

**AWS EC2 (Elastic Compute Cloud) **

Table of Contents

1. Introduction to EC2
2. Key Features
3. EC2 Instance Lifecycle
4. Launching an EC2 Instance (Steps and Code)
5. EC2 Pricing Options
6. Storage Options
7. Security Features
8. Common Use Cases
9. Common Interview Questions

1. Introduction to EC2

- **Amazon EC2**: A web service that provides scalable compute capacity in the AWS cloud.
- Use EC2 to launch virtual servers (instances) quickly and manage them easily.

2. Key Features

- **Scalability**: Adjust instance size and count based on demand.
- **Pay-as-you-go**: Billed for the compute time used.
- **Variety of Instance Types**: Tailored for different workloads (compute-optimized, memory-optimized, etc.).
- **Integration**: Works with other AWS services like S3, VPC, and RDS.
- **Security**: Supports IAM, security groups, and key pairs.

3. EC2 Instance Lifecycle

- **Launch**: Start a new instance from an Amazon Machine Image (AMI).

- **Stop**: Halts the instance; data remains intact, no compute charges.
- **Terminate**: Deletes the instance and its data.

4. Launching an EC2 Instance (Steps and Code)

Steps to Launch an EC2 Instance

1. **Log into AWS Console**.
2. **Select EC2** from the Services menu.
3. **Click "Launch Instance"**.
4. **Choose an AMI** (e.g., Amazon Linux 2).
5. **Select Instance Type** (e.g., t2.micro).
6. **Configure Instance Details**: Set network, IAM role, etc.
7. **Add Storage**: Define EBS volume.
8. **Add Tags**: Optionally, add metadata.
9. **Configure Security Group**: Set firewall rules.
10. **Review and Launch**: Choose a key pair for SSH access.

Code to Launch an EC2 Instance using AWS CLI

```
```bash
aws ec2 run-instances \
--image-id ami-0123abcd \
--count 1 \
--instance-type t2.micro \
--key-name MyKeyPair \
--security-group-ids sg-0123456789abcdef0 \
--subnet-id subnet-6e7f829e
````
```

5. EC2 Pricing Options

- **On-Demand**: Pay per hour or second with no long-term commitment.
- **Reserved Instances**: Significant discounts for 1 or 3-year commitments.

- **Spot Instances**: Bid for unused capacity at lower costs.
- **Dedicated Hosts**: Physical servers dedicated to your use.

6. Storage Options

- **Elastic Block Store (EBS)**: Persistent storage volumes attached to instances.
- **Instance Store**: Temporary storage; data is lost if the instance stops.
- **Elastic File System (EFS)**: Scalable file storage accessible by multiple instances.

7. Security Features

- **Security Groups**: Define rules to control traffic to instances.
- **Key Pairs**: SSH keys for secure access to instances.
- **IAM Roles**: Manage permissions for accessing AWS resources.
- **VPC (Virtual Private Cloud)**: Isolate instances within a virtual network.

8. Common Use Cases

- **Web Hosting**: Host dynamic websites and web applications.
- **Batch Processing**: Run jobs that can be scaled across multiple instances.
- **Big Data Analytics**: Process large data sets with tools like Hadoop.
- **Development and Testing**: Use isolated environments to test applications.

9. Common Interview Questions

1. **What is EC2?**

- A web service for scalable virtual servers in the cloud.

2. **What are Security Groups?**

- Virtual firewalls to control traffic to and from instances.

3. **Difference between stopping and terminating an instance?**

- **Stopping**: Halts the instance, data persists, no compute charges.
- **Terminating**: Permanently deletes the instance and its data.

4. **What is an AMI?**

- Amazon Machine Image, a pre-configured template for launching instances.

5. **What is a Spot Instance?**

- Instances that use excess capacity at lower costs, but can be interrupted.

10. Code Examples for Common Tasks

Connect to an EC2 Instance Using SSH

```bash

```
ssh -i /path/to/key-pair.pem ec2-user@your-ec2-public-ip
```

```

Stopping an EC2 Instance

```bash

```
aws ec2 stop-instances --instance-ids i-0123456789abcdef0
```

```

Terminating an EC2 Instance

```bash

```
aws ec2 terminate-instances --instance-ids i-0123456789abcdef0
```

```

Adding a Security Group Rule

```bash

```
aws ec2 authorize-security-group-ingress \
--group-id sg-0123456789abcdef0 \
--protocol tcp \
```

```
--port 22 \
--cidr 0.0.0.0/0
```
---
---
```

This guide provides a compact overview of AWS EC2, making it easier to understand its core components and operations, and gives you practical code examples for common tasks. Let me know if you need further information or more detailed examples!