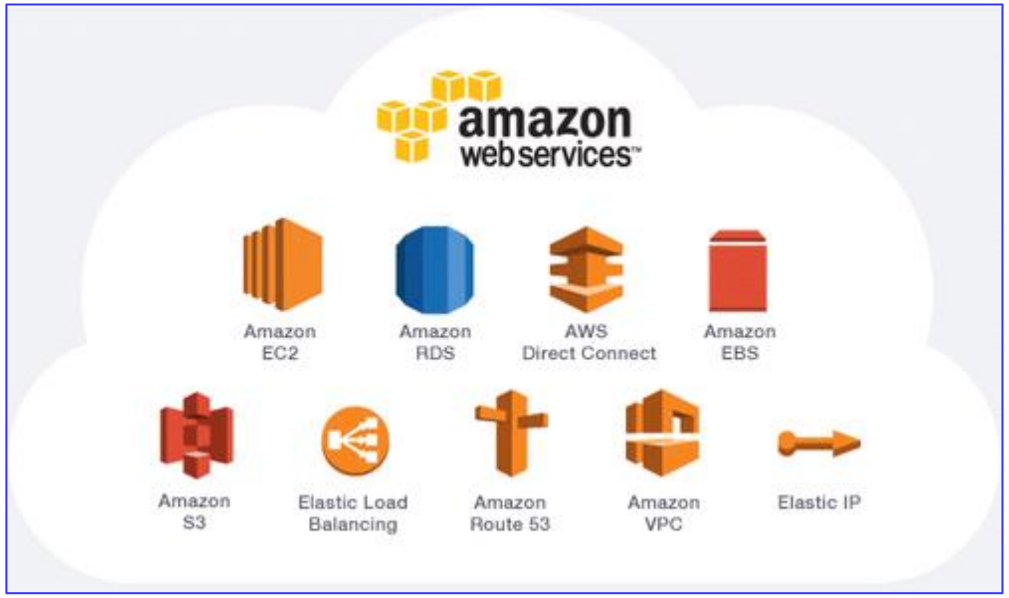
Week 9

Amazon Cloud (AWS)



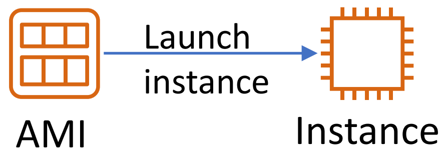
Section 1

EC2 Review
Questions and Discussions

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*

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 EC2 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

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

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

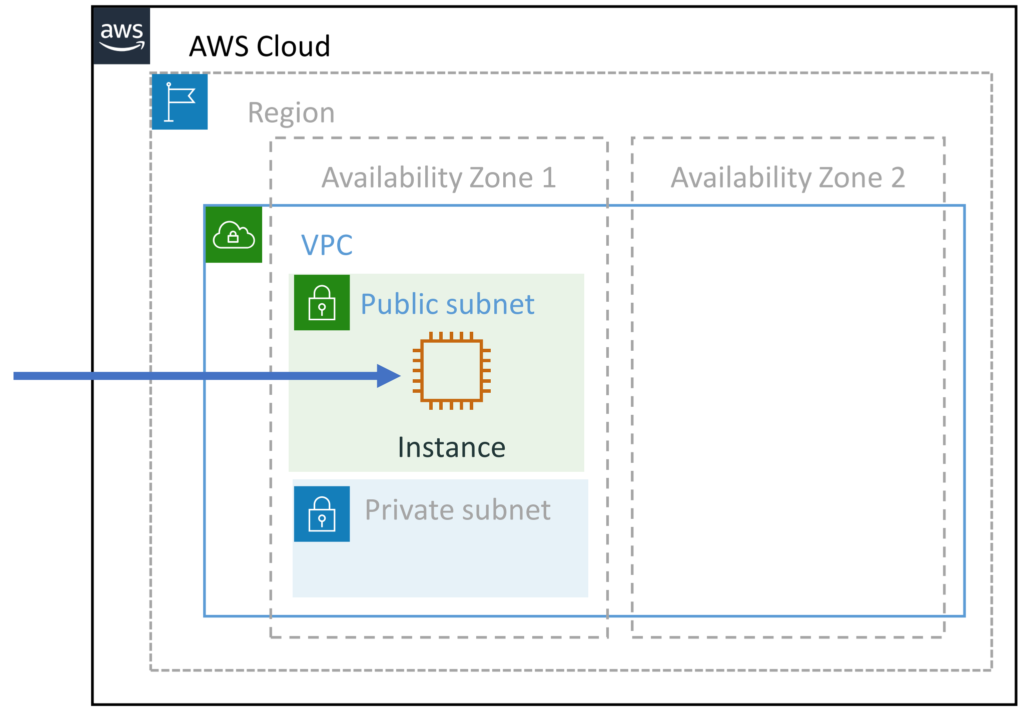
3. Specify network settings

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



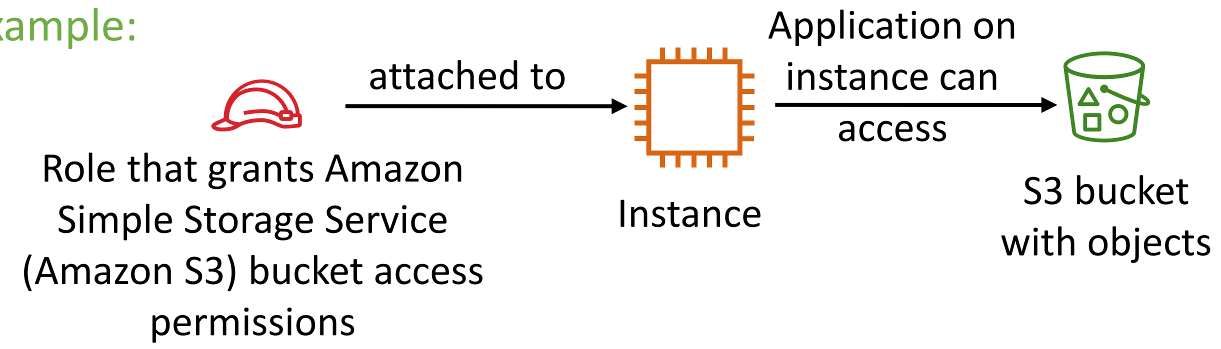
4. Attach IAM role (optional)

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.

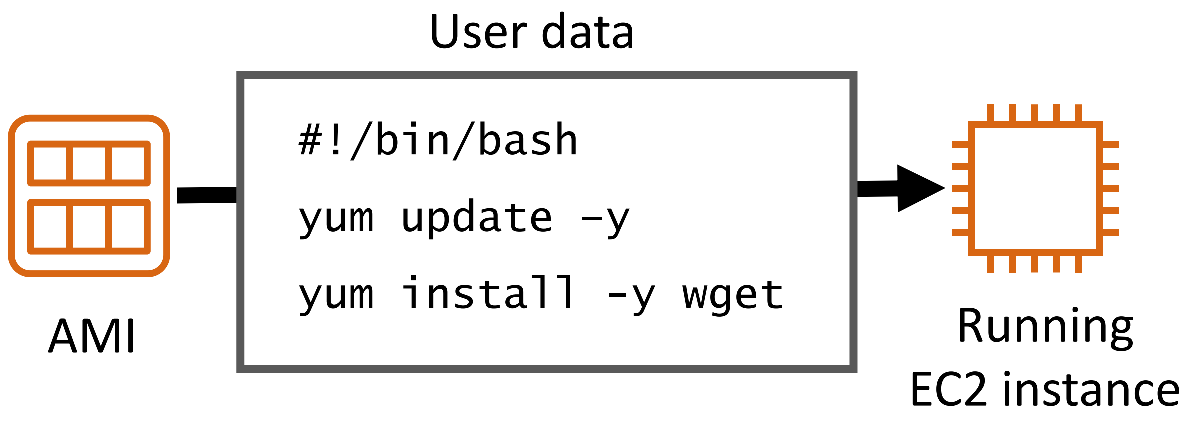
Example:


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

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**



```
#!/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 executes the first time the instance starts
- Can be used strategically
  - For example, reduce the number of custom AMIs that you build and maintain

EC2 instance **userdata**

Script executes the first time the instance starts

- Use **user data** scripts to customize the runtime environment of your instance
- A linux machine is running but no one can access it. How do we access it?

Optional Lab 2#

Your company has a Linux machine is running but no one can access it. Develop a solution to solve this problem.

Present the following:

1. You solution including codes/scripts used in the solution
2. A short demo-video showing your solution using AWS-console or command line
3. Due on the final exam day.
4. 3 points total.

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.

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

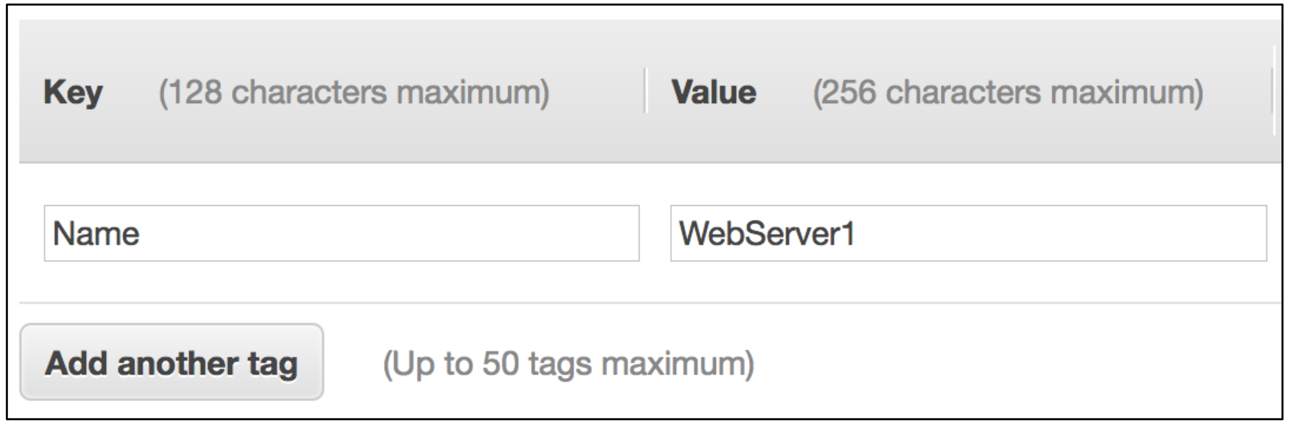
7. Add tags

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:



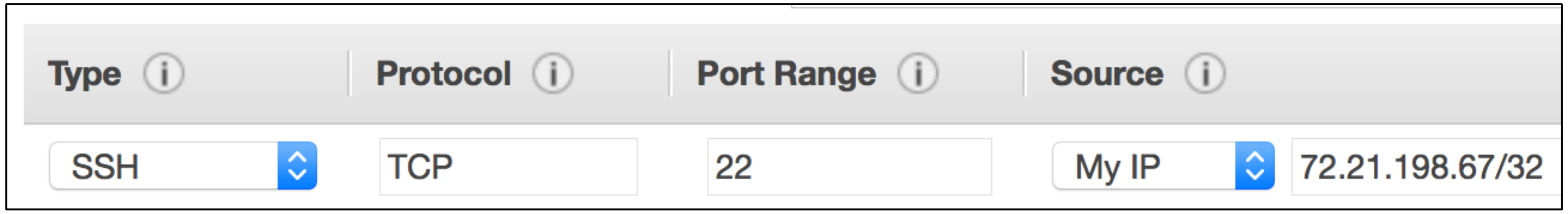
Key (128 characters maximum)	**Value** (256 characters maximum)
Name	WebServer1

8. Security group settings

Choices made by using the Launch Instance Wizard:

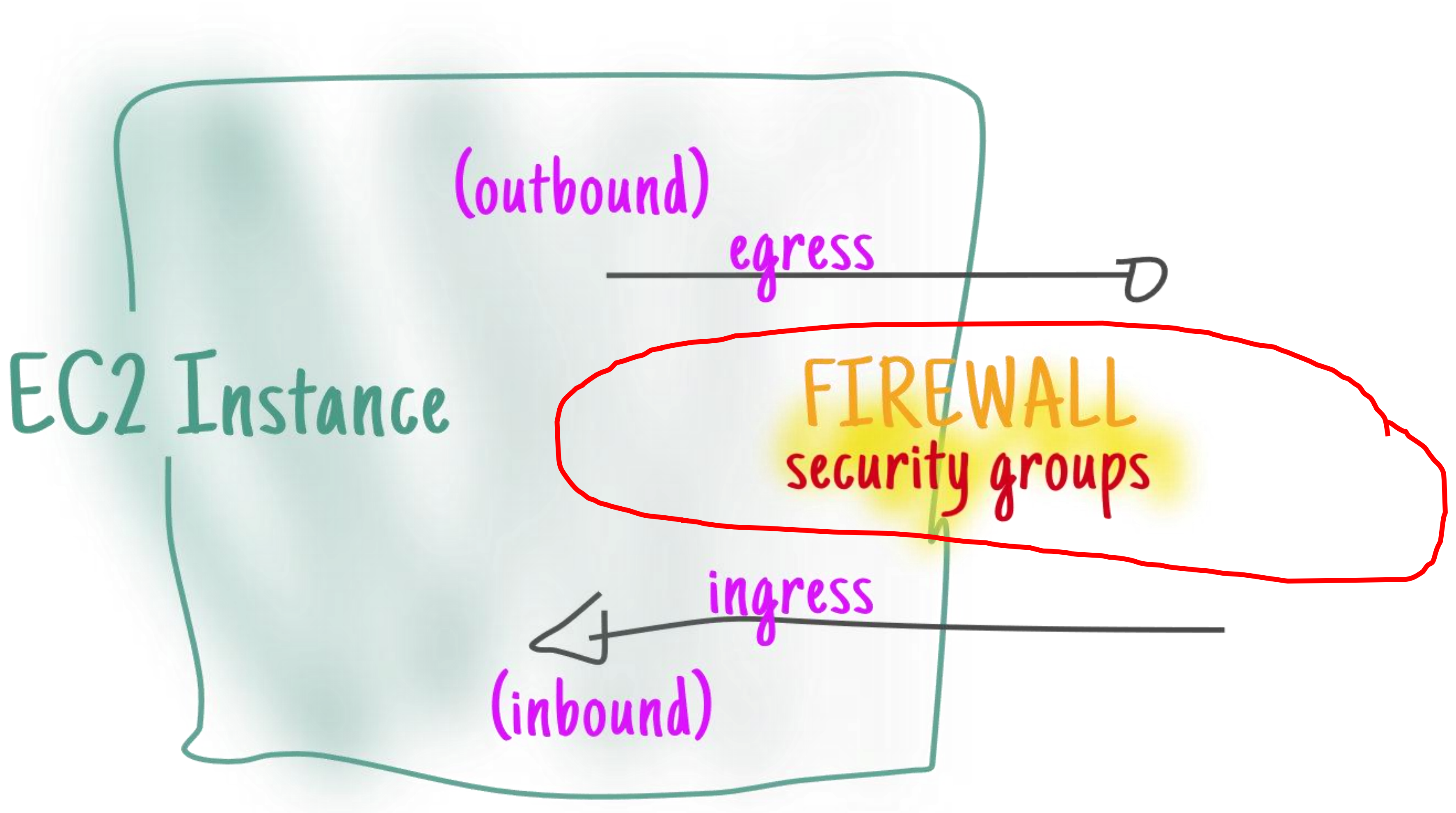
1. **AMI**
2. **Instance Type**
3. **Network settings**
4. **IAM role**
5. **User data**
6. **Storage options**
7. **Tags**
8. **Security group**
9. **Key pair**

- 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.

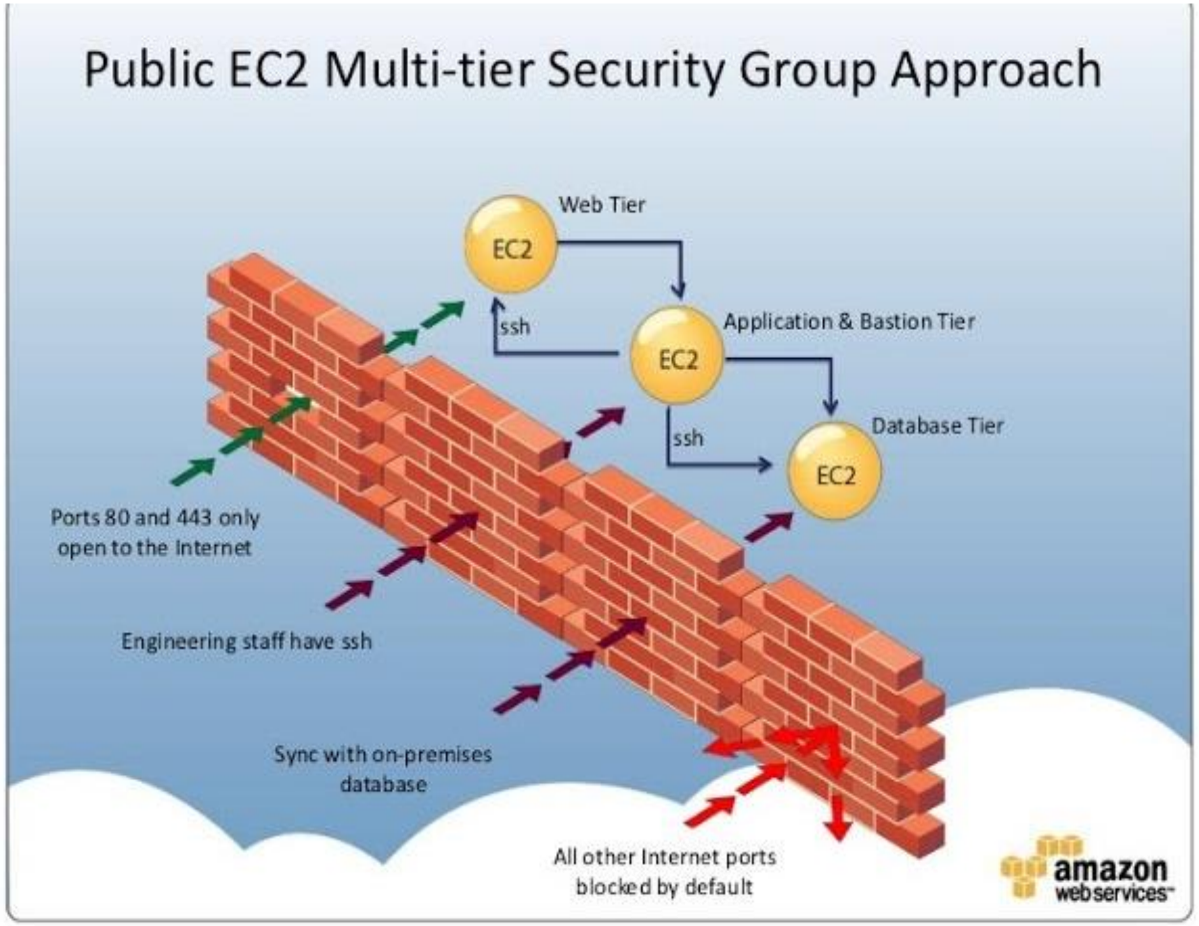


Type	Protocol	Port Range	Source	
SSH	TCP	22	My IP	72.21.198.67/32

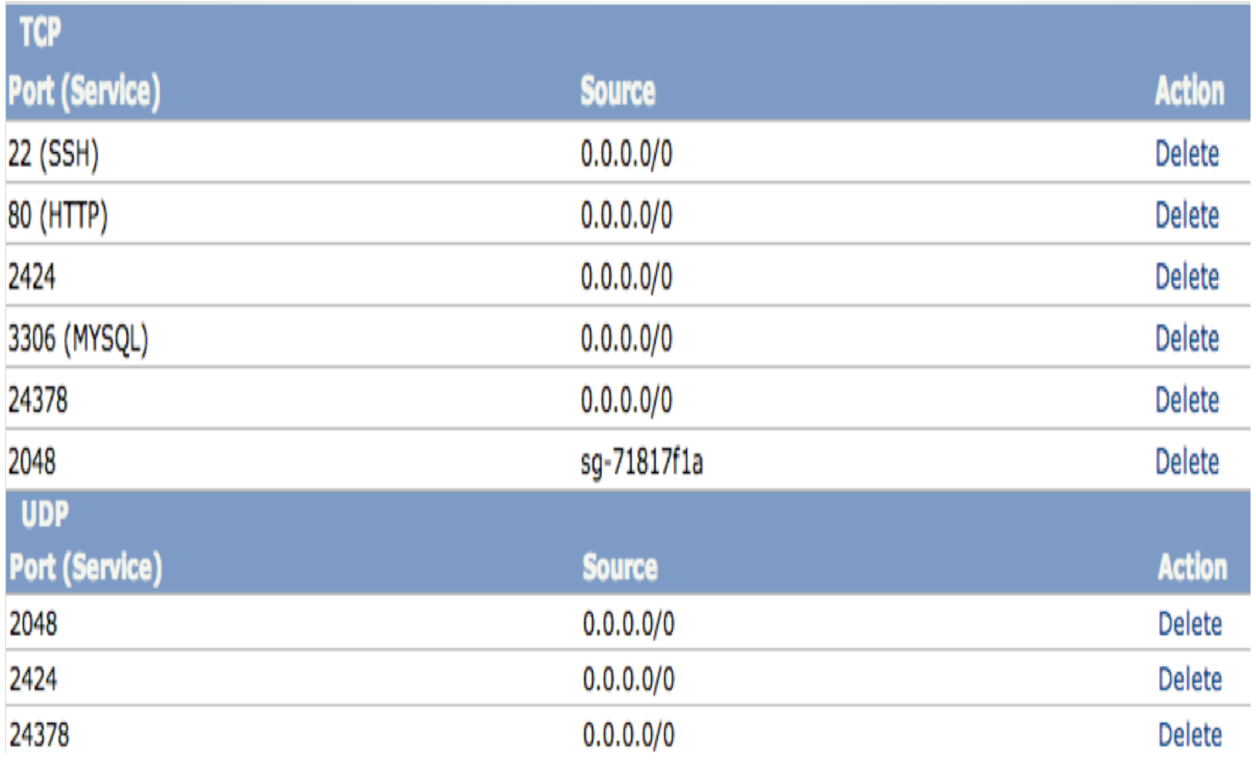
EC2 Security



EC2 security (SG)


