Slack Bot Deployment Documentation

1. Introduction

- **Purpose**: To automate and manage interactions with Slack using a Slack bot deployed on an AWS EC2 instance.
- **Target Audience**: Developers and system administrators who are setting up, running, and maintaining the Slack bot.
- **Overview**: This bot allows integration with Slack APIs for sending and receiving messages, handling events, and automating tasks directly within Slack channels.

2. Prerequisites

- **Requirements**:
  - An AWS EC2 instance.
  - Access to SSH for connecting to the EC2 instance.
  - Python 3.9+ installed on the EC2 instance.
  - `slack_bolt` Python package (and other required packages like `socket-mode`).
- **Environment Variables**:
  - `SLACK_BOT_TOKEN` (provided by Slack for the bot).
  - `SLACK_APP_TOKEN` (required for Socket Mode).
  - `EC2_SSH_KEY` (SSH private key for connecting to the EC2 instance).
  - `OPENAI_API_KEY` (your OpenAI API key).

3. Setup Instructions

Step 1: Prepare the EC2 Instance

- **Connect to the EC2 instance**:

  ```
  ssh -i "your-ssh-key.pem" ec2-user@your-ec2-public-ip
  ```

- **Install Python**:

  ```
  sudo yum install python3 -y
  ```

- **Set up virtual environment**:

```
python3 -m venv slackbot-env
source slackbot-env/bin/activate
```

Step 2: Install Dependencies

- Navigate to your bot's directory:

  ```
  cd ~/slackbot
  ```

- Install the required Python packages:

  ```
  pip install -r requirements.txt
  ```

Step 3: Set Environment Variables

- **Using GitHub Secrets**:
  - In your GitHub repository, go to `Settings` -> `Secrets` -> `New repository secret`.
  - Add the following secrets:
    - `SLACK_BOT_TOKEN` (Slack bot token).
    - `SLACK_APP_TOKEN` (Slack app token).
    - `EC2_SSH_KEY` (your SSH private key).
    - `OPENAI_API_KEY` (your OpenAI API key).
- **On EC2**:
  - Set the `SLACK_BOT_TOKEN` and `SLACK_APP_TOKEN` as environment variables on the EC2 instance:

    ```
    export SLACK_BOT_TOKEN=$SLACK_BOT_TOKEN
    export SLACK_APP_TOKEN=$SLACK_APP_TOKEN
    export OPENAI_API_KEY=$OPENAI_API_KEY
    ```

Step 4: Start the Bot

- Run the bot:

  ```
  nohup python slackBot.py &
  ```

4. Configuration

Configuring the Slack App Token

- Navigate to the Slack API page and create a new Slack app.
- Under `Bot Token Scopes`, add the necessary permissions.

- Note down the generated `SLACK_BOT_TOKEN` and `SLACK_APP_TOKEN`.
- **Environment Variables**:
  - Provide instructions on how to set up these environment variables on the EC2 instance using GitHub Secrets and command line:

    ```
    export SLACK_BOT_TOKEN=$SLACK_BOT_TOKEN
    export SLACK_APP_TOKEN=$SLACK_APP_TOKEN
    export OPENAI_API_KEY=$OPENAI_API_KEY
    ```

5. Running the Bot

Starting the Bot

- Explain the command to start the bot:

  ```
  nohup python slackBot.py &
  ```

- How to stop the bot (useful for troubleshooting or updates):

  ```
  pkill -f slackBot.py
  ```

Checking the Bot Status

- Guide on monitoring the bot:

  ```
  tail -f nohup.out
  ```

- What to look for in the log files to ensure the bot is running correctly.

6. Troubleshooting

Common Errors

- `KeyError: 'SLACK_APP_TOKEN'`: How to resolve this by setting the environment variable.
- `BoltError: Either an env variable SLACK_BOT_TOKEN or token argument in the constructor is required`: How to fix this by ensuring the appropriate tokens are set.
- **Bot Not Responding**:
  - Check the logs for error messages.
  - Ensure environment variables are correctly set.

    - Restart the bot process if necessary.

7. Maintenance

Updating the Bot

- How to update dependencies.
- Procedure to restart the bot after updating.

Scaling

- Considerations for scaling the bot on multiple EC2 instances.
- Load balancing and managing connections.

8. References

- **Slack API Documentation**:
    - Slack API.
- **`slack_bolt` Python library**:
    - slack_bolt GitHub repository.
- **SSH Key Management**:
    - Managing SSH Keys.

CI/CD for Slack Bot with GitHub Actions

9. GitHub Actions Setup for CI/CD

- **Overview**:
    - Automate the deployment of your Slack bot to the EC2 instance using GitHub Actions.
- **Workflow File**:
    - **`deploy.yml`**:
        - This workflow file is located in `.github/workflows/` in your GitHub repository.
        - It is triggered on pushes to the `main` branch and deploys the Slack bot to your EC2 instance.

`deploy.yml`:

```
yaml
```

```
Copy code
name: Deploy SlackBot to EC2

on:
  push:
    branches:
      - main

jobs:
  deploy:
    runs-on: ubuntu-latest

    steps:
    - name: Checkout Code
      uses: actions/checkout@v3

    - name: Set up SSH
      uses: webfactory/ssh-agent@v0.5.3
      with:
        ssh-private-key: ${{ secrets.EC2_SSH_KEY }}

    - name: Copy Files to EC2
      run: |
        scp -o StrictHostKeyChecking=no -r * ec2-user@${{
secrets.EC2_PUBLIC_IP }}:~/slackbot/

    - name: Set OpenAI API Key
      run: |
        echo "OPENAI_API_KEY=${{ secrets.OPENAI_API_KEY }}"
>> ~/slackbot/.env

    - name: Install Dependencies on EC2
      run: |
        scp -o StrictHostKeyChecking=no -r * ec2-user@${{
secrets.EC2_PUBLIC_IP }}:~/slackbot/
        pip3 install -r requirements.txt
        pkill -f slackBot.py || true
        nohup python3 slackBot.py &
        EOF
```

- **Usage**:
  - This workflow automatically updates and deploys the Slack bot whenever changes are pushed to the `main` branch. It sets up SSH, copies files, installs dependencies, and starts the bot on the EC2 instance.
- **How to Monitor**:
  - Use the GitHub Actions UI to check the status of deployments.
  - Review the logs for any issues related to deployment on the EC2 instance.

- **Troubleshooting**:
  - If the deployment fails, check the error messages in the GitHub Actions logs.
  - Update environment variables (`SLACK_BOT_TOKEN`, `SLACK_APP_TOKEN`) directly in the EC2 instance if they are missing.
  - Ensure that the bot is running as expected by monitoring the logs on the EC2 instance:

    ```
    tail -f ~/slackbot/nohup.out
    ```

10. Appendices

- **Sample Configurations**:
  - Example configuration files or scripts for managing tokens and environment variables.
- **FAQ**:
  - Address common questions that may arise.
- **Change Log**:
  - Document updates and changes made to the bot over time.