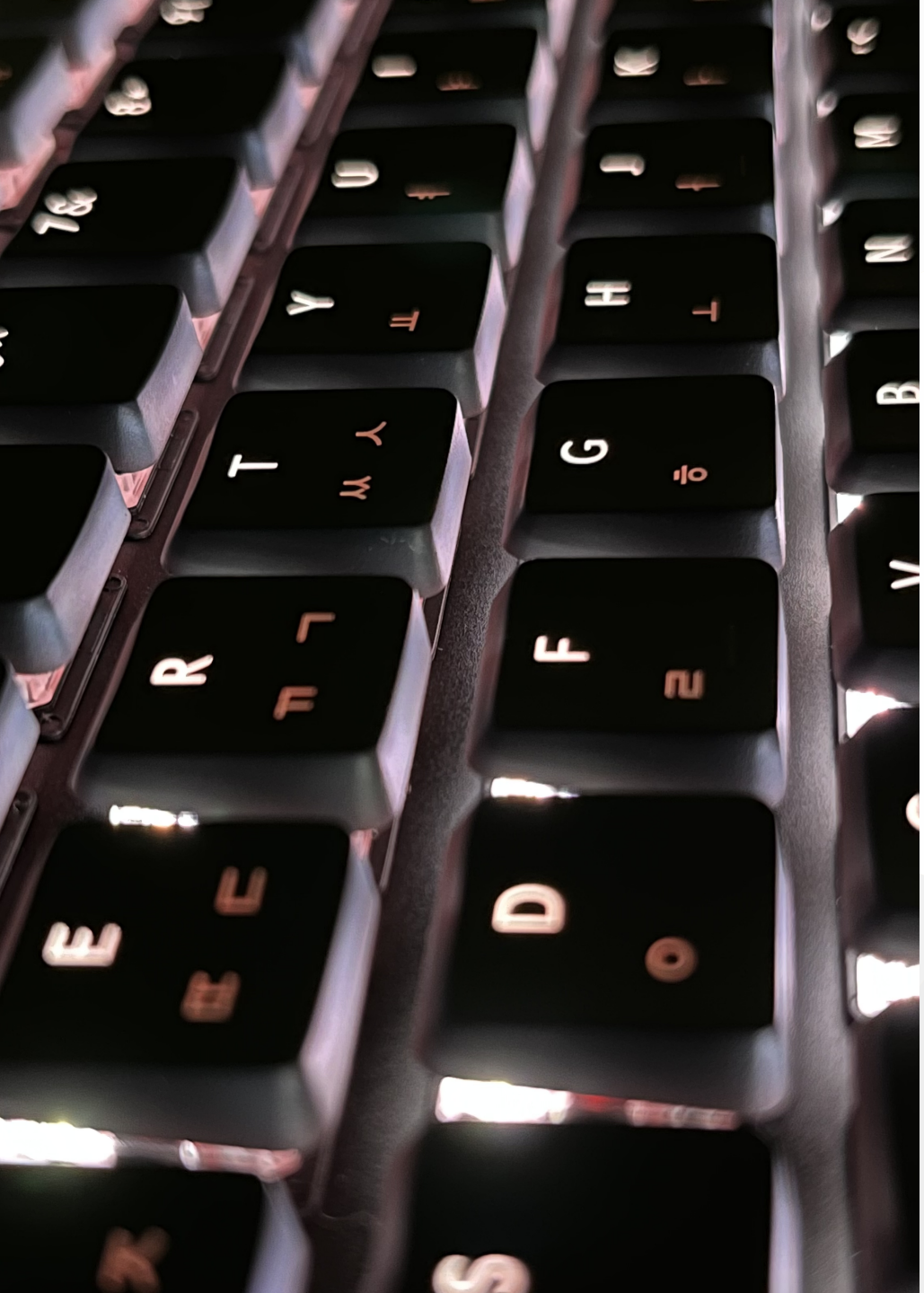

Contents

- kubectl 기본 명령어 사용

kubectl 기본 명령어 사용

기본적인 쿠버네티스 명령어들에 대해 알아보겠습니다.
제일먼저 도움말을 볼까요?

```
ubuntu@ip-10-0-1-14:~$ kubectl --help
kubectl controls the Kubernetes cluster manager.

 Find more information at: https://kubernetes.io/docs/reference/kubectl/overview/

Basic Commands (Beginner):
  create        Create a resource from a file or from stdin
  expose        Take a replication controller, service, deployment or pod and expose it as a new Kubernetes service
  run           Run a particular image on the cluster
  set           Set specific features on objects

Basic Commands (Intermediate):
  explain       Get documentation for a resource
  get           Display one or many resources
  edit          Edit a resource on the server
  delete        Delete resources by file names, stdin, resources and names, or by resources and label selector
... 생략 ...
Usage:
  kubectl [flags] [options]

Use "kubectl <command> --help" for more information about a given command.
Use "kubectl options" for a list of global command-line options (applies to all commands).
```