Public EC2 Multi-tier Security Group Approach
Web Tier
EC2
ssh
Application & Bastion Tier
EC2
ssh
Database Tier
EC2
Ports 80 and 443 only open to the Internet
Engineering staff have ssh
Sync with on-premises database
All other Internet ports blocked by default
amazon webservices

Security Group (inbound)



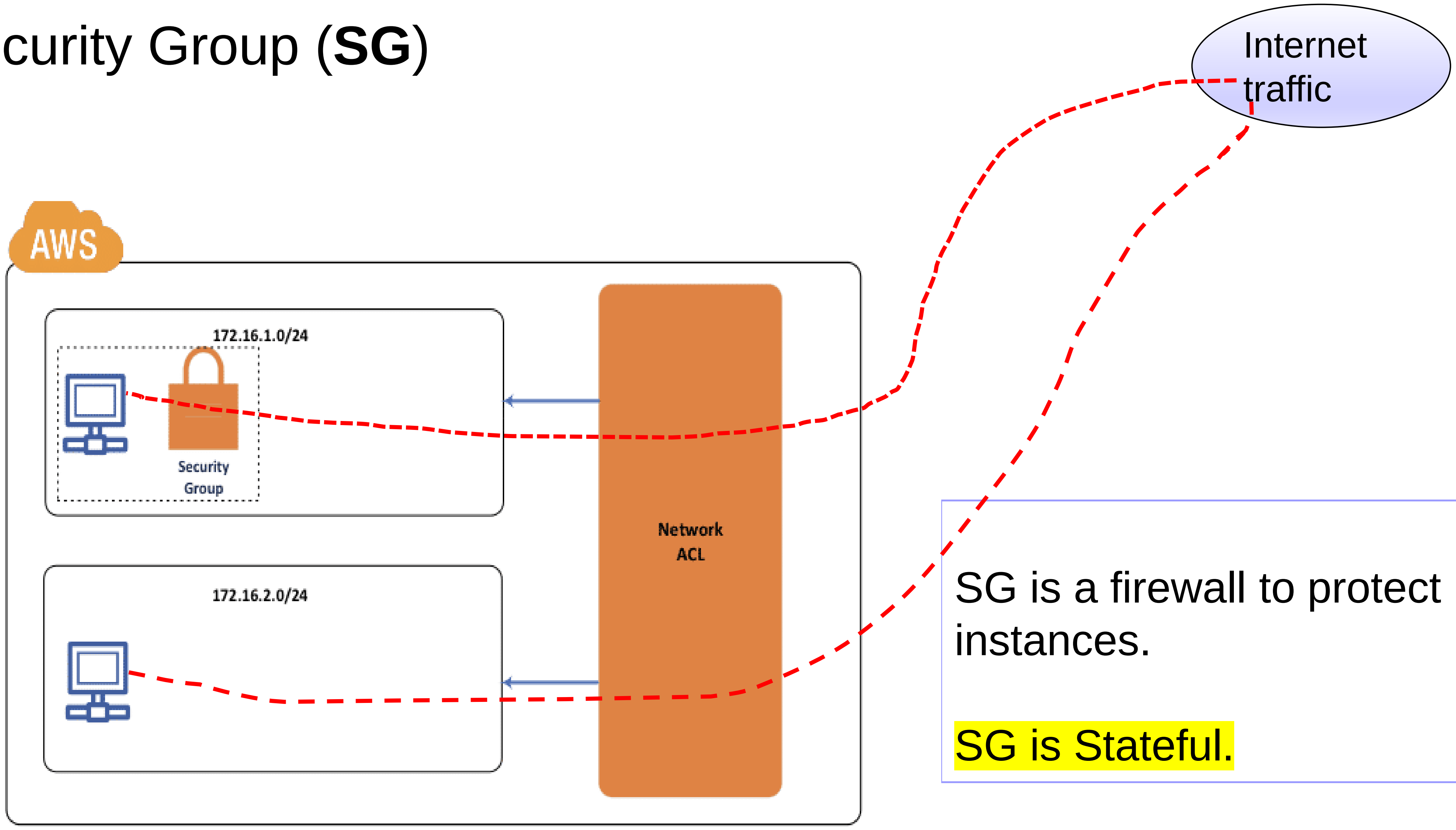
TCP

Port (Service)	Source	Action
22 (SSH)	0.0.0.0/0	Delete
80 (HTTP)	0.0.0.0/0	Delete
2424	0.0.0.0/0	Delete
3306 (MYSQL)	0.0.0.0/0	Delete
24378	0.0.0.0/0	Delete
2048	sg-71817f1a	Delete

UDP

Port (Service)	Source	Action
2048	0.0.0.0/0	Delete
2424	0.0.0.0/0	Delete
24378	0.0.0.0/0	Delete

Security Group (**SG**)



SG is a firewall to protect EC2 instances.

SG is Stateful.

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

Cloud Shell – a Free instance shell

Q: What is AWS CloudShell?

AWS CloudShell is a browser-based shell that makes it easier to securely manage, explore, and interact with your AWS resources. CloudShell is pre-authenticated with your console credentials. Common development and operations tools are pre-installed.

Q: What can I do with CloudShell?

CloudShell gets you started with the AWS CLI more quickly, so you can automate tasks, manage infrastructure, and interact with AWS services. You can use CloudShell to clone repositories containing commonly used scripts, make edits to those scripts, and store them for future use.

Q: What's pre-installed in CloudShell?

CloudShell runs on Amazon Linux 2 and contains common AWS command line interfaces, including AWS CLI, Amazon ECS CLI, AWS SAM CLI, along with runtimes and AWS SDKs for Python and Node.js.

Q: What is the pricing for CloudShell?

There is no additional charge for CloudShell.

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
```

