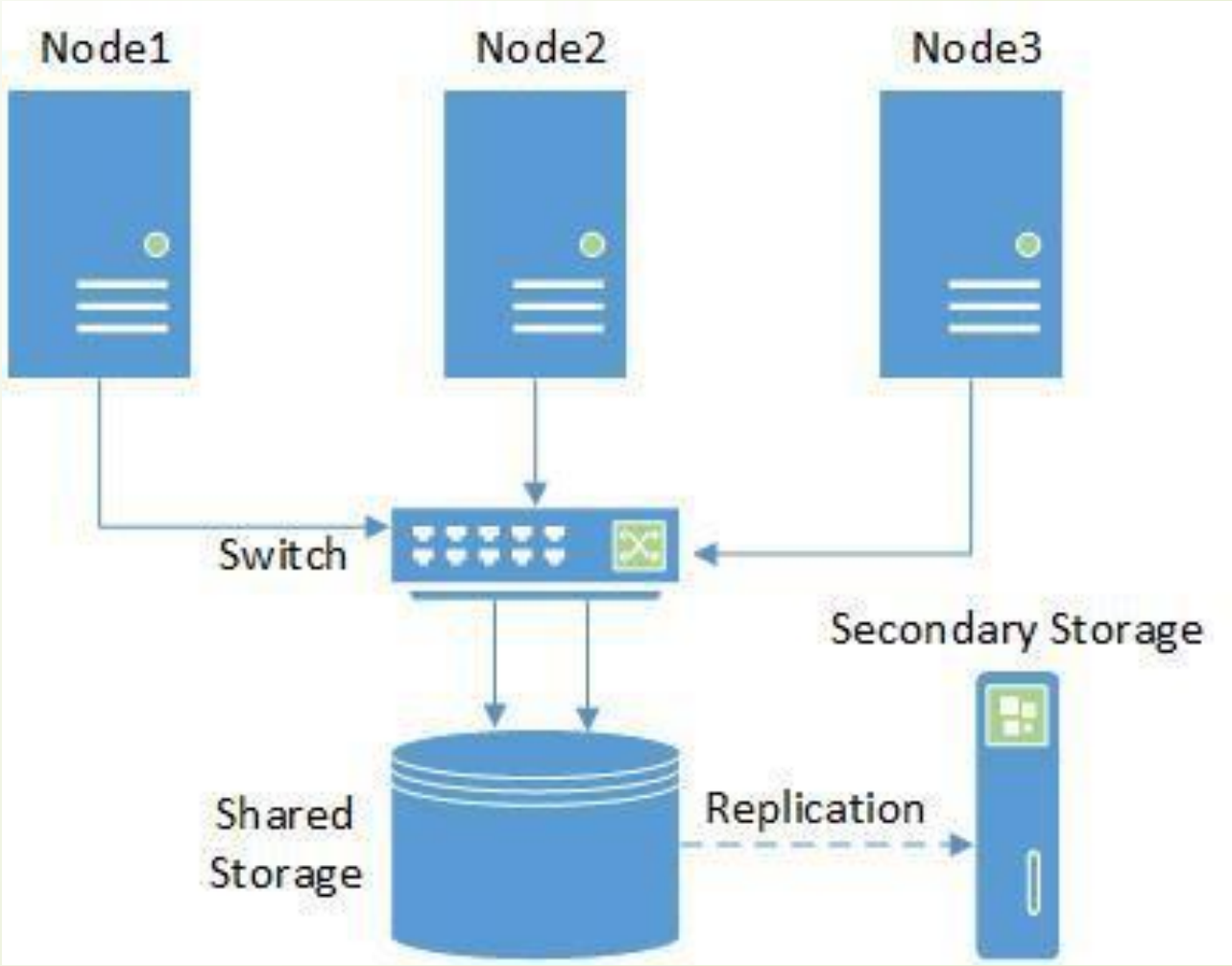
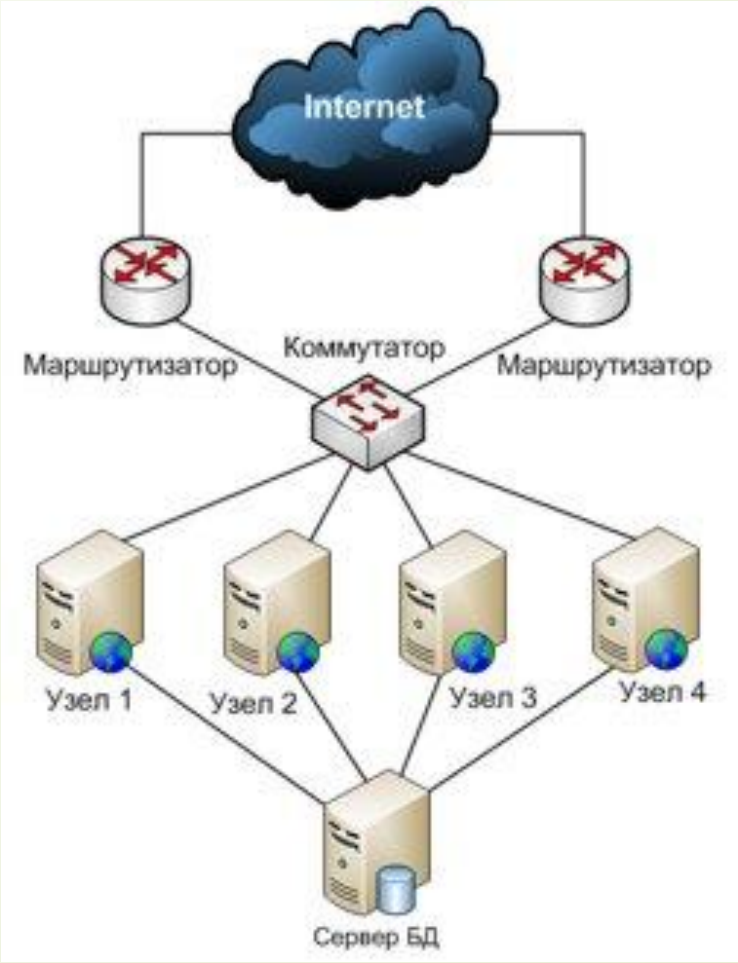
