

Praxis: Jenkins mit AWS EC2 Instance

Zielsetzung

- Statische Website mit React
- CI/CD mit Jenkins auf einer AWS EC2 Instance

1. Vorbereitung

1.1 AWS EC2 Instance

- Starte eine AWS EC2 Instance (t2.small) mit Ubuntu, öffne die benötigten Ports (SSH: 22, Jenkins UI: 8080) und weise einen SSH key zu.
- Notiere dir die öffentliche IP-Adresse der Instanz.

1.2 Jenkins Installation

- Verbinde dich per SSH mit der EC2 Instance.

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

- Installiere docker und docker compose auf der EC2 Instanz

```
for pkg in docker.io docker-doc docker-compose docker-compose-v2 podman-docker containerd runc; do sudo apt-get install $pkg; done

# Add Docker's official GPG key:
sudo apt-get update
sudo apt-get install ca-certificates curl gnupg
sudo install -m 0755 -d /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg
sudo chmod a+r /etc/apt/keyrings/docker.gpg

# Add the repository to Apt sources:
echo \
"deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu $(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
sudo apt-get update

sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin

sudo groupadd docker
sudo usermod -aG docker $USER
```

- Starte deine SSH Verbindung neu und teste, ob du folgendes Command ausführen kannst, falls nicht starte die EC2 Instanz neu (`sudo reboot`):

```
docker ps
```

- **Auf der EC2 Instanz:** Erstelle einen neuen Ordner und `docker-compose` file:

```
sudo mkdir -p /var/jenkins_home
mkdir $HOME/jenkins
```

```
vim $HOME/jenkins/docker-compose.yml
```

```
version: '3'
services:
  jenkins:
    image: jenkins/jenkins:lts
    restart: always
    privileged: true
    user: root
    ports:
      - 8080:8080
      - 50000:50000
    container_name: jenkins
    volumes:
      - /var/jenkins_home:/var/jenkins_home
      - /usr/bin/docker:/usr/bin/docker
      - /var/run/docker.sock:/var/run/docker.sock
```

- Immer noch auf der EC2 Instanz und im `$HOME/jenkins` Ordner, starte den Container

```
sudo docker compose up
```

- Achte auf den Log und kopiere das initiale Password:

```
jenkins | Jenkins initial setup is required. An admin user has been created and a password generated.
jenkins | Please use the following password to proceed to installation:
jenkins |
jenkins | 0d63c16741c445839c0182dc29e0efee
```

- Öffne Jenkins in deinem Webbrowser, indem du die öffentliche IP-Adresse der EC2 Instance gefolgt von Port 8080 eingibst (z.B., `http://<public-ip>:8080`). Gebe hier das kopierte Password ein.
- Klicke jetzt auf `Select Plugins to install`
- Entferne den Haken bei:
 - Ant
 - Gradle
 - LDAP
 - EMAIL Extension
 - Mailer
- Klicke auf OK und warte bis die Plugins installiert sind
- Erstelle den ersten Admin user:

```
Username: admin
Password: admin
Confirm Password: admin
Full Name: Admin
EMail Address: admin@test.de
```

- Klicke auf `Save and Finish` und dann `Start using Jenkins`

2. Pipeline

2.1 Repository

- Navigiere im Browser zu folgendem Github Repo und erstelle einen Fork in deinem Account:
<https://github.com/0xfabio/jenkins-react-app>

- Clone den Fork auf deinen lokalen Rechner

2.2 Erstelle die Jenkins Pipeline

- Im Jenkins Frontend klicke auf `Create a Job`
- Nenne deinen Job `ReactApp` **KEINE LEERZEICHEN IM JOB NAME**
- Wähle die `Pipeline` Konfiguration aus
- Wähle den Punkt `Github Project` and und gebe die Url zu deinem Fork an
- Klicke auf `Poll SCM` und gebe folgendes Schedule ein: `H/5 * * * *` (Alle 5 Minuten) -> Jenkins überprüft alle 5 Minuten ob sich dein Code verändert hat und startet eine neue Pipeline
- Unter dem Abschnitt Pipeline, wähle `Pipeline script from SCM` um ein Pipeline Script aus dem Repo zu verwenden
- Wähle unter SCM `git` und gib wieder die URL zu deinem Fork an (muss mit `.git` enden)
- Klicke auf `Save`

2.3 Erster Test Job

- Im Menü des `React App` Jobs, klicke auf `Build now`
- Schaue dir den Log an und stelle sicher, das alle Schritte durchlaufen
 - Rechts unten klicke auf Build Id (`#1`)
 - Klicke auf `Console Output`

2.4 Anpassen des Jenkinsfile

- Öffne deinen Fork mit einem Texteditor
- Bearbeite die `Jenkinsfile` Datei, sodass neben dem `npm ci` auch ein Build der React App ausgeführt wird
- Füge deine Änderungen nach dem echo command ein
- Committe und Pushe deine Änderungen
- Nach spätestens 5 Minuten sollte ein neuer Job starten