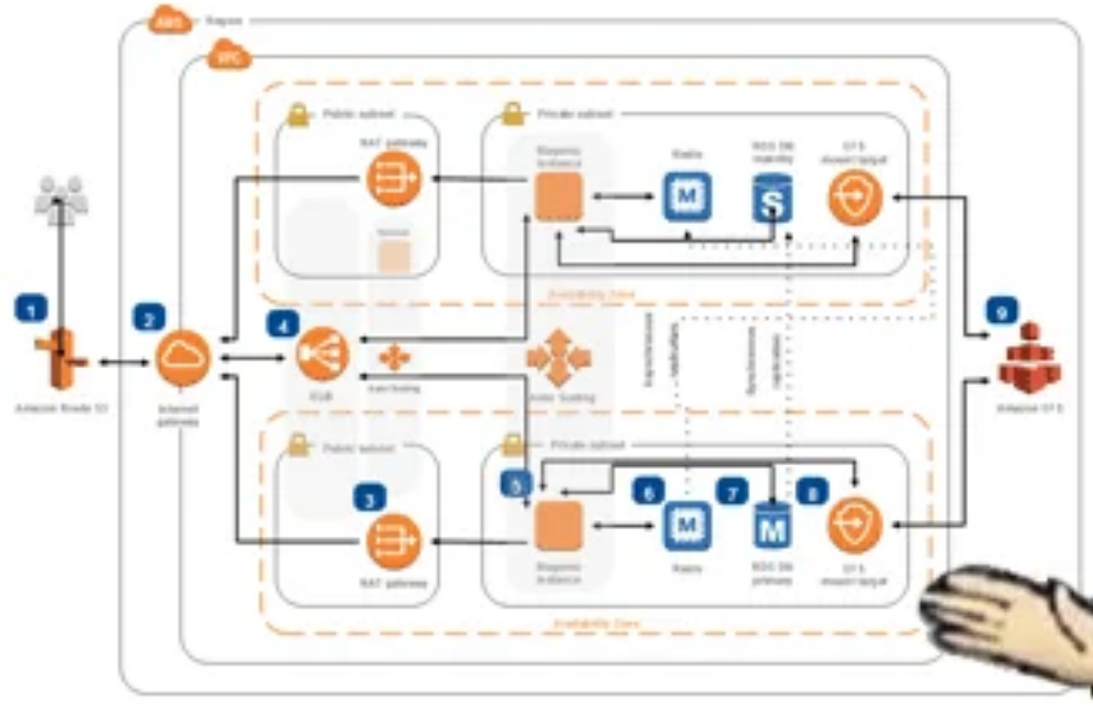
AWS 101

IAM, S3, VPC and EC2 -1

Proton 2024/10

So, What will you learn in this workshop ?

ME: I JUST NEED TO HOST 'HELLO WORLD' ON THE CLOUD.



AWS: NO PROBLEM. HAVE YOU CHECKED ALL OF OUR COOL NAMED PRODUCTS YOU'LL NEVER UNDERSTAND?

Let’s Get Started

Some Basic Knowledge

Tools

aws-cli and Terraform

- Every operation on the AWS Console GUI can be considered as an API call
    - aws-cli, a command line tool that can be used to control every aspect on your AWS account
- Terraform (from HashiCorp)
    - An Infrastructure-as-Code (IaC) tool that supports multiple different clouds.
    - IaC let you create reproducible and maintainable Infrastructure compare to using GUI to control and deploy resources
    - AWS does provide a native tool called CloudFormation, or use AWS CDK if the programming you prefer supports that

Tools

aws-cli config

- ~/.aws/config (Windows: C:\Users\Username\.aws\config
  - Config storage
  - Example Syntax

```
[profile default]
region = us-west-2
[sso-session example]
sso_start_url = https://something.awsapps.com/start#
sso_region = us-west-2
sso_registration_scopes = sso:account:access
[profile workshop]
sso_session = example
sso_account_id = 123456780000
sso_role_name = AdministratorAccess
```

Tools

aws-cli credentials

- ~/.aws/credentials (Windows: C:\Users\Username\.aws\credentials)
  - credential storage, sso session does not need this
  - Example syntax [default]

    aws_access_key_id = AKIAUBBSEEEBAUEDBX

    aws_secret_access_key = iojdzsfimcfdfsoiuhadsiuof

- You can use aws [--profile name] configure / sso configure to use the guided setup

Tools

Basic Terraform configs

- main.tf
  - Main part of your configurations
  - Can separate configs into submodules, then import from main.tf
- terraform.tfstate
  - Backend state file generate by Terraform, which saves the current infrastructure states, and will be compare to when applying new configs
  - Normally stored in save places, i.g. S3 Bucket, GCP Storage, for multiple access