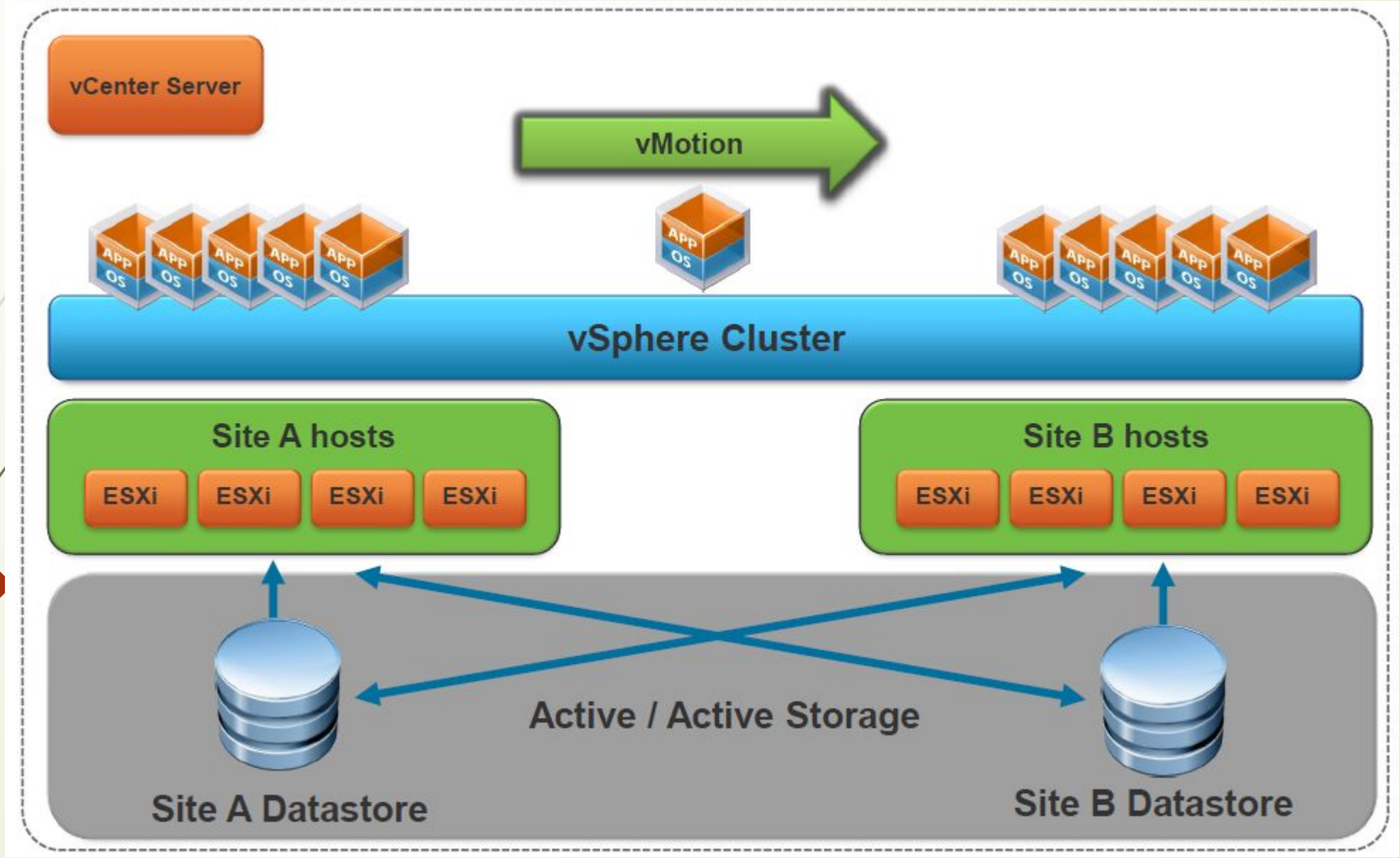
Devops Lesson 10


