Module 1, Section 1: Cloud Concepts Overview

aws academy

What's In This Module

- Part 1: What is Cloud Computing?
- Part 2: Six Advantages of Cloud Computing
- Part 3: What is Amazon Web Services (AWS)?
- Part 4: The AWS Cloud Adoption Framework (CAF)

Module Objectives

Discuss key concepts related to cloud computing and the advantages of cloud computing:

- Define different types of cloud computing.
- Describe six advantages of cloud computing.
- Describe cloud deployment models.
- Review the AWS Cloud Adoption Framework (CAF).

Part 1: What is Cloud Computing?

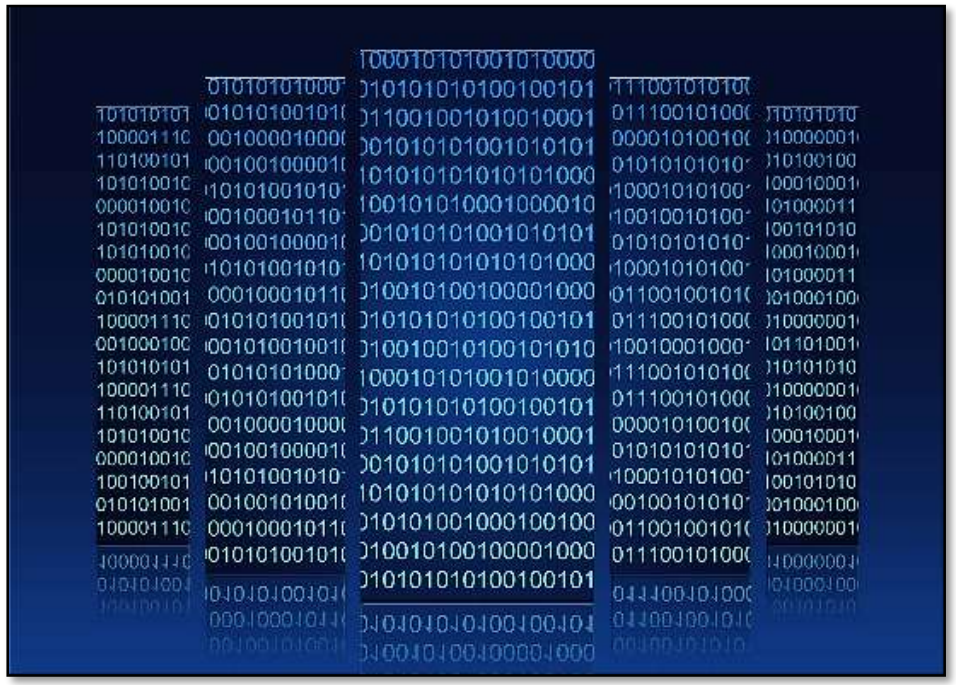
What is Cloud Computing?

What is Cloud Computing?

Cloud computing is the **on-demand** delivery of compute power, database storage, applications, and other IT resources through a cloud services platform **via the internet** with **pay-as-you-go** pricing.

Before Cloud Computing

Cloud computing enables you to **stop thinking of your infrastructure as hardware**, and instead **think of it (and use it) as software.**



Before Cloud Computing

- Hardware solutions are **physical**. This means they require:
  - Space
  - Staff
  - Physical security
  - Planning
  - Capital expenditure
- Guess at theoretical maximum peaks
  - Is there enough resource capacity?
  - Do we have sufficient storage?

What if your needs change?

You have to go through the **time, effort, and cost** required to change all these.

Utilizing Cloud Computing

Software is flexible.

If your needs change, your software can change much more **quickly, easily, and cost-effectively** than your hardware.

Three Models of Cloud Computing

IaaS

Infrastructure
as a Service

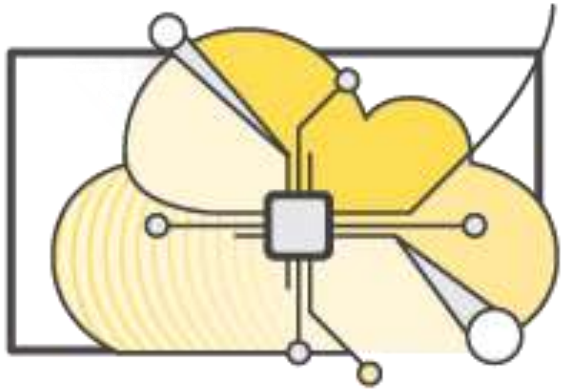
PaaS

Platform
as a Service

SaaS

Software
as a Service

Three Cloud Deployment Models



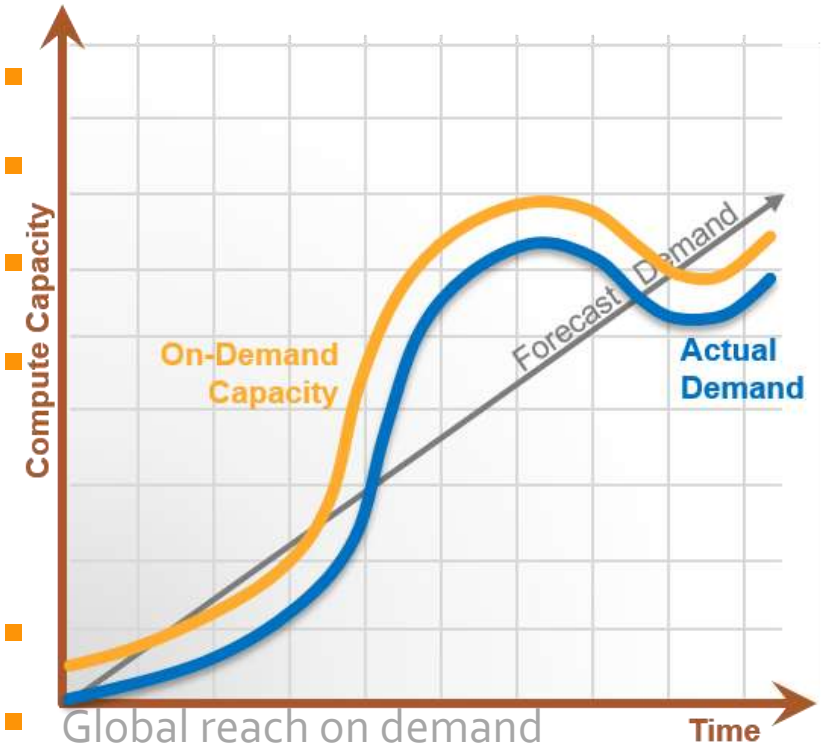
All-In Cloud

Hybrid

Private Cloud (On-premises)

All-In Cloud versus On-Premises

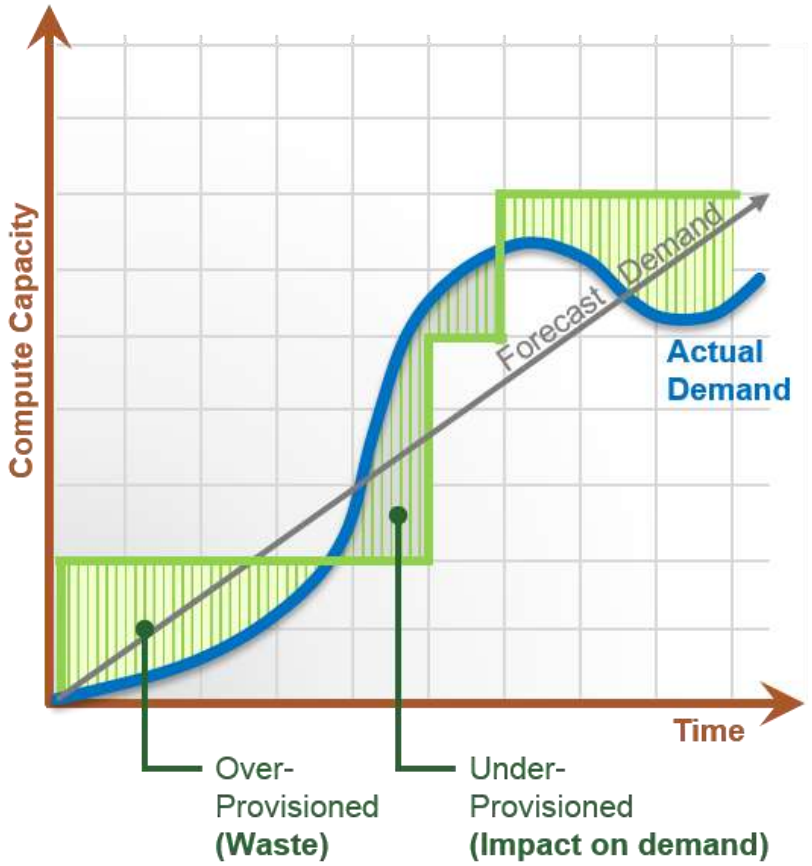
All-In Cloud



Global reach on demand

- ✓ No waste
- ✓ Meets demand

On-Premises



All-In Cloud versus On-Premises

All-In Cloud

- No upfront investment
- Low ongoing costs
- Focus on innovation
- Flexible capacity
- Speed and agility
- Global reach on demand

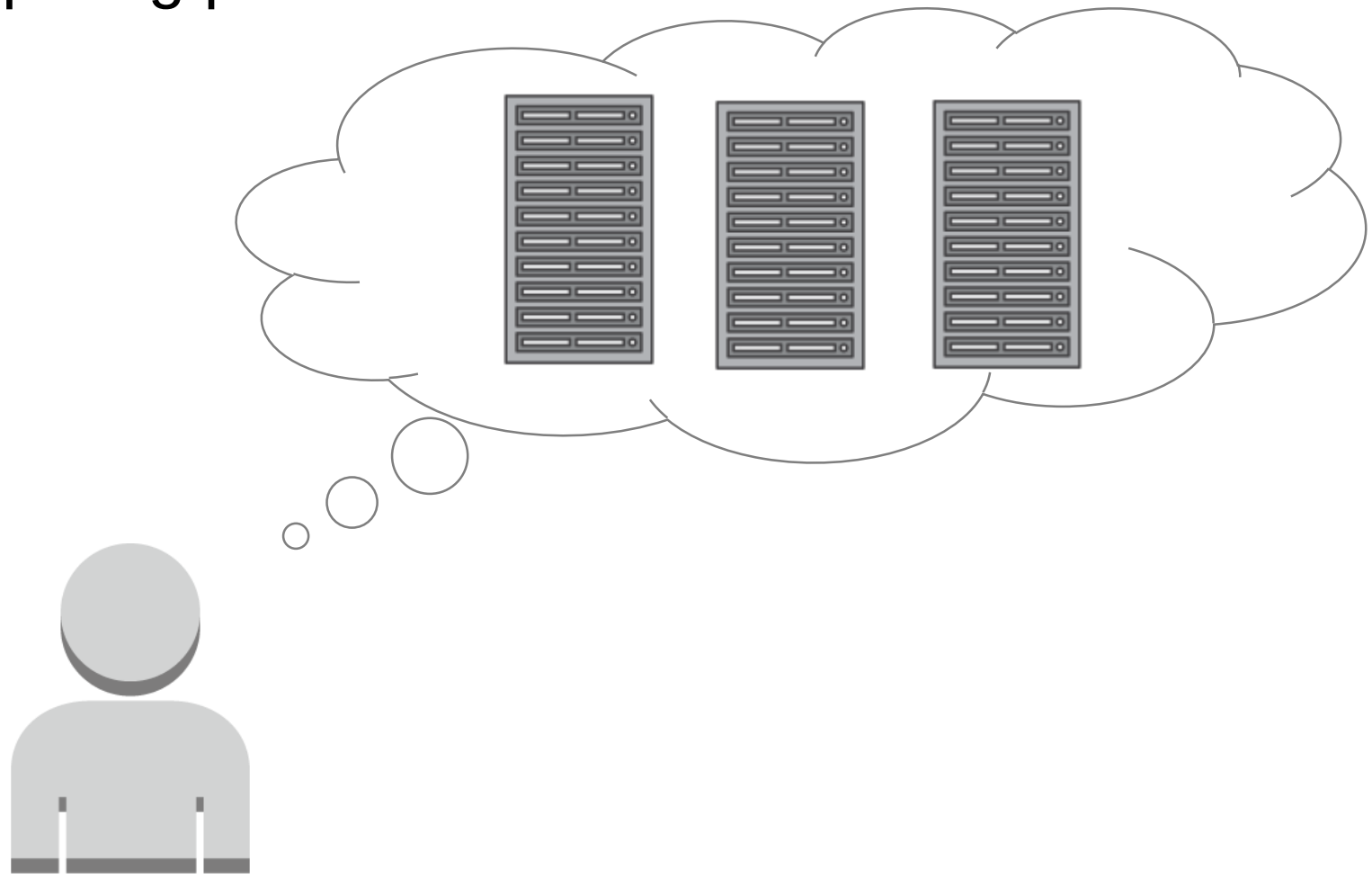
On-Premises

- Large initial purchase
- Labor, patches, and upgrade cycles
- Systems administration
- Fixed capacity
- Long procurement cycle and setup
- Limited geographic regions