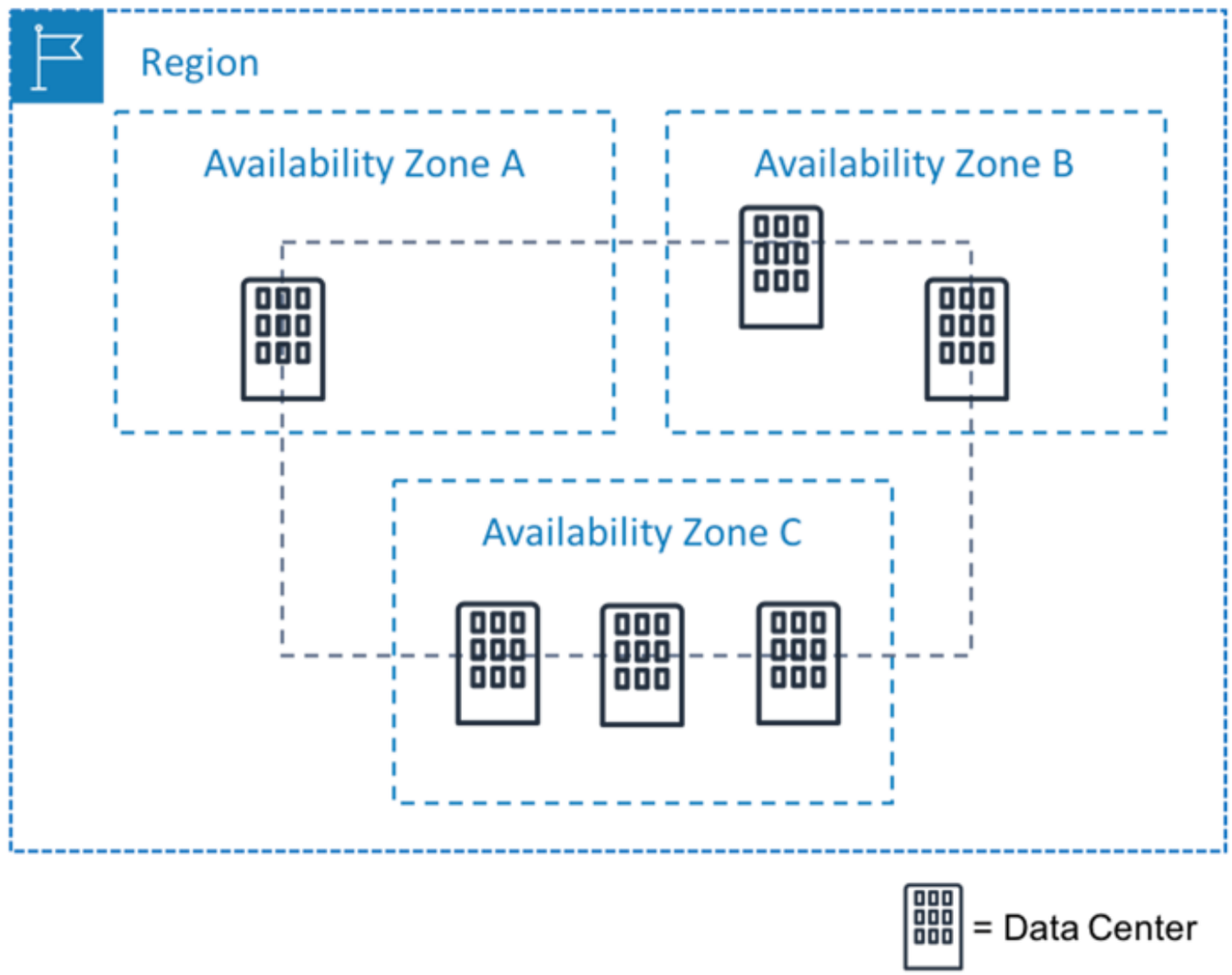
AWS Regions

And Availability Zones

- Region: A cluster of data centers in a specific geographic area

- Availability Zones: standalone or a set of data centers within a Region

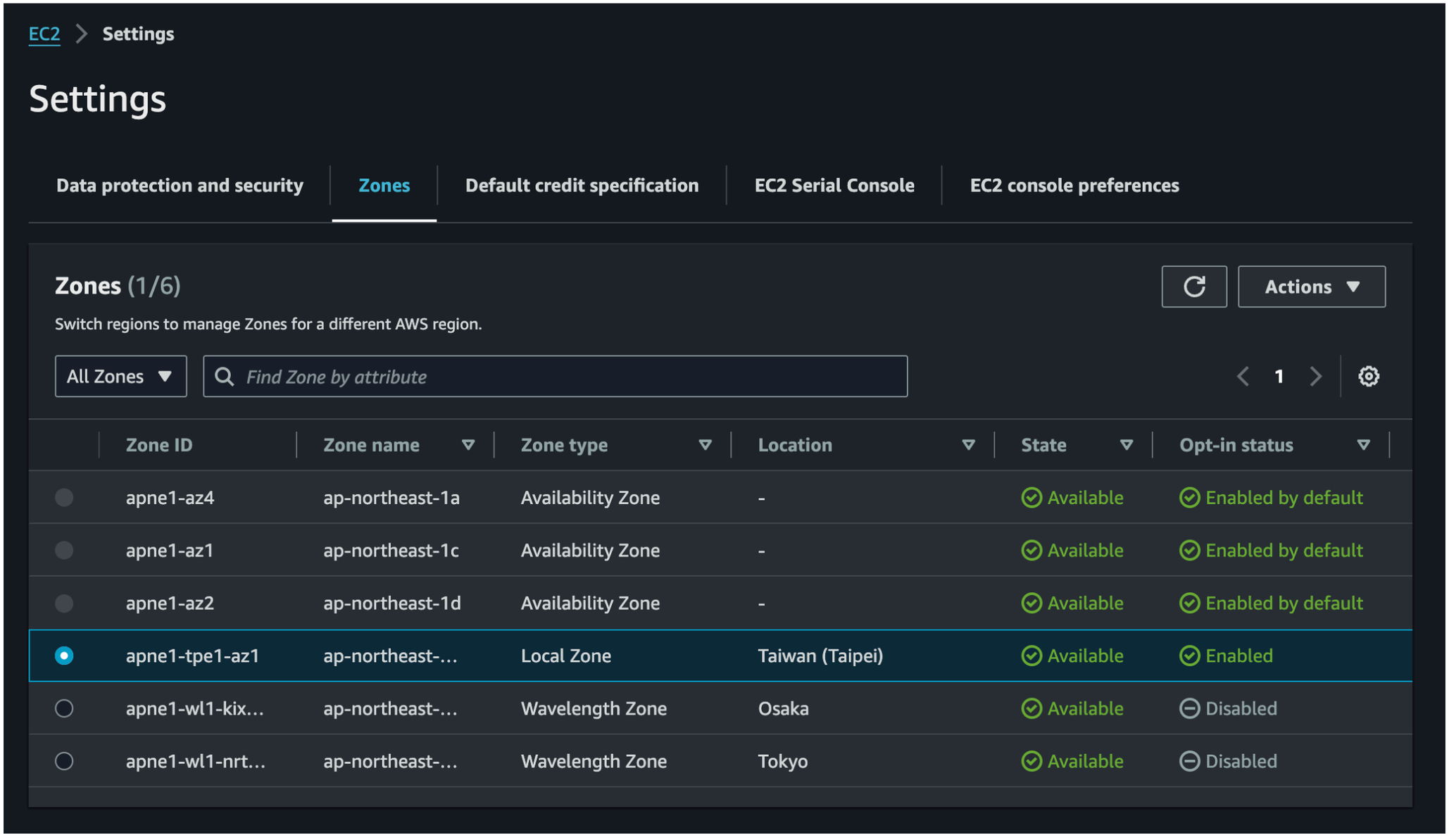
- Most of the time, a AWS region will contain at least 3 different availability zones



https://aws.amazon.com/about-aws/global-infrastructure/

AWS Local Zones

- Not a full-scale region, but have datacenter located at certain places
- Won’t have full AWS resources, but at least have EC2 support
- A better choice if you want to get closer to your end users, reducing latency



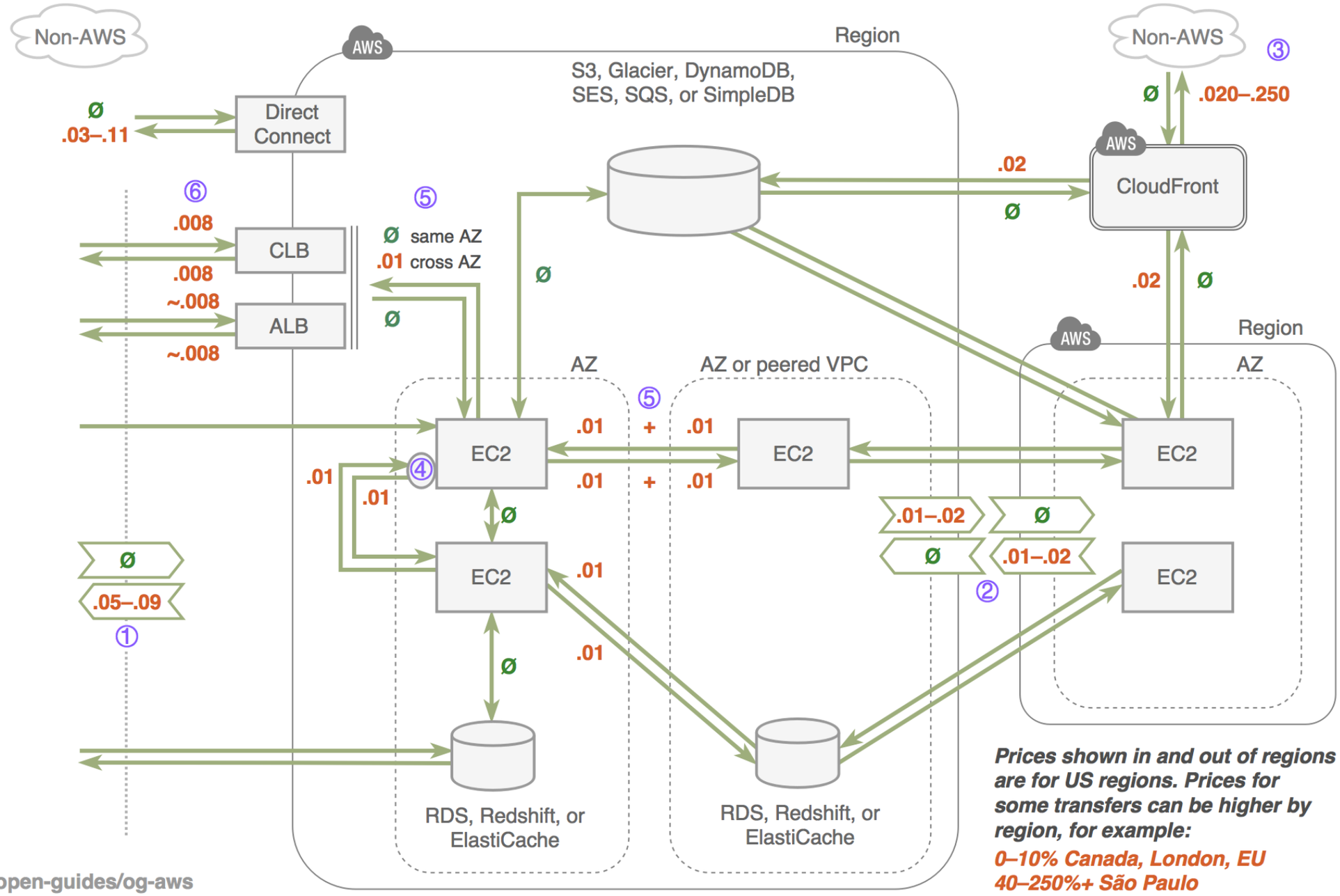
Data transfer Cost

Data transfers are not free, and can be expensive : (

AWS DATA TRANSFER COSTS

Numbers are data transfer in $/GB. Transaction and hourly prices are not shown. See notes.

- Ø **Free. Inbound traffic is mostly free —you pay on the way out. Some but not all internal traffic is free.**
- ① **Direct outbound data starts at $.09/GB for <10TB, and discounts with volume. First 1GB free.**
- ② **Region-to-region traffic is $.02/GB when it exits a region for indicated services except between us-east-1 and us-east-2, where it's $.01/GB.**
- ③ **Outbound CloudFront prices are highly variable by geography and regional edge cache and start at $.085/GB in US/Canada.**
- ④ **Internal traffic via public or elastic IPs incurs additional fees in both directions.**
- ⑤ **Cross-AZ EC2 traffic within a region costs as much as region-to-region! ELB-EC2 traffic is free except outbound crossing AZs.**
- ⑥ **Elastic Load Balancing: Classic LB is priced per GB. Application LB costs are in LCUs, not $/GB.**



Prices shown in and out of regions are for US regions. Prices for some transfers can be higher by region, for example:

0–10% Canada, London, EU
40–250%+ São Paulo

Credits and latest version: **github.com/open-guides/og-aws**
Last update: 2017-08-14

AWS Identity and Access Management

AWS IAM

What is this ?

- How does AWS describe this ?