> 명령어 : `kubectl --help`

여러가지 명령어들을 볼 수 있고, 사용법을 알 수 있습니다.

Docker & Kubernetes - [Hands-on] 07. Kubernetes Overview

몇 가지 명령어들을 알아볼까요?

버젼을 알아보려면,

```
ubuntu@ip-10-0-1-14:~$ kubectl version
Client Version: version.Info{Major:"1", Minor:"23", GitVersion:"v1.23.3", GitCommit:"816c97ab8cff8a1c72eccca1026f7820e93e0d25", GitTreeState:"clean", BuildDate:"2022-01-25T21:25:17Z", GoVersion:"go1.17.6", Compiler:"gc", Platform:"linux/amd64"}
Server Version: version.Info{Major:"1", Minor:"23", GitVersion:"v1.23.3", GitCommit:"816c97ab8cff8a1c72eccca1026f7820e93e0d25", GitTreeState:"clean", BuildDate:"2022-01-25T21:19:12Z", GoVersion:"go1.17.6", Compiler:"gc", Platform:"linux/amd64"}
```

> 명령어 : `kubectl version` 또는 `kubectl version --output yaml`

쿠버네티스 클러스터 정보를 확인하려면,

```
ubuntu@ip-10-0-1-14:~$ kubectl cluster-info
Kubernetes control plane is running at https://192.168.49.2:8443
CoreDNS is running at https://192.168.49.2:8443/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
```

> 명령어 : `kubectl cluster-info`

우리 클러스터의 노드목록은 아래 명령어로 알아볼 수 있습니다.

```
ubuntu@ip-10-0-1-14:~$ kubectl get nodes
NAME       STATUS   ROLES                  AGE     VERSION
minikube   Ready    control-plane,master   4d23h   v1.23.3
```

> 명령어 : `kubectl get nodes`

`--output wide` 옵션(또는, `-o wide`)을 사용하면 더 많은 정보를 보여줍니다.

```
ubuntu@ip-10-0-1-14:~$ kubectl get nodes --output wide
NAME       STATUS   ROLES                  AGE     VERSION   INTERNAL-IP    EXTERNAL-IP   OS-IMAGE             KERNEL-VERSION    CONTAINER-RUNTIME
minikube   Ready    control-plane,master   4d23h   v1.23.3   192.168.49.2   <none>        Ubuntu 20.04.2 LTS   5.15.0-1013-aws   docker://20.10.12
```

> 명령어 : `kubectl get nodes --output wide`

help와 유사하게 쿠버네티스 리소스들의 정의와 설명을 보려면 `kubectl explain` 명령을 사용하면 됩니다.

예를들어 POD에 대해 알아보려면 아래와 같이 실행하면 됩니다.

```
ubuntu@ip-10-0-1-14:~$ kubectl explain pod
KIND:     Pod
VERSION:  v1

DESCRIPTION:
     Pod is a collection of containers that can run on a host. This resource is
     created by clients and scheduled onto hosts.

FIELDS:
   apiVersion   <string>
     APIVersion defines the versioned schema of this representation of an
     object. Servers should convert recognized schemas to the latest internal
     value, and may reject unrecognized values. More info:
     https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
   ... 생략 ...
```

> 명령어 : `kubectl explain pod`

이제 우리 클러스터에 존재한는 리소스들 중 Pod에 대해 좀 더 자세히 알아볼까요?

먼저 현재 존재하는 POD 목록은 아래와 같이 조회합니다.

```
ubuntu@ip-10-0-1-14:~$ kubectl get pods
No resources found in default namespace.
```

> **명령어** : `kubectl get pods`

음. 아무것도 없군요...
지금은 **default 네임스페이스**에서 조회를 한 경우입니다.