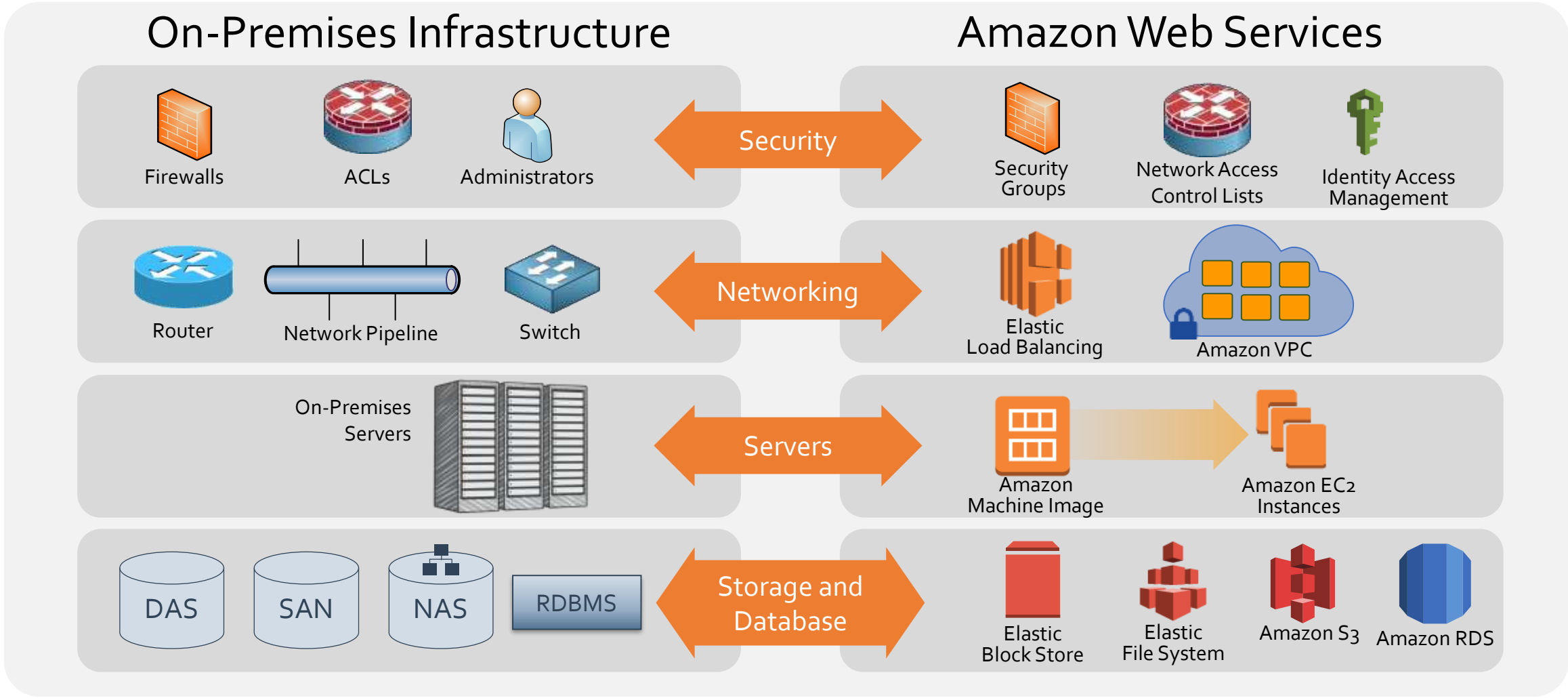
What can you do in the cloud?

You can use a cloud computing platform for:

- Application Hosting
- Backup and Storage
- Content Delivery
- Websites
- Enterprise IT
- Databases



On-Premises and AWS Comparison



Important Cloud Terminology

- **High Availability (Highly Available):**
  - Accessible when you need it
- **Fault Tolerance (Fault Tolerant):**
  - Ability to withstand a certain amount of failure and still remain functional
- **Scalability (Scalable):**
  - Ability to easily grow in size, capacity, and/or scope when required
  - Growth is (usually) based on demand
- **Elasticity (Elastic):**
  - Ability to grow (scale) when required and to reduce in size when resources are no longer needed

Summary

- Cloud computing is the on-demand delivery of IT resources online with pay-as-you-go pricing.
- Three models of cloud computing are:
  - Infrastructure as a Service (IaaS)
  - Platform as a Service (PaaS)
  - Software as a Service (SaaS)
- All-in cloud, hybrid, and private cloud are three cloud deployment models.
- Cloud services are available to replace traditional on-premises computing activities.

Part 2: Six Benefits of Cloud Computing

Advantage #1: Capex to Variable Expense

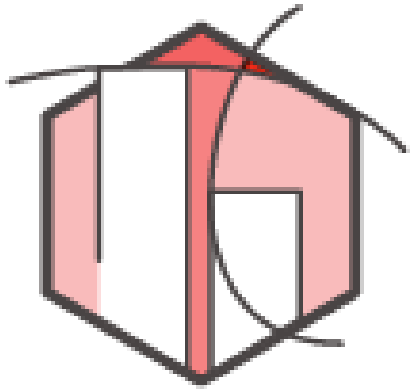
Trade **capital expense** for **variable expense**.

Capital Expense vs. Variable Expense

- **Capital expense** (capex): Funds used by a company to acquire, upgrade, and maintain physical assets such as property, industrial buildings, or equipment.

- **Variable expense**: A variable expense is an expense that is easily altered or avoided by the person bearing the cost.

Advantage #2: Economies of Scale

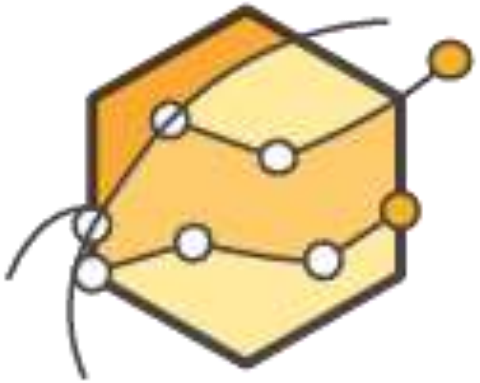


Benefit from **massive economies of scale**.

Economies of Scale

- Hardware solutions are **physical** and require:
  - Space
  - Staff
  - Physical security
- Significant cost to procure and house these resources.
- No purchasing power.
- Cloud providers leverage hundreds of thousands of customers to achieve economies of scale.

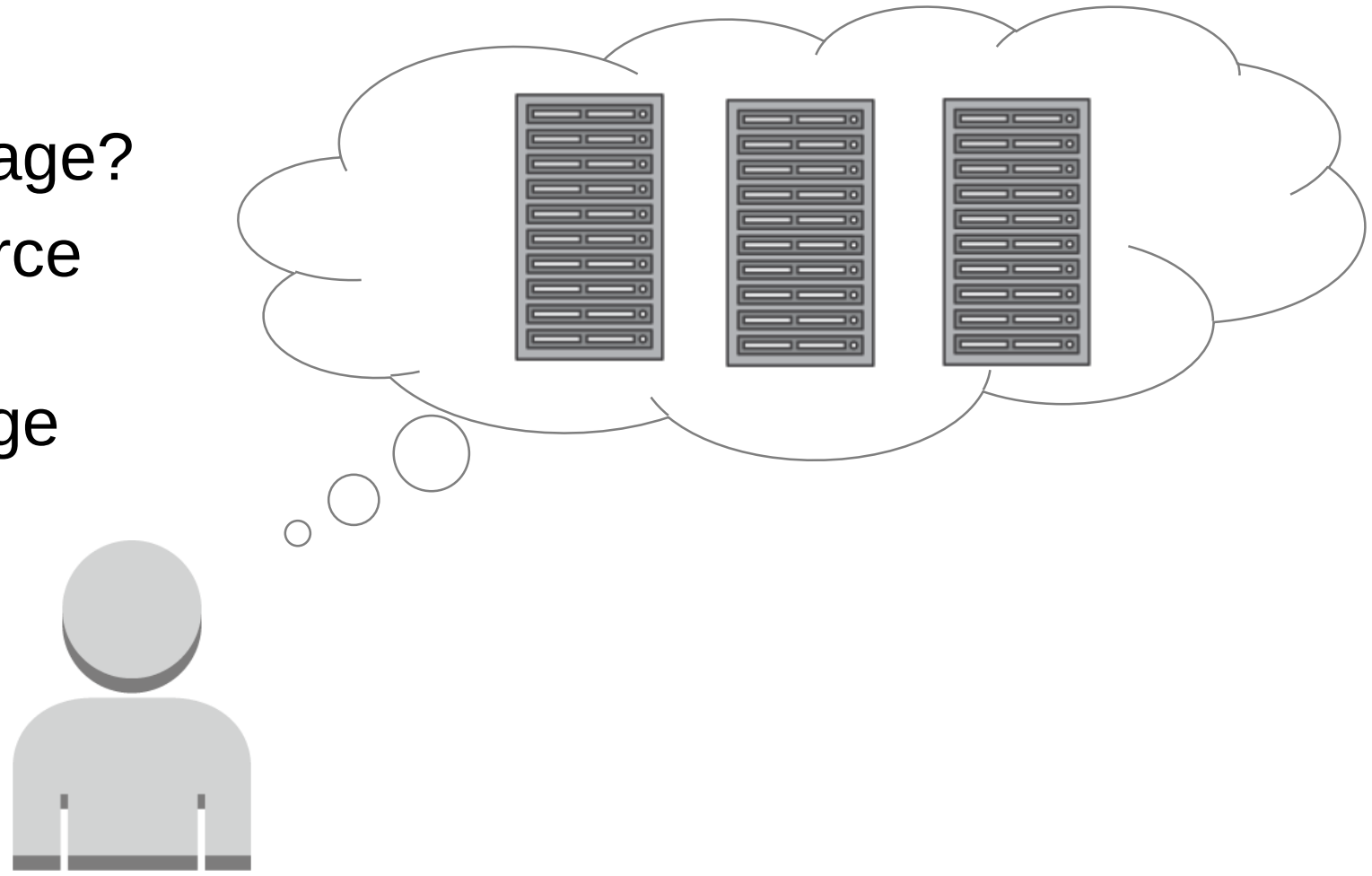
Advantage #3: Capacity Planning



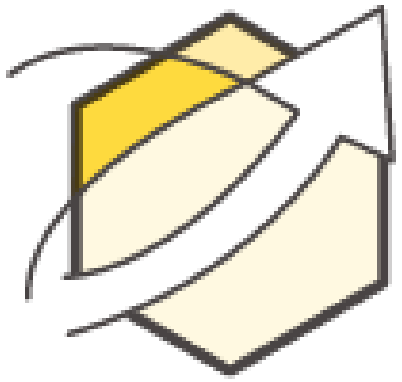
Eliminate guessing on your capacity needs.