Securely managed identities and access to AWS services and resources

- by AWS (written on their website)

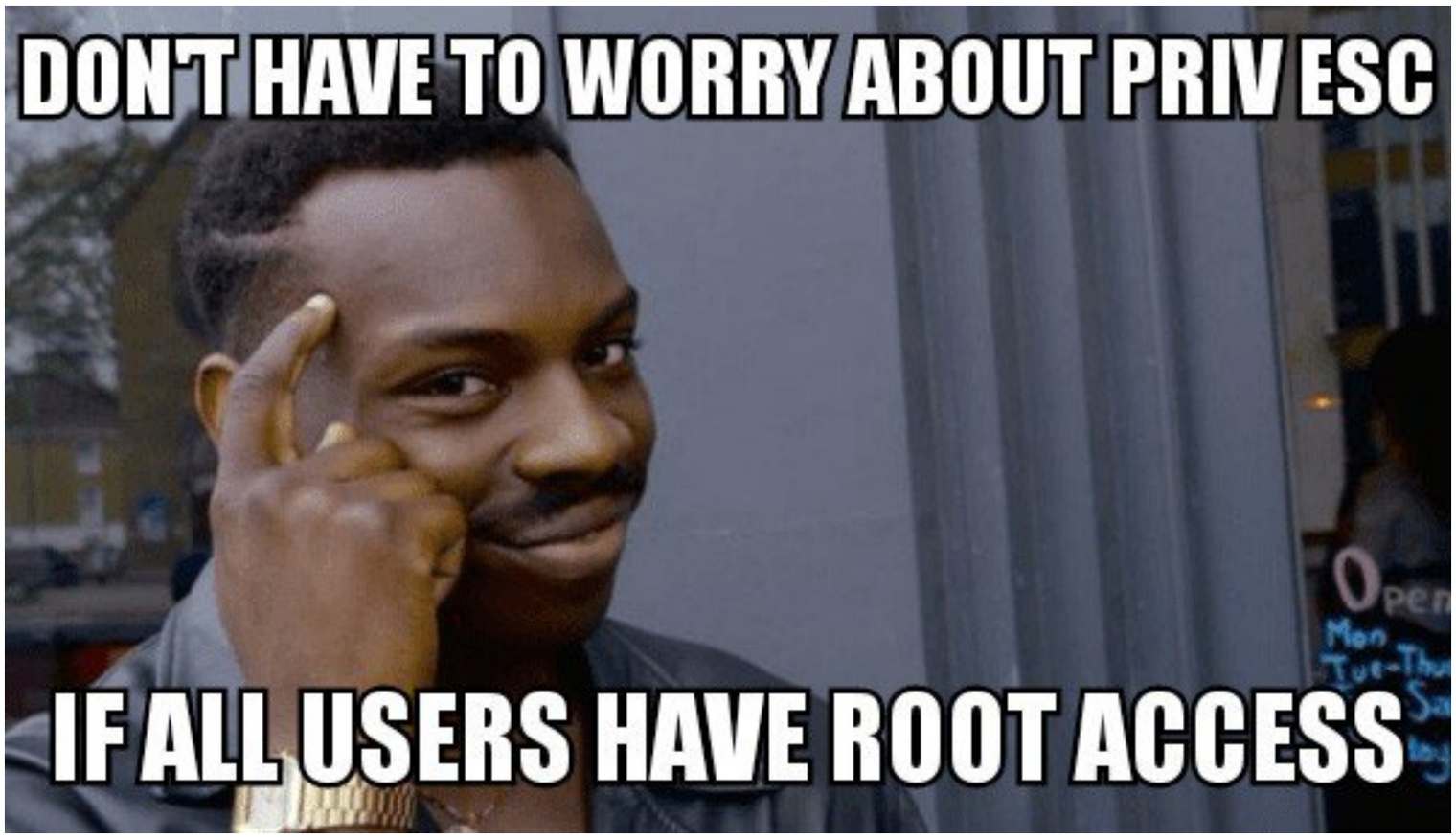
AWS IAM

So, what exactly does it do ?

- It provides methods to authenticate users
- It gives you control over your AWS resources permission sets
- Let you follow least privilege model during daily operations
- In Plain English:
    - IAM controls Who (Authentication) can do What (Permissions) in your AWS account

AWS IAM

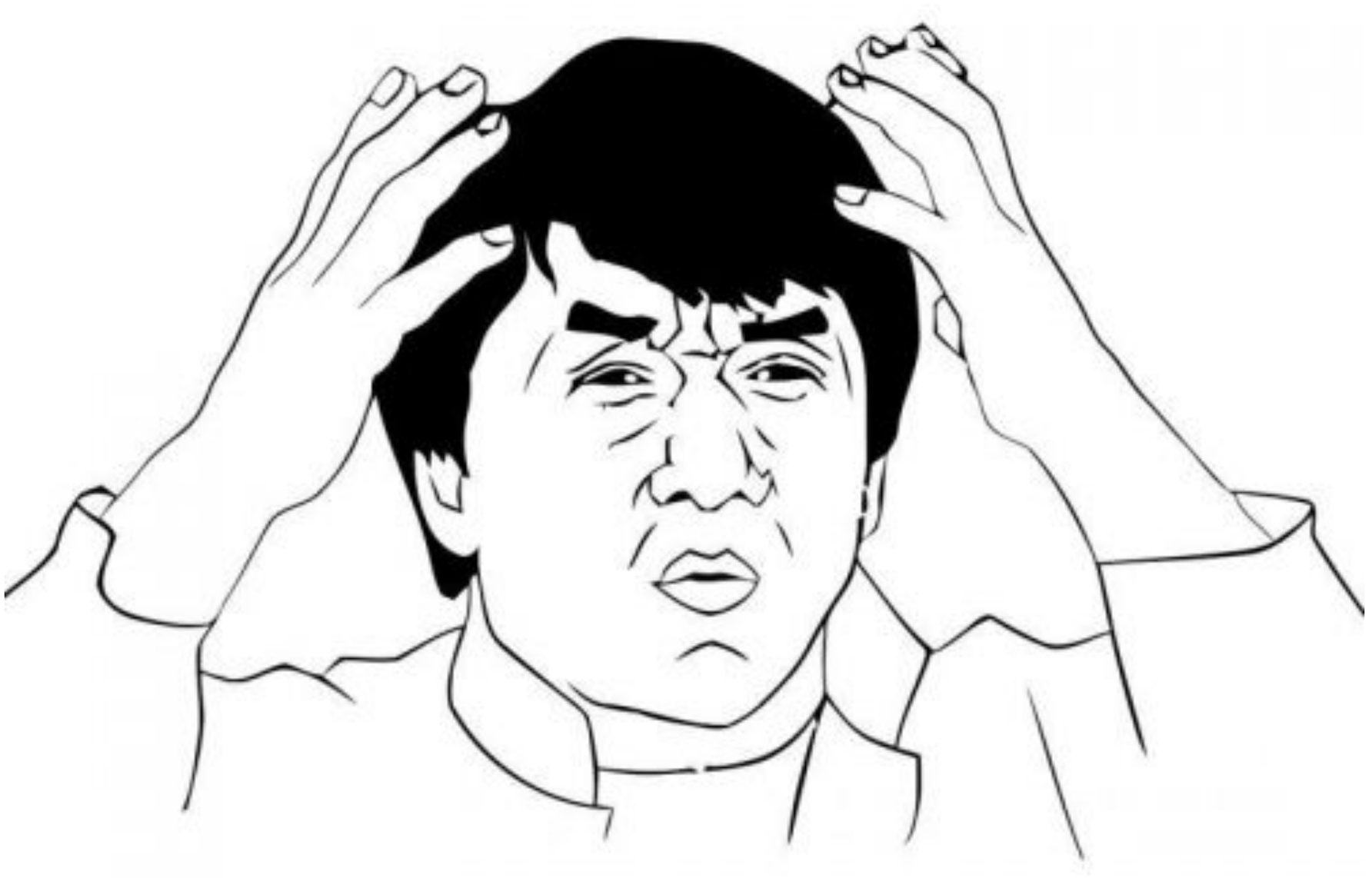
Least privileges ?