Questions?

Keypair:

1. We used the same keypair for them, what are the differences when we access the window/Linux?
2. What happens if someone obtained your keypair? Can the attacker access the EC2?
3. What happens if someone obtained your AWS account (12digt) login/password?

Security Group:

1. Why did we use ssh and RDP (protocol)? What protocol will we use if the EC2 will be our webserver?
2. Why did we use My IP (73.44.45.67/32) (source) but not Anywhere 0.0.0.0/0? When will we need to use 0.0.0.0/0?
3. What does it mean that Security group is stateful?

AMI (Amazon Machine Image):

1. Why do we need an AMI?
2. I used t2.micro and created an EC2 last week. I installed a database and I need to upgrade the computer. What do I do now?

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.

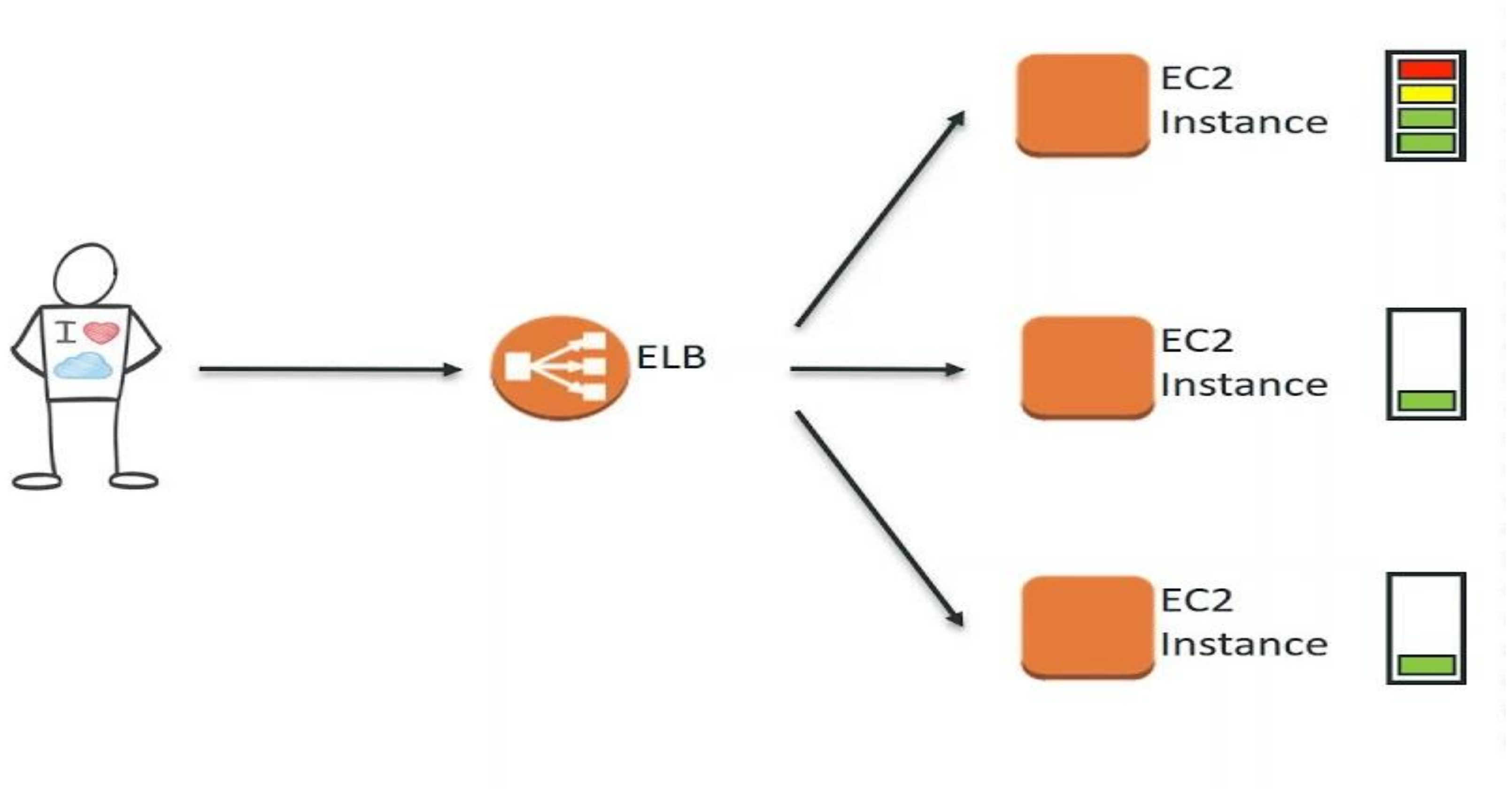
Elastic IP Address

Section 2

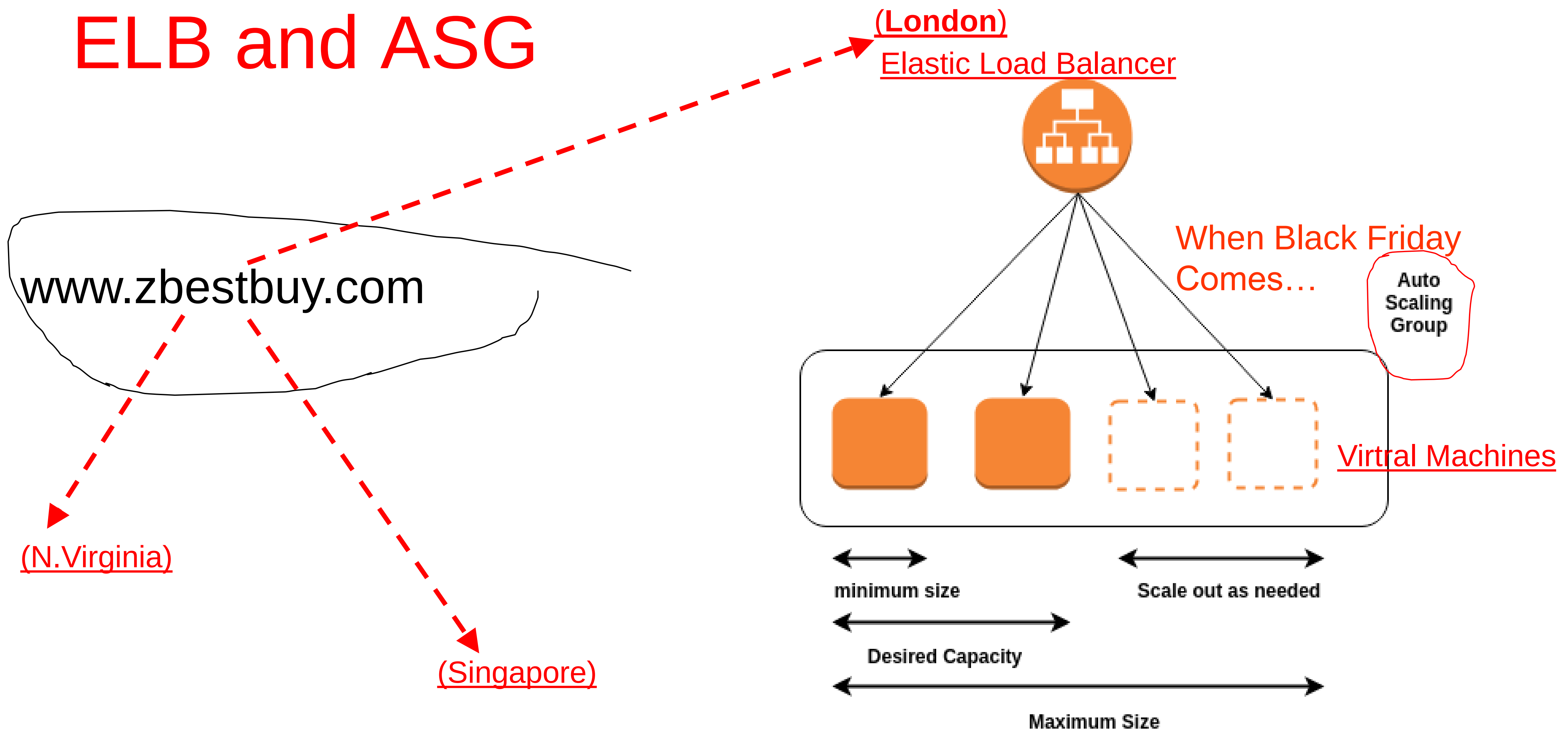
ELB and ASG

- ✓ Elastic Load Balancer
- ✓ Auto Scaling Group

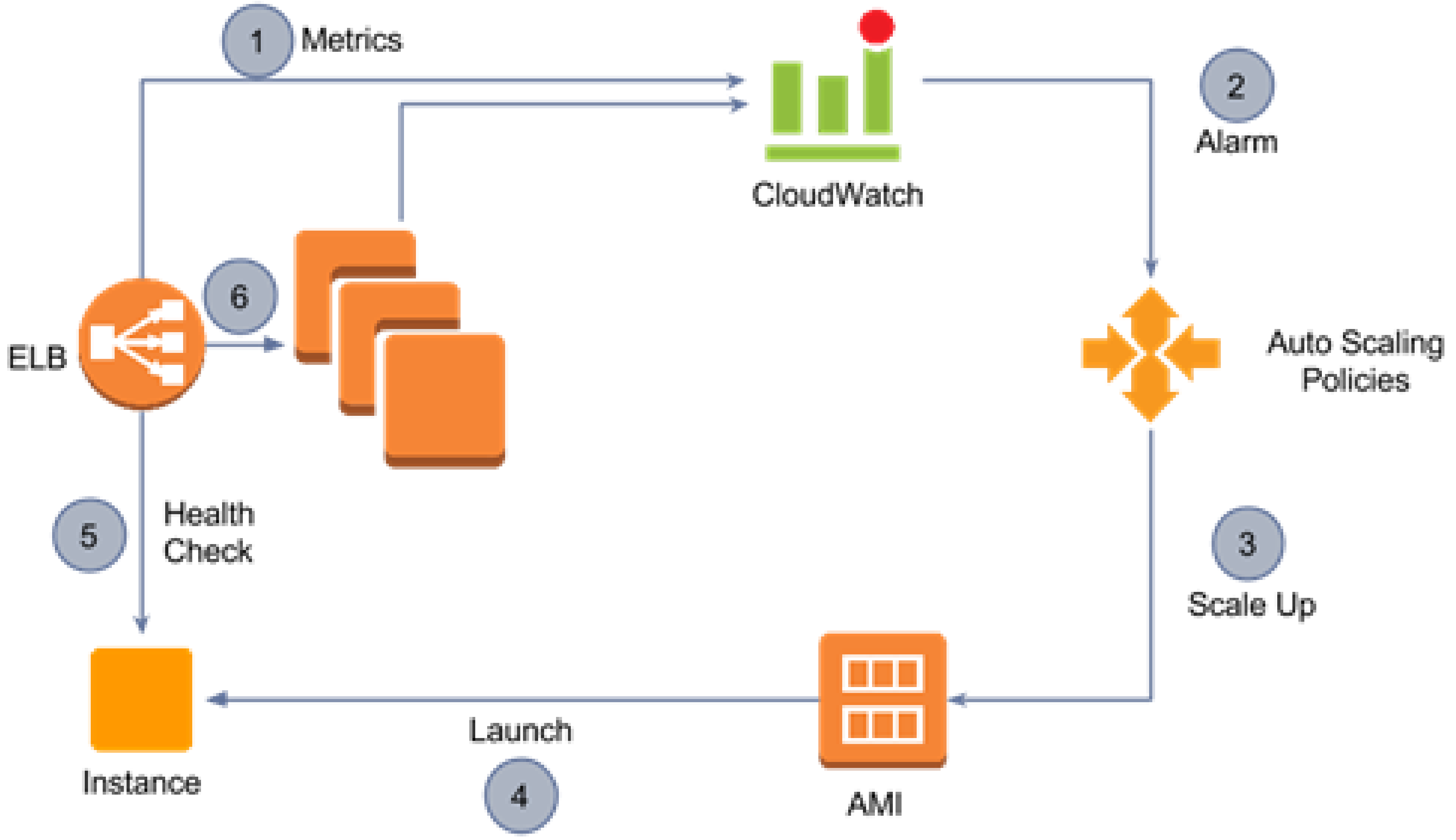
AWS Elastic Load Balancer (ELB)



ELB and ASG


(London)
Elastic Load Balancer
www.zbestbuy.com
(N.Virginia)
(Singapore)
When Black Friday Comes…
Auto Scaling Group
Virtral Machines
minimum size
Scale out as needed
Desired Capacity
Maximum Size

ELB and ASG



1
Metrics
CloudWatch
2
Alarm
Auto Scaling
Policies
3
Scale Up
AMI
Launch
4
Instance
5
Health
Check
ELB
6

Section 3