Guessing about Capacity

- What are the potential maximum peaks in usage?
- Is there enough resource capacity at peak?
- Is the amount of storage sufficient?



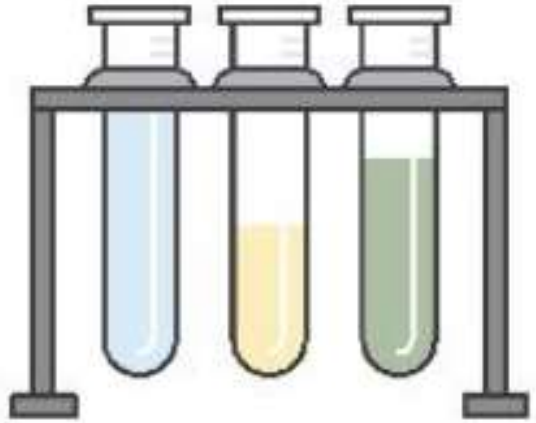
Advantage #4: Speed and Agility



Increase **speed** and **agility**.

Increase Speed and Agility

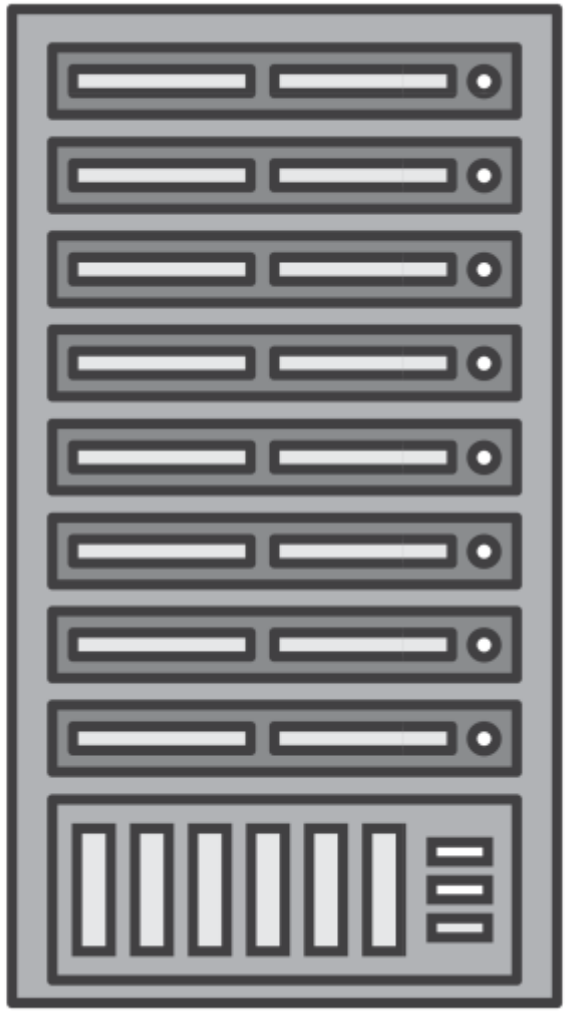
- Rapid availability of new resources
  - Provision resources in minutes, not weeks.
- Increase Innovation
  - Quick, low cost experimentation.
  - Leverage pre-fabricated functionality without requiring in-house expertise. (i.e., data warehousing, analytics)
- Increase experimentation
  - Explore new avenues of business with minimal risk and expense.
  - Test with different configurations.



Advantage #5: Spend Strategically

Stop spending money on running and maintaining data centers.

Stop Spending Money on Data Centers



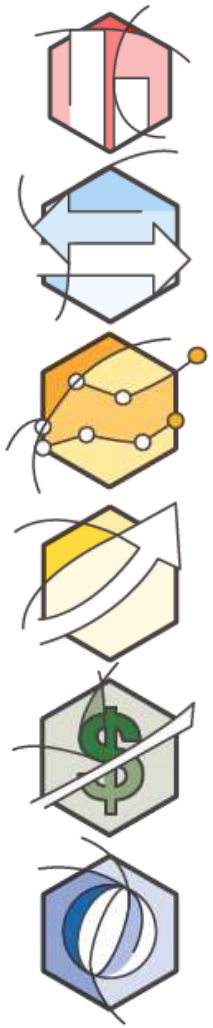
- Focus on customers
- Focus on projects that differentiate the business
- Delegate the racking, stacking and powering of servers to the cloud provider

Advantage #6: Ease of Deployment

Go global in minutes.

Go Global in Minutes

Summary



Trade **capital expense** for **variable expense**.

Benefit from **massive economies of scale**.

Eliminate guessing on your capacity needs.

Increase **speed** and **agility**.

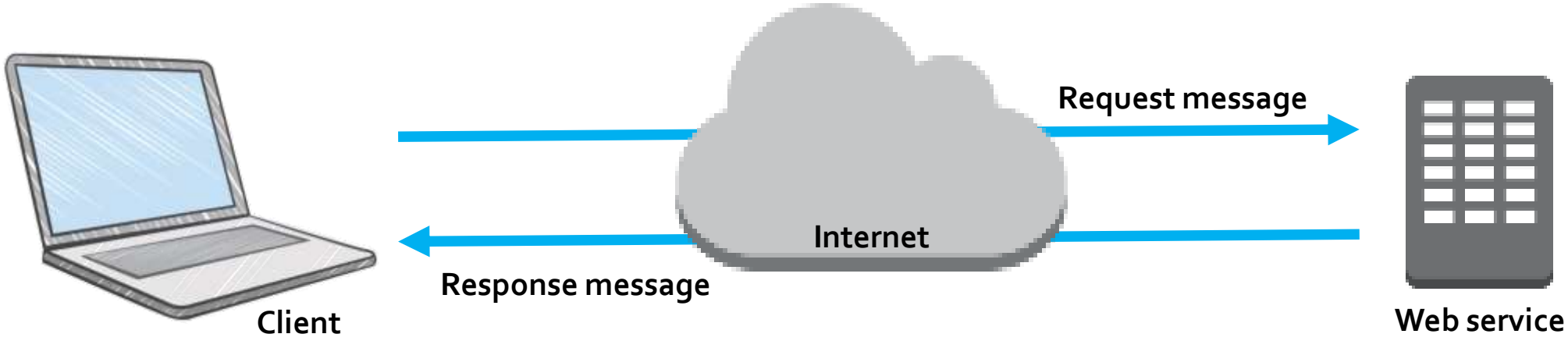
Stop spending money to run and maintain data centers.

Go global in minutes.

Part 3: What is Amazon Web Services (AWS)?

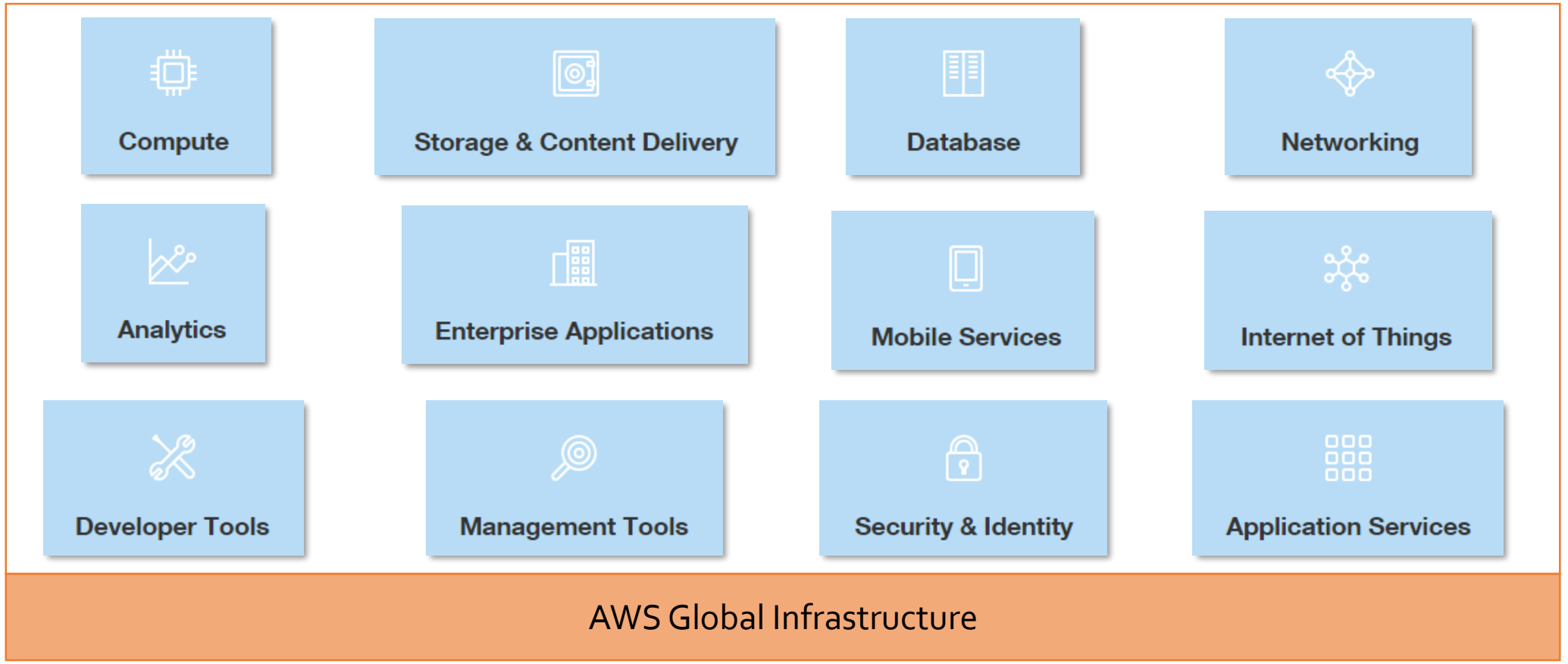
What are Web Services?

A **web service** is any piece of software that makes itself available over the internet and uses a **standardized format** (XML or JSON) for the request and the response of an **API interaction**.



What is AWS?

AWS is a secure cloud platform with more than 165 different services that include solutions for:



AWS by Category: Core Services

Compute


Amazon EC2


AWS Lambda


Auto Scaling


AWS Elastic Beanstalk


Amazon Elastic Container Registry


Amazon Elastic Container Service


Amazon Lightsail


AWS Batch

Networking


Amazon VPC


Amazon Route 53


AWS Direct Connect


Elastic Load Balancing

Storage


Amazon S3


Amazon EBS


Amazon CloudFront


Amazon Glacier


Amazon Elastic File System


AWS Snowball


Storage Gateway


AWS Snowmobile

Database


Amazon RDS


Amazon DynamoDB


Amazon Redshift


AWS Database Migration Service


Amazon ElastiCache

AWS by Category: Foundational Services

Analytics


Amazon EMR


AWS Data Pipeline


Amazon Elasticsearch


Amazon Kinesis


Amazon Machine Learning


Amazon QuickSight


Amazon Redshift


Amazon Athena

Enterprise Apps


Amazon WorkSpaces


Amazon WorkMail

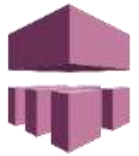

Amazon WorkDocs

Mobile Services


AWS Mobile Hub


Amazon SNS


Amazon Cognito


AWS Device Farm


Amazon Mobile Analytics


AWS Mobile SDKs


Amazon Pinpoint

Internet of Things


AWS IoT


AWS Greengrass

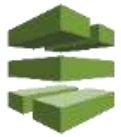
AWS by Category: Developer and Operations Services