This is definitely an ISO27001 violation

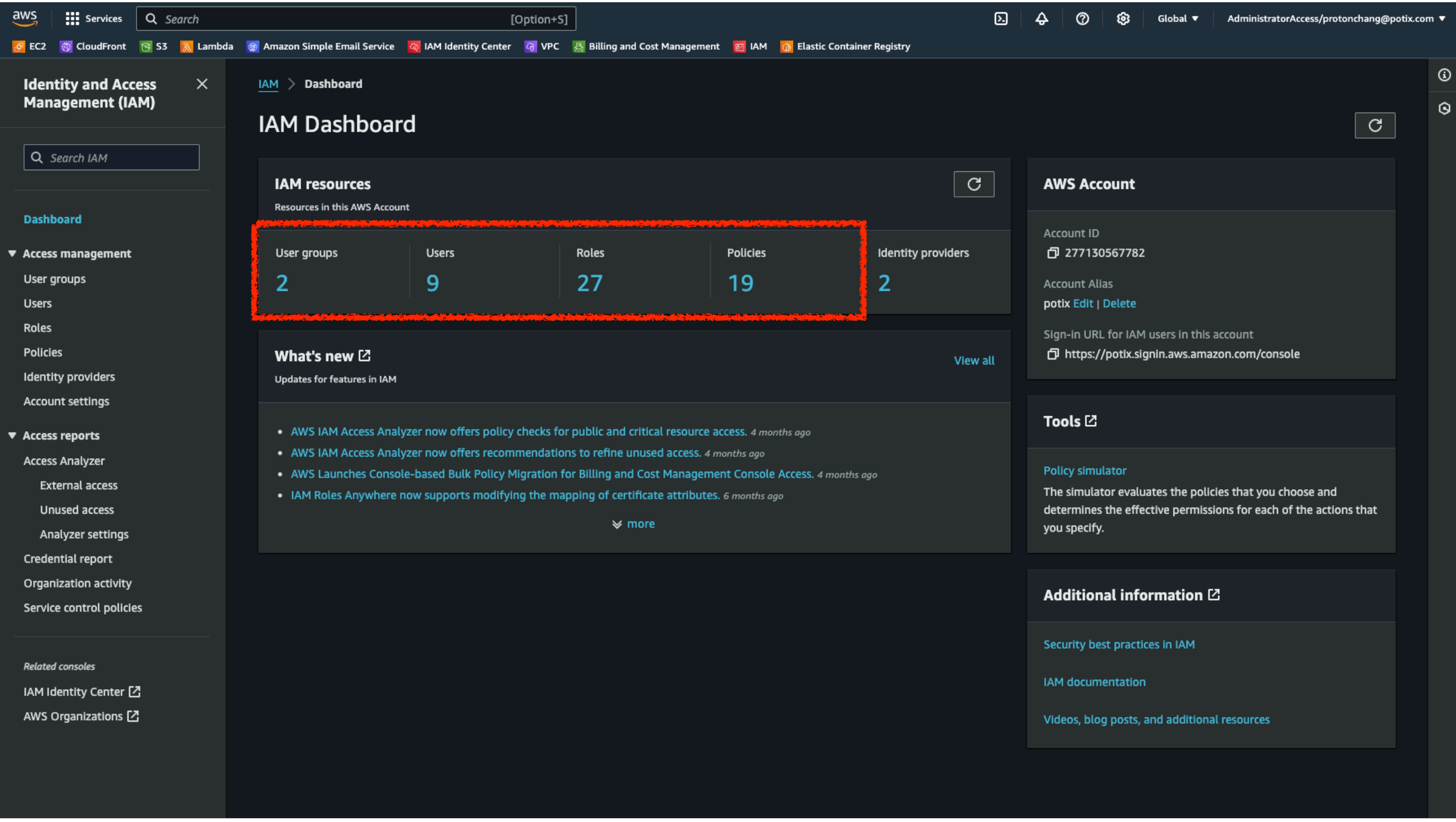
AWS IAM

So, how exactly does this work ?



AWS IAM

Hands-on



AWS IAM

Policies

- The place where you define permissions
- Ability to do “fine-grained” permission configurations
- Either use pre-defined permission sets by AWS, or configure your own

AWS IAM

Users - Just think them as end users

- Added user will have delegation permission to control your account
- Permission boundary can be set using policies
- IAM Users has Fixed Access Key that can be use repeatedly
- Real World Usage: Separate users can have different permission settings
  - i.g. Finance dept can have access to billing details, but can’t control other resources in the same account
- AWS Recommendations:
  - Create a user with IAM, grant Admin Permission, and stop using your Root Account

AWS IAM

Roles

- Identity within your AWS accounts, that has specific permissions
- Similar to IAM user, just not associated to person
- In Plain English:
    - It grants permission to specific “actors” for a set of duration of time
    - Actors can be authenticated by AWS or trusted external system (i.g. external SAML authentication, Github Action run group etc…)

AWS IAM

Roles

- AWS supports 3 Role Types for different scenarios
  - AWS service Roles: Provide access between AWS resources
    - i.g. Let EC2 access S3 data
  - Cross-Account Access:
    - Grant permission to users from other AWS account
  - Identity Provider Access: Grant permission to users authenticated by a trusted external systems (identity federation)
    - AWS support either OpenID connect or SAML 2.0

AWS IAM

IAM User vs IAM Role

	IAM User	IAM Role
Can have password/ Access Key	Yes	No
Can belong to Groups	Yes	No
Can be associated with AWS Resources	No	Yes

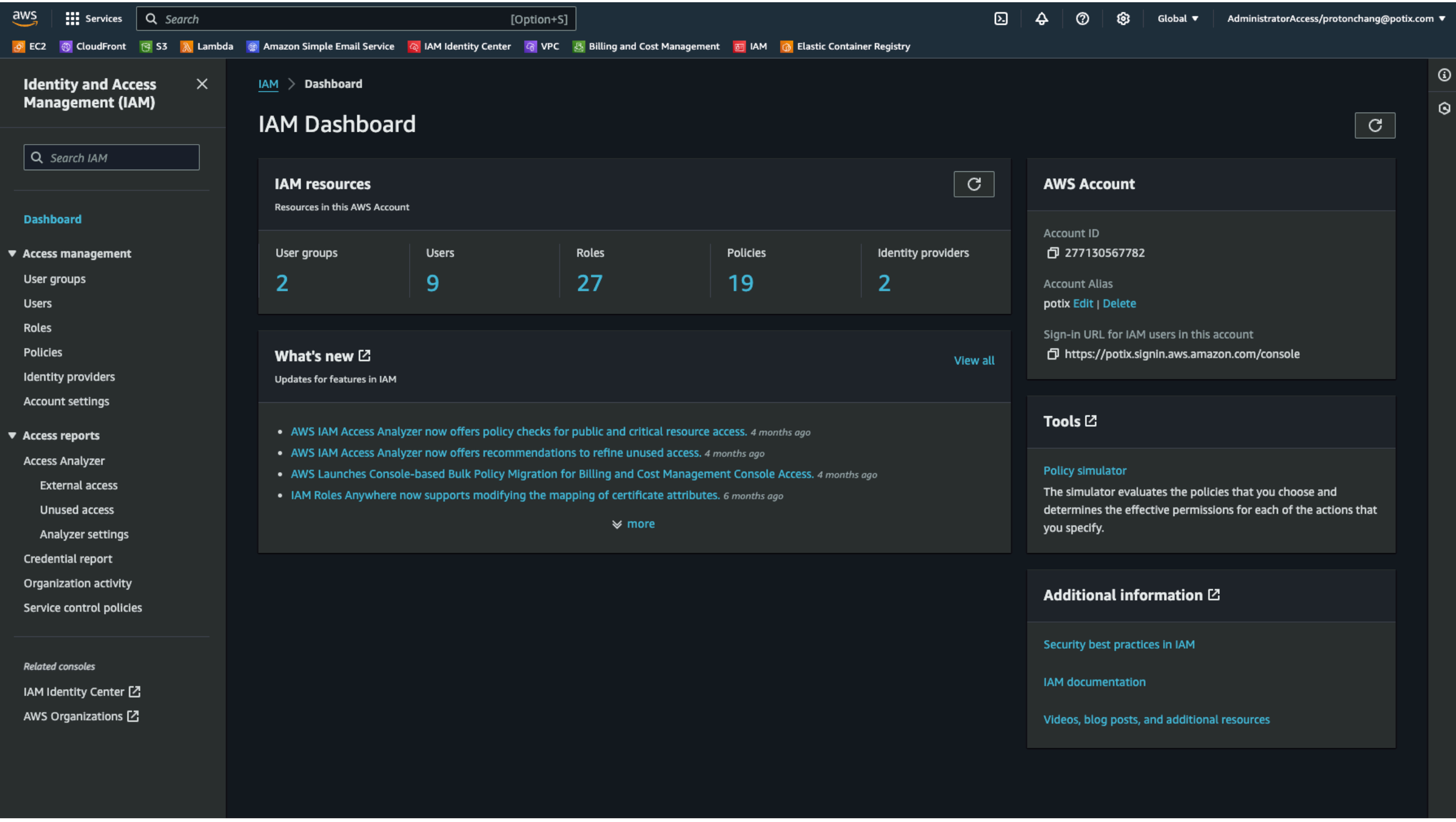
AWS IAM

Recap

- Policies controls what you can do on AWS
- Users and roles let you authenticate with AWS
- Users authenticate using password or access key (API Key), roles authenticate by AWS or by trusted external systems
- Security Reminders:
  - Whatever you do, always keep your Access Key safe, do not commit them into your version control software, THEY ARE YOUR SECRETS !!!!
  - AWS does detect leaked access keys, especially when you commit them to Github, automatically quarantine them, and will notify users to reset their access key

AWS IAM

Workshop hands-on



AWS IAM

Workshop hands-on

- Your Objectives
    - Create a user and a role with admin privileges that can only control AWS S3 services
    - Limit the user allowed IP, request can only be coming from company network IP (or your local IP)
    - We will be using the user credential during the next chapter :)
    - Add this credential to a new aws-cli profile, and try accessing S3 command with the credential (aws s3 ls)