Node1
Node2
Node3
Switch
Secondary Storage
Shared
Storage
Replication



https://www.youtube.com/watch?v=Kt2VR5a9AF4


vCenter Server
vMotion
APP
OS
vSphere Cluster
Site A hosts
ESXi
ESXi
ESXi
ESXi
Site B hosts
ESXi
ESXi
ESXi
ESXi
Active / Active Storage
Site A Datastore
Site B Datastore

Infrastructure as a Service (IaaS):

Contains the basic building blocks for cloud IT:
Eg. VPC, EC2, EBS

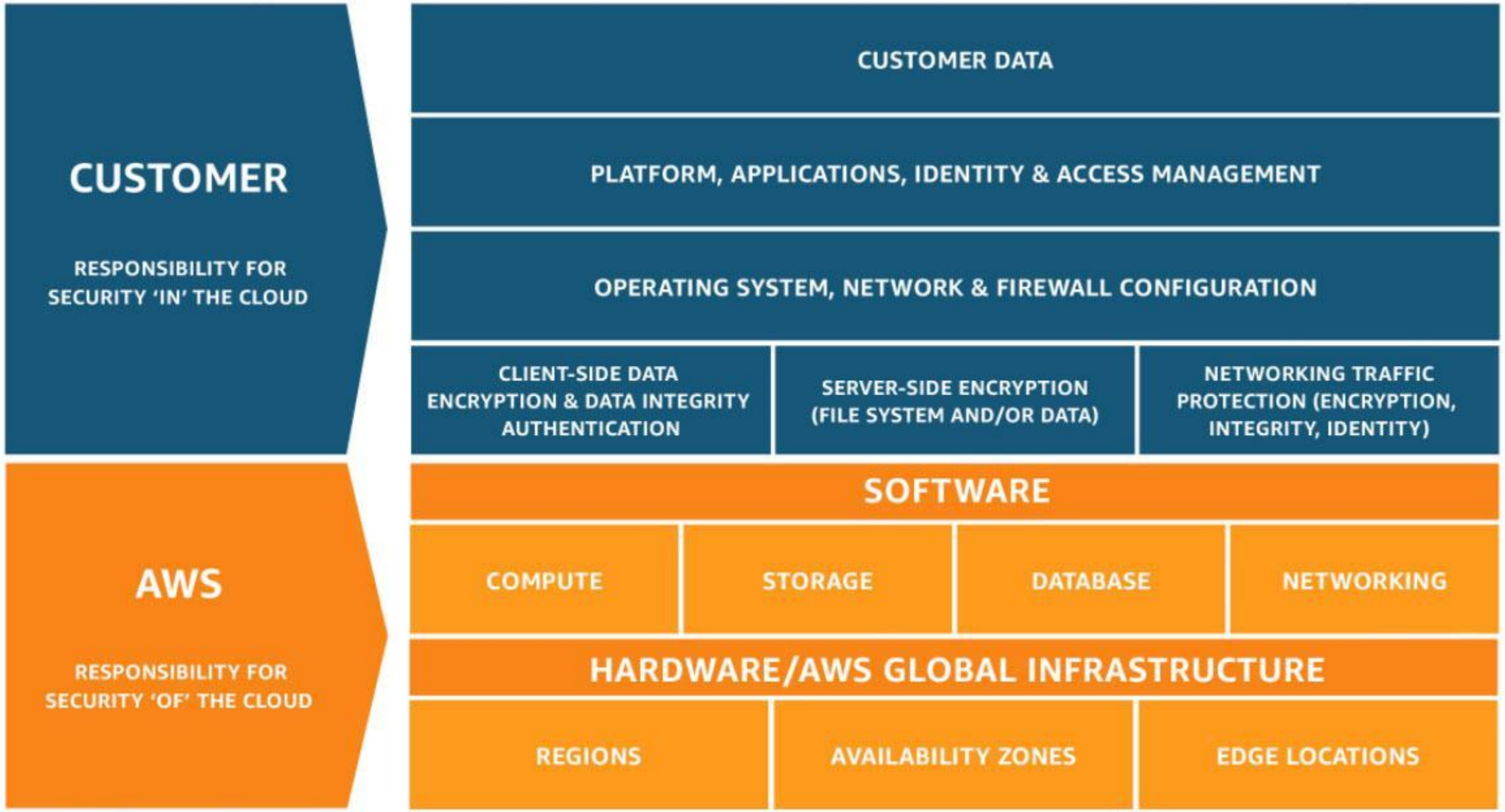
Platform as a Service

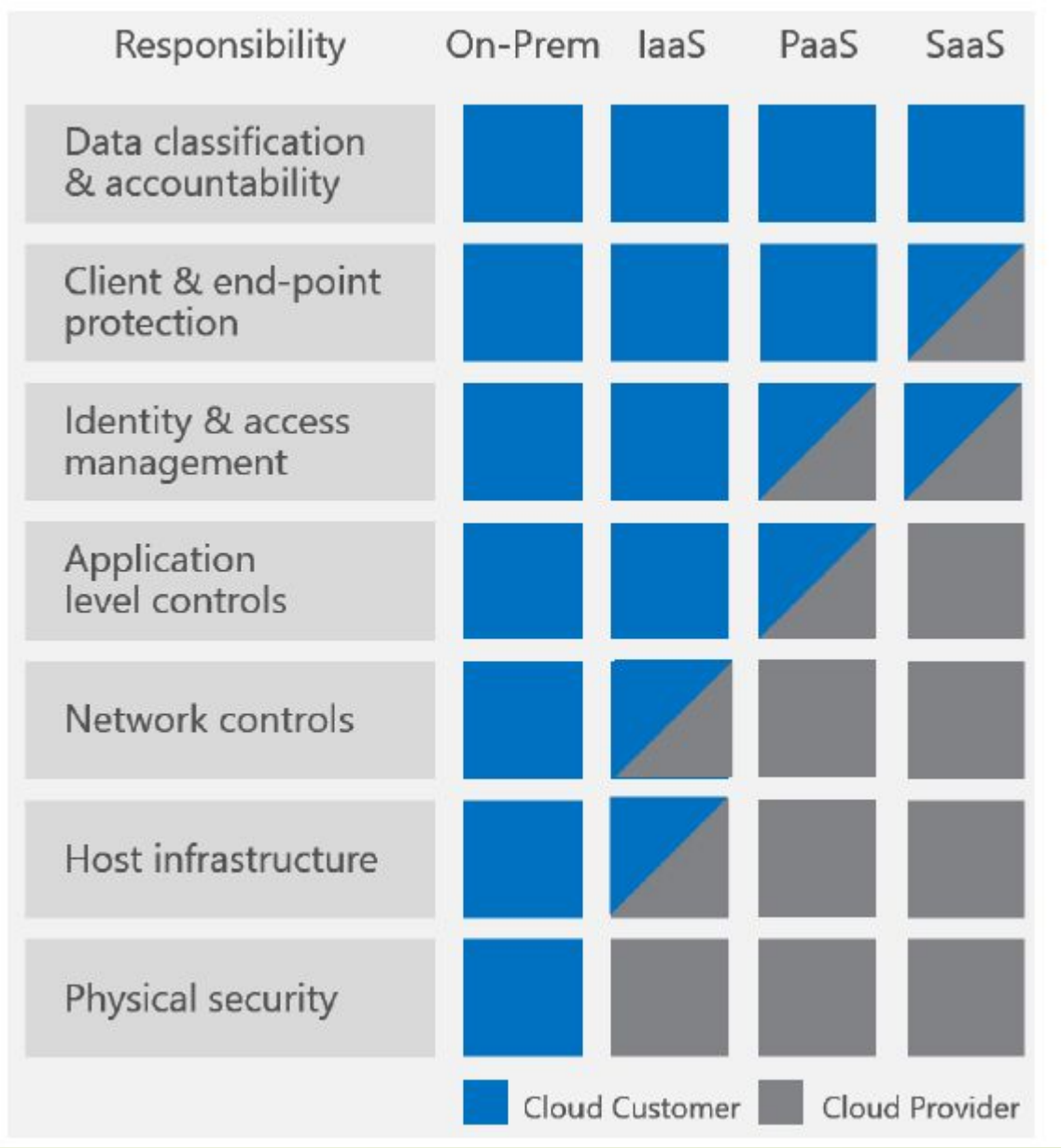
AWS manages the underlying infras
operating systems)
Eg. RDS, EMR, ElasticSearch

Software as a Service (SaaS):

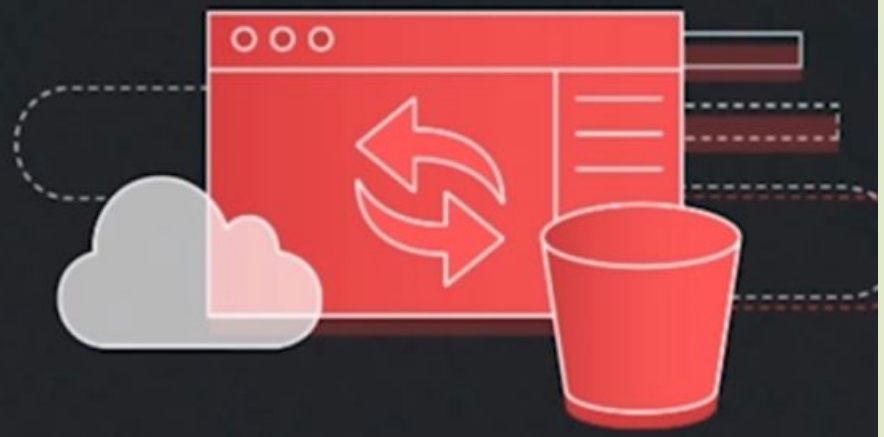
Completed product that is run and managed by the service provider. Mostly refers