aws academy

Developer Tools


AWS CodeCommit


AWS CodeDeploy


AWS CodePipeline


AWS CodeBuild


AWS X-Ray

Management Tools


Amazon CloudWatch


AWS CloudFormation


AWS CloudTrail


AWS Config


AWS OpsWorks


AWS Service Catalog


AWS Trusted Advisor


AWS Organizations

Security & Identity


AWS Identity and Access Management


AWS Directory Service


Amazon Inspector


AWS CloudHSM


AWS Key Management Service


AWS WAF


AWS Certificate Manager


AWS Shield

App Services


Amazon API Gateway


Amazon AppStream


Amazon CloudSearch


Amazon Elastic Transcoder


Amazon SES


Amazon SNS


Amazon SQS


Amazon SWF

Core Services: The Basics

Compute


Amazon EC2


AWS Lambda

Auto Scaling


AWS Elastic Beanstalk

Amazon Elastic Container Registry

Amazon Elastic Container Service

Amazon Lightsail

AWS Batch

Networking


Amazon VPC


Amazon Route 53

AWS Direct Connect

Elastic Load Balancing

Storage


Amazon S3


Amazon Glacier


Amazon CloudFront

Amazon EBS

Amazon Elastic File System


AWS Snowball

Storage Gateway

AWS Snowmobile

Database


Amazon RDS


Amazon DynamoDB

Amazon Redshift

AWS Database Migration Service

Amazon ElastiCache

Core Services: The Basics

Management Tools

Amazon CloudWatch

AWS CloudFormation

AWS CloudTrail

AWS Config

AWS OpsWorks

AWS Service Catalog

AWS Trusted Advisor

Security & Identity

AWS Identity and Access Management

AWS Directory Service

Amazon Inspector

AWS CloudHSM

AWS Key Management Service

AWS WAF

AWS Organizations

AWS Certificate Manager

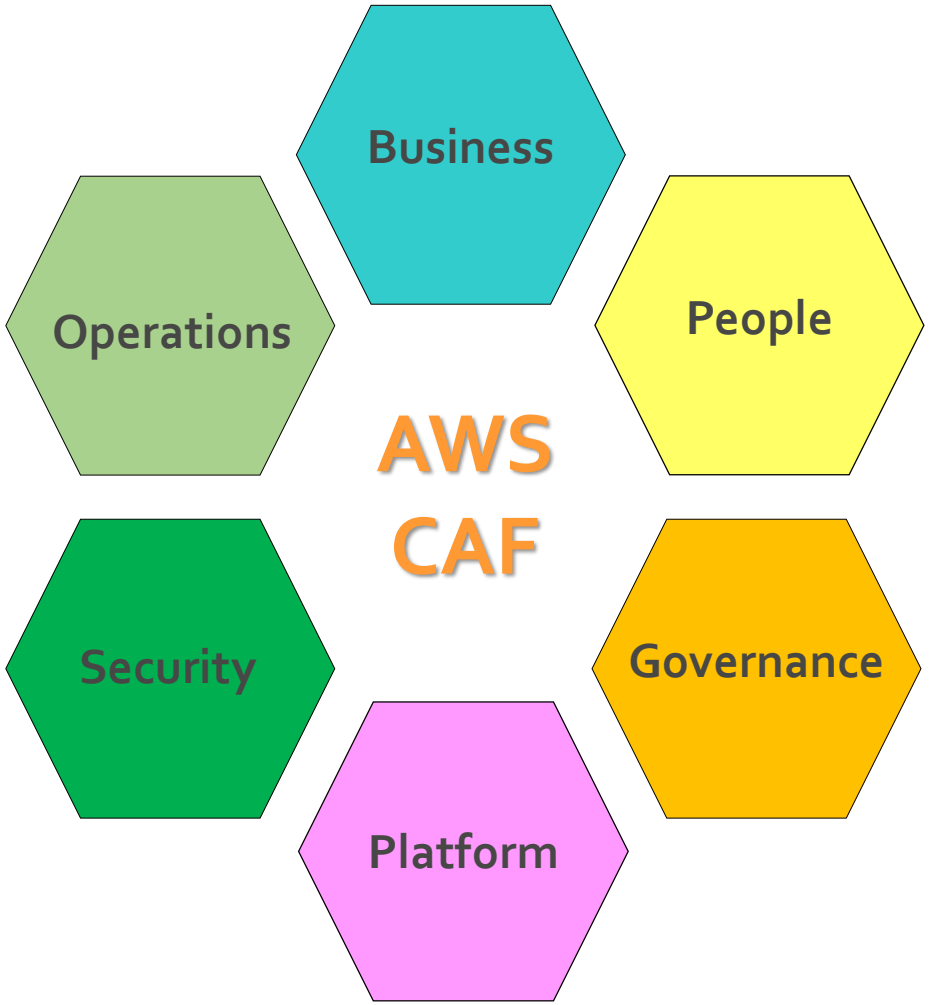
AWS Shield

Access to AWS Services

- AWS Management Console
  - Access on the go with AWS Console Mobile App
- AWS Command Line Interface (AWS CLI)
- Software Development Kits (SDK)

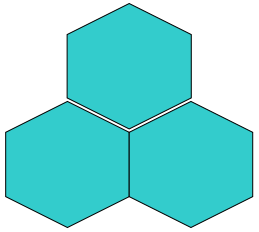
Part 4: The AWS Cloud Adoption Framework

AWS Cloud Adoption Framework (CAF)



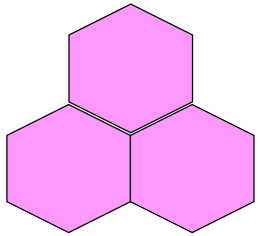
- Perspectives in planning, creating, managing, and supporting a modern IT service
- Guidelines for establishing, developing, and running AWS environments
- Structure for business and IT teams to work together

Six Core Perspectives



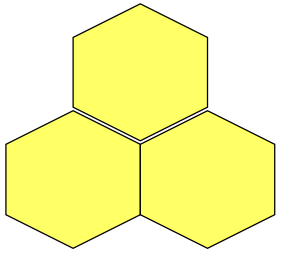
Business Perspective

How will your architectural approaches **align technical delivery to business imperatives**?



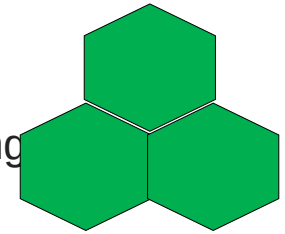
Platform Perspective

What patterns, guidance, and tools are necessary to optimize your use of **technology services** on AWS?



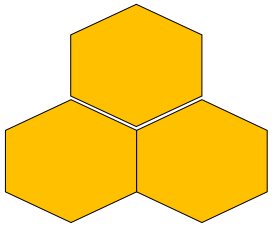
People Perspective

What **skills** are needed in order to adopt the AWS cloud platform? Examples include guiding processes of role descriptions, training, certification, and mentoring.



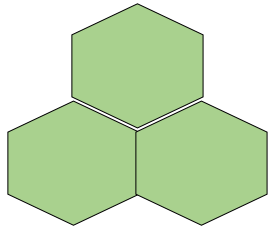
Security Perspective

How will you define and implement the required levels of security, governance, and risk management to **achieve compliance**?



Governance Perspective

How to update the staff skills and **organizational processes** necessary to ensure business governance in the cloud, and manage and measure cloud investments to evaluate business outcomes?



Operations Perspective

How will you provide process, guidance, and tools for optimum **operational service management** of the AWS environment?

Summary

- Defined cloud computing and alternative implementation models
- Described the advantages of cloud computing
- Explored AWS services
- Discussed the AWS CAF

To finish this module:

- Complete:  **Knowledge Assessment**

FUNDAMENTALS OF AWS

What is Cloud?

Someone else's Computer?

A collection of Services that create a platform

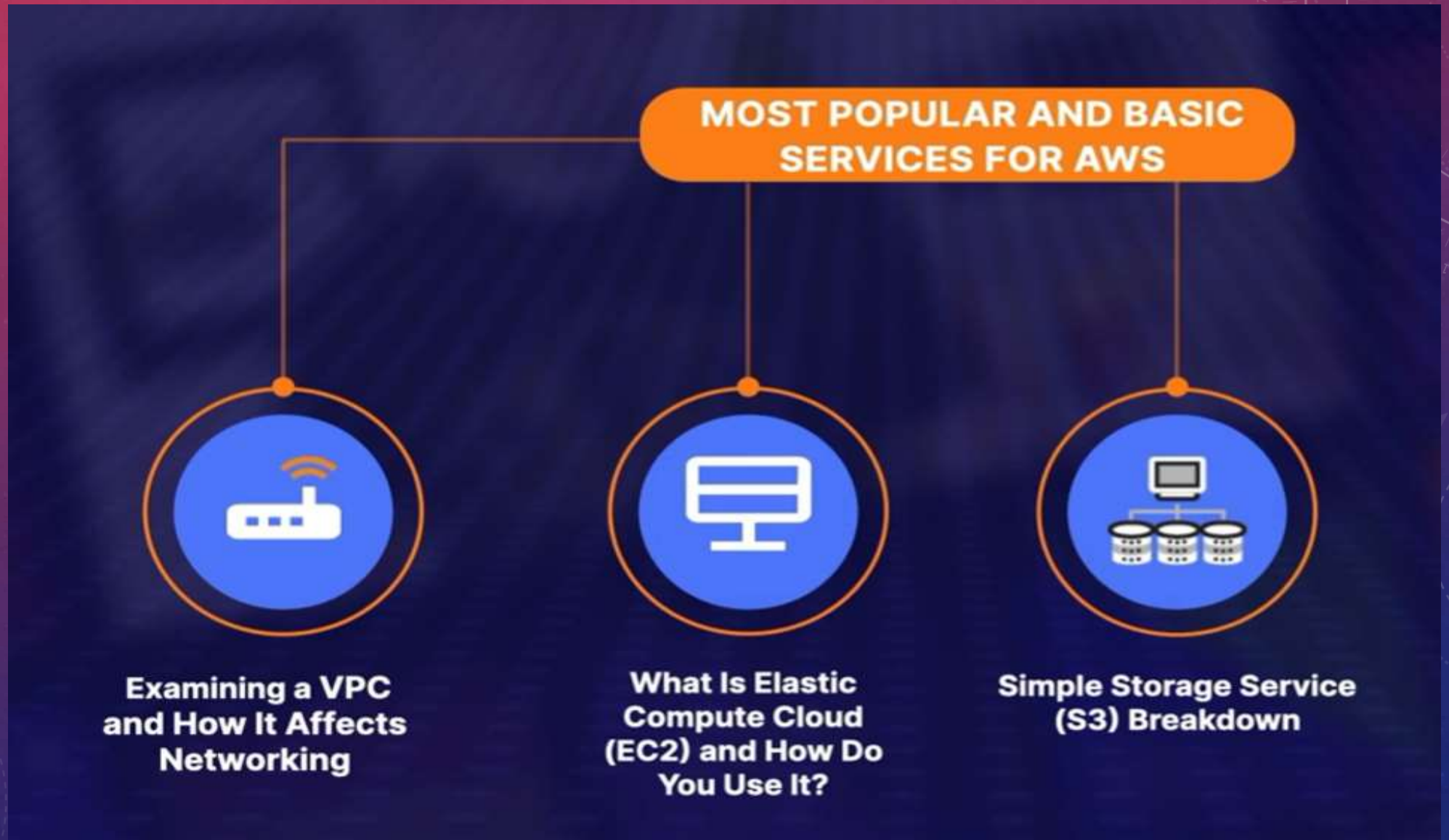
- Media Services
- Databases
- Security
- Virtual Servers
- Networking
- Machine Learning
- Data Storage
- Analytics

DEFINITION

"AWS is the world's most comprehensive and broadly adopted cloud platform, offering over 200 fully featured services from data centers globally. Millions of customers, including the fastest growing startups, largest enterprises and leading government agencies are using AWS to lower costs, become more agile and innovate faster."

- Billed by second for using Services
- 60 seconds minimum
- Use 83 seconds, get charged for 83 seconds
- Lambda recently moved to per millisecond billing

Most Popular and Basic Services for AWS



aws

ASHOK IT

Learn Here... Lead Anywhere...!!

AWS Account Setup

in 2 Mins


aws
IAM
Identity And Access
Management

HOW IDENTITY AND ACCESS MANAGEMENT WORKS?

AWS(Amazon Web Services) will allows you to maintain the fine-grained permissions to the AWS account and the services provided Amazon cloud. You can manage the permissions to the individual users or you can manage the permissions to certain users as group and roles will helps you to manage the permissions to the resources.