Amazon Simple Storage Service

Amazon Simple Storage Service

aka Amazon S3

- Object storage service (Not Block Storage)
- No maximum bucket size limit, with maximum single file size up to 5TB
- Durability up to 99.999999999% (11 9s)

Amazon S3

Introduction

- Pay-as-you-go pricing model
- Multiple Storage Classes depends on your need
- Can setup Lifecycle policy to automatically move objects between different Storage Class
- Supports file versioning
- Comes with Server-side encryption by default
- Can be use to host static websites
- Compatible with AWS CDN services

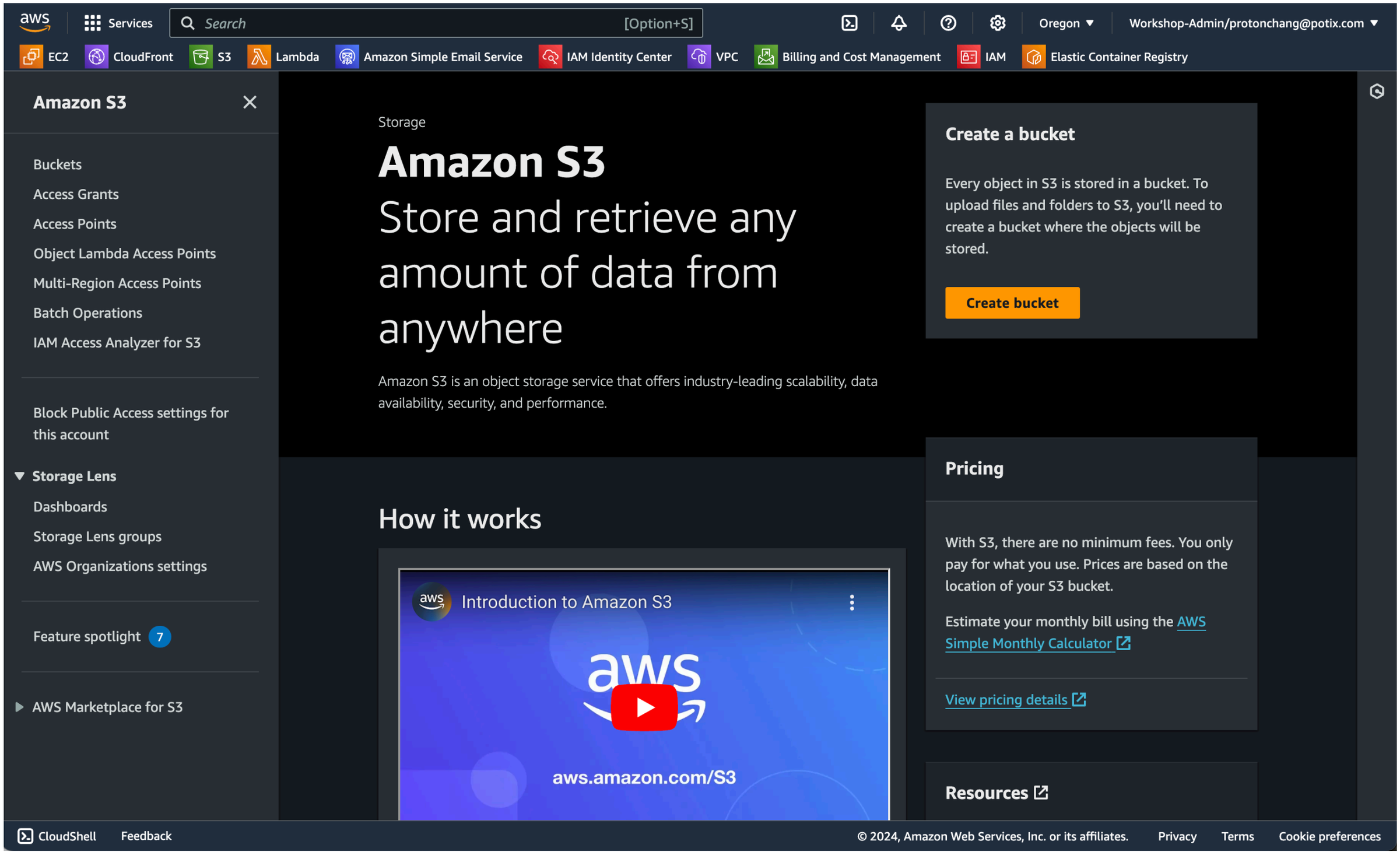
Amazon S3

Storage Class and Pricing model

Storage Class	Availability Zones	Min Storage Duration	Retrieval Fees	Price per GB
Standard	≥3	No limitation	None	0.023 per GB
Intelligent-Tiering	≥3	No limitation	None	Storage Tier + 0.0025 per 1k object
Standard-IA	≥3	30 days	0.01 per GB	0.0125 per GB
One Zone-IA	1	30 days	0.01 per GB	0.01 per GB
Glacier Instant Retrieval	≥3	90 days	0.03 per GB	0.004 per GB
Glacier Flexible Retrival	≥3	90 days	0.01 to 0.03 per GB	0.0036 per GB
Glacier Deep Archive	≥3	180 days	0.02 per GB	0.00099 per GB

Amazon S3

Workshop hands-on



Amazon S3

Workshop hands-on

- Your Objectives
  - Create a bucket within Region: Oregon
  - Upload files to your bucket with GUI and aws-cli (try Amazon S3 commands)
  - Share files with pre-signed URL, either generate with aws-cli or from GUI
  - Change the Bucket ACL to let external users get the file without any authentication
  - Create another bucket and try to enable the static web hosting function

Amazon S3

Workshop hands-on

- Now, try creating the s3 bucket with Terraform

Amazon VPC

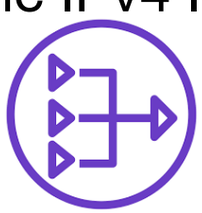
Amazon VPC

Amazon Virtual Private Cloud

- Defines your own virtual networks in the cloud
- Central control for all the network related configurations, including
  - Subnets (Define your own virtual IP block)
  - IP addressing (You can bring your own IP Blocks to AWS, or use IP from AWS)
  - Routing
  - Gateway and Endpoints
  - VPC Peering (Connections between different VPCs)
  - Transit Gateways (More on this later)
  - VPN Connections (You can create VPN tunnels to connect your on-premises networks to AWS)
  - DNS Hostname: Instances with Public IP gets corresponding DNS Hostnames
  - DNS Resolution: Amazon-provided DNS server

Amazon VPC

Basic Components

- Subnets
  - Your custom private subnets config
  - Typical config will contain both public and private subnet
- Route Tables
  - Routing tables can be associated to subnets, to create your own private network config
- Internet Gateways 
  - Gateway between VPC and Internet (inbound and outbound)
- Elastic IPs
  - Allocatable Public IPv4 IP
- NAT Gateways 
  - An outbound-only gateway that can be deployed in a public subnet to provide internet access for instances located in private subnets

Amazon VPC

Additional Notable Components

- VPC Peering
  - Let you connect between two VPC
  - VPCs does not have to belong to the same user
  - Two peers should have different VPC subnets
- Network ACLs
  - Stateless ACL rules that can be bounded to subnets
- Security Groups
  - Stateful connection rules that can be bounded to EC2 instances
- Transit Gateway
  - A central hub for VPCs to simplifies your network and puts an end to complex peering relationships
  - In Plain English: Scalable cloud router

