AWS - Ec2 - Volume

- **EC2 volume** refers to a block-level storage device that you can attach to an **EC2 instance** (a virtual server).
- These volumes are typically used to store data, applications, or the OS for the EC2 instance.
- EBS volumes are persistent storage, meaning that data remains intact even if you stop or terminate your EC2 instance.

Types:

- **General Purpose SSD (gp3)**: Good for most workloads, balances price and performance.
- **Provisioned IOPS SSD (io2, io2 Block Express)**: For applications requiring high-performance storage.
- **Throughput Optimized HDD (st1)**: Ideal for large, sequential workloads.
- **Cold HDD (sc1)**: Low-cost storage for infrequently accessed data.

Ec2-EBS-Encryption

- EBS uses KMS keys when creating encrypted volumes and snapshots.
- This volume data is encrypted using the industry-standard AES-256.
- Can attach both encrypted and unencrypted volumes to an instance simultaneously.
- Can encrypt the boot and data volumes of an Ec2 instance

Following Types of data are encrypted

- Data at rest inside the volume
- All data moving between the volume and the instance
- All snapshots created from the volumes
- All volumes created from snapshot

AWS - Ec2 and Key-Pairs

An **EC2 Key Pair** is a set of cryptographic keys that you use to securely connect to your **Amazon EC2 instances**. EC2 key pairs consist of a **private key** and a **public key**. The public key is embedded into your EC2 instance when you launch it, while you download the private key (usually as a .pem file) to your local machine. The private key is then used to authenticate your login to the EC2 instance.

Key Concepts of EC2 Key Pairs

1. **Public and Private Key**:
   - **Public Key**: Stored on the EC2 instance. It's used by AWS to encrypt data that only the corresponding private key can decrypt.
   - **Private Key**: Stored locally on your machine. You use it to decrypt the data sent to your EC2 instance, like for SSH login (Linux) or RDP login (Windows).
2. **Key Pair Generation**:
   - You can generate an EC2 key pair when launching an instance from the AWS Management Console, AWS CLI, or using an AWS SDK.
   - Once generated, the private key is downloaded immediately to your local machine. It's **crucial** to store the private key securely, as AWS **cannot** retrieve it again for you.
3. **Authentication for SSH (Linux) or RDP (Windows)**:
   - **Linux instances**: You use the private key to authenticate with the EC2 instance via SSH.
   - **Windows instances**: You use the private key to decrypt the Administrator password (once the instance is running) and then use RDP to connect.

How EC2 Key Pairs Work

1. **When You Launch an Instance**:
   - You specify a key pair while launching an EC2 instance. This public key is copied to the instance, and you receive a .pem file containing the private key for your local machine.
2. **Connecting to the Instance**:
   - When you try to connect to the EC2 instance, the SSH or RDP client uses the **private key** for authentication.
   - The **public key** on the instance matches the **private key** you use, allowing for secure access.
3. **Key Pair Use Cases**:
   - **Secure login**: The most common use is for SSH access to Linux instances or RDP access to Windows instances.
   - **Encryption**: Some services use key pairs for secure data transmission.

Creating an EC2 Key Pair (AWS Management Console)

Here’s how to create an EC2 key pair using the **AWS Management Console**:

1. **Navigate to EC2 Console**:
   - Go to the **EC2 Dashboard** in the AWS Management Console.
2. **Create a Key Pair**:
   - Under the **Network & Security** section, click on **Key Pairs**.
   - Click **Create Key Pair**.

3. **Configure the Key Pair**:
    - Enter a name for the key pair (e.g., MyKeyPair).
    - Select the **key pair type** (the default is RSA).
    - Choose the **private key format** (PEM for Linux/SSH or PPK for PuTTY on Windows).
4. **Download the Private Key**:
    - Click **Create**. The private key will be downloaded as a .pem file.
    - **Save the private key** securely on your local machine because it will not be available again from AWS.
5. **Launch Your EC2 Instance**:
    - When launching an EC2 instance, select the key pair you just created under the **Key Pair** section.

Connecting to an EC2 Instance Using SSH (Linux Example)

1. **Open Terminal**:
   Open a terminal on your local machine.

- **Set Permissions on the Private Key**:
  Before you can use the .pem file for SSH, ensure it has the correct permissions:

  bash

  CopyEdit

```
chmod 400 /path/to/your-key.pem
```

2.

- **Connect to Your EC2 Instance**:
  Use the ssh command with the private key to connect to your EC2 instance:

  bash

CopyEdit

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

3. Replace `<ec2-public-ip>` with the public IP or DNS name of your EC2 instance.

4. **Verify Connection**:
   If everything is configured correctly, you should be logged into your EC2 instance.

Connecting to a Windows EC2 Instance Using RDP

1. **Get the Administrator Password**:

   - After your Windows instance is running, go to the EC2 dashboard, select the instance, and click **Get Windows Password**.

   - You'll be asked to upload the `.pem` private key you used to launch the instance to decrypt the password.

2. **Decrypt the Password**:

   - Once decrypted, copy the **Administrator password**.

3. **RDP Connection**:

   - Use a Remote Desktop Protocol (RDP) client (like Remote Desktop on Windows or Microsoft Remote Desktop on macOS) to connect to your instance using the following information:

     - **Public IP address** of the instance.

     - **Administrator username** (usually `Administrator`).

     - **Decrypted password**.

Managing Key Pairs in AWS CLI

You can also create and manage EC2 key pairs using the **AWS CLI**.

- **Create a Key Pair**:

  bash
  CopyEdit

  ```
  aws ec2 create-key-pair --key-name MyKeyPair --query "KeyMaterial" --output text > MyKeyPair.pem
  ```

1. This command creates a key pair called MyKeyPair and saves the private key to a .pem file.

- **Import an Existing Key Pair**:
  If you have an existing public key that you want to import into AWS:

  bash
  CopyEdit

  ```
  aws ec2 import-key-pair --key-name MyExistingKey --public-key-material file://my-public-key.pub
  ```

2.

Important Security Considerations

1. **Keep Your Private Key Secure**:

   - **Never share** your private key with anyone. It's the primary way to access your EC2 instances.

   - If the key is compromised, you should **regenerate the key pair** and update your instances with the new key.

2. **No Password Authentication**:

   - By default, EC2 instances are set up so that they do **not** allow password-based SSH or RDP logins (only key-based authentication). This makes them more secure.

3. **Key Rotation**:

   - Periodically rotate your keys and remove unused or compromised keys.

4. **Access Control**:

    - Ensure that only authorized users or systems have access to the private key. Consider using **AWS Identity and Access Management (IAM)** roles and policies to restrict access to EC2 instances.

Replacing or Adding Key Pairs After Instance Launch

If you've lost the private key or want to change it, you cannot directly modify the key pair on an existing EC2 instance. However, you can:

1. **Create a new key pair** and attach it by updating the instance's authorized keys.

2. **Use EC2 Instance Connect** (for Amazon Linux 2 or Ubuntu) or **Systems Manager Session Manager** to access the instance and manually add a new SSH key to the ~/.ssh/authorized_keys file.

Summary

- **EC2 Key Pairs** are essential for securely connecting to EC2 instances.

- The **public key** is embedded on the instance, while you use the **private key** for authentication.

- You create and manage key pairs in the **AWS Console** or **CLI**.

- SSH (Linux) and RDP (Windows) use key pairs for authentication.

- Always store your private key securely and avoid sharing it.