Eg. Web-based email, Office 365, Salesforce.com


CUSTOMER
RESPONSIBILITY FOR SECURITY 'IN' THE CLOUD
CUSTOMER DATA
PLATFORM, APPLICATIONS, IDENTITY & ACCESS MANAGEMENT
OPERATING SYSTEM, NETWORK & FIREWALL CONFIGURATION
CLIENT-SIDE DATA ENCRYPTION & DATA INTEGRITY AUTHENTICATION
SERVER-SIDE ENCRYPTION (FILE SYSTEM AND/OR DATA)
NETWORKING TRAFFIC PROTECTION (ENCRYPTION, INTEGRITY, IDENTITY)
AWS
RESPONSIBILITY FOR SECURITY 'OF' THE CLOUD
SOFTWARE
COMPUTE
STORAGE
DATABASE
NETWORKING
HARDWARE/AWS GLOBAL INFRASTRUCTURE
REGIONS
AVAILABILITY ZONES
EDGE LOCATIONS


Responsibility
On-Prem
IaaS
PaaS
SaaS
Data classification & accountability
Client & end-point protection
Identity & access management
Application level controls
Network controls
Host infrastructure
Physical security
Cloud Customer
Cloud Provider

AWS Storage Services



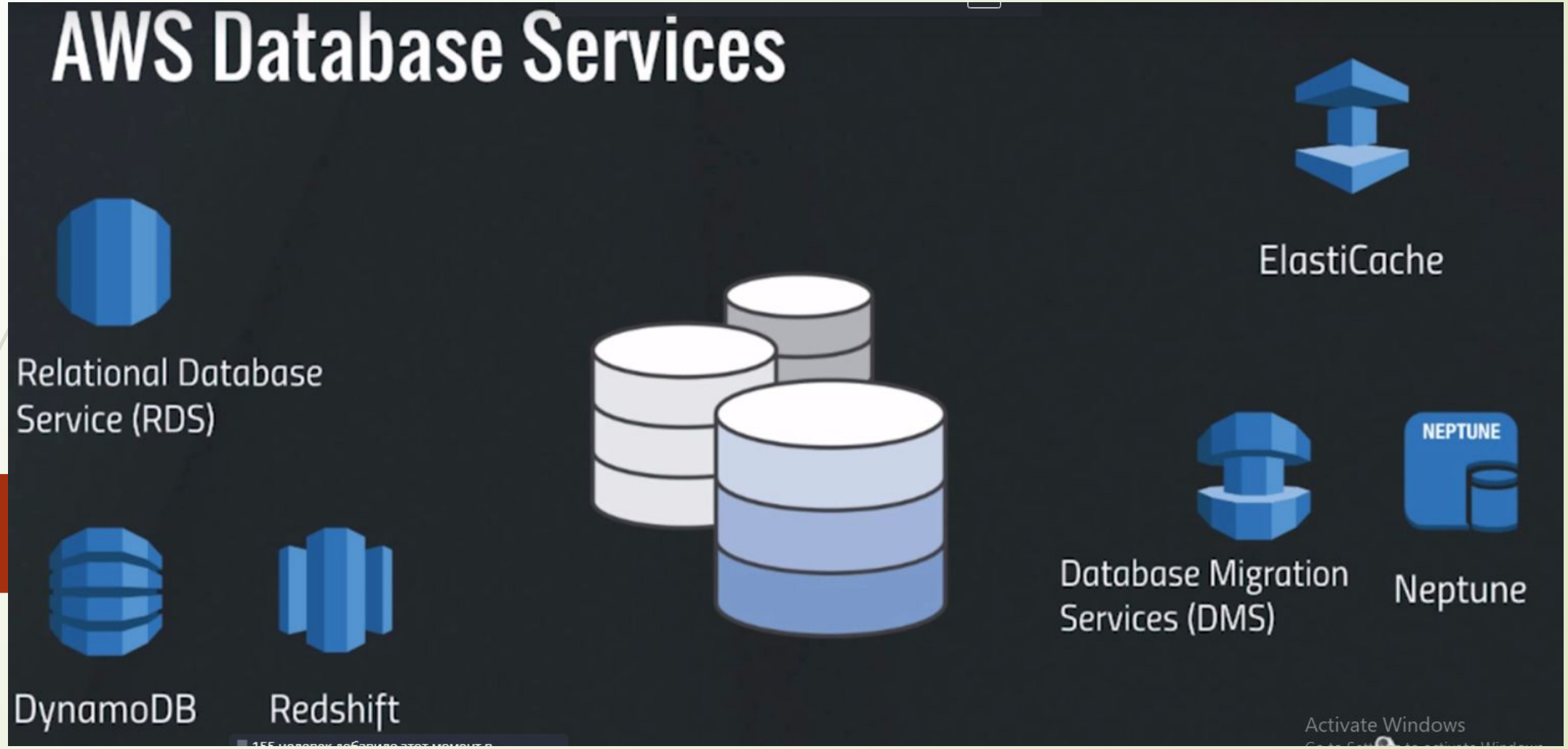
Simple Storage Service (S3)

Glacier

Elastic Block Store (EBS)

Elastic File System (EFS)

Storage Gateway


AWS Database Services
ElastiCache
Relational Database Service (RDS)
NEPTUNE
Database Migration Services (DMS)
Neptune
DynamoDB
Redshift
Activate Windows

AWS Compute Services

Elastic Compute Cloud (EC2)

EC2 Autoscaling

Elastic Container Service (ECS)

Amazon Lightsail

AWS Lambda

Networking & Content Delivery

CloudFront

Virtual Private Cloud (VPC)

Direct Connect

Elastic Load Balancing (ELB)

Route 53

API Gateway

AWS OpsWorks – это сервис управления конфигурациями, который предоставляет управляемые инстансы Chef и Puppet. Chef и Puppet – это платформы автоматизации, позволяющие использовать программный код для автоматического конфигурирования серверов.

