

# 1. OpenFaas 설치

## 실행 환경 확인

```
cat /proc/version
```

## Docker 설치

- 공식 문서 : <https://docs.docker.com/engine/install/ubuntu/>

### Set up the repository

- 패키지 인덱스 업데이트 및 필요 패키지 설치

```
$ sudo apt-get update

$ sudo apt-get install \
    ca-certificates \
    curl \
    gnupg \
    lsb-release
```

- GPG Key 추가

```
$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg
```

- stable repository set up

```
$ 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
```

## Docker 엔진 설치

- 최신버전 설치

```
$ sudo apt-get update  
$ sudo apt-get install docker-ce docker-ce-cli containerd.io
```

- 설치 확인

```
$ sudo docker run hello-world
```

## 권한 설정

- root가 아닌 일반 유저도 Docker를 사용할 수 있도록 설정

```
$ sudo usermod -a -G docker $USER  
$ sudo service docker restart
```

- 설정이 반영되도록 재시작

```
$ sudo reboot
```

# Minikube 설치

- 싱글 노드 쿠버네티스 클러스터를 생성하기 위해 kubernetes 대신 minikube 설치
- 공식 문서 : <https://minikube.sigs.k8s.io/docs/start/>

## 최소 사양

- 2 CPU core

```
$ cat /proc/cpuinfo | grep processor
```

- 2GB Memory

```
$ free
```

- 20GB Disk Space

```
$ df -h
```

- Internet connection
- Container manager (Docker)

## 설치

```
$ curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64
$ sudo install minikube-linux-amd64 /usr/local/bin/minikube
```

## 설치 확인

```
minikube start
```

# OpenFaas CLI 설치

- 공식 문서 : <https://docs.openfaas.com/deployment/kubernetes/>
- Minikube에서의 배포 참고 문서 : <https://faun.pub/getting-started-with-openfaas-on-minikube-634502c7acdf>

## OpenFaaS CLI 설치

```
$ curl -sL https://cli.openfaas.com | sudo sh
```

## Helm CLI 설치

```
$ brew install kubernetes-helm
```

## minikube 시작

```
$ minikube stop  
$ minikube start
```

## OpenFaaS namespace 생성

```
$ minikube kubectl -- apply -f https://raw.githubusercontent.com/openfaas/faas-netes/master/namespaces.yml  
  
$ minikube kubectl -- get namespaces
```

## OpenFaaS helm repository 추가 및 업데이트

- OpenFaaS를 helm repository에 추가

```
$ helm repo add openfaas https://openfaas.github.io/faas-netes/
```

- helm chart 업데이트

```
$ helm repo update
```

## Password 설정

- random password 생성 및 확인

```
$ export PASSWORD=$(head -c 12 /dev/urandom | shasum | cut -d' ' -f1)
```

- password의 secret 생성

```
$ minikube kubectl -- -n openfaas create secret generic basic-auth --from-literal=basic-auth-user=admin --from-literal=basic-auth-password="$PASSWORD"
```

## chart를 이용한 openfaas 설치

```
$ helm upgrade openfaas --install openfaas/openfaas --namespace openfaas --set functionName=openfaas-fn --set basic_auth=true
```

## OpenFaaS URL 설정

```
$ export OPENFAAS_URL=$(minikube ip):31112
```

## CLI를 통한 로그인

```
$ echo -n $PASSWORD | faas-cli login -g http://$OPENFAAS_URL -u admin --password-stdin
```

or

```
$ faas-cli login -g http://$OPENFAAS_URL -u admin -p $PASSWORD
```

## openfaas 설치 확인

```
$ minikube kubectl -- get pods -n openfaas
```

## UI 실행

- 포트포워딩

```
$ minikube kubectl -- port-forward svc/gateway -n openfaas 9888:8080 --address='0.0.0.0' &
```

- 필요하다면 방화벽 해제

## Monitoring Dashboard 실행

- openfaas namespace에서 grafana 실행

```
$ minikube kubectl -- -n openfaas run \
--image=stefanprodan/faas-grafana:4.6.3 \
--port=3000 \
grafana
```

- NodePort 설정

```
$ minikube kubectl -- -n openfaas expose pod grafana \
--type=NodePort \
--name=grafana
```

- Grafana NodePort 주소 찾기

```
$ GRAFANA_PORT=$(minikube kubectl -- -n openfaas get svc grafana -o jsonpath="{.spec.ports  
[0].nodePort}")  
$ GRAFANA_URL=http://$(minikube ip):$GRAFANA_PORT/dashboard/db/openfaas
```

- 포트포워딩

```
$ minikube kubectl -- port-forward pod/grafana -n openfaas 9889:3000 --address='0.0.0.0' &
```

- 필요하다면 방화벽 설정