SK telecom

Kubernetes and OpenStack-Helm

July 2017

NIC R&D Center 안승규

Partner for New Possibilities

3 minutes
Demo

오늘 발표 내용은?

1. Kubernetes
2. Helm
3. OpenStack-Helm

Kubernetes 용어

[Pod]

- 컨테이너를 담고 있는 그룹 (여러개의 컨테이너가 포함될 수 있음)
- 같은 Pods 안에서의 여러 컨테이너가 같은 네트워크 네임스페이스와 ip 를 가짐 (Apache -> (localhost, port) -> Tomcat)
- 같은 Pods 안에서의 여러 컨테이너가 같은 볼륨을 본다.

[Replica Set]

- Pod 개수를 관리

[Deployment]

- Pod 와 Replica Set 을 통합하여 배포할 수 있는 단위
- 배포 히스토리를 버전별로 관리

[Service]

- Route to pod (using labels) – 내부 IP로 Pod 에 대한 Load Balancing (기본기능)
- 외부에서 접근할려면 아래 두 타입을 활용하여 가능
- 타입 : Load balancer (GCE), NodePort (iptables)

[ConfigMap and Secret]

- ConfigMap : Application 의 Configuration, 혹은 shell script
- Secret : 보안 값

Pod

- 컨테이너 배포 단위로 컨테이너를 담고 있는 그릇
  - 여러 개의 컨테이너가 포함될 수 있음
- 하나의 Pod 안에서의 여러 컨테이너는 같은 docker ip 를 가짐
  - pause 컨테이너가 하나씩 생김
  - Pod 내부에서 컨테이너간 통신은 localhost & 포트로 통신
  - Docker Networking 의 Mapped Container Mode

    ```
    docker run -d --name pause pause_image
    docker run -d --name web -net=container:pause web_image
    ```
- 하나의 Pods 안에서의 여러 컨테이너는 같은 볼륨을 볼 수 있다.

ReplicaSet

- Pod 의 개수를 지정해서 실행
  - 실행 중인 Pod 의 수를 항상 보장
- Pod 를 명령어로 삭제해도 ReplicaSet 에 의해 자동 복구됨
- Horizontal Pod Autoscaler 가 autoscale 을 할 때 ReplicaSet 활용
- ReplicaSet = Pod + replicas 수 (Pod 개수)

```
apiVersion: autoscaling/v1
kind: HorizontalPodAutoscaler
metadata:
  name: frontend-scaler
spec:
  scaleTargetRef:
    kind: ReplicaSet
    name: frontend
  minReplicas: 3
  maxReplicas: 10
  targetCPUUtilizationPercentage: 50
```

Deployment

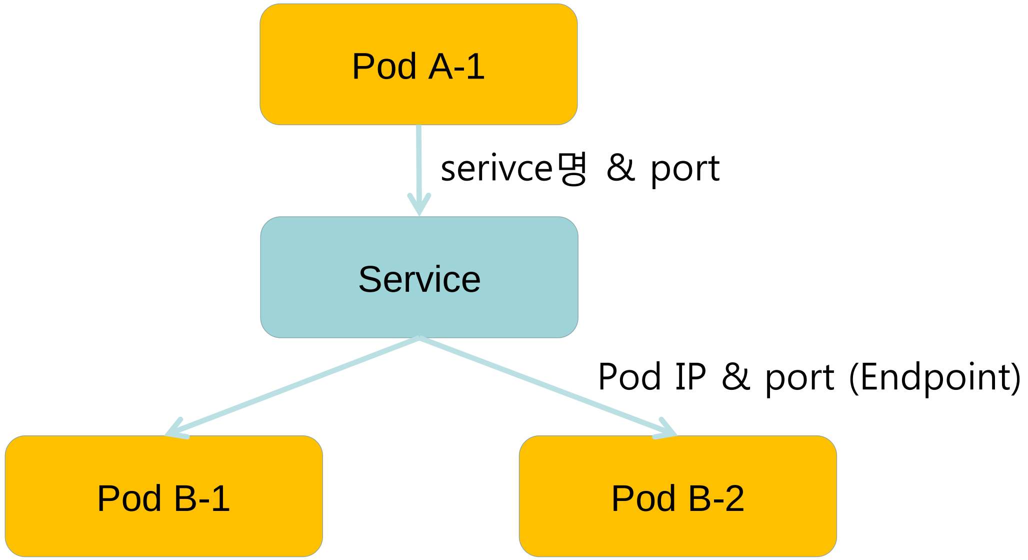
- Deployment = ReplicaSet + History (Revision)
- Pod 배포에 대한 버전 관리가 가능

```
$ kubectl create -f nginx.yaml

$ kubectl rollout history deployment/nginx-deployment

$ kubectl rollout history deployment/nginx-deployment --revision=2

$ kubectl rollout undo deployment/nginx-deployment --to-revision=2
```

```yaml
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: nginx-deployment
  namespace: default
spec:
  replicas: 3
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:1.7.9
        ports:
        - containerPort: 80
```

