This document explains **AWS compute options**, focusing on **Amazon EC2** (Elastic Compute Cloud). It covers **different compute models, EC2 components, security considerations, storage, AMIs, and lifecycle management**. Below is a complete breakdown:

1. AWS Compute Options

AWS offers five cloud compute models:

a. Virtual Machines (VMs) – Amazon EC2

- Provides **scalable, resizable virtual servers** in the cloud.
- Allows complete control over the server's **operating system, applications, and network settings**.
- Common use cases: **Web servers, databases, and machine learning workloads**.

b. Containers – Amazon ECS & Amazon EKS

- **Amazon Elastic Container Service (ECS):** Runs **Docker** containers.
- **Amazon Elastic Kubernetes Service (EKS):** Managed **Kubernetes** for container orchestration.
- Used for **microservices, scalable applications, and DevOps**.

c. Virtual Private Servers (VPS) – Amazon Lightsail

- **Simpler alternative to EC2**, designed for **small projects, websites, and applications**.
- Includes **compute, storage, networking, and load balancing** at a fixed monthly price.

d. Platform as a Service (PaaS) – AWS Elastic Beanstalk

- Automates **deployment, scaling, and maintenance** of applications.
- Supports multiple languages: **Java, .NET, PHP, Python, Node.js, Ruby, Go, Docker**.
- **Best for developers** who don't want to manage infrastructure.

e. Serverless – AWS Lambda & AWS Fargate

- **AWS Lambda:** Runs **code without managing servers**. Supports **Java, Python, Node.js, C#, Go**.

- **AWS Fargate:** Serverless container compute, eliminating the need to manage instances.
- Ideal for **event-driven applications, automation, and microservices**.

Comparison of Compute Models

Compute Model	Management Responsibility	Scalability	Best Use Cases
EC2 (VMs)	Full control over OS, storage, and network	Manual or Auto Scaling	General workloads, databases
Containers	Control over containers but not the underlying VM	Highly scalable	Microservices, DevOps
VPS (Lightsail)	Limited control, easy setup	Auto scaling within limits	Small websites, e-commerce
PaaS (Elastic Beanstalk)	Minimal infrastructure management	Auto scaling	Web applications
Serverless (Lambda, Fargate)	No server management	Automatically scales	Event-driven apps, automation

2. Amazon EC2 (Elastic Compute Cloud)

a. Overview of EC2

- EC2 provides **on-demand virtual machines** with full control.
- Supports multiple **operating systems** (Windows, Linux, macOS).
- Enables **horizontal and vertical scaling** of compute resources.
- **Pay-as-you-go pricing**, allowing cost efficiency.

b. Components of an EC2 Instance

1. **Host Computer:** Physical hardware where VMs run.
2. **Hypervisor:** Software layer that manages VM access to physical resources.
3. **VM (Instance):** The virtual machine itself.

4. **Operating System (OS):** Installed OS inside the VM (e.g., Amazon Linux, Ubuntu).
5. **Instance Storage:**
    - **Instance Store (Ephemeral Storage):** Temporary data, lost on reboot.
    - **Elastic Block Store (EBS):** Persistent storage, remains after instance shutdown.
6. **Networking:** Connects EC2 to **other instances, AWS services, and the internet**.
7. **Security Groups:** Firewall rules defining **inbound and outbound traffic**.

3. EC2 Instance Types & Pricing

a. EC2 Instance Families

EC2 offers different **instance types** optimized for different workloads:

Instance Type	Description	Best Use Cases
General Purpose (T, M series)	Balanced CPU, memory, storage	Web servers, small databases
Compute Optimized (C series)	High CPU performance	High-performance computing, gaming
Memory Optimized (R, X series)	Large RAM capacity	Big data, in-memory databases
Storage Optimized (I, D series)	High disk I/O performance	Data warehousing, analytics
Accelerated Computing (P, G series)	GPUs for AI/ML workloads	Machine learning, deep learning

b. EC2 Pricing Models

1. **On-Demand Instances:** Pay per hour/second, **best for short-term workloads**.
2. **Reserved Instances: 1-3 year commitment** with up to **75% cost savings**.
3. **Spot Instances: Up to 90% cheaper**, but can be terminated anytime.
4. **Savings Plans:** Flexible pricing model with **discounts based on usage**.

5. **Dedicated Hosts:** Physical servers reserved for a **single customer**.

4. Security & Access Control in EC2

a. Key Pair Authentication (SSH & RDP Access)

- Uses **public-private key pairs** for **secure remote access**.
- Required for **SSH (Linux) or RDP (Windows) connections**.

b. Security Groups

- Act as a **firewall**, controlling **inbound and outbound traffic**.
- Example: Allow **port 22 for SSH, port 80 for web traffic**.

c. IAM Roles & Policies

- **IAM Roles:** Grant EC2 instances access to AWS services **without storing credentials**.
- **IAM Policies:** Define **fine-grained permissions** for EC2 actions.

d. Additional Security Best Practices

- **Enable Auto Updates** for OS and applications.
- **Use IAM roles instead of root access.**
- **Limit SSH/RDP access to specific IPs.**
- **Monitor with AWS CloudTrail and GuardDuty.**

5. Amazon Machine Images (AMIs) & EC2 Lifecycle

a. What is an AMI?

- An AMI is a **pre-configured template** for launching EC2 instances.
- Includes **OS, applications, libraries, and configurations**.

b. Benefits of AMIs

1. **Repeatability:** Ensures identical instance setups.
2. **Reusability:** Launch multiple instances with the same configuration.

3. **Recoverability:** Use AMIs for disaster recovery.

c. AMI Sources

1. **AWS Quick Start AMIs:** Provided by AWS, includes Windows & Linux distributions.
2. **Custom AMIs:** Create from existing EC2 instances.
3. **AWS Marketplace:** Third-party AMIs for specific applications.
4. **Community AMIs:** Publicly shared AMIs (use with caution).

6. EC2 Instance Lifecycle

a. Lifecycle Stages of an EC2 Instance

1. **Pending:** Instance is launching.
2. **Running:** Instance is active and accessible.
3. **Stopping:** Instance is shutting down.
4. **Stopped:** Instance is off but retains data (for EBS-backed instances).
5. **Terminating:** Instance is being deleted.
6. **Terminated:** Instance is permanently removed.

b. Hibernate vs. Stop vs. Terminate

- **Stop:** Saves EBS data but deallocates resources.
- **Hibernate:** Saves in-memory data and resumes where it left off.
- **Terminate:** Deletes instance and all attached storage.

7. EC2 Image Builder (Automated AMI Creation)

- Introduced in **December 2019** to automate **AMI creation and management**.
- Benefits:
    - **Automated security patching.**
    - **Built-in testing before deployment.**
    - **Version control for images.**