EC2 Cost Optimization

- ✓ Amazon EC2 pricing models
- ✓ The four pillars of cost optimization

AWS EC2

- AMI: Amazon Machine Image

  Operating System? Windows, Linux?

  What Pkgs? Any databases? Webserver? Java?
- Instance Type

  How many CPUs, RAM, Network, Storage, special feature?
- Security

  Who have access to EC2 instances (Security Group, SSH keypair).

Pricing model: Reserved, on-demand, spot, spot-block

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 your use.

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 you reserve.
- Discount on hourly charge for that instance.
- 1-year or 3-year term.

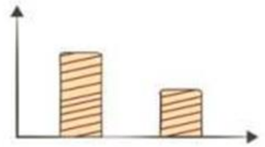
Scheduled Reserved Instances

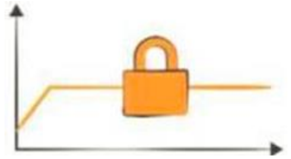
- Purchase a capacity reservation that is always available on a recurring schedule you specify.
- 1-year term.

Spot Instances

- Instances run as long as they are available and your 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 you have flexibility in when your applications can run.

Amazon EC2 pricing models: Benefits

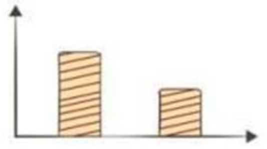




On-Demand Instances	Spot Instances	Reserved Instances	Dedicated Hosts
• Low cost and flexibility	• Large scale, dynamic workload	• Predictability ensures compute capacity is available when needed	• Save money on licensing costs • Help meet compliance and regulatory requirements

