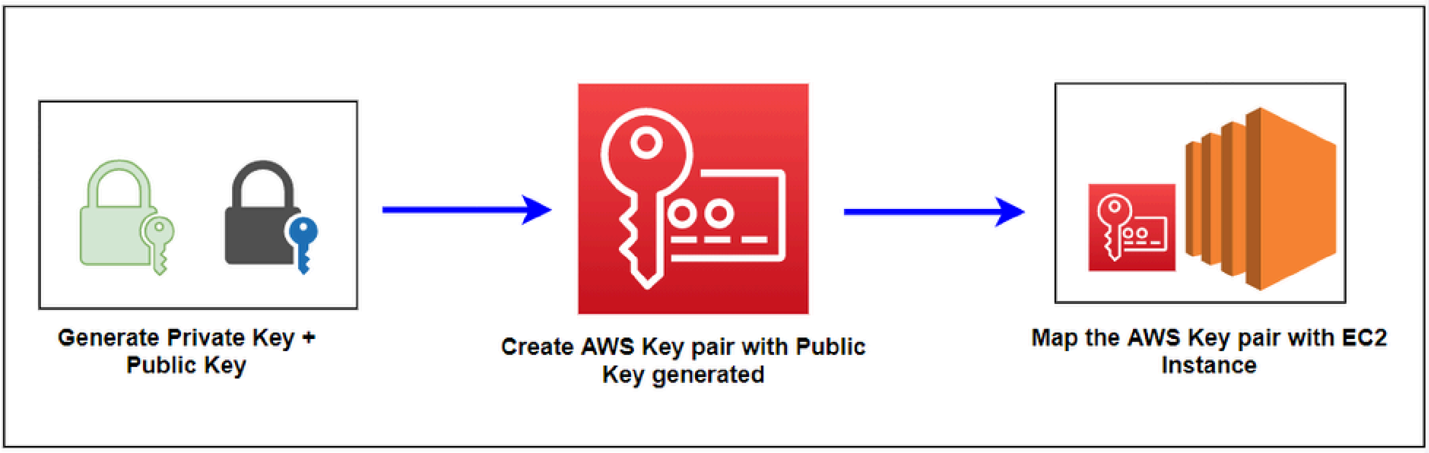
Key Pair in AWS

A **Key Pair** in AWS is a set of security credentials used to securely connect to Amazon EC2 instances. The key pair consists of two parts:

1. **Public Key** – Stored in AWS and associated with the instance.
2. **Private Key** – Downloaded by the user and stored locally. This private key is used to authenticate access to the instance using Secure Shell (SSH).


Generate Private Key + Public Key
Create AWS Key pair with Public Key generated
Map the AWS Key pair with EC2 Instance

Key Features of AWS Key Pairs

1. **Authentication**
   - Key pairs are used to authenticate the user when connecting to EC2 instances.
2. **Secure Communication**
   - Ensures secure access via SSH without the need to use traditional passwords.
3. **Instance Association**
   - A key pair can be associated with one or more EC2 instances at the time of instance launch.
4. **Regional Scope**
   - Key pairs are specific to an AWS region. A key pair created in one region cannot be used in another region unless explicitly copied.
5. **Key Management**
   - AWS does not retain the private key. Users must download and securely store the private key at the time of creation.

Creating a Key Pair

1. AWS Management Console

1. Go to the **EC2 Dashboard**.
2. Navigate to **Key Pairs** under **Network & Security**.
3. Click on **Create Key Pair**.
4. Enter a key pair name and choose the key type:
   - **RSA** (default, widely supported).
   - **ED25519** (newer, more secure, faster).

5. Choose the private key format:
    - **.pem** (for OpenSSH, used on Linux/Unix systems).
    - **.ppk** (for PuTTY, used on Windows systems).
6. Click **Create Key Pair**.
7. The private key file (.pem or .ppk) is downloaded automatically.

2. AWS CLI

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

3. AWS SDKs

- Use AWS SDKs like Boto3 (Python) or AWS SDK for Java to programmatically create and manage key pairs.

Best Practices for Key Pair Management

1. **Secure Storage**
    - Store the private key in a secure location, such as a password-protected directory or a dedicated key management system.
2. **Access Permissions**
    - Restrict access to the private key file using file system permissions:
    - chmod 400 MyKeyPair.pem
3. **Backups**
    - Keep a secure backup of the private key. AWS does not store the private key and cannot recover it.
4. **Use Key Rotation**
    - Regularly rotate key pairs for security. To rotate a key pair:
        1. Create a new key pair.
        2. Add the new public key to the instance’s **~/.ssh/authorized_keys** file.
        3. Remove the old public key after validating access.
5. **Limit Key Pair Sharing**
    - Avoid sharing the private key across multiple users. Instead, use unique key pairs for each user.
6. **Multi-factor Authentication (MFA)**
    - Combine SSH key-based authentication with AWS Identity and Access Management (IAM) for enhanced security.
7. **Use Instance Connect**
    - Consider using **EC2 Instance Connect** for temporary access, which doesn’t require a permanent key pair.

Managing Key Pairs

1. **Delete Unused Key Pairs**
    - Remove unused or unnecessary key pairs from your AWS account.
    - **AWS CLI:**
    - aws ec2 delete-key-pair --key-name MyKeyPair

2. **List Existing Key Pairs**
    - **AWS CLI:**
    - aws ec2 describe-key-pairs
3. **Replace a Key Pair**
    - If the private key is lost:
        1. Create a new key pair.
        2. Use an existing user with access to the instance to update the **~/.ssh/authorized_keys** file with the new public key.

Common Issues and Troubleshooting

1. **Permission Denied Error**
    - Ensure the private key file has proper permissions:
    - chmod 400 MyKeyPair.pem
2. **Lost Private Key**
    - If the private key is lost:
        - Use a user account with access to the instance.
        - Update the instance's **authorized_keys** file with a new public key.
3. **Wrong Key Pair Association**
    - Ensure the correct key pair is associated with the instance during launch.
4. **Key Pair Not Found**
    - Verify you are using the correct region where the key pair was created.

Alternatives to Key Pairs

1. **IAM Roles**
    - Use IAM roles and **AWS Systems Manager Session Manager** for secure and keyless access to EC2 instances.
2. **EC2 Instance Connect**
    - For instances with Amazon Linux 2 or Ubuntu, use EC2 Instance Connect for temporary SSH access without requiring a key pair.
3. **Third-party Tools**
    - Use tools like HashiCorp Vault or AWS Secrets Manager for managing and distributing SSH keys securely.

IMP Points:

1. ***1 ec2 instance can have only 1 key pair***
2. ***we can attach 1 key pair to multiple ec2 instances***
3. ***for windows instance username : administrator; password : you will get through key-pair***
4. ***for Linux instance username : ec2-user; password : you will get through key-pair***