

Jobsheet 12 – CI/CD

CLOUD COMPUTING

JURUSAN TEKNOLOGI INFORMASI



POLITEKNIK NEGERI MALANG

OKTOBER 2021

BAB 12 : CI/CD

Praktikum

1. Hosting Static Website Menggunakan CI/CD Pipeline

Pada praktikum yang pertama akan dilakukan hosting static website di layanan Object Storage dari Oracle menggunakan GitHub Actions.

1.1. Menyiapkan Bucket Object Storage

- Masuk ke akun <https://cloud.oracle.com> dan navigasi ke menu Storage -> Buckets
- Tekan tombol Create Bucket, dan lengkapi Bucket Name dengan os-react. Biarkan nilai lainnya dengan nilai default kemudian tekan tombol Create.
- Ubah visibility bucket dengan menekan tombol Edit Visibility, dan ubah menjadi public. Tekan tombol Save Changes untuk menyimpan perubahan.
- Pada dashboard bucket, perhatikan nilai Namespace. Nilai ini nanti akan digunakan untuk pengaturan deployment.

1.2. Menyiapkan Project

- Pada praktikum ini, akan digunakan React sehingga harus diinstall terlebih dahulu Node. Untuk langkah instalasi Node silahkan gunakan tautan berikut sebagai referensi <https://nodejs.org/en/download/> atau <https://nodejs.org/en/download/package-manager/>
- Setelah Node terinstall, silahkan lakukan Langkah generate project dengan menggunakan perintah berikut.

```
npx create-react-app os-react
```

- Buatlah repository pada GitHub kemudian push project tersebut ke dalam repository.

1.3. Menambahkan GitHub Workflow

- Pipeline CI/CD pada GitHub dibuat dengan format yaml dan diletakkan pada lokasi .github/workflows. Buatlah file yaml pada lokasi tersebut, sebagai ilustrasi silahkan perhatikan filename berikut

```
.github/workflows/build.yml
```

- Salin konfigurasi pipeline berikut.

```
name: Hosting Static Website

on:
  push:
    branches:
      - '*'
jobs:
  build:
    name: Build Job
    runs-on: ubuntu-latest
    steps:
      - name: 'Start build job'
        run: |
          echo "Starting the build job"

      - name: 'Checkout'
        uses: actions/checkout@v2

      - name: 'Setup nodejs'
        uses: actions/setup-node@v2
        with:
          node-version: '16'
```

```

- run: yarn install
- run: yarn build

- name: 'Write config'
  run: |
    mkdir ~/.oci
    echo "[DEFAULT]" >> ~/.oci/config
    echo "user=${{secrets.OCI_USER_OCID}}" >>
~/.oci/config
    echo "fingerprint=${{secrets.OCI_FINGERPRINT}}" >>
~/.oci/config
    echo "region=${{secrets.OCI_REGION}}" >> ~/.oci/config
    echo "tenancy=${{secrets.OCI_TENANCY_OCID}}" >>
~/.oci/config
    echo "key_file=~/.oci/oci_api_key.pem" >>
~/.oci/config
    echo "${{secrets.OCI_KEY_FILE}}" >>
~/.oci/oci_api_key.pem
    echo "${{secrets.OCI_KEY_PUBLIC}}" >>
~/.oci/oci_api_key_public.pem

- name: 'Install OCI CLI'
  run: |
    curl -L -O
https://raw.githubusercontent.com/oracle/oci-
cli/master/scripts/install/install.sh
    chmod +x install.sh
    ./install.sh --accept-all-defaults
    echo "/home/runner/bin" >> $GITHUB_PATH
    exec -l $SHELL

- name: 'Fix Config File Permissions'
  run: |
    oci setup repair-file-permissions --file
/home/runner/.oci/config
    oci setup repair-file-permissions --file
/home/runner/.oci/oci_api_key.pem

- name: 'Deploy Into Object Storage'
  run: |
    oci os object bulk-delete -bn ${{secrets.OCI_BUCKET}}
--prefix static --force
    oci os object put -bn ${{secrets.OCI_BUCKET}} --file
./build/manifest.json --content-type application/json --force
    oci os object bulk-upload -bn ${{secrets.OCI_BUCKET}}
--src-dir ./build --content-type text/html --include *.html --
overwrite
    oci os object bulk-upload -bn ${{secrets.OCI_BUCKET}}
--src-dir ./build --content-type image/jpeg --include *.jpg --
overwrite
    oci os object bulk-upload -bn ${{secrets.OCI_BUCKET}}
--src-dir ./build --content-type text/javascript --include *.js
--overwrite
    oci os object bulk-upload -bn ${{secrets.OCI_BUCKET}}
--src-dir ./build --content-type text/css --include *.css --
overwrite
    oci os object bulk-upload -bn ${{secrets.OCI_BUCKET}}
--src-dir ./build --content-type text/plain --exclude *.js --
exclude *.html --exclude *.jpg --exclude *.css --exclude
./build/manifest.json --overwrite

