**1. Connecting to AWS Instance**

**Command:**

bash

Command

ssh ubuntu@<IP\_ADDRESS> -i "<PATH\_TO\_PEM\_FILE\_FROM\_LOCAL\_SYSTEM>"

* **Purpose:**  
  Connects to an AWS instance using the Secure Shell (SSH) protocol.
* **Key Details:**
  + Replace <IP\_ADDRESS> with the public IP address of the AWS instance.
  + Replace <PATH\_TO\_PEM\_FILE\_FROM\_LOCAL\_SYSTEM> with the local path to your private key file (PEM file).
  + The IP address of the instance will change every time you start or stop the instance unless you use an elastic IP.

**2. Installing Git on the AWS Server**

**Command:**

bash

Command

sudo apt-get install git

* **Purpose:**  
  Installs Git, a version control tool, on the AWS server.
* **Key Details:**
  + Necessary for cloning repositories from GitHub or other Git hosting services.

**3. Cloning a GitHub Repository**

**Command:**

bash

Command

git clone <GITHUB\_REPO\_PATH>

* **Purpose:**  
  Clones a repository from GitHub to the AWS server.
* **Key Details:**
  + Replace <GITHUB\_REPO\_PATH> with the HTTPS or SSH URL of your GitHub repository.

**4. Checking Current Directory and Listing Files**

**Commands:**

bash

Command

pwd

ls

* **Purpose:**
  + pwd: Prints the current working directory.
  + ls: Lists files and directories in the current directory.
* **Example Output:**
  + /home/ubuntu: The typical home directory for the ubuntu user.
  + Files like TEST\_BACKEND and react-01-login.zip may appear depending on what you’ve added to the directory.

**5. Installing ZIP Utility on the Server**

**Command:**

bash

Command

sudo apt install zip

* **Purpose:**  
  Installs the zip utility for compressing and decompressing files.

**6. Unzipping a File**

**Command:**

bash

Command

unzip react-01-login.zip

* **Purpose:**  
  Extracts the contents of the react-01-login.zip file.
* **Key Details:**
  + Ensure the ZIP file is in the directory where you run the command.

**7. Installing PostgreSQL on AWS Server**

**Command:**

bash

Command

sudo apt install postgresql postgresql-contrib

* **Purpose:**  
  Installs PostgreSQL, an open-source relational database, along with additional tools (postgresql-contrib).
* **Key Details:**
  + PostgreSQL will be required if your application uses a database for storing data.

**8. Checking PostgreSQL Status**

**Command:**

bash

Command

sudo systemctl status postgresql

* **Purpose:**  
  Verifies if PostgreSQL is running on the server.
* **Key Details:**
  + A successful installation will show PostgreSQL as active and running.

**9. Uploading a ZIP File to the AWS Instance via SFTP**

**Steps:**

1. **Connect to the AWS Instance Using SFTP**  
   **Command:**

bash

Command

sftp -i "<PATH\_TO\_PEM\_FILE\_FROM\_LOCAL\_SYSTEM>" ubuntu@<IP\_ADDRESS>

* + Connects to the AWS instance via Secure File Transfer Protocol (SFTP).
  + Replace <IP\_ADDRESS> with the public IP of the instance and <PATH\_TO\_PEM\_FILE\_FROM\_LOCAL\_SYSTEM> with the PEM file's path.

1. **Upload the ZIP File**  
   **Command:**

bash

Command

sftp> PUT "C:\Users\hp\Downloads\react-01-login.zip" /home/ubuntu

* + Uploads the file react-01-login.zip from your local system to the /home/ubuntu directory on the server.
  + Ensure the paths for both the local file and server destination are accurate.

**General Notes**

* Each step is part of setting up an AWS instance to deploy or test an application.
* The process involves connecting to the instance, preparing the server environment, and managing files and databases.