Application Integration

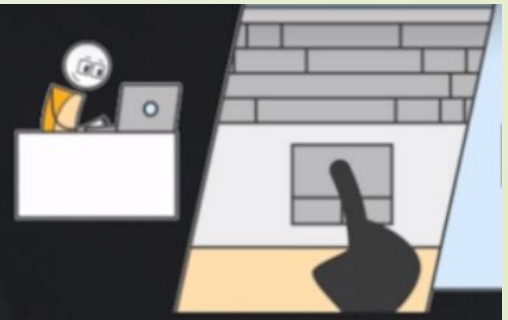
Simple Notification Service (SNS)

Step Functions

Simple Queue Service (SQS)

Simple WorkFlow Service (SWF)

Analytics



Amazon EMR

Athena

Amazon
Elasticsearch Service

Kinesis

QuickSight

Security, Identity & Compliance

AICPA SOC
ISO 27001
PCI Security Standards Council
C5
irap
FedRAMP

AWS Certified Cloud Practitioner

Подтвердите знания и опыт работы в облаке признанными в отрасли документами:

РЕГИСТРИРУЕМСЯ В АМАЗОНЕ

https://aws.amazon.com/ru/ и просто нажать кнопку регестрации по центре екрана, там пройти регистрацию, надо будет ввести кучу данных там город штут и так далее

Центр ресурсов для начала работы

10-минутные учебные пособия	Проекты	Курсы для самостоятельного обучения	Видео	Инструмент SD

Примеры использования

Веб-сайты и интернет-приложения

3 учебных пособия, 3 курса для самостоятельного изучения, 3 видео, 6 проектов Подробнее »

DevOps

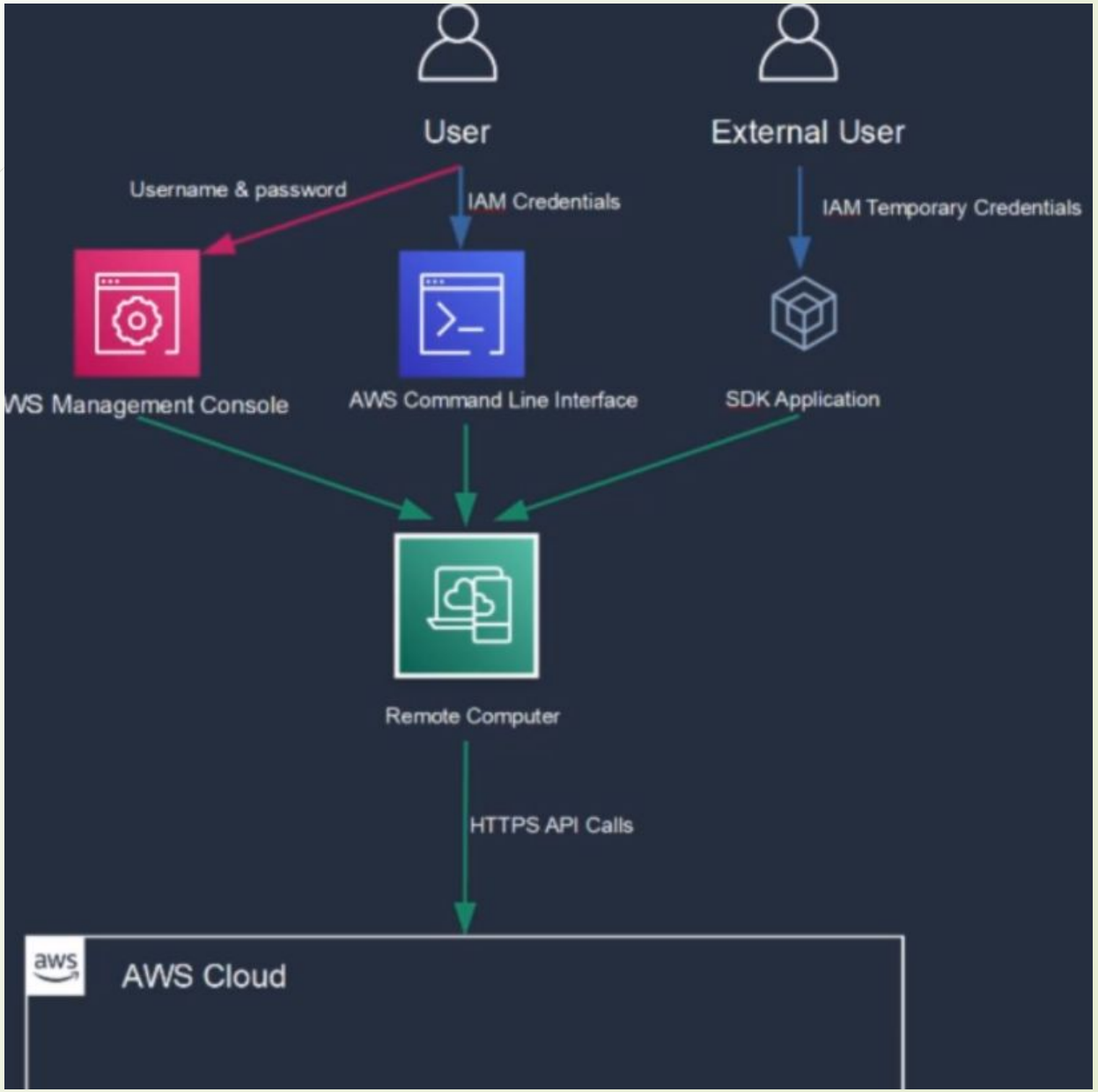
2 учебных пособия, 3 курса для самостоятельного изучения, 6 видео, 3 проекта Подробнее »

Резервное копирование и восстановление

2 учебных пособия, 3 курса для самостоятельного изучения, 3 видео, 3 проекта Подробнее »

Большие данные и аналитика

2 учебных пособия, 3 курса для самостоятельного изучения, 3 видео, 3 проекта Подробнее »


User
External User
Username & password
IAM Credentials
IAM Temporary Credentials
WS Management Console
AWS Command Line Interface
SDK Application
Remote Computer
HTTPS API Calls
aws
AWS Cloud

Pre-requisites

1- AWS CLI requires either Python 2.6.5+ or Python 3.3+ to be installed on the system. We can install Python with the following command,