Service

- Type: ClusterIP(default), LoadBalancer, NodePort, ExternalName
- Pod 를 대표하는 DNS 이름
- ClusterIP 가 할당됨 (Virtual IP)
- kube-proxy 가 iptables 에 Cluster IP 세팅
- Simple Load Balance (default : Round Robin)
- selector 를 지정하면 Endpoint 가 생김



Pod A-1

serivce명 & port

Service

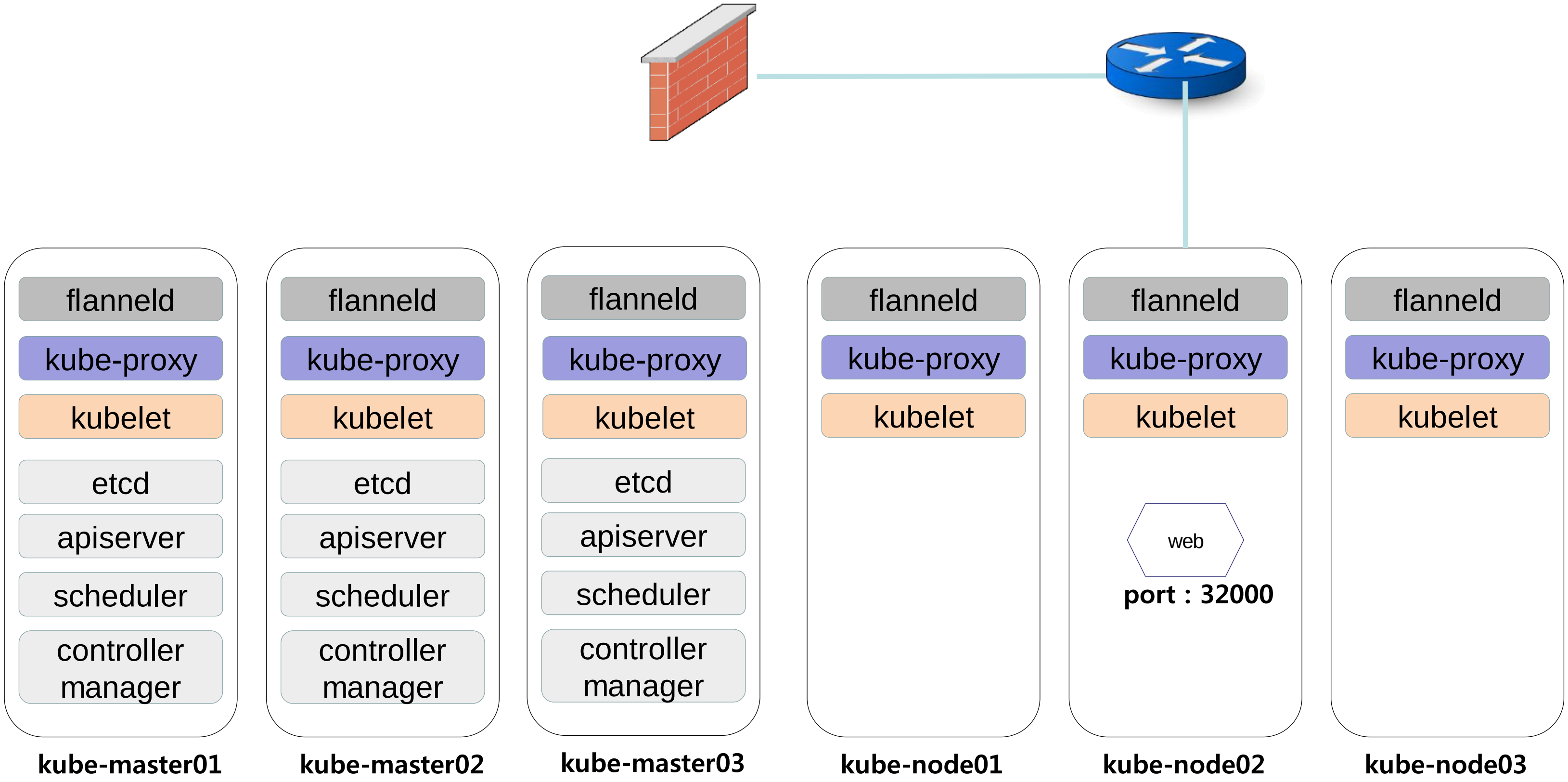
Pod IP & port (Endpoint)

Pod B-1

Pod B-2

```
apiVersion: v1
kind: Service
metadata:
  name: frontend
  labels:
    app: guestbook
    tier: frontend
spec:
  type: NodePort
  ports:
  - port: 80
    nodePort: 31000
  selector:
    app: guestbook
    tier: frontend
```