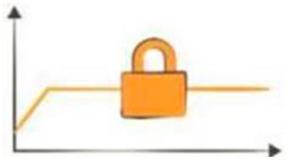
Amazon EC2 pricing models: Use cases

Spiky Workloads



Time-Insensitive Workloads

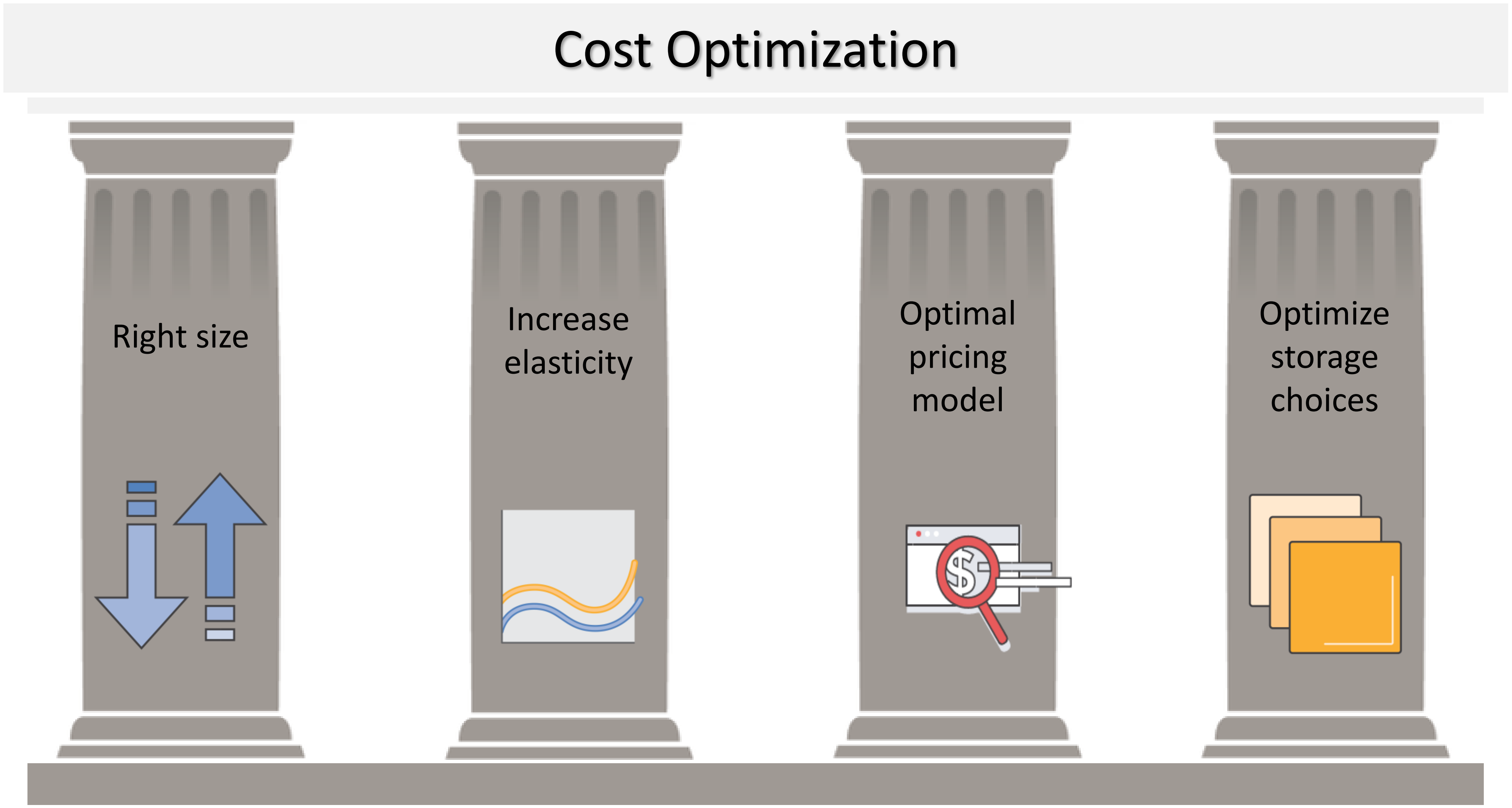
Steady-State Workloads



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 and end times • Applications only feasible at very low compute prices • Users with urgent computing needs for large amounts of additional 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) • Compliance and regulatory restrictions • Usage and licensing tracking • Control instance placement

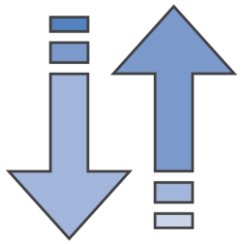
The four pillars of cost optimization



Pillar 1: Right size

Pillars:

1. Right size
2. Increase elasticity
3. Optimal pricing model
4. Optimize storage choices



✓Provision instances to match the need

- CPU, memory, storage, and network throughput
- Select appropriate instance types for your use

✓Use Amazon CloudWatch metrics

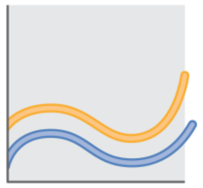
- How idle are instances? When?
- Downsize instances

✓Best practice: Right size, then reserve

Pillar 2: Increase elasticity

Pillars:

1. Right-Size
2. Increase Elasticity
3. Optimal pricing model
4. Optimize storage choices



✓**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

Pillar 3: Optimal pricing model

Pillars:

1. Right-Size
2. Increase Elasticity
3. Optimal pricing model
4. Optimize storage choices

✓Leverage the right pricing model for your use case

- Consider your usage patterns

✓Optimize and *combine* purchase types

✓Examples:

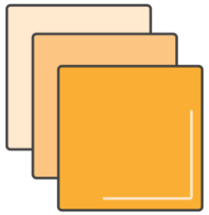
- Use **On-Demand Instance** and **Spot Instances** for variable workloads
- Use **Reserved Instances** for predictable workloads

✓Consider serverless solutions (AWS Lambda)

Pillar 4: Optimize storage choices

Pillars:

1. Right-Size
2. Increase Elasticity
3. Optimal pricing model
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.
- ✓ 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

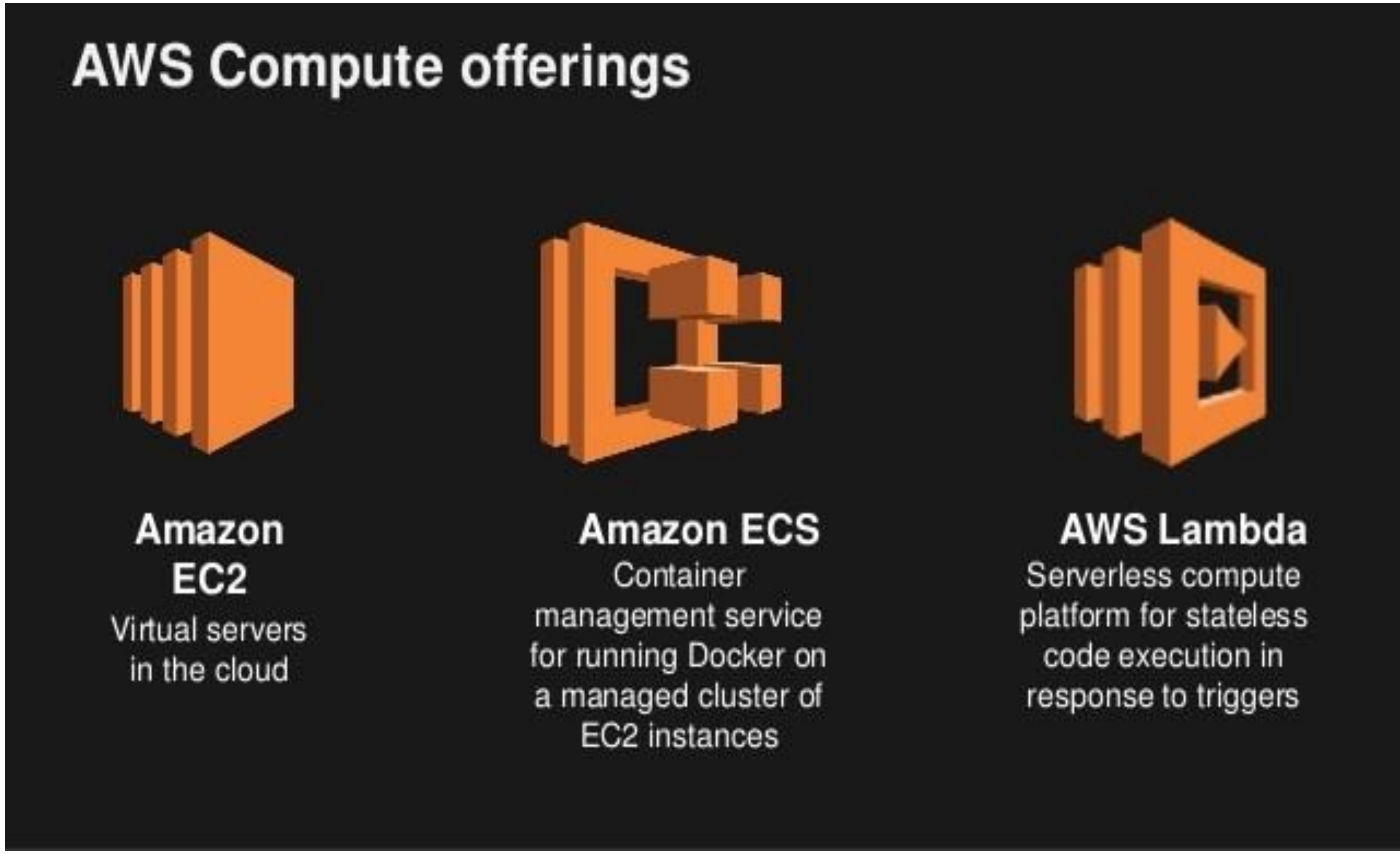
- Cost optimization is an ongoing process.
- Recommendations –
  - Define and enforce **cost allocation tagging**.
  - Define metrics, set targets, and review regularly.
  - Encourage teams to **architect for cost**.
  - Assign the responsibility of optimization to an individual or to a team.

