JobsSync

Prerequisites

1. Infrastructure Setup

- **Jenkins Server: An Ubuntu instance where Jenkins will be installed and configured.**
- **Deployment Server: An Ubuntu instance where the application will be deployed.**
- **Ensure both instances are accessible over SSH.**

---

2. Software Requirements

Jenkins Server:

- **Jenkins installed and running.**
- **Docker installed.**
- **Maven installed.**
- **AWS CLI installed and configured.**
- **SSH access to the deployment server.**

Deployment Server:

- **Docker installed.**
- **SSH access enabled for the Jenkins server.**

*******Follow README.md for setting up both the servers.*******

Install Required Plugins

I

nstall the following Jenkins plugins:

- **Docker & Docker Pipeline**: For Docker integration.
- **AWS Credentials**: For managing AWS credentials.
- **SSH Agent**: For SSH key-based authentication.

---

3. Configure Global Tools

- **Maven**: Install Maven and configure it in Jenkins (Manage Jenkins → Global Tool Configuration).
- **Docker**: Ensure Docker is installed on the Jenkins server and the Jenkins user has permissions to run Docker commands.
- **AWS CLI**: Install the AWS CLI on the Jenkins server and configure it with the necessary credentials.

4. Set Up Credentials

Add the following credentials in Jenkins (Manage Jenkins → Credentials → System → Global credentials):

1. **Docker Hub Credentials**:
    - Credential ID: **docker-hub-creds**
    - Username: Your Docker Hub username.
    - Password: Your Docker Hub password.
2. **AWS Credentials**:
    - Credential ID: **aws-creds**
    - Access Key ID: Your AWS access key.
    - Secret Access Key: Your AWS secret key.
3. **EC2 SSH Key**:
    - Credential ID: **ec2-ssh-key**
    - Private Key: The SSH private key for accessing the EC2 instance.

---

5. Configure Environment Variables

Set up the following environment variables in Jenkins (Manage Jenkins → Configure System → Global properties → Environment variables):

- DOCKER_HUB_USER: Your Docker Hub username.
- DOCKER_HUB_REPO: Your Docker Hub repository name.
- APP_NAME: The name of your application (e.g., jobsync).
- S3_BUCKET: The S3 bucket name for storing artifacts.
- POSTGRES_USER: PostgreSQL username.
- POSTGRES_PASSWORD: PostgreSQL password.
- POSTGRES_DB: PostgreSQL database name.
- EC2_INSTANCE_IP: The IP address of your EC2 instance.
- EC2_SSH_USER: The SSH user for the EC2 instance (e.g., ubuntu).

---

6. Configure Email Notifications

Set up email notifications in Jenkins (Manage Jenkins → Configure System → Extended E-mail Notification):

- SMTP server: Configure your email server (e.g., Gmail SMTP).

- Default Recipients: codesai127.0.0.1@gmail.com.
- Test the configuration to ensure emails are sent successfully.

---

7. Configure Jenkins Pipeline Job

1. **Create a New Pipeline Job:**
    - **Go to Jenkins → New Item → Enter a name for your job → Select Pipeline → Click OK.**
2. **Configure the Pipeline:**
    - **In the pipeline configuration, scroll down to the Pipeline section.**
    - **Select Pipeline script from SCM.**
    - **Choose your SCM (e.g., Git).**
    - **Enter the Repository URL (e.g., https://github.com/your-username/your-repo.git).**
    - **Specify the Branch (e.g., main or master).**
    - **Set the Script Path to Jenkinsfile (this is the default path if your Jenkinsfile is in the root of the repository).**
3. **Save the Job.**

---

8. Prepare the EC2 Instance

- Ensure Docker is installed on the EC2 instance.

    1. sudo apt update
    2. sudo apt install -y apt-transport-https ca-certificates curl software-properties-common
    3. curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg
    4. echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
    5. sudo apt update
    6. sudo apt install -y docker-ce docker-ce-cli containerd.io
    7. sudo systemctl start docker
    8. sudo systemctl enable docker
    9. sudo usermod -aG docker $USER

- Open the necessary ports (e.g., 8081 for the application).
- Ensure the SSH key is configured for the Jenkins user to access the EC2 instance.

---

9. Prepare the S3 Bucket

- Create an S3 bucket named **jobsync-artifacts** (or the name you specified in S3_BUCKET).

- Ensure the Jenkins server has the necessary permissions to upload files to the bucket.

---

10. Run the Pipeline

- Trigger the pipeline manually or configure it to run automatically on Git changes.
- Monitor the pipeline execution in the Jenkins console.

---

11. Verify Deployment

- After the pipeline runs successfully, verify that the application is running on the EC2 instance:
    - Access the application at http://<EC2_INSTANCE_IP>:8081.
    - Check the Docker container logs on the EC2 instance for any errors.

---

12. Troubleshooting

- If the pipeline fails, check the Jenkins console output for errors.
- Ensure all credentials, environment variables, and tools are correctly configured.
- Verify that the Jenkins server has network access to Docker Hub, AWS, and the EC2 instance.