$ sudo apt-get install python3 **(Ubuntu/Debian)**

$ sudo yum install python **(CentOS/RHEL)**

$ dnf install python **(Fedora)**

$ curl -O https://bootstrap.pypa.io/get-pip.py

& then execute,

$ python get-pip.py

$ sudo apt-get install awscli **or**

$ sudo yum install awscli

But these are not the updated versions. For latest aws cli installation, run the following PIP command from the terminal,

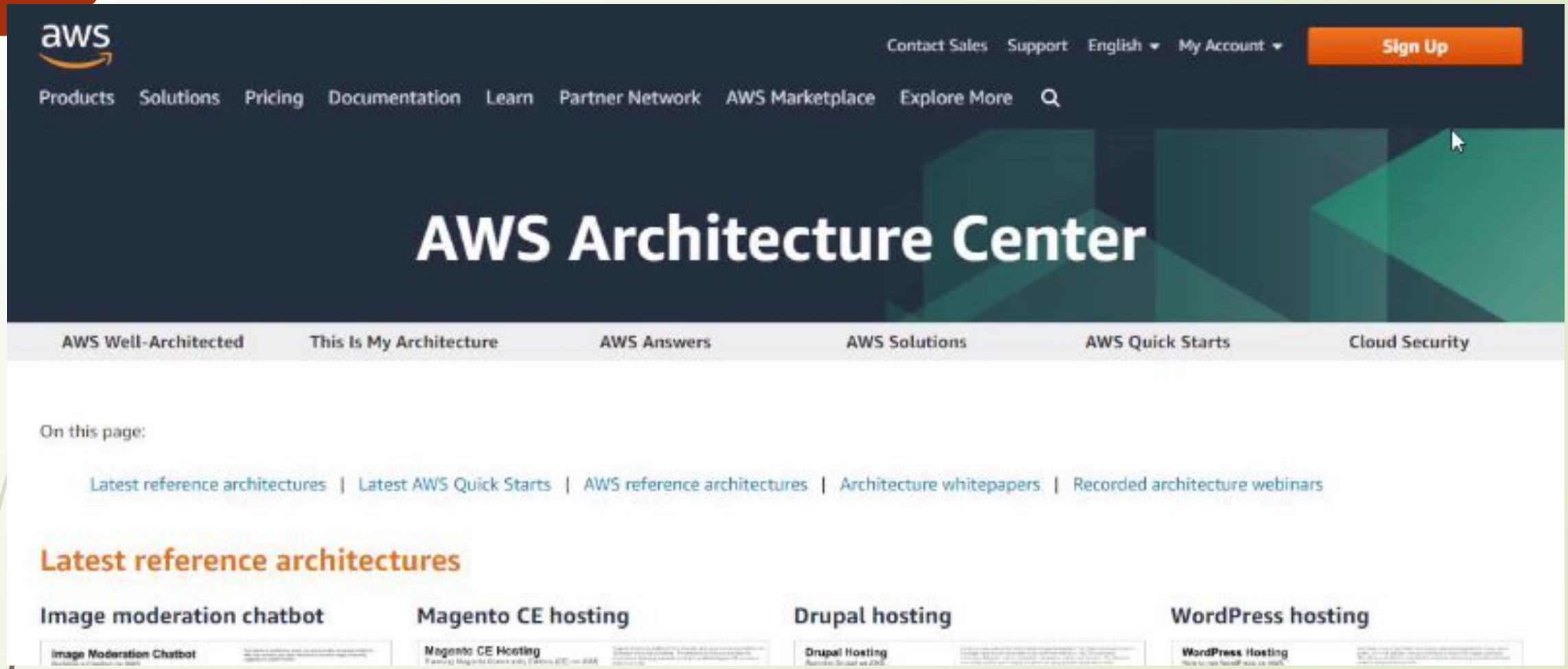
$ pip install awscli

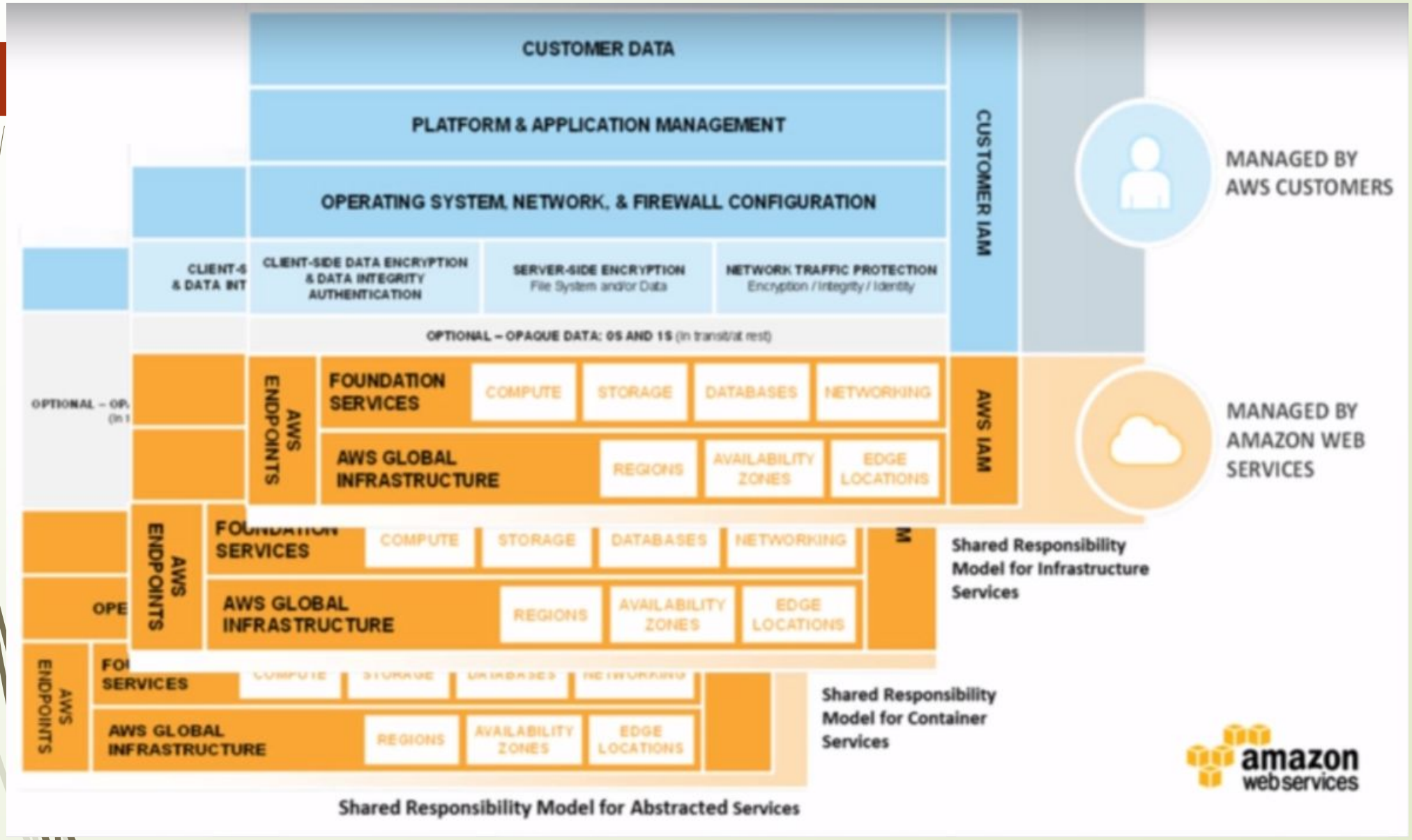
To upgrade the aws cli to the latest version,