- Manage users and their level of access
- Used to set users, permissions and roles
- Grant access to the different parts of the aws platform

How Does AWS IAM Work?

- Create a directory of users called identities, and they can be users inside of your AWS account or users outside of your AWS account. Some of these users may need access to your website and services. Along with managing this directory, you manage what they can do once they log in.
- You can group identities to manage shared permissions and assign access permissions, using security policies and roles

Understand the difference between authentication and authorization.

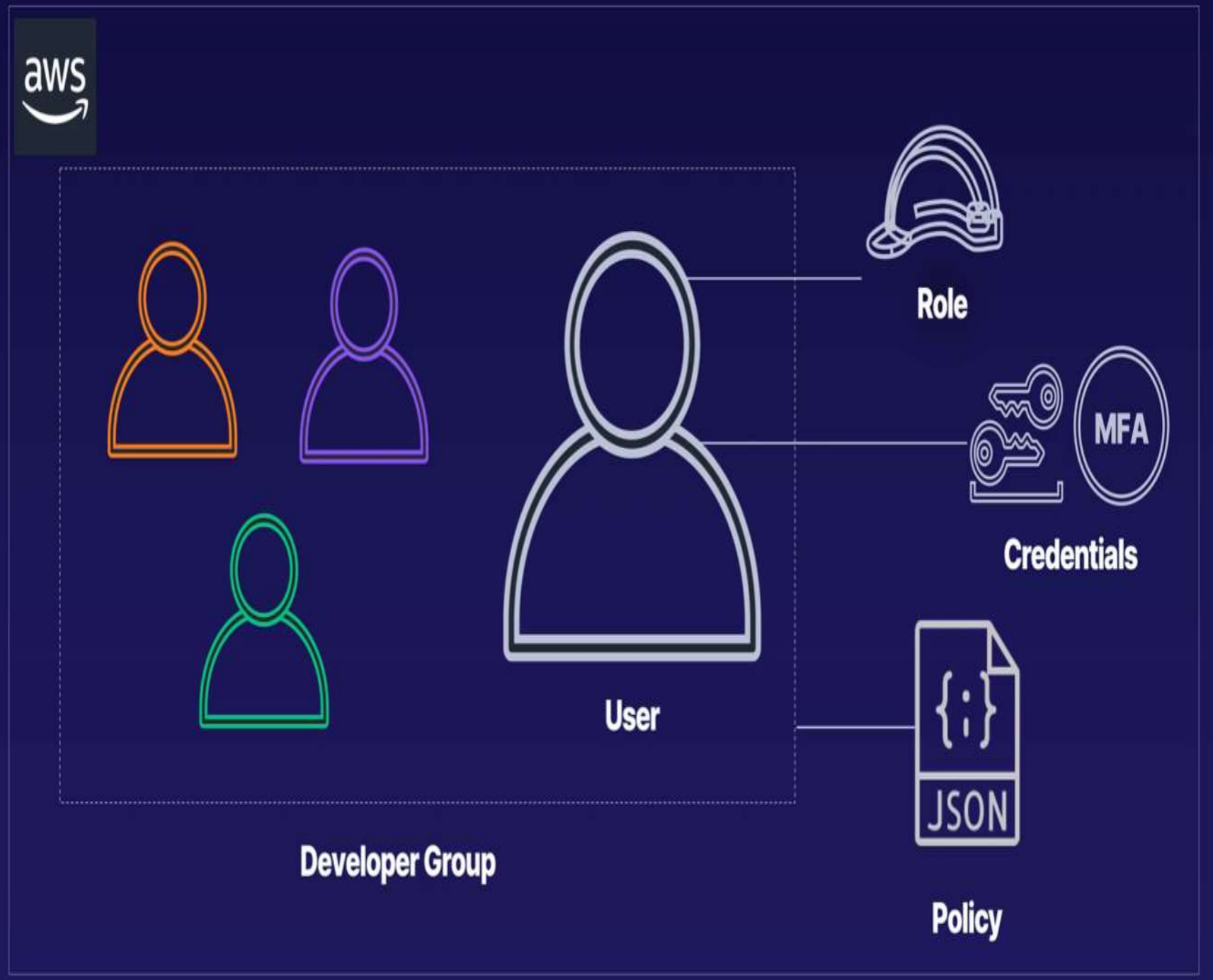
AUTHENTICATION

Authentication identifies **who** you are.

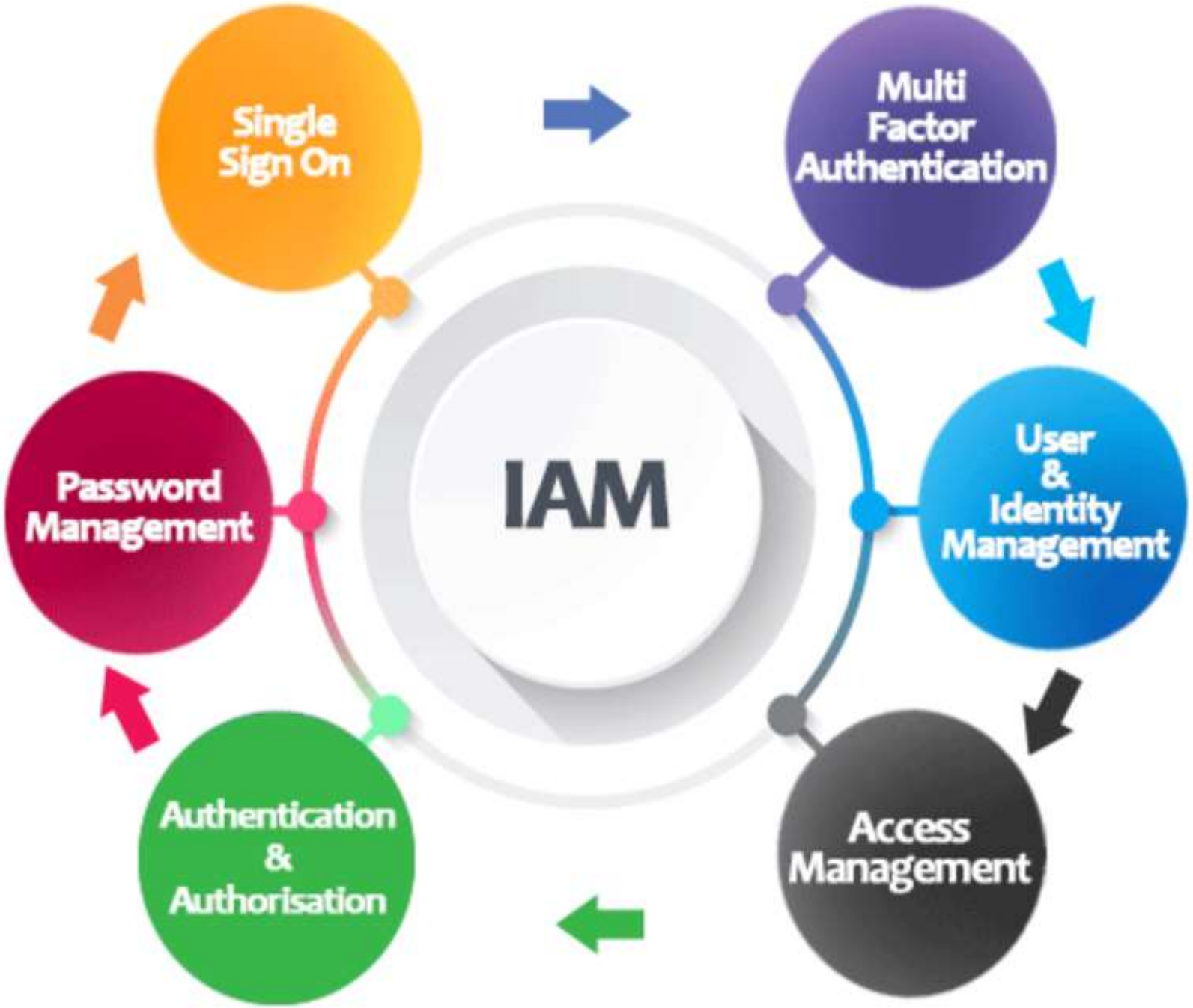
VS.

AUTHORIZATION

Authorization defines **what** you have access to.



- User is an identity.
- Users can be placed in groups. All users within a group share the same permissions.
- Perform a specific duty using a role.
- Policy with granular level permissions.



Key Components of IAM

How to create an IAM User

Simple Storage Service (Amazon S3)

Simple Storage Services (S3)

- Store infinite amount of data
- Highly scalable, reliable, fast, inexpensive data storage infrastructure
- Storage for:
  - Internet applications
  - Backup and recovery
  - Disaster recovery data archives
  - Data lakes for analytics
  - Hybrid cloud storage.

AWS S3 Overview

Object based Storage system - Files are stored as Objects inside Buckets

Key Concepts:

- Buckets
- Objects
- Consistency
- Storage Tiers
- S3 Lifecycle rules
- S3 Performance
- S3 Encryption
- S3 Security

Buckets

- Objects/Files are stored in Buckets (Directories)
- Bucket is an AWS S3 resource that you create
- Buckets are created in a particular region
- Buckets are named unique globally
- Naming convention for buckets:

✓ Bucket names must be between 3 and 63 characters long.

✓ Bucket names can consist only of lowercase letters, numbers, dots (.), and hyphens (-).

✓ Bucket names must begin and end with a letter or number.

✓ Bucket names must not be formatted as an IP address (for example, 192.168.5.4).

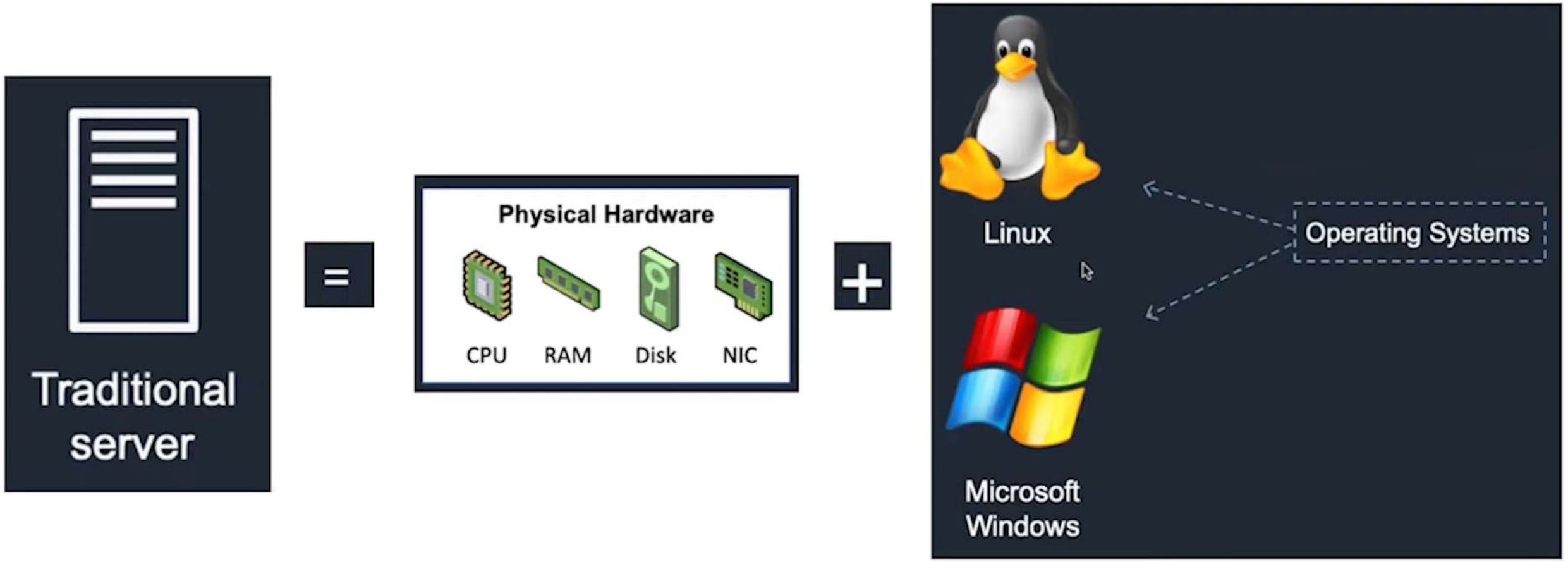
Objects (Files)

- **Objects or files have two key attributes:**
  1. Key : The name that you assign to an object. You use the object key to retrieve the object.
     e.g., example-bucket/directory1/my_file.txt
  2. Value: The content that you are storing.
- Maximum size of an object is **5 TB**
- **Metadata** – A set of name-value pairs with which you can store information regarding the object.
- **Version ID** – Within a bucket, a key and version ID uniquely identify an object.
- **Access Control Information** – You can control access to the objects you store in Amazon S3.