`--namespace`로 네임스페이스를 지정하지 않으면 **default 네임스페이스**를 기본으로 합니다.

다른 네임스페이스는 뭐가 있을까요?
네임스페이스를 보려면 아래 명령어를 사용하면 됩니다.

```
ubuntu@ip-10-0-1-14:~$ kubectl get namespaces
NAME                   STATUS   AGE
default                Active   4d23h
kube-node-lease        Active   4d23h
kube-public            Active   4d23h
kube-system            Active   4d23h
kubernetes-dashboard   Active   4d23h
```

> **명령어** : `kubectl get namespaces`

이번에는 Pod목록을 조회하는데, `--all-namespaces` 옵션을 추가해볼까요?

```
ubuntu@ip-10-0-1-14:~$ kubectl get pods --all-namespaces --output wide
NAMESPACE              NAME                                         READY   STATUS    RESTARTS       AGE     IP             NODE       NOMINATED NODE   READINESS GATES
kube-system            coredns-64897985d-hkjv2                      1/1     Running   5 (21m ago)    4d23h   172.17.0.2     minikube   <none>           <none>
kube-system            etcd-minikube                                1/1     Running   5 (10h ago)    4d23h   192.168.49.2   minikube   <none>           <none>
kube-system            kube-apiserver-minikube                      1/1     Running   5 (10h ago)    4d23h   192.168.49.2   minikube   <none>           <none>
kube-system            kube-controller-manager-minikube             1/1     Running   5 (10h ago)    4d23h   192.168.49.2   minikube   <none>           <none>
kube-system            kube-proxy-nhkrh                             1/1     Running   5 (10h ago)    4d23h   192.168.49.2   minikube   <none>           <none>
kube-system            kube-scheduler-minikube                      1/1     Running   5 (10h ago)    4d23h   192.168.49.2   minikube   <none>           <none>
kube-system            storage-provisioner                          1/1     Running   11 (20m ago)   4d23h   192.168.49.2   minikube   <none>           <none>
kubernetes-dashboard   dashboard-metrics-scraper-58549894f-7gj7g    1/1     Running   5 (10h ago)    4d23h   172.17.0.4     minikube   <none>           <none>
kubernetes-dashboard   kubernetes-dashboard-ccd587f44-9pqvf         1/1     Running   9 (20m ago)    4d23h   172.17.0.3     minikube   <none>           <none>
```

> 명령어 : `kubectl get pods --all-namespaces --output wide`

시스템이 사용하는 Pod들을 보려면 kube-system 네임스페이스를 보면 됩니다.

```
ubuntu@ip-10-0-1-14:~$ kubectl get pods --namespace kube-system --output wide
NAME                               READY   STATUS    RESTARTS       AGE     IP             NODE       NOMINATED NODE   READINESS GATES
coredns-64897985d-hkjv2            1/1     Running   5 (22m ago)    4d23h   172.17.0.2     minikube   <none>           <none>
etcd-minikube                      1/1     Running   5 (10h ago)    4d23h   192.168.49.2   minikube   <none>           <none>
kube-apiserver-minikube            1/1     Running   5 (10h ago)    4d23h   192.168.49.2   minikube   <none>           <none>
kube-controller-manager-minikube   1/1     Running   5 (10h ago)    4d23h   192.168.49.2   minikube   <none>           <none>
kube-proxy-nhkrh                   1/1     Running   5 (10h ago)    4d23h   192.168.49.2   minikube   <none>           <none>
kube-scheduler-minikube            1/1     Running   5 (10h ago)    4d23h   192.168.49.2   minikube   <none>           <none>
storage-provisioner                1/1     Running   11 (21m ago)   4d23h   192.168.49.2   minikube   <none>           <none>
```