$ pip install awscli –upgrade

$ aws configure

You will now be asked to enter the ‘AWS Access Key ID’, then ‘AWS Secret Access Key’ & lastly ‘Default Region Name’. All this information can be obtained from AWS Dashboard. Once all the information has been entered, we will be able to provide resources directly from our terminal, rather than from the AWS Dashboard.


aws
Contact Sales
Support
English
My Account
Sign Up
Products
Solutions
Pricing
Documentation
Learn
Partner Network
AWS Marketplace
Explore More
AWS Architecture Center
AWS Well-Architected
This Is My Architecture
AWS Answers
AWS Solutions
AWS Quick Starts
Cloud Security
On this page:
Latest reference architectures | Latest AWS Quick Starts | AWS reference architectures | Architecture whitepapers | Recorded architecture webinars
Latest reference architectures
Image moderation chatbot
Magento CE hosting
Drupal hosting
WordPress hosting


CUSTOMER DATA
PLATFORM & APPLICATION MANAGEMENT
OPERATING SYSTEM, NETWORK, & FIREWALL CONFIGURATION
CLIENT-SIDE DATA ENCRYPTION & DATA INTEGRITY AUTHENTICATION
SERVER-SIDE ENCRYPTION File System and/or Data
NETWORK TRAFFIC PROTECTION Encryption / Integrity / Identity
OPTIONAL – OPAQUE DATA: 0S AND 1S (In transit/at rest)
CUSTOMER IAM
MANAGED BY AWS CUSTOMERS
AWS ENDPOINTS
FOUNDATION SERVICES
COMPUTE
STORAGE
DATABASES
NETWORKING
AWS GLOBAL INFRASTRUCTURE
REGIONS
AVAILABILITY ZONES
EDGE LOCATIONS
AWS IAM
MANAGED BY AMAZON WEB SERVICES
Shared Responsibility Model for Infrastructure Services
Shared Responsibility Model for Container Services
Shared Responsibility Model for Abstracted Services
amazon web services

What is IAM?

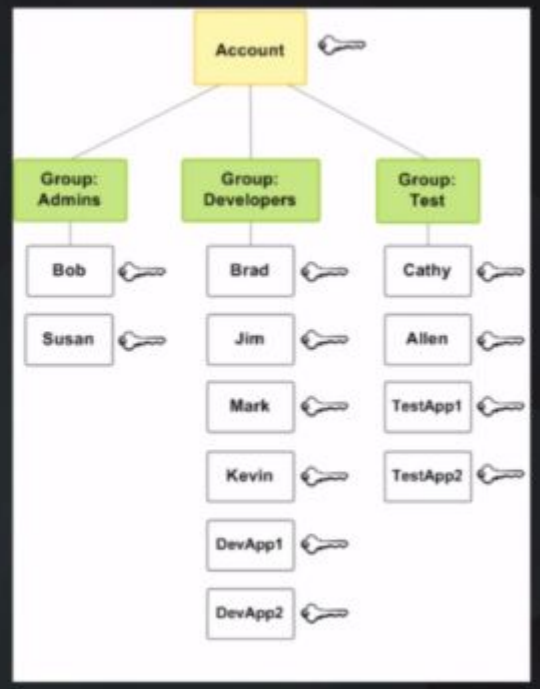
- A web service that allows you to securely control individual and group access to your AWS resources.
- Create and manage user identities ("IAM users") and grant permissions.
- Features:
  - Shared access to your AWS account
  - Granular permissions
  - Secure access to AWS resources for applications that run on Amazon EC2
  - Identity federation to grant permissions for users outside of AWS
  - Payment Card Industry (PCI) Data Security Standard (DSS) Compliance
  - Access log auditing using CloudTrail
  - Eventually Consistent
  - Free to use

- Represent the person or service accessing your account
- Consists of a name and credentials
- Users are identified by:
  - A "friendly name" eg "Bill"
  - Amazon Resource Name (ARN)
    - arn:aws:iam::account-ID-without-hyphens:user/Bill
  - Unique identifier which is returned only when you use the API, SDKs, Tools for Windows PowerShell, or AWS CLI to create the user.
- Credentials can be associated to a user:
  - Console password. User will have a url link to login to the console.
  - Access keys (access key ID and a secret access key), max 2.
- Never use root user **to access resources** unless absolutely essentials. Create admin users with required permissions. Always enable multi-factor authentication of the root user.

- You can use a password policy to do these things:
  - Set a minimum password length.
  - Require specific character types.
  - Allow all IAM users to change their own passwords.
  - Password expiration.
  - Prevent users from reusing previous passwords.
  - Force users to contact an account administrator when the password has expired.

Groups

- Collection of IAM users.
- Users assume the permissions of the group.
- Users can belong to multiple groups.
- Groups can only contain users, cannot be nested.