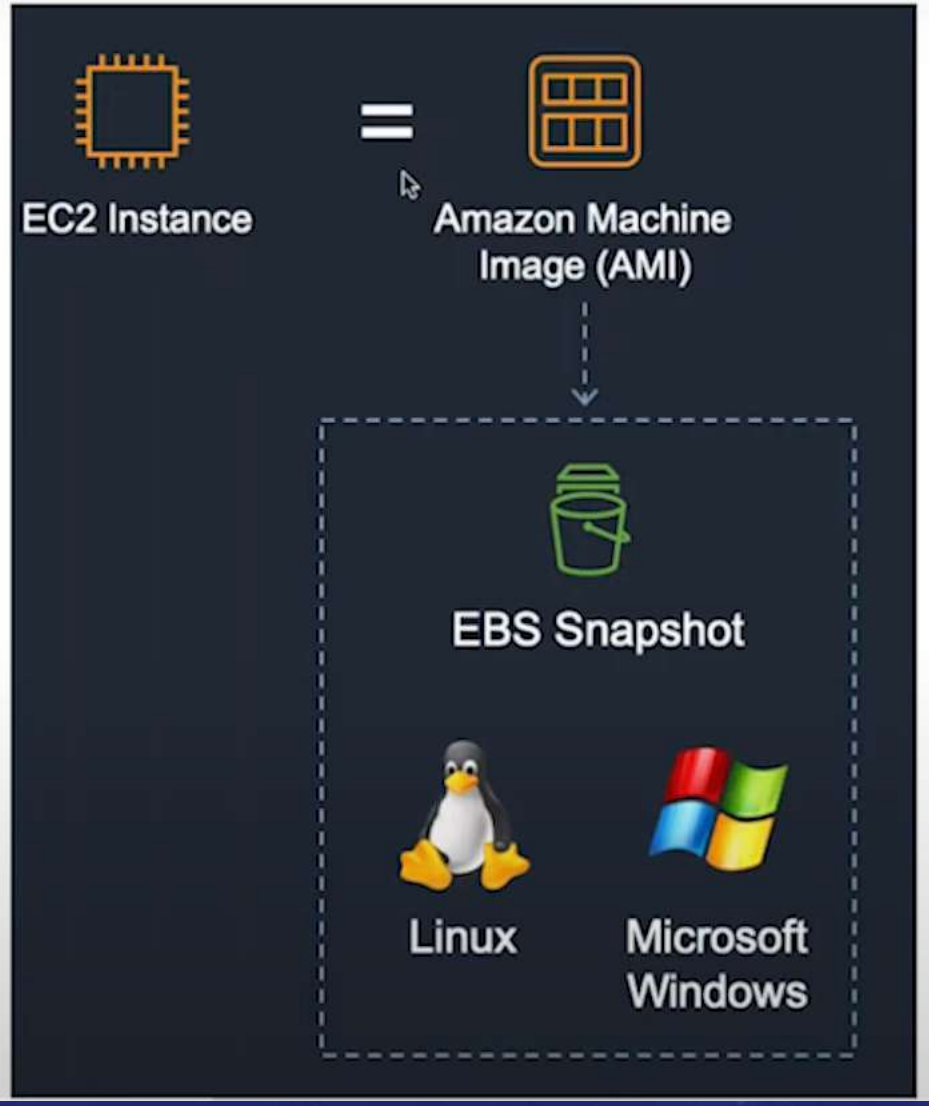
Play from 7:00

Amazon EC2

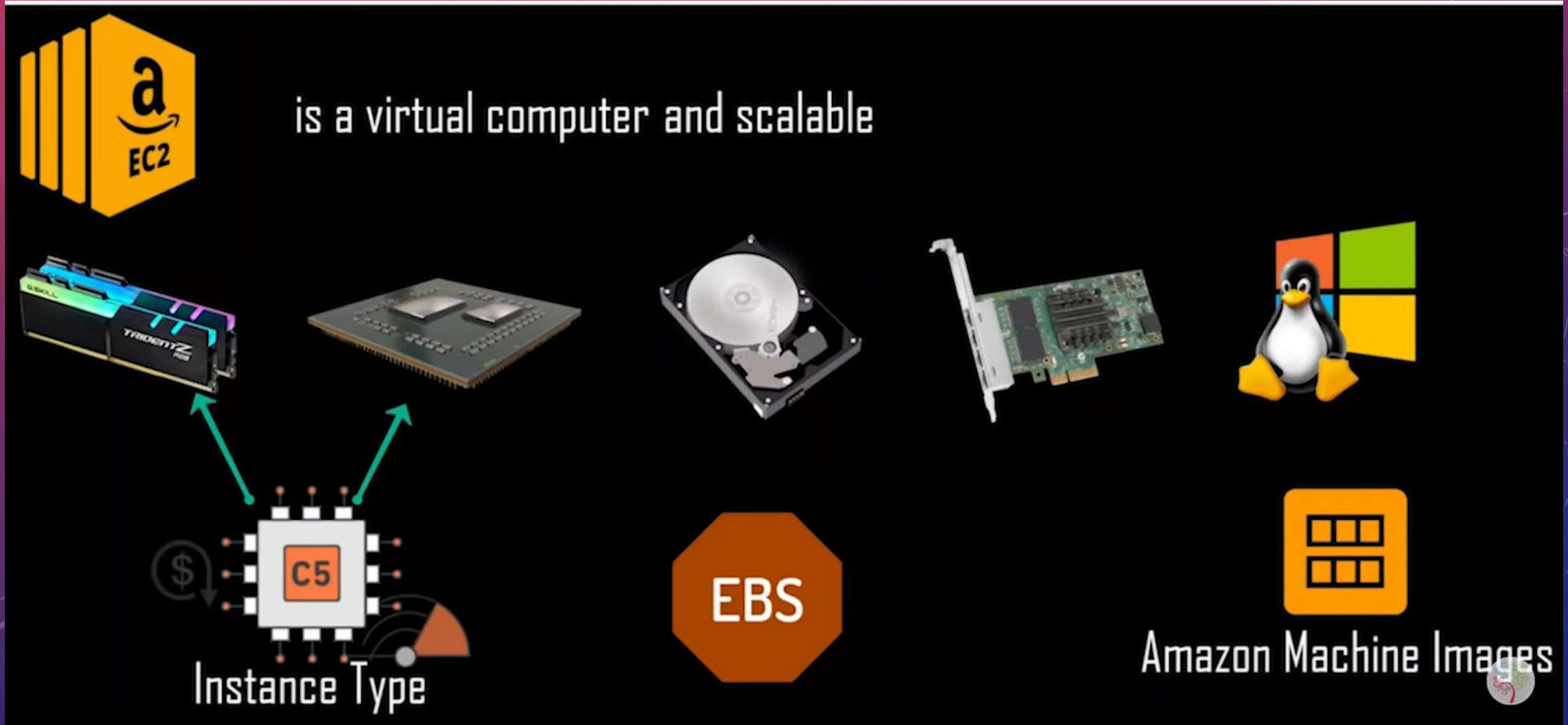
Elastic Compute Cloud EC2



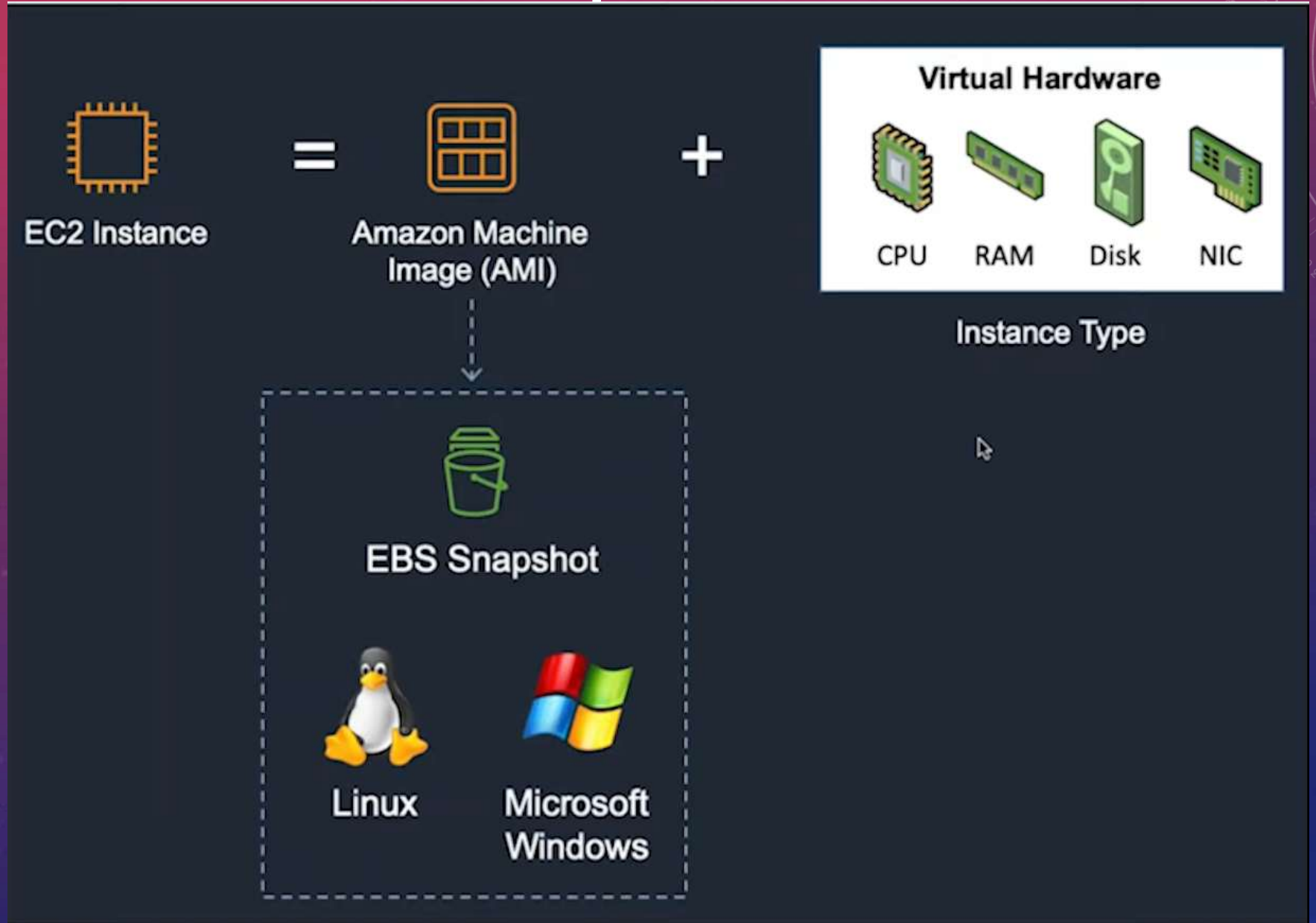
Elastic Compute Cloud EC2



Elastic Compute Cloud EC2



Elastic Compute Cloud EC2



Elastic Compute Cloud EC2

- EC2 Instances – Virtual servers in AWS(billed by Second)
- EC2 service – Provision EC2 instances or virtual servers
- You can create new instances based on AMIs.
- AMIs are preconfigured templates that include base operating systems and any additional software in a single package.
- It provides various combinations of CPU, memory, storage, and networking capacity for provisioning instances. These combinations are called instance types.