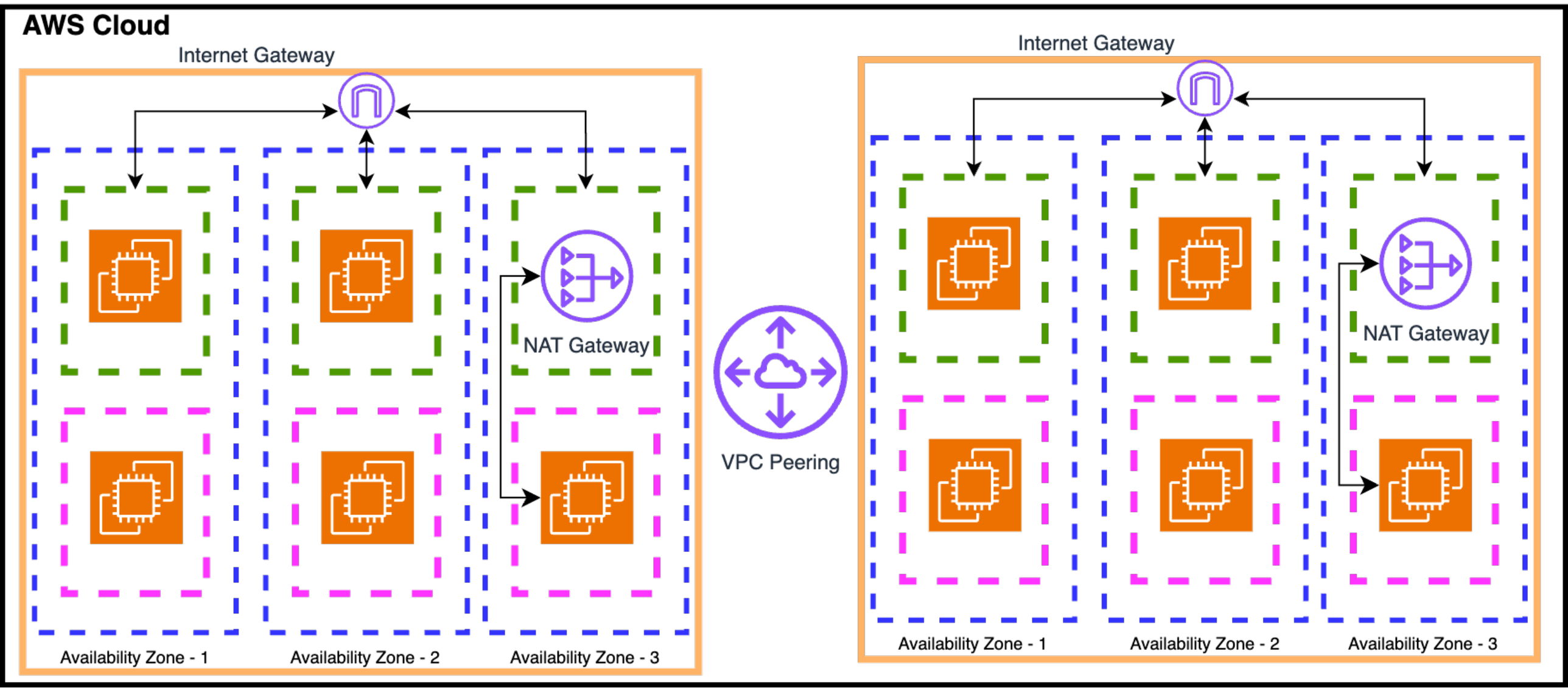
Amazon VPC

Additional Notable Components

- VPC Flow Log
    - Log traffic flow and send data to Cloudwatch, S3 or Amazon Data Firehose
    - Ability to monitor traffic logs, without causing impact to network performance