Roles

- Defined permissions that can be assumed by users or resources.
- Allow EC2 instances to access other AWS resources.
- Grant access to your resources to users in another AWS account
- Can be used to allow users to temporarily assume a role with least privilege access to critical resources. Identity federation using:
  - AWS Cognito
  - OAUTH (Facebook, Google etc)
  - Enterprise Single Sign On with LDAP or Active Directory

AWS Organisations

- S3 Whitelisted
- IAM Policy access for S3 & EC2

-> Bob can access S3 but not EC2

- "*" Whitelisted
- IAM Blacklisted
- IAM Policy access for S3

-> Bob can access S3 but not IAM

Amazon Resource Names (ARN)

The access policy language requires you to specify the resource or resources using the following Amazon Resource Name (ARN) format:

arn:aws:iam::account:resource (note region missing)

Examples:

An IAM user in the account: arn:aws:iam::123456789012:user/Bob

An IAM group: arn:aws:iam::123456789012:group/Developers

An IAM role: arn:aws:iam::123456789012:role/S3Access

An instance profile that can be associated with an EC2 instance: arn:aws:iam::123456789012:instance-profile/Webserver

A federated user identified in IAM as "Bob": arn:aws:sts::123456789012:federated-user/Bob

- Lock Away Your AWS Account Root User Access Keys
- Create Individual IAM Users
- Use Groups to Assign Permissions to IAM Users
- Use AWS Defined Policies to Assign Permissions Whenever Possible
- Grant Least Privilege
- Use Access Levels to Review IAM Permissions (List, Read, Write, or Permissions management)
- Configure a Strong Password Policy for Your Users
- Enable Multi-Factor Authentication (MFA) for Privileged Users
- Delegate by Using Roles Instead of by Sharing Credentials
- Use Roles for Applications That Run on Amazon EC2 Instances
- Rotate Credentials Regularly
- Remove Unnecessary Credentials
- Use Policy Conditions for Extra Security (eg MFA login)
- Monitor Activity in Your AWS Account (eg CloudTrail)

EC2 Purchasing Options

- **On-Demand Instances**
  - Pay, by the second with no up-front or terminating costs.
- **Spot Instances**
  - Request unused EC2 instances, which can lower your Amazon EC2 costs significantly. Generally cheapest option although not always.
  - Maximum price that you are willing to pay per hour per instance.
  - AWS can interrupt them when needed (Spot Instance interruption) or when spot price exceeds your max price.
  - If your Spot instance is terminated or stopped by Amazon EC2 in the first instance hour, you will not be charged for that usage. Otherwise charged to the nearest second.

EC2 Purchasing Options

- **Reserved Instances**
  - Purchase, at a significant discount, instances that are always available, for a term from one to three years.
- **Scheduled Instances**
  - Purchase instances that are always available on the specified recurring schedule, for a one-year term.
- **On Demand Capacity Reservations**
  - Reserve capacity for your EC2 instances in a specific Availability Zone for any duration.
- **Dedicated Instances**
  - Pay, by the hour, for instances that run on single-tenant hardware.
- **Dedicated Hosts**
  - Pay for a physical host that is fully dedicated to running your instances.

EC2 Instance Types

- **General Purpose**
  - Small and mid-size databases, data processing tasks that require additional memory, caching fleets, and for running backend servers for SAP, Microsoft SharePoint, cluster computing, and other enterprise applications. (T2, M3, M4)
- **Compute Optimized**
  - High performance front-end fleets, web-servers, batch processing, distributed analytics, high performance science and engineering applications, ad serving, MMO gaming, and video-encoding. (C3, C4)
- **Memory Optimized**
  - High performance databases, distributed memory caches, in-memory analytics, genome assembly and analysis, larger deployments of SAP, Microsoft SharePoint, and other enterprise applications. (X1, R3, R4)
- **GPU / Accelerated Computing**
  - 3D application streaming, machine learning, video encoding, and other server-side graphics or GPU compute workloads. (G3, G2)
- **Storage Optimized**
  - NoSQL databases like Cassandra and MongoDB, scale out transactional databases, data warehousing, Hadoop, and cluster file systems. (I3, I2). Massively Parallel Processing (MPP) data warehousing, MapReduce and Hadoop distributed computing, distributed file systems, network file systems, log or data-processing applications. (D2)

Choice of Linux or Windows

RedHat Linux, Windows Server, SuSE Linux, Ubuntu, Fedora, Debian, Cent OS, Gentoo Linux, Oracle Linux, and FreeBSD

More info

https://aws.amazon.com/ec2/instance-types

Amazon Machine Images (AMI)

Provides the information required to launch an instance:

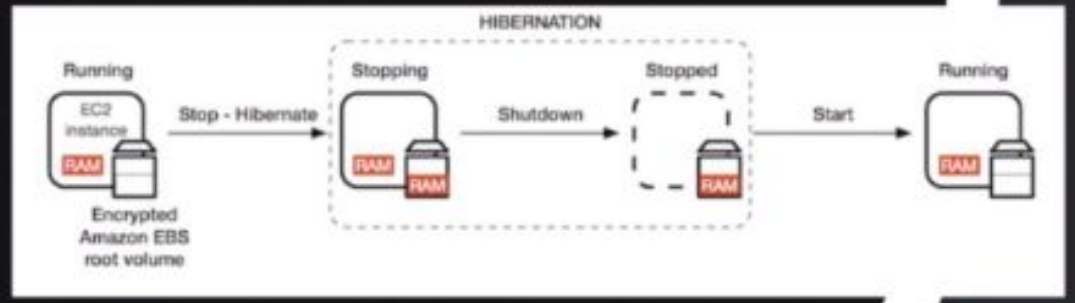
- A template for the **root volume** for the instance (for example, an operating system, an application server, and applications)
- **Launch permissions** that control which AWS accounts can use the AMI to launch instances
- A **block device mapping** that specifies the volumes to attach to the instance when it's launched

awsmarketplace

- Market for Amazon Machine Images (AMI)
- Paid, free, trial and BYO license software

EC2 Instance States

- Start
- Stop (EBS Backed only)
  - Instance is shut down with no instance charges
  - Still charged for EBS volumes
  - Minimum of 1 minute charge on restart
- Stop-Hibernate (EBS Backed only)
  - Suspend-to-disk
  - Saves RAM to EBS
- Reboot
  - Operating system reboot.
- Terminate



EC2 Storage Options

Elastic Block Store (EBS)

- Most common.
- Replicated within AZ
- EBS volumes attached at instance launch are deleted when instance terminated.*
- EBS volumes attached to a running instance are not deleted when instance is terminated but are detached with data intact.*

*Unless delete on terminate flag modified

Instance Store

- Physically attached to the host server
- Data NOT LOST when OS is rebooted.
- Data LOST when:
  - Underlying drive fails
  - Instance is terminated
- Do not rely on for valuable, long-term data.
- You cannot detach and attach to another instance

H work