Elastic Compute Cloud EC2

Create an Amazon EC2 instance

Step by Step Tutorial

Amazon EC2

Amazon VPC

WHAT IS VIRTUAL PRIVATE CLOUD?

A Virtual Private Cloud (VPC) is a special implementation of public cloud that uses a corporate firewall with virtual data isolation and is managed by a cloud provider.

In a VPC, the cloud provider allocates and provisions a portion of its public cloud infrastructure for a single user of VPC, thus keeping the data of a VPC user isolated from other public cloud users.

FEATURES OF A VIRTUAL PRIVATE CLOUD

- User data is stored on a public cloud but isolated from other users using virtualization.
- Offers very high data security with encryption chip security, tunneling, and private IP addressing.
- User gets dedicated cloud server and virtual networks in a VPC.
- It's fully managed by the cloud provider.

Introduction to AWS IAM

Agenda

- What is IAM?
- IAM as a service
- IAM Best Practices
- Elements of IAM policy
- UseCase
- Policy Simulator

What is IAM?

Identity and access management (IAM) is the security discipline that enables the right individuals to access the right resources at the right times for the right reasons.

IAM enables you to securely control access to your application or product services and resources for your users.

Using IAM, you can create and manage users and groups and use permissions to allow and deny their access to the resources.

AWS: IAM as a service

AWS Identity and Access Management

AWS IAM roles are a web service that gives you secured "Control Access" to AWS services for your users. IAM policies specify which services/actions are allowed or denied.

You attach policies to group, users, and roles, which are then subject to permissions that you define.

With IAM, you can centrally manage users, security credentials such as access keys, and permissions that control which AWS resources users can access.

Policies can be granted either from AWS API programmatically or the AWS management console.

IAM gives you following features :

- Shared access to your AWS account.
- Granular permission.
- Secure access to your AWS resources.
- Identity Information.
- Integrated with many AWS resources.
- Free to use.

Types of Policies

- **Managed Policy** - Standalone policies that you can attach to multiple users, groups, and roles in your AWS account. Managed policies apply only to identities (users, groups, and roles) - not resources.
- **Inline Policy** - Policies that you create and manage, and that are embedded directly into a single user, group, or role. Resource-based policies are another form of inline policy.

IAM Groups

- A collection of IAM users.
- You assign permission to group, all IAM user in the inherit those permission.

IAM Users

- Can have username/password to login to aws console.
- Can have aws credentials for making API calls to interact with aws services.
- New IAM user have no permission to do anything, permission must be explicilty granted.

IAM Roles / Instance profile

- The permission of an IAM role can be granted/assigned to EC2 instance.
- All AWS Sdk has buit-in way to auto discover AWS credentials on AWS EC2.

  -Credential file

  -Environmental variable

  -Instance profile

IAM Policies

- When you create a IAM group,user you associate an IAM policy with it which specify the permission that you want to grant.
- IAM policies are JSON formatted document that defines AWS permission.

Elements of policy

- Version - Version specifies the current version of the policy language. It must specify it before the statement element. In this case, our version is "2012-10-17."
- Statement - The Statement element is the main element of the policy. This element is required. The Statement element contains an array of individual statements. Each individual statement is a JSON block enclosed in braces { }.
- Effect - The Effect element is required and specifies whether the statement will result in an allow or an explicit deny.
- Action - The Action element describes the specific action or actions that will be allowed or denied. Each AWS service has its own set of actions that describe tasks that you can perform with that service.
- Resource - The Resource element specifies the object or objects that the statement covers. Statements must include either a Resource or a NotResource element. You specify a resource using an ARN.
- Principal – The Principal element specifies the identity. It is use to specify the User,AWS account that is allowed or denied acces to a resource.