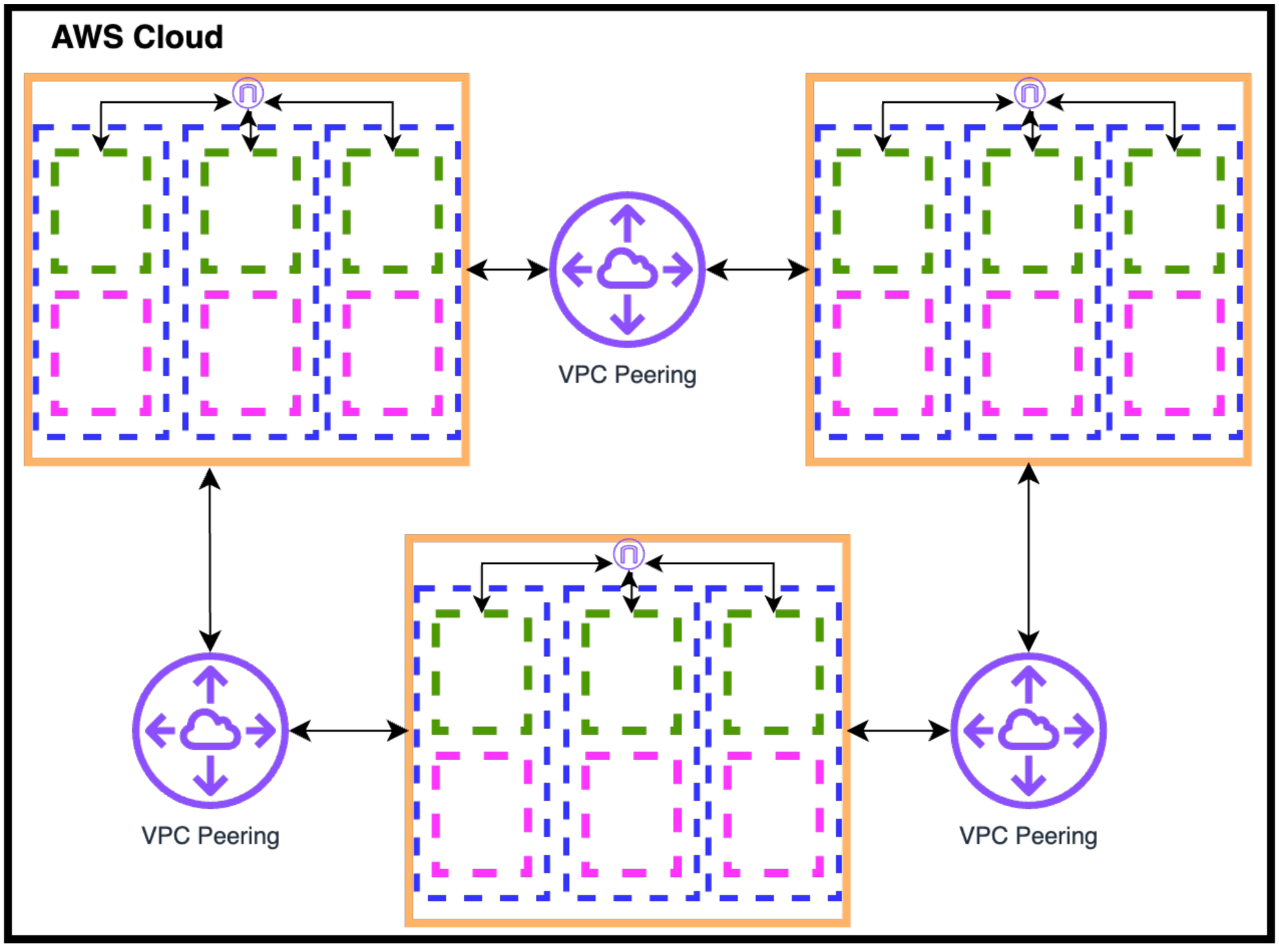
Amazon VPC

VPC Peering



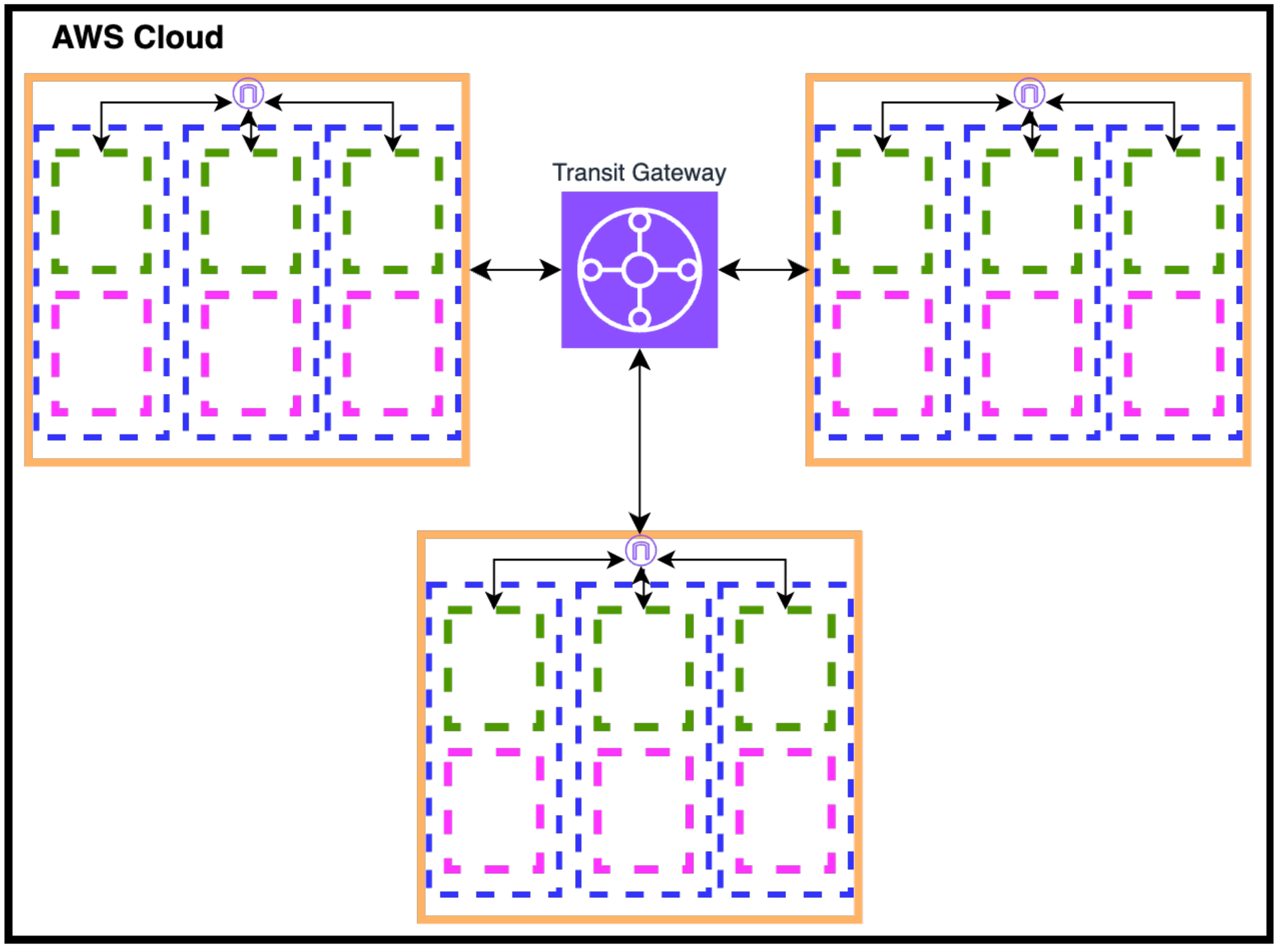
Amazon VPC

VPC Peering



Amazon VPC

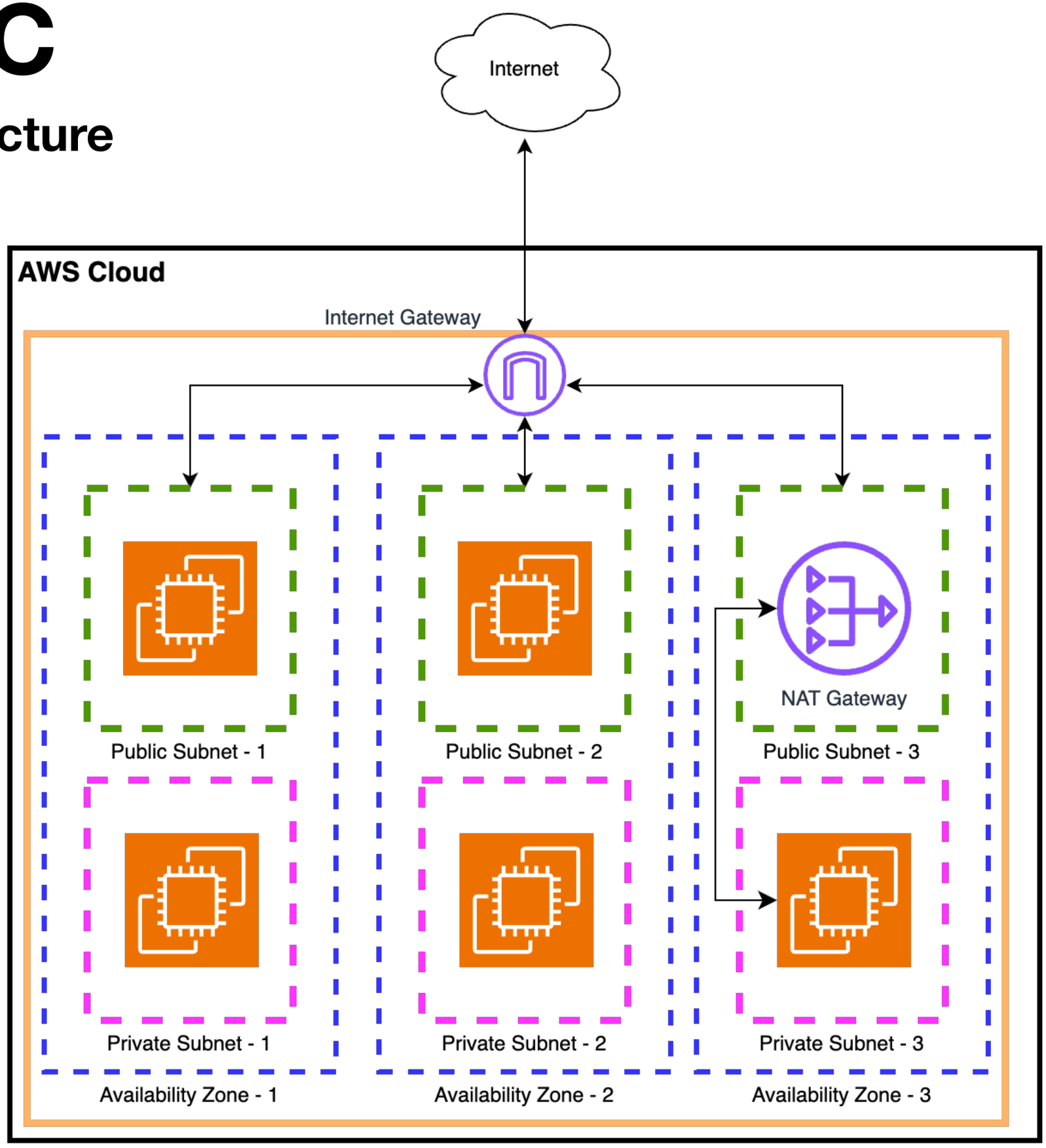
Transit Gateway



Amazon VPC

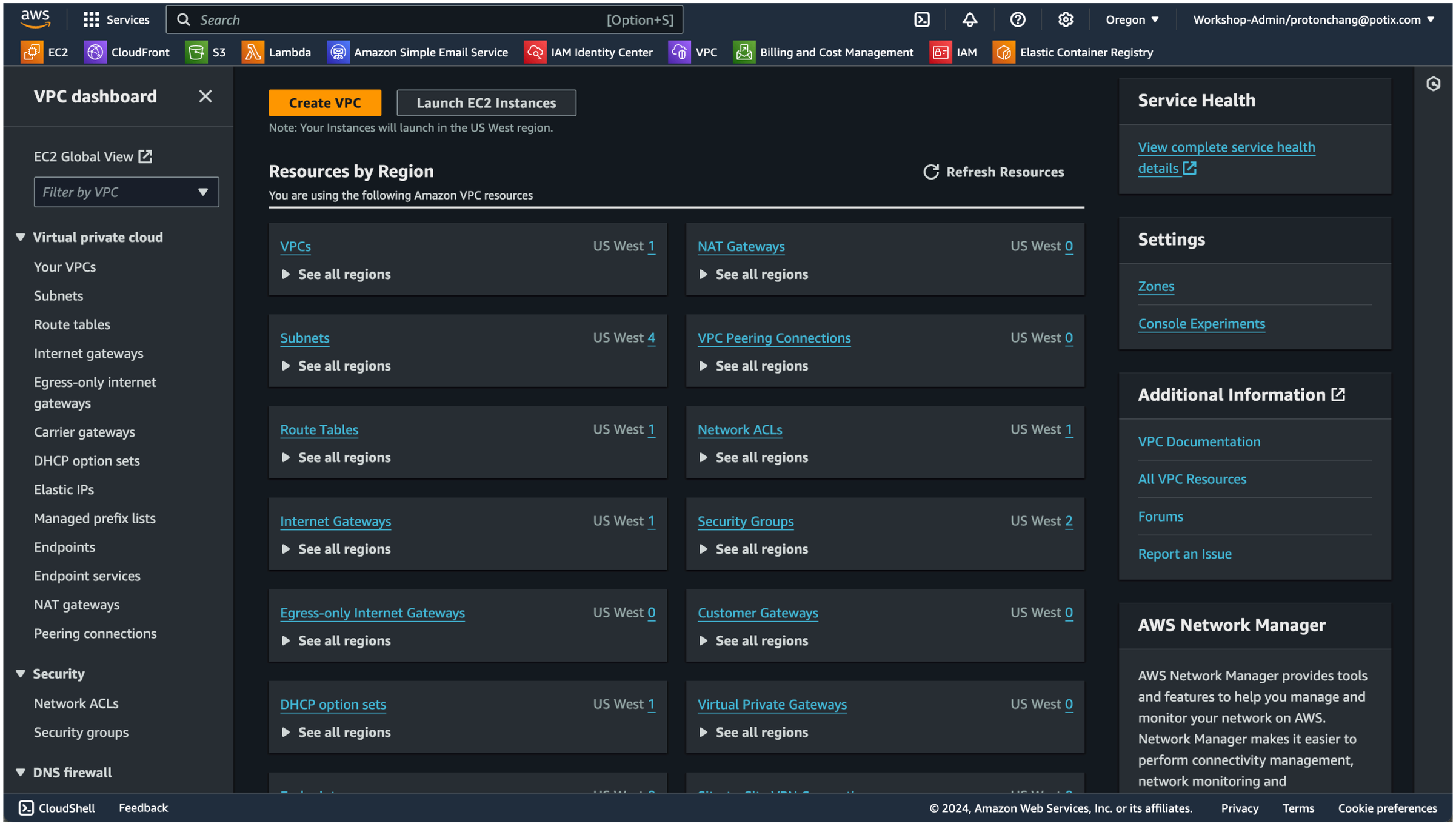
Recommended Architecture

- Each AZ should have at least 2 subnet, public and private
- Only public subnet should have direct access to the Internet
- Use NAT Gateway when necessary ($$)



