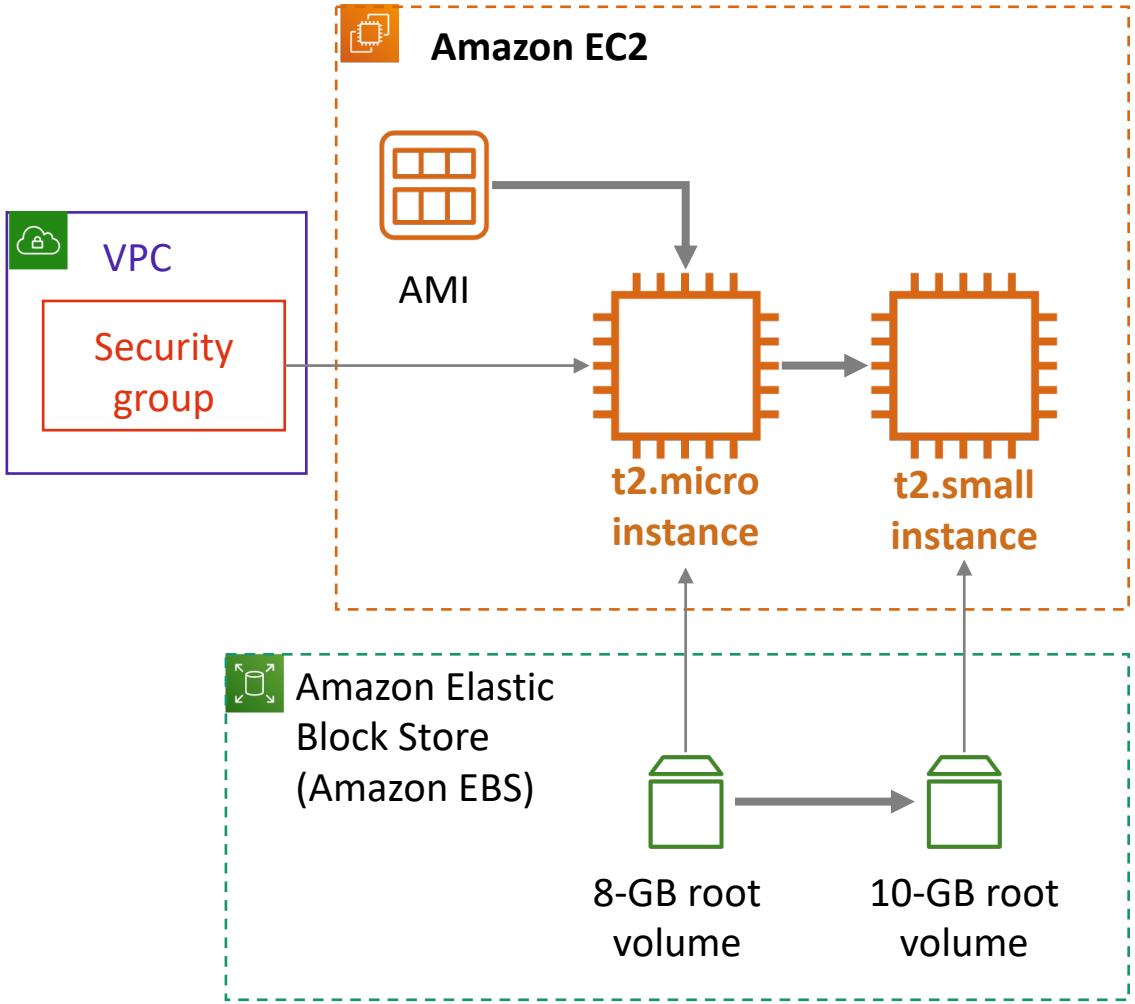
Lab 3: Tasks

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

Lab 3: Final product

By the end of the lab, you will have:

1. Launched an instance that is configured as a web server
2. Viewed the instance system log
3. Reconfigured a security group
4. Modified the instance type and root volume size



Additional resources

- Amazon EC2 Documentation
- Amazon EC2 Pricing
- Amazon ECS Workshop
- Running Containers on AWS
- Amazon EKS Workshop
- AWS Lambda Documentation
- AWS Elastic Beanstalk Documentation
- Cost Optimization Playbook

Thank you

aws academy