

# 3. Prometheus & Grafana 실습

1. Prerequisites

2. How to Install

1) minikube

2) kube-prometheus-stack Helm Repo 추가

3) kube-prometheus-stack 설치

3. How to Use

## 1. Prerequisites

- k8s 환경
  - minikube v1.22.0
- helm binary
  - helm v3

## 2. How to Install

### 1) minikube

```
minikube start --driver=docker --cpus='4' --memory='4g'
```

### 2) kube-prometheus-stack Helm Repo 추가

- <https://github.com/prometheus-community/helm-charts/tree/main/charts/kube-prometheus-stack>
- Prometheus, Grafana 등을 k8s 에 쉽게 설치하고 사용할 수 있도록 패키징된 Helm 차트
  - 버전 : kube-prometheus-stack-19.0.2

```
# helm repo 추가
helm repo add prometheus-community https://prometheus-community.github.io/helm-charts

# helm repo update
helm repo update
```

### 3) kube-prometheus-stack 설치

```
# helm install [RELEASE_NAME] prometheus-community/kube-prometheus-stack

helm install prom-stack prometheus-community/kube-prometheus-stack
# 모든 values 는 default 로 생성됨
# https://github.com/prometheus-community/helm-charts/blob/main/charts/kube-prometheus-stack/values.yaml

# 정상 설치 확인
# 최초 설치 시 docker image pull 로 인해 수 분의 시간이 소요될 수 있음
kubectl get pod -w
```

- 실무에서 admin password, storage class, resource, ingress 등의 value 를 수정한 뒤 적용하는 경우라면, charts 를 clone 한 뒤, `values.yaml` 을 수정하여 git 으로 환경별 히스토리 관리

## 3. How to Use

### • 포트포워딩

- 새로운 터미널을 열어 포트포워딩
- Grafana 서비스

- `kubectl port-forward svc/prom-stack-grafana 9000:80`

- Prometheus 서비스

- `kubectl port-forward svc/prom-stack-kube-prometheus-prometheus 9091:9090`

### • Prometheus UI Login

- `localhost:9091` 으로 접속
- 다양한 PromQL 사용 가능 (Autocomplete 제공)

- `kube_pod_container_status_running`

- running status 인 pod 출력

- `container_memory_usage_bytes`

- container 별 memory 사용 현황 출력

- 다양한 AlertRule 이 Default 로 생성되어 있음

- expression 이 PromQL 을 기반으로 정의되어 있음

- 해당 AlertRule 이 발생하면 어디로 어떤 message 를 보낼 것인지도 정의할 수 있음

- message send 설정은 default 로는 설정하지 않은 상태

- alertmanager configuration 을 수정하여 설정할 수 있음

- <https://github.com/prometheus-community/helm-charts/blob/7c5771add4ef2e92f520158078f8ea842c626337/charts/kube-prometheus-stack/values.yaml#L167>

- **Grafana UI Login**

- [localhost:9000](http://localhost:9000) 으로 접속
- 디폴트 접속 정보
  - admin / prom-operator

```
kubectl get secret --namespace default prom-stack-grafana -o jsonpath="{.data.admin-user}" | base64 --decode ; echo
```

```
kubectl get secret --namespace default prom-stack-grafana -o jsonpath="{.data.admin-password}" | base64 --decode ; echo
```

- Configuration - Data sources 탭 클릭
  - Prometheus 가 default 로 등록되어 있음
    - Prometheus 와 통신하는 URL 은 쿠버네티스 service 의 DNS 로 세팅
      - Grafana 와 Prometheus 모두 쿠버네티스 내부에서 통신
- Dashboards - Manage 탭 클릭
  - 다양한 대시보드가 default 로 등록되어 있음
    - **Kubernetes/Compute Resources/Namespaces(Pods)** 확인
  - Time Range 조절 가능
  - Panel 별 PromQL 구성 확인 가능
  - 우측 상단의 Add Panel 버튼
    - Panel 추가 및 수정 가능
  - 우측 상단의 Save dashboard 버튼
    - 생성한, 수정한 Dashboard 를 영구히 저장하고 공유 가능
      - Dashboards - Manage 탭
        - Upload JSON file
        - Import from grafana.com