```

- c. Pada pengaturan pipeline, terdapat nilai secrets yang diatur melalui pengaturan pada repository GitHub.
- d. Buka alamat repository GitHub, dan masuk ke menu Settings -> Secrets
- e. Sebagai panduan, silahkan gunakan table berikut untuk pengisian secrets

Nama Key	Keterangan
OCI_BUCKET	Nama bucket, pada praktikum ini digunakan nama os-react
OCI_BUCKET_NAMESPACE	Namespace pada bucket, nilai ini dapat dilihat pada Bucket Information. Contoh: axlecckwcju3
OCI_REGION	Nilai region dari akun oracle, Contoh: ap-melbourne-1
OCI_TENANCY_OCID	Nilai tenancy dapat dilihat pada menu Profile -> Tenancy. Contoh: ocid1.tenancy.oc1..aaaaaaa26ywtdlgaxl3p3fn2sknly3wdb553tmzgel4z222hk4kplcmcuq
OCI_USER_OCID	Nilai dapat dilihat pada User Information pada menu Identity & Security -> Users
OCI_FINGERPRINT	Untuk mendapatkan nilai ini, pada User Information, pilih API Keys kemudian tekan tombol Add API Key. Pilih Generate API Key Pair, dan unduh private key dan public key. Pada berkas public key, diperlukan konversi format dengan menggunakan perintah ssh-keygen. Masukkan perintah berikut: ssh-keygen -I -m PKCS8 -f <public-key-file.pub>
OCI_KEY_FILE	
OCI_KEY_PUBLIC	

- f. Lengkapi nilai semua secrets yang dibutuhkan. Kemudian push perubahan ke repository GitHub.
- g. Perhatikan tab Actions pada halaman repository. Cek hasil proses deployment apakah terdapat kesalahan. Jika terjadi kesalahan, koreksi kembali nilai secrets yang dimasukkan.
- h. Jika proses deployment sudah mendapatkan tanda centang hijau, silahkan kembali ke dashboard bucket object storage. Centang berkas index.html, kemudian tekan menu titik tiga dan pilih View Object Details. Perhatikan URL Path dengan pola sebagai berikut <https://objectstorage.<region>.oraclecloud.com/n/<namespace>/b/<bucket-name>/o/index.html>.
- i. Salin URL Path tanpa nilai index.html, kemudian buka kembali project React. Cari berkas package.json dan tambahkan key "homepage" dengan nilai URL Path pada Object Details. Ilustrasi berkas hasil akhir package.json dapat dilihat sebagai berikut.

```
{
  "name": "hello-react",
  "version": "0.1.0",
  "private": true,
  "homepage": "https://objectstorage.region.oraclecloud.com/n/namespace/b/bucket/o/",
}
```

- j. Jika mengalami kesulitan, silahkan gunakan repository <https://github.com/dhanifudin/hello-react> sebagai rujukan.

2. Deploy Simple Website Menggunakan Docker CI/CD Pipeline

Pada praktikum ini akan dilakukan proses deploy pada OCI Compute dengan menggunakan docker. Pada Langkah di bawah mengasumsikan telah dibuat VCN dengan akses pada port HTTP (80). Selain itu diasumsikan telah mempunyai akun docker hub.

2.1. Menyiapkan Project Repository

- a. Buat baru project React dengan memasukkan perintah

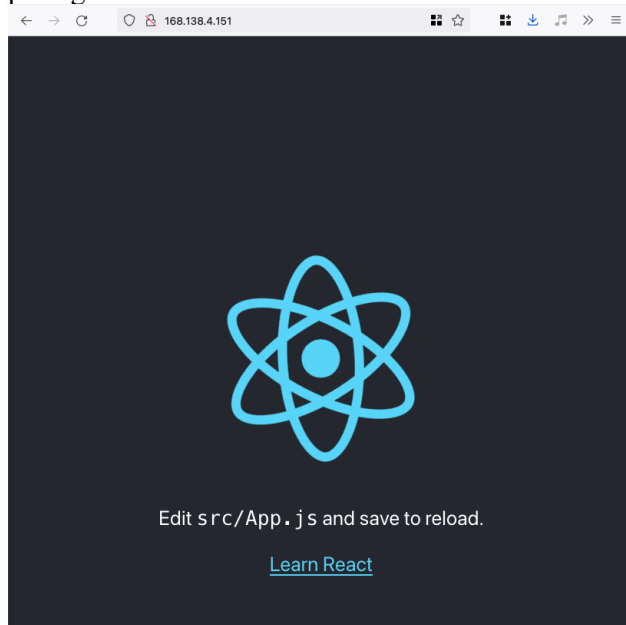
```
npx create-react-app docker-react
```