> 명령어 : `kubectl get pods --namespace kube-system --output wide`

그 중에 하나, kube-scheduler를 좀 더 자세히 볼까요?
정보를 yaml형태로 볼 수 도 있구요. (`--output yaml` 옵션을 사용)

```
ubuntu@ip-10-0-1-14:~$ kubectl get pod kube-scheduler-minikube --namespace kube-system --output yaml
apiVersion: v1
kind: Pod
metadata:
  annotations:
    kubernetes.io/config.hash: be132fe5c6572cb34d93f5e05ce2a540
    kubernetes.io/config.mirror: be132fe5c6572cb34d93f5e05ce2a540
    kubernetes.io/config.seen: "2022-06-28T03:00:51.759906088Z"
    kubernetes.io/config.source: file
    seccomp.security.alpha.kubernetes.io/pod: runtime/default
  creationTimestamp: "2022-06-28T03:00:59Z"
  labels:
    component: kube-scheduler
    tier: control-plane
  name: kube-scheduler-minikube
  namespace: kube-system
  ownerReferences:
  - apiVersion: v1
    controller: true
    kind: Node
    name: minikube
    uid: 62972b18-7ace-41ed-8101-1e799dc7039b
  resourceVersion: "107807"
  uid: 496eb351-e550-47d7-b681-1c1a4db07ee2
  ... 생략 ...
```

명령어 : `kubectl get pod kube-scheduler-minikube --namespace kube-system --output yaml`

`kubectl describe` 명령으로 오브젝트의 자세한 정보를 조회할 수도 있습니다.

```
ubuntu@ip-10-0-1-14:~$ kubectl describe pod kube-scheduler-minikube --namespace kube-system
Name:                 kube-scheduler-minikube
Namespace:            kube-system
Priority:             2000001000
Priority Class Name:  system-node-critical
Node:                 minikube/192.168.49.2
Start Time:           Sun, 03 Jul 2022 02:26:05 +0000
Labels:               component=kube-scheduler
                      tier=control-plane
Annotations:          kubernetes.io/config.hash: be132fe5c6572cb34d93f5e05ce2a540
                      kubernetes.io/config.mirror: be132fe5c6572cb34d93f5e05ce2a540
                      kubernetes.io/config.seen: 2022-06-28T03:00:51.759906088Z
                      kubernetes.io/config.source: file
                      seccomp.security.alpha.kubernetes.io/pod: runtime/default
Status:               Running
IP:                   192.168.49.2
IPs:
  IP:           192.168.49.2
Controlled By:  Node/minikube
Containers:
  kube-scheduler:
    Container ID:  docker://611f03eee9bda2543f503017e1e6a93ac784eefc22bfb6a86a6407a54fdc0e13
    Image:         k8s.gcr.io/kube-scheduler:v1.23.3
    Image ID:      docker-pullable://k8s.gcr.io/kube-scheduler@sha256:32308abe86f7415611ca86ee79dd0a73e74ebecb2f9e3eb85fc3a8e62f03d0e7
    Port:          <none>
    Host Port:     <none>
    ... 생략 ...
```

명령어 : `kubectl describe pod kube-scheduler-minikube --namespace kube-system`

Pod의 로그를 보려면 아래와 같이 하시면 됩니다.

```
ubuntu@ip-10-0-1-14:~$ kubectl logs -n kube-system kube-scheduler-minikube
I0703 02:26:09.300292       1 serving.go:348] Generated self-signed cert in-memory
W0703 02:26:09.824641       1 authentication.go:345] Error looking up in-cluster authentication configuration: Get "https://192.168.49.2:8443/api/v1/namespaces/kube-system/configmaps/extension-apiserver-authentication": dial tcp 192.168.49.2:8443: connect: connection refused
W0703 02:26:09.824698       1 authentication.go:346] Continuing without authentication configuration. This may treat all requests as anonymous.
W0703 02:26:09.824710       1 authentication.go:347] To require authentication configuration lookup to succeed, set --authentication-tolerate-lookup-failure=false
I0703 02:26:09.920711       1 server.go:139] "Starting Kubernetes Scheduler" version="v1.23.3"
I0703 02:26:09.941545       1 configmap_cafile_content.go:201] "Starting controller" name="client-ca::kube-system::extension-apiserver-authentication::client-ca-file"
I0703 02:26:09.941603       1 secure_serving.go:200] Serving securely on 127.0.0.1:10259
I0703 02:26:09.941631       1 tlsconfig.go:240] "Starting DynamicServingCertificateController"
I0703 02:26:09.952426       1 shared_informer.go:240] Waiting for caches to sync for client-ca::kube-system::extension-apiserver-authentication::client-ca-file

... 생략 ...
```

> 명령어 : `kubectl logs -n kube-system kube-scheduler-minikube`

여기까지, 기본적인 kubectl 명령어들을 알아보았습니다. 더 많은 내용은 차차 알아볼게요~ ٩(ˊᗜˋ*)و