Section 4

From VM to Container and Serverless

- ✓ AWS compute Spectrum
- ✓ Container
- ✓ AWS Lambda

AWS Compute Spectrum



AWS Compute offerings

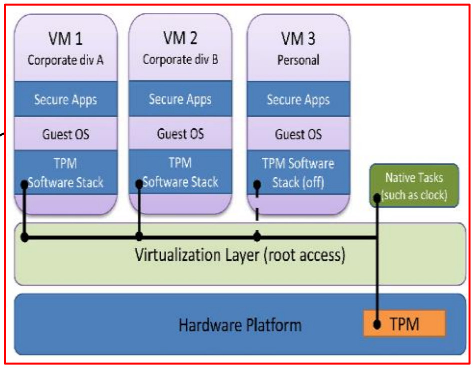
Amazon EC2
Virtual servers in the cloud

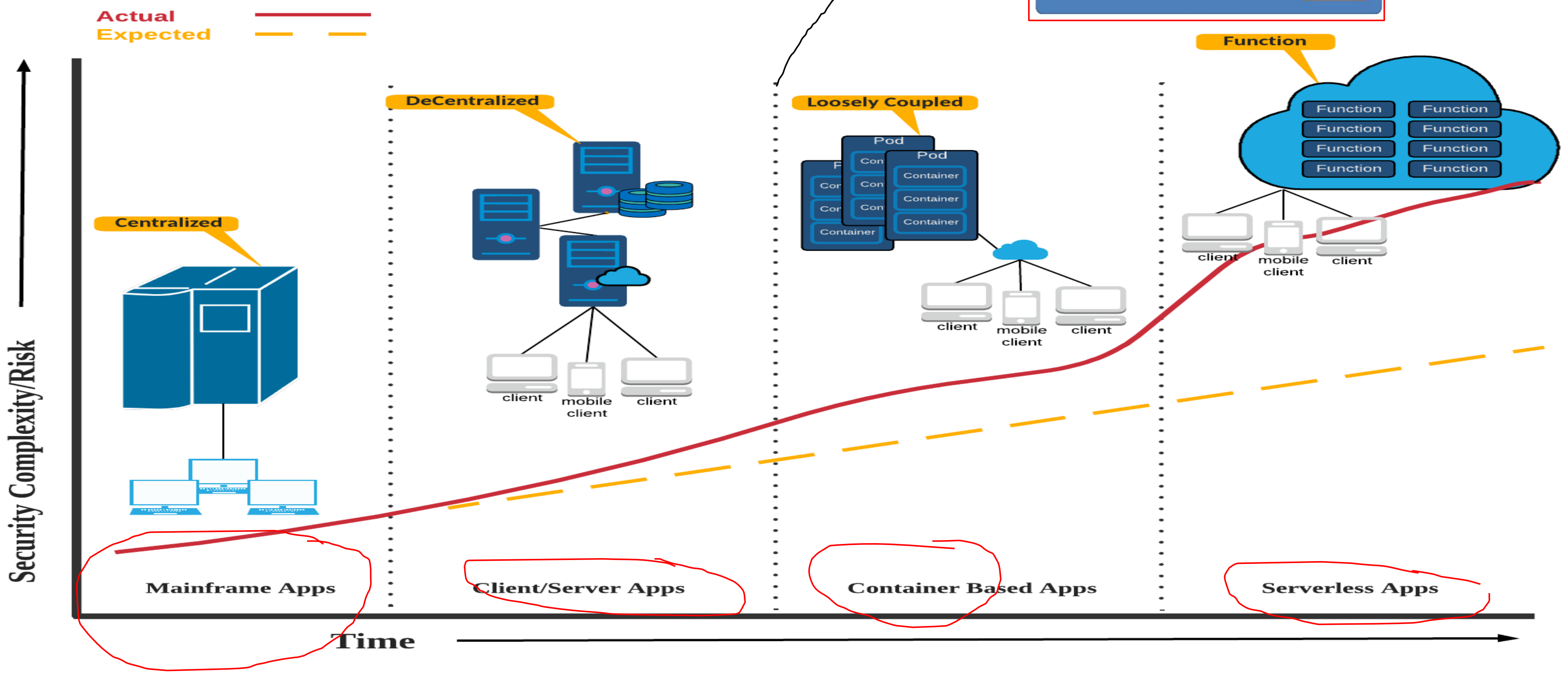
Amazon ECS
Container management service for running Docker on a managed cluster of EC2 instances

AWS Lambda
Serverless compute platform for stateless code execution in response to triggers