Amazon VPC

Workshop hands-on

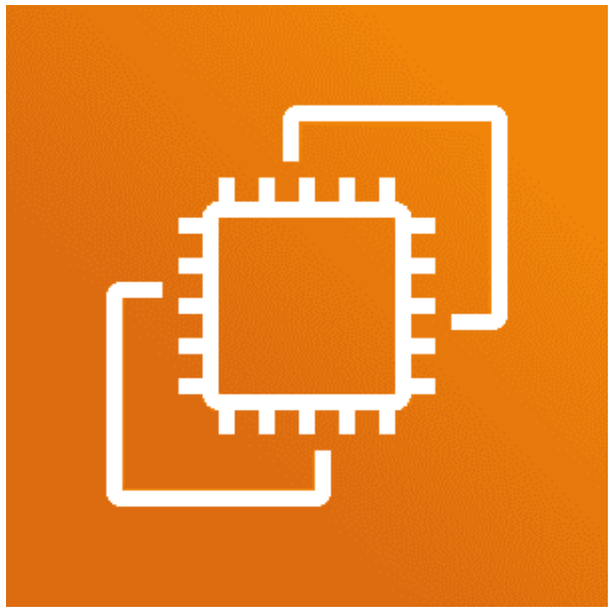


Amazon VPC

Workshop hands-on

- Your Objectives (IPv4 only)
    - Create a new VPC with name Workshop-VPC
    - Allocate subnets for at least 2 Availability Zones
    - Each Availability Zones should have 2 subnets, public and private
    - Create and configure routing tables and internet gateways, so your public subnet can have access to the internet
    - Instances in private subnet should have access to instances in public subnet
    - Configs will be verified at the next chapter (When we create EC2 instances)
    - It’s OK to use VPC wizard to create the new VPC :)

Amazon Elastic Compute Cloud



Amazon Elastic Compute Cloud (EC2)

Introduction

- Virtual Servers in the Cloud
- Multiple specifications to choose from, depends on your need
- Multiple pricing-model, not just on-demand, but also spot requests, saving plans, and reserved instances
- Can do auto-scaling based on your settings (We will talk about this next time)
- Built-in Load balancing functionality that can be enabled

Amazon Elastic Compute Cloud (EC2)

Components - Instance Pricing Type

- On-demand
    - Most used type, launch machine at any given time, without lead time or interruption
- Spot Requests
    - Unused EC2 resources, cheaper than on-demand (50% ~ 90% discount)
    - Dynamic Pricing, you just set the maximum price you would like to offer when asking for spot request
    - When using spot requests, your workload must be fault-tolerant, as AWS might claim back the resources, and interrupt your instances

Amazon Elastic Compute Cloud (EC2)

Components - Instance Pricing Type

- Reserved Instances
    - Pre-paying or set commitment for a 1 or 3-years, in exchange for discounted hourly rate (commitment on instances)
    - Tied to specific region and instance type, specs are immutable during the commitment time

Amazon Elastic Compute Cloud (EC2)

Components - Instance Pricing Type

- Saving Plans
    - Set usage commitment for 1 or 3 years (commitment on spendings), can be pre-paid or post-paid
    - Can select between Compute saving plans and EC2 saving plans
- Compute Saving Plans
    - Get discounts on ANY EC2 instances, regardless of instance family
- EC2 Instance saving plans
    - More like Reserved Instance, but locks you into a specific region and instance family, instead of specific instance type

Amazon Elastic Compute Cloud (EC2)

Reserved Instances vs Saving Plans

	Reserved Instances	Saving Plans
Discount	Up to 75% off on-demand prices	Up to 66% off on-demand prices
Flexibility	Lock-in for a specific instance type, and region	Can applies to different instance types and region
Payment Options	No Upfront Partial Upfront Full Upfront	No Upfront Partial Upfront Full Upfront
Ideal For	Long-term workloads When you know exact instance requirements	Workloads that changes over time Broad flexibility + Cost Savings

Amazon Elastic Compute Cloud (EC2)

Components - Storage

- Elastic Block Storage (EBS)
  - Disks for your virtual machines
  - Also multiple specs to choose from, depends on your needs
    - Ranged from SSD or NVMe based to HDD based disks
- Ephemeral Storage
  - Some Instance type provide on-device SSD directly attached to the instance
  - This kind of storage has the fastest performance, but will suffer data loss when instance reboot

Amazon Elastic Compute Cloud (EC2)

Components - EBS Snapshot and AMI

- EBS Snapshot
  - Create a snapshot of your current EBS disk states
  - Incremental Backup, first snapshot might take a very long time
- AMI
  - Base image of your instances
  - Can create your own AMI with custom presets

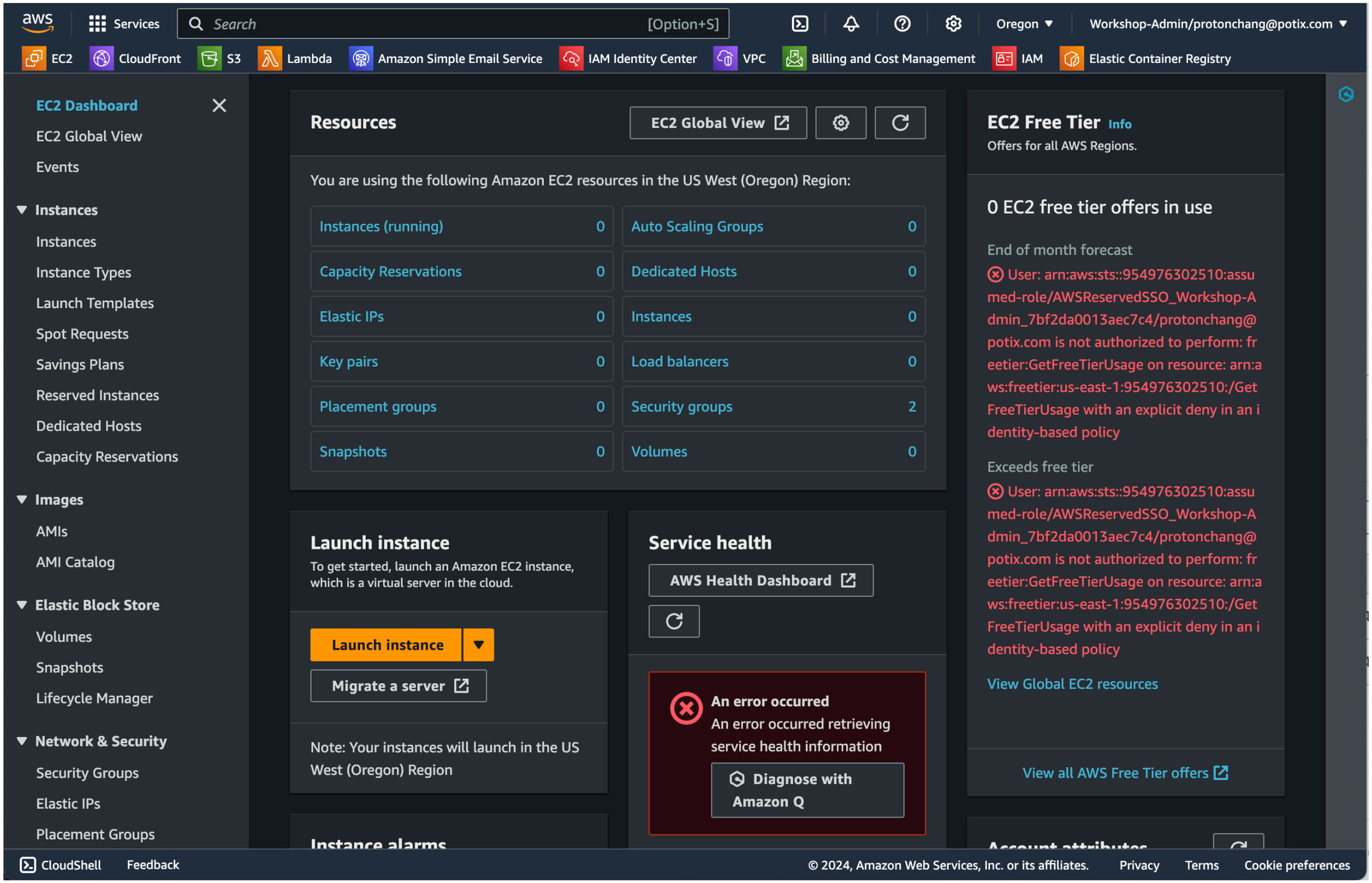
Amazon Elastic Compute Cloud (EC2)

Components - Userdata, T series machine type, Instance Profile

- Userdata
  - Can be shell script or cloud-init directives
  - By default will only run once during the boot cycle when you first launch the instance
    - Will run as root, and can’t do any interactive inputs
- T series machines
  - Have certain baseline performance, depends on the machine type
  - Does not have unlimited CPU power, but have CPU credit
  - Support unlimited mode, with additional cost
- Instance Profile
  - A Container for you IAM role

Amazon Elastic Compute Cloud (EC2)

Workshop hands-on



Amazon Elastic Compute Cloud (EC2)

Workshop hands-on

- Your objectives
  - Create two t2.micro instances, with Ubuntu 24.04
  - Do not allow SSH connection for anywhere, allow it only from your own network, and your VPC subnet
  - We will use the VPC created from the last chapter, make sure one of your instance is in public subnet and another one in private subnet
  - Install nginx on the instance located in public subnet
  - Try to establish SSH connection to the instance in the private subnet, with the instance in public subnet as the bastion host
  - Try applying IAM role (Instance Profile) to the instance, and use aws-cli to access previous created S3 bucket