

Deployment auf GCP (Google Cloud Platform)

Mithilfe diesen Schritten, kann die App in einer GCP Cloud Compute VM-Resource betrieben werden. Dafür wird die gcloud-CLI vorausgesetzt. Ebenso muss das Login in GCP bereits erfolgt sein. Zusätzlich muss terraform installiert sein.

Als erstes wird ein neues Projekt erstellt für das Billing.

```
gcloud projects create vcid
```

Die erstellte Projekt-ID auslesen.

```
gcloud projects list
```

Projekt auswählen für das Deployment.

```
gcloud config set project vcid-435813
```

In dem Projekt muss die Compute Engine API aktiviert sein.

```
gcloud services enable compute.googleapis.com
```

Erstelle der Terraform Konfigurationsdatei (main.tf) mit nachfolgendem Inhalt. Mithilfe dieser Datei werden

```
resource "google_compute_instance" "default" {  
  name      = "vcidvm"  
  machine_type = "n2-standard-2"  
  zone      = "us-central1-a"  
  
  boot_disk {  
    initialize_params {  
      image = "debian-cloud/debian-11"  
    }  
  }  
}
```

Für das Deployment mit terraform, mit nachfolgenden schritten, muss die Location im Ordner mit der oben erstellten main.tf Datei sein. Als erstes muss Terraform initialisiert werden mit nachfolgendem Befehl.

```
terraform init
```

Sobald terraform initialisiert ist, kann mithilfe von «plan» die zu erstellende Infrastruktur angezeigt werden.

```
terraform plan
```

Danach kann mithilfe von apply die Ressource erstellt werden.

```
terraform apply
```

Sobald die Ressource in GCP erstellt ist, kann mithilfe von SSH darauf zugegriffen werden. Dafür bietet Google Cloud CLI einen Command.

```
gcloud compute ssh vcidvm
```

Als erstes muss auf der VM Docker installiert werden. Dafür können nachfolgende Commands verwendet werden. Diese Commands stammen von der offiziellen Docker Seite (<https://docs.docker.com/engine/install/debian/>)

```
sudo apt-get update
sudo apt-get install ca-certificates curl
sudo install -m 0755 -d /etc/apt/keyrings
sudo curl -fsSL https://download.docker.com/linux/debian/gpg -o /etc/apt/keyrings/docker.asc
sudo chmod a+r /etc/apt/keyrings/docker.asc
# Add the repository to Apt sources:
echo "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc]
https://download.docker.com/linux/debian \
$(. /etc/os-release && echo "$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
```

Nach der Installation kann ein Ort definiert werden in welchem das GitHub-Repository hin geklont werden kann. Dies wurde in diesem Fall unter /VCID geklont. Danach kann in das geklonte Verzeichnis navigiert werden.

```
# VM Setup Create App Dir and access it
mkdir /VCID
cd /VCID

# VM Setup Clone App repository
git clone https://github.com/gigersam/VCID\_Praxisarbeit\_Public.git

# VM Setup Navigate to Project directory
cd VCID_Praxisarbeit_Public
```

Nach dem klonen des Projekts, muss der App Docker-Container gebildet werden. Das bilden des Containers wird mit nachfolgendem Befehl ausgeführt.

```
sudo docker compose build
```

Für den Test, dass die App gestartet werden kann, kann der nachfolgende Befehl ausgeführt werden.

```
sudo docker compose up
```

Damit bei einem allfälligen Reboot der Maschine die Container automatisch starten, kann das Compose File als Service registriert werden. Dafür wird die Datei docker-App-VCID.service benötigt. In dieser Datei wird definiert, dass der Service docker.service vorhanden sein muss. Der Service immer neugestartet wird, als user root ausgeführt werden muss und member von der Gruppe docker sein muss. Das WorkingDirectory gibt an in welchem Kontext der Service gestartet wird. Bei einem Start wird zuerst die Docker Compose Konfiguration gestoppt und danach gestartet. Beim Stoppen legentlich gestoppt.

```
[Unit]
Description=Docker Compose App VICD Service
Requires=docker.service
After=docker.service

[Service]
Restart=always
User=root
Group=docker
WorkingDirectory=/VCID/VCID_Praxisarbeit_Public
ExecStartPre=docker compose -f docker-compose.yml stop
ExecStart=docker compose -f docker-compose.yml up
ExecStop=docker compose -f docker-compose.yml stop

[Install]
WantedBy=multi-user.target
```

Die Datei docker-App-VCID.service kann vom Projekt in den entsprechenden Ordner kopiert werden. Danach muss der systemctl daemon neu geladen werden. Sobald dies geschehen ist, kann der Service enabled werden. Sobald dies geschehen ist, wird der Service bei jedem Reboot automatisch gestartet. Zum Test kann der Service neugestartet werden.

```
cp docker-App-VCID.service /etc/systemd/system/docker-App-VCID.service
sudo systemctl daemon-reload
sudo systemctl enable docker-App-VCID.service
sudo systemctl restart docker-App-VCID.service
```