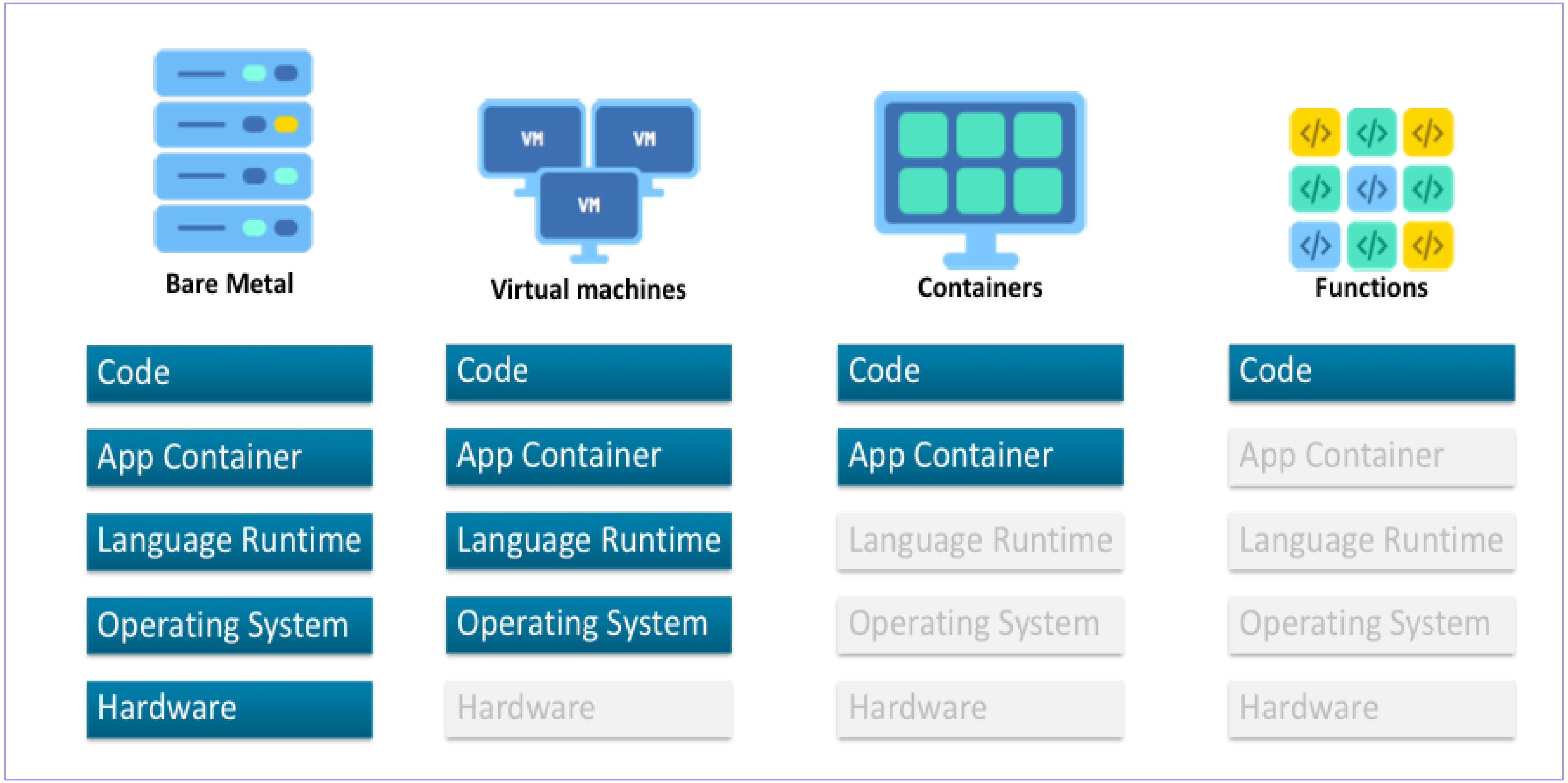
Compute Transformation

Virtual Machines





Computing Transformation

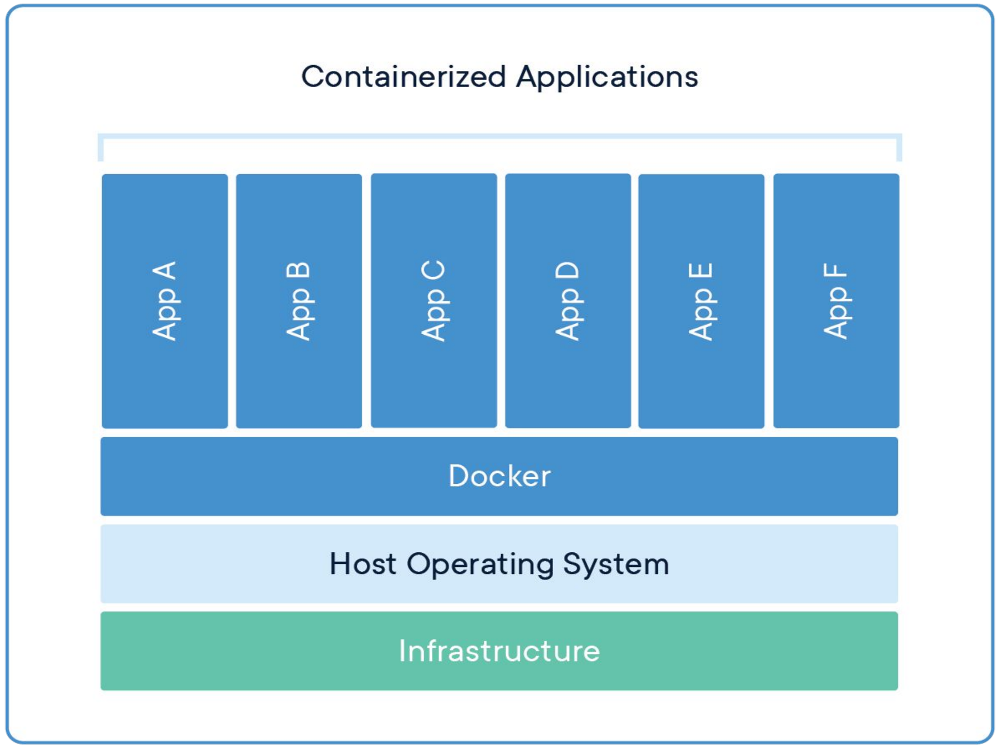


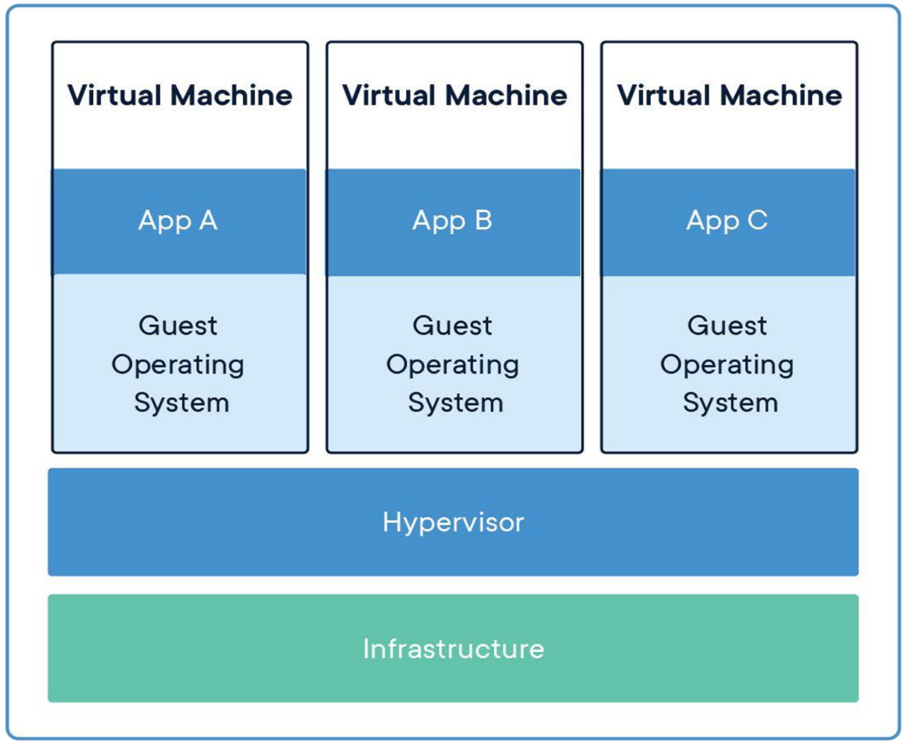
Compute Transformation

- From physical machine to Virtual Machine
  - Bare-metal Hypervisor
- From VM to Docker/Container
  - Container is packaging processes/lib/namespace
- From Container to Serverless
  - Serverless is managed services hiding all the details

Container and VM

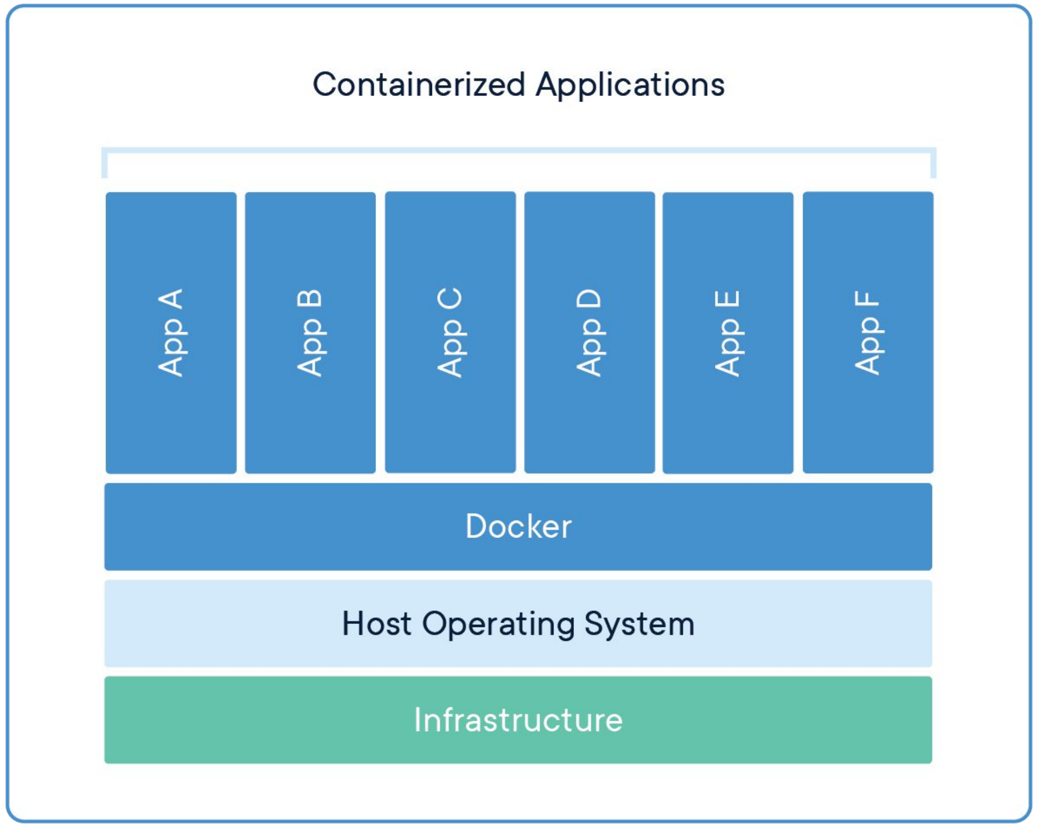
- Similar resource isolation and allocation benefits,
- Containers virtualize the operating system instead of hardware.
- Containers are more portable and efficient.





What is a container

- A docker engine virtualize an OS to multiple apps/containers.
- A docker image is a lightweight, standalone, executable package of software that includes everything needed to run an application: code, runtime, system tools, system libraries and settings.
- Multiple containers can run on the same virtual machine and share the OS kernel with other containers, each running as isolated processes in user space.



Benefits of Server-less Computing

- No Servers to Manage
- Continuous Scaling
- Dynamic allocation of resources
- Avoid overallocation of resources
- Never Pay for Idle: pay-per-usage

AWS Lambda

AWS Lambda is an event-driven, serverless computing platform provided by Amazon as a part of Amazon Web Services. It is a computing service that runs code in response to events and automatically manages the computing resources required by that code. It was introduced in November 2014.

AWS Lambda

47

We expanded the cloud compute spectrum

- **EC2**
- **Container (ECS)**
- **Lambda (Server less)**

Questions?

1. EC2 Dive Deeper

2. ELB and ASG

3. EC2 Cost Optimization

4. AWS Compute Spectrum

5. Container and Serverless

THE UNIVERSITY OF TEXAS AT DALLAS
Naveen Jindal School of Management

FINANCE TRADING LAB

SPEAKERS LIST FINHACK SPRING'22

Date: 03/26/2022
Time: 9 am to Noon
Venue: JSOM 1.107

9 am - 9:10 am **Dr. Harold Zhang**

Department Chair of Finance

Opening Remarks

9:10 - 9:55 am **Mr. Kevin Kirksey**

President, ALM First Analytics

Data Intelligence in M&A and Private Equity

9:55 - 10:40 am **Mr. Tieyi Guo**

VP Cognitive Computing Platform, Fidelity

Fintech in the Financial Services Industry

10:40 - 11:10 am **Mr. Logan Song**

Chief Cloud Architect, Dito

Hack and Unlock the Triangle

11:10 - 11:40 am **Mr. Vivekpandian Veerapandian**

Data Scientist, Forbes

Data Visualization & Modeling using Python

11:40 - 12:10 pm **Ms. Anomita Chandra**

Business Solutions Analyst, Terminix

Tableau in Business Analytics