- b. Buatlah project baru pada GitHub, dan push project React tersebut.
- 2.2. Menyiapkan Akses Docker Hub
 - a. Silahkan login pada akun Docker pada <https://hub.docker.com>
 - b. Untuk memberikan akses push ke dalam registry, perlu dibuat Access Token. Silahkan masuk pada halaman <https://hub.docker.com/settings/security> kemudian tekan tombol New Access Token.
 - c. Masukkan GitHub Actions pada nama token, simpan nilai token yang telah didapatkan.
- 2.3. Menambahkan GitHub Workflow
 - a. Buat sebuah file dengan berkas yaml, pada lokasi .github/workflows/deploy.yml.
 - b. Unduh konfigurasi berkas pada tautan <https://github.com/dhanifudin/hello-react-docker/raw/master/.github/workflows/deploy.yml> dan simpan pada lokasi tersebut.
 - c. Tambahkan nilai-nilai secrets yang terdapat pada GitHub Workflow tersebut. Untuk nilai yang diawali prefix OCI_ dapat digunakan table sebelumnya sebagai referensi.
 - d. Perhatikan table berikut sebagai referensi untuk pengisian nilai secrets

Nama Key	Keterangan
DOCKERHUB_USERNAME	Username akun docker hub
DOCKERHUB_TOKEN	Access token akun docker hub
VM_COMPARTMENT_OCID	Nilai ini sama dengan tenancy id
VM_AVAILABILITY_DOMAIN	Pilih menu Compute -> Instances dan perhatikan nilai Availability domain yang berada di pojok kiri bawah. Contoh: INJg:AP-MELBOURNE-1-AD-1
VM_SHAPE	Pada praktikum ini digunakan shape VM.Standard.E2.1.Micro
VM_CUSTOM_IMAGE_OCID	Buka https://docs.oracle.com/en-us/iaas/images/ Contoh: ocid1.image.oc1.ap-melbourne-1.aaaaaaaalpnwbccpcvukmqfus2rn5jcexymxjjqbiybzhshnheyfojitermyq
VM_SUBNET_OCID	Pilih menu Networking -> Virtual Cloud Network masuk ke vcn yang telah dibuat. Informasi id dapat dilihat dalam Subnet Information pada subnet yang tersedia.

- 2.4. Menyiapkan Dockerfile
 - a. Buatlah berkas dengan nama Dockerfile. Gunakan tautan berikut sebagai referensi: <https://github.com/dhanifudin/hello-react-docker/blob/master/Dockerfile>
 - b. Pastikan penamaan berkas sudah sesuai.
- 2.5. Menyiapkan Docker Compose
 - a. Buatlah berkas dengan nama docker-compose.yml
 - b. Gunakan tautan <https://github.com/dhanifudin/hello-react-docker/blob/master/docker-compose.yml> sebagai referensi untuk membuat docker-compose.yml. Jangan lupa untuk mengubah nilai image sesuai dengan username docker hub.
- 2.6. Melakukan Deployment
 - a. Setelah berkas deployment dan nilai secrets telah selesai diatur, lakukan push ke repository.
 - b. Amati log deployment pada tab Actions, dan pastikan proses menghasilkan tanda centang hijau. Jika terjadi error, periksa kembali konfigurasi dan nilai secrets yang dimasukkan.

- c. Setelah proses deployment berhasil, untuk mendapatkan nilai IP public yang dapat diakses silahkan masuk ke dashboard pada halaman Compute Instances.
- d. Pada hasil akhir, jika tidak terdapat kesalahan akan didapatkan halaman website seperti pada gambar berikut.



Tugas

1. Buatlah laporan dan dokumentasi dari praktikum yang Anda lakukan.
2. Tambahkan proses automated testing sebelum dilakukan deployment
3. Buatlah pipeline deployment untuk project web dinamis.
4. Kirim laporan dalam bentuk pdf dengan format :

Kelas_Absen_Nama_KomputasiAwan_jobsheet07.pdf

contoh : TI3A_25_Alonso_KomputasiAwan_jobsheet10.pdf

Referensi

- [1] J. Nickoloff, *Docker in action*. Shelter Island, NY: Manning Publications, Co, 2016.
- [2] S. Gallagher, *Mastering Docker: rethink what's possible with Docker - become an expert in the innovative containerization tool to unlock new opportunities in the way you use and deploy software*. Birmingham Mumbai: Packt Publishing, 2015.
- [3] J. Krochmalski, *Docker and Kubernetes for Java developers: scale, deploy, and monitor multi-container applications*, First published. Birmingham Mumbai: Packt Publishing, 2017.