Sample JSON

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Action": "s3:ListBucket",
        "Resource": "arn:aws:s3:::example_bucket"
    }
}
```

IAM Best Practices

- Protect the “root” account.
- Create the individual IAM user.
- Create and use groups.
- Set up a strong password policy.
- Use multifator authentication.
- Use Roles/Instance profiles.
- Rotate credentials often.
- Monitor IAM activity.

Protect the “root” account.

- Root account in AWS has full access to any service.
- By design it permission cannot be restricted.
- Never create AWS credentials keys.

Create the individual IAM user.

- Each user get his/her account.
- Makes managing of users easy.
- Makes defining policies of each user easy.

Create and use groups

- Group allows you to logically define set of users.
- Groups can define different set of policies.
- Users can be part of multiple groups. They will inherit permission for both the groups.

Set up a strong password policy.

- Users should have permission to manage their own passwords.
- You can define a strong password policy that enforces things like minimum length, complexity, periodic rotation etc.

Use multifator authentication

- In addition to using a username and password, IAM has an option setting up a second factor.

Use Roles/Instance profiles.

- If you have an app/script that needs to make an API call to AWS, as far as possible avoid using static access keys.
- Instead use Roles/Instance Profiles.
- AWS automatically expires the credentials.

Rotate credentials often

- Enforce all the keys and passwords to be changed often.
- Passwords should be changed once in 90 days.
- Keys could be rotated much more often.
- All keys should have the permission to create another set of keys and delete the old ones.

Monitor IAM activity

- AWS gives you logs for ALL IAM operations.
- Typically this will be in Cloud Trail. Logs are sent to S3 bucket you define.
- Use this information to keep a close watch on what's happening.
- Setup alerts on interesting activities.

THANK YOU

Amazon EC2

Amazon Elastic Compute Cloud (EC2)

- Amazon Machine Images (**AMIs**) are the basic building blocks of Amazon EC2
- An AMI is a template that contains a software configuration (operating system, application server and applications) that can run on Amazon's computing environment
- AMIs can be used to launch an ***instance***, which is a copy of the AMI running as a virtual server in the cloud.

Getting Started with Amazon EC2

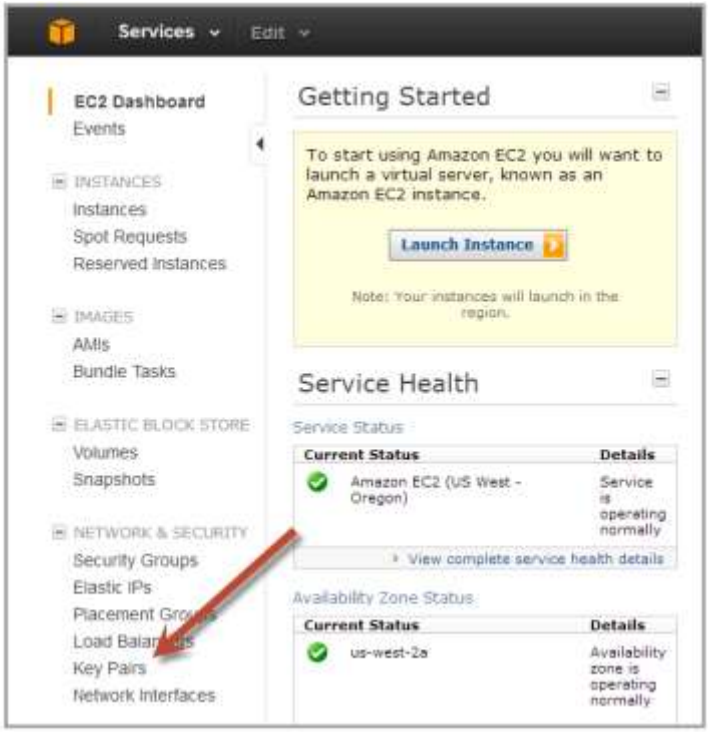
- Step 1: Sign up for Amazon EC2
- Step 2: Create a key pair
- Step 3: Launch an Amazon EC2 instance
- Step 4: Connect to the instance
- Step 5: Customize the instance
- Step 6: Terminate instance and delete the volume created

Creating a key pair

- AWS uses public-key cryptography to encrypt and decrypt login information.
- AWS only stores the public key, and the user stores the private key.
- There are two options for creating a key pair:
  - Have Amazon EC2 generate it for you
  - Generate it yourself using a third-party tool such as OpenSSH, then import the public key to Amazon EC2

Generating a key pair with Amazon EC2

1. Open the Amazon EC2 console at http://console.aws.amazon.com/ec2/
2. On the navigation bar select region for the key pair
3. Click **Key Pairs** in the navigation pane to display the list of key pairs associated with the account

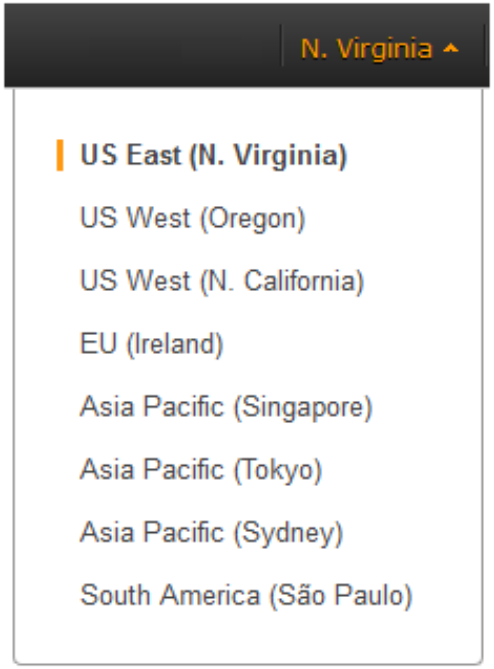


Generating a key pair with EC2 (cont.)

4. Click **Create Key Pair**
5. Enter a name for the key pair in the **Key Pair Name** field of the dialog box and click **Create**
6. The private key file, with .pem extension, will automatically be downloaded by the browser.

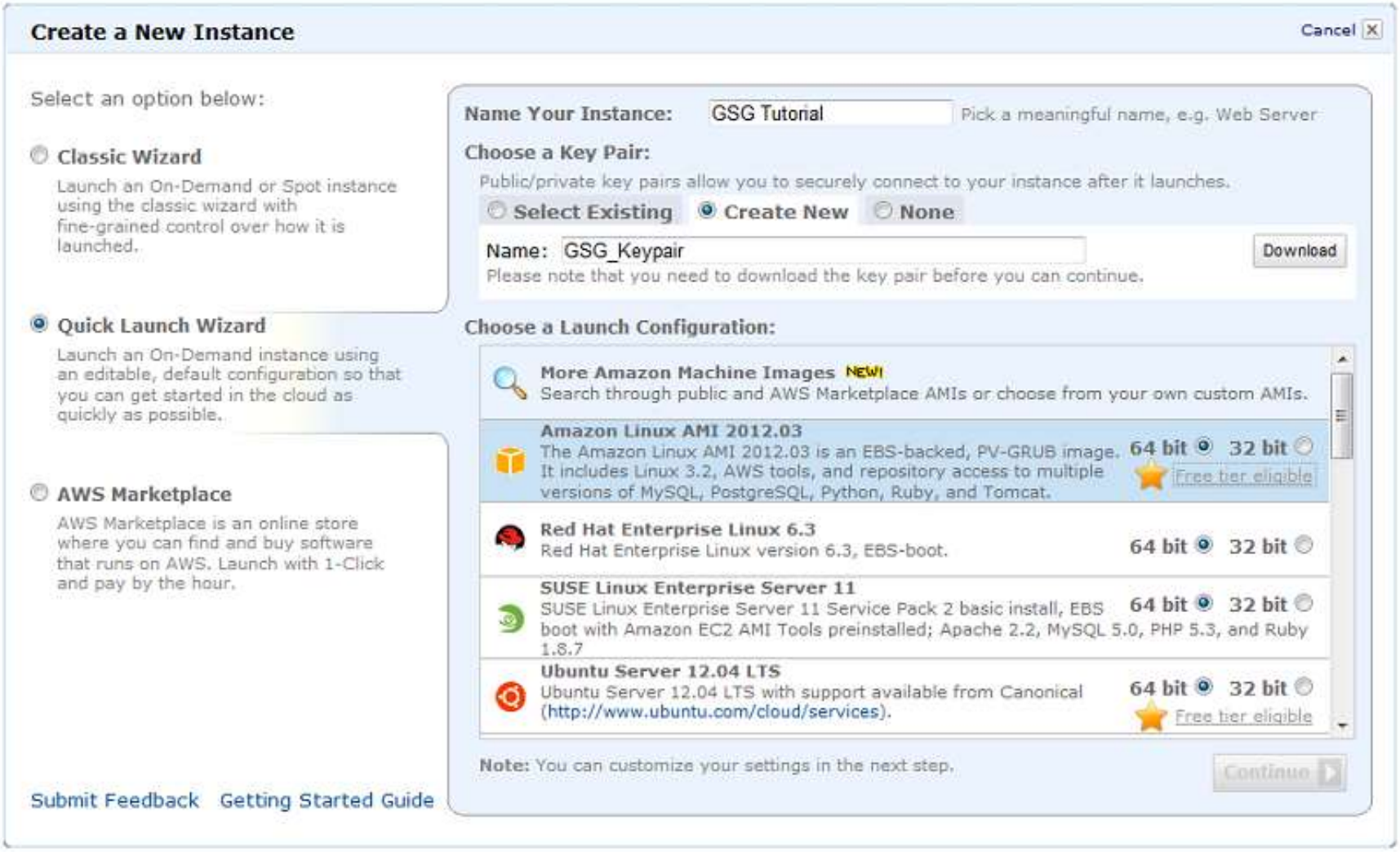
Launching an Amazon EC2 instance

1. Sign in to AWS Management Console and open the Amazon EC2 console at http://console.aws.amazon.com/ec2/
2. From the navigation bar select the region for the instance



Launching an Amazon EC2 instance (cont.)

3. From the Amazon EC2 console dashboard, click **Launch Instance**

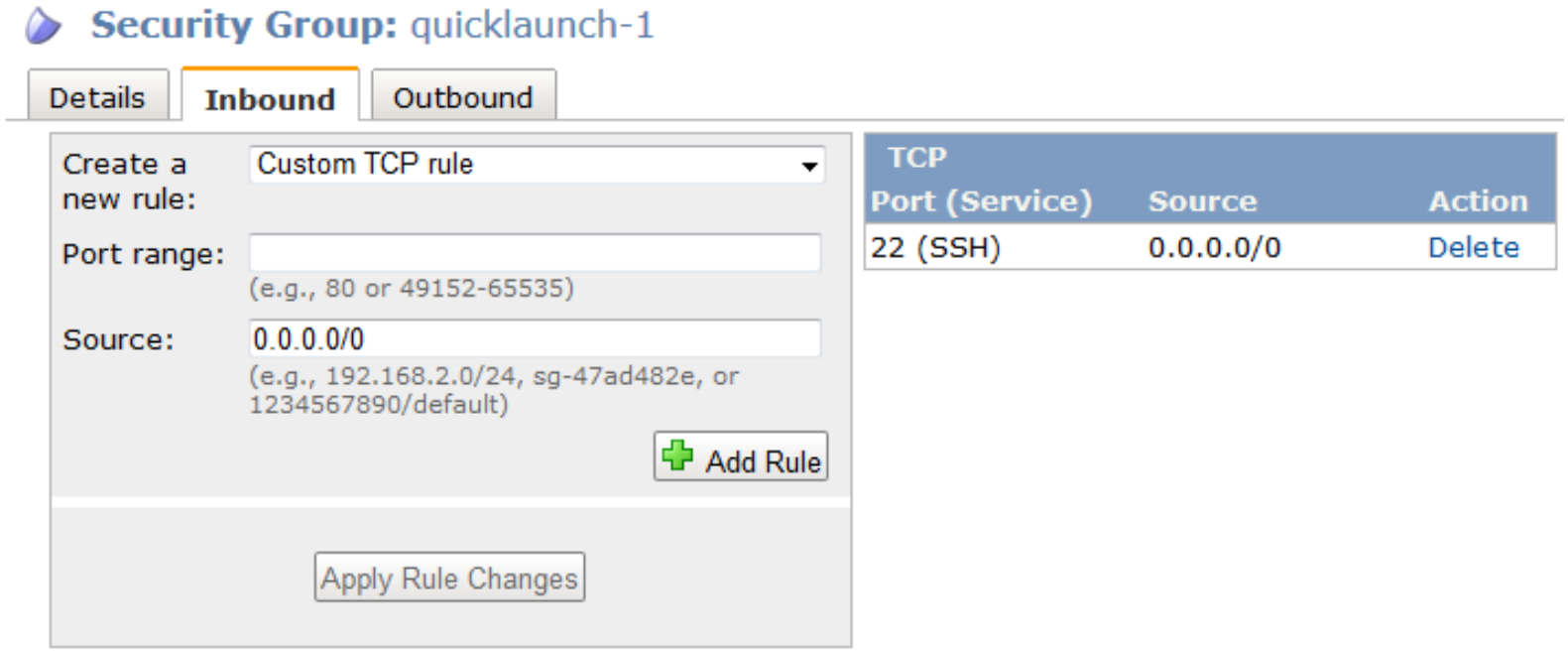


Launching an Amazon EC2 instance (cont.)

4. On the **Create a New Instance** page, click **Quick Launch Wizard**
5. In **Name Your Instance**, enter a name for the instance
6. In **Choose a Key Pair**, choose an existing key pair, or create a new one
7. In Choose a Launch Configuration, a list of basic machine configurations are displayed, from which an instance can be launched
8. Click continue to view and customize the settings for the instance

Launching an Amazon EC2 instance (cont.)

9. Select a security group for the instance. A **Security Group** defines the firewall rules specifying the incoming network traffic delivered to the instance. Security groups can be defined on the Amazon EC2 console, in **Security Groups** under **Network and Security**



Launching an Amazon EC2 instance (cont.)

10. Review settings and click **Launch** to launch the instance
11. Close the confirmation page to return to EC2 console
12. Click **Instances** in the navigation pane to view the status of the instance. The status is **pending** while the instance is launching

	Name	Instance	AMI ID	Root Device	Type	State	Public DNS
	GSG Tutorial	i-e1ab569a	ami-aecd60c7	ebs	t1.micro	pending	

After the instance is launched, its status changes to **running**

	Name	Instance	AMI ID	Root Device	Type	State	Public DNS
	GSG Tutorial	i-e1ab569a	ami-aecd60c7	ebs	t1.micro	running	ec2-50-19-54-72.compute-1.amazonaws.com

Connecting to an Amazon EC2 instance

- There are several ways to connect to an EC2 instance once it's launched.

- **Remote Desktop Connection** is the standard way to connect to Windows instances.

- An **SSH client** (standalone or web-based) is used to connect to Linux instances.

Connecting to Linux/UNIX Instances from Linux/UNIX with SSH

Prerequisites:

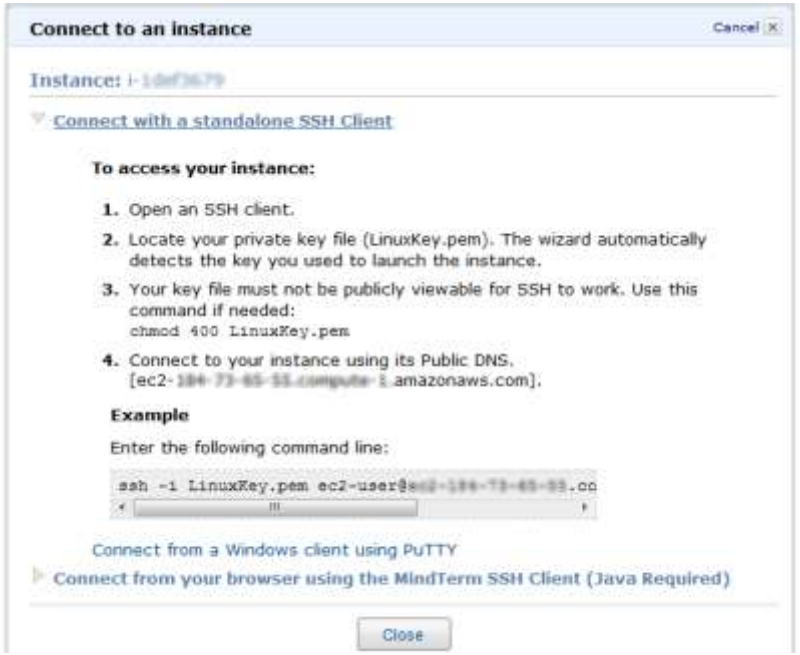
- Most Linux/UNIX computers include an SSH client by default, if not it can be downloaded from openssh.org
- Enable SSH traffic on the instance (using security groups)
- Get the path the private key used when launching the instance

1. In a command line shell, change directory to the path of the private key file
2. Use the **chmod** command to make sure the private key file isn't publicly viewable

```
chmod 400 My_Keypair.pem
```

Connecting to Linux/UNIX Instances(cont.)

3. Right click on the instance to connect to on the AWS console, and click **Connect**.
4. Click **Connect using a standalone SSH client.**
5. Enter the example command provided in the Amazon EC2 console at the command line shell




```
ssh -i <your key a name>.pem ec2-user@ec2-184-72-204-112.compute-1.amazonaws.com
```

Transfering files to Linux/UNIX instances from Linux/UNIX with SCP

Prerequisites:

- Enable SSH traffic on the instance
- Install an SCP client (included by default mostly)
- Get the ID of the Amazon EC2 instance, public DNS of the instance, and the path to the private key

If the key file is My_Keypair.pem, the file to transfer is samplefile.txt, and the instance's DNS name is ec2-184-72-204-112.compute-1.amazonaws.com, the command below copies the file to the ec2-user home

```
scp -i My_Keypair.pem samplefile.txt ec2-user@ec2-184-72-204-112.compute-1.amazonaws.com:~
```

Terminating Instances

- If the instance launched is not in the free usage tier, as soon as the instance starts to boot, the user is billed for each hour the instance keeps running.
- A terminated instance cannot be restarted.
- To terminate an instance:
  1. Open the Amazon EC2 console
  2. In the navigation pane, click **Instances**
  3. Right-click the instance, then click **Terminate**
  4. Click **Yes, Terminate** when prompted for confirmation

Google App Engine Quick Start

adapted from
https://developers.google.com/appengine/docs/whatisgoogleappengine

Google App Engine (GAE)

- GAE lets users run web applications on Google's infrastructure
- GAE data storage options are:
  - Datastore: a NoSQL schemaless object datastore
  - Google Cloud SQL: Relational SQL database service
  - Google Cloud Storage: Storage service for objects and files
- All applications on GAE can use up to 1 GB of storage and enough CPU and bandwidth to support an efficient application serving around 5 million page views a month for free.
- Three runtime environments are supported: **Java**, **Python** and **Go**.

Developing Java Applications on GAE

- The easiest way to develop Java applications for GAE is to use the Eclipse development environment with the Google plugin for Eclipse.
- App Engine Java applications use the Java Servlet standard for interacting with the web server environment.
- An application's files, including compiled classes, JARs, static files and configuration files, are arranged in a directory structure using the WAR standard layout for Java web applications.

Running a Java Project

- The App Engine SDK includes a web server application to test applications. The server simulates the complete GAE environment.
- The project can be run using the "**Debug As > Web Application**" option of Eclipse or using Ant.
- After running the server, the application can be tested by visiting the server's URL in a Web browser.

Uploading an Application to GAE

- Applications are created and managed using the Administration Console at https://appengine.google.com.
- Once an application ID is registered for an application, the application can be uploaded to GAE using the Eclipse plugin or a command-line tool in the SDK.
- After uploading, the application can be accessed from a Web browser. If a free appspot.com account was used for registration, the URL for the application will be http://app_id.appspot.com/, where app_id the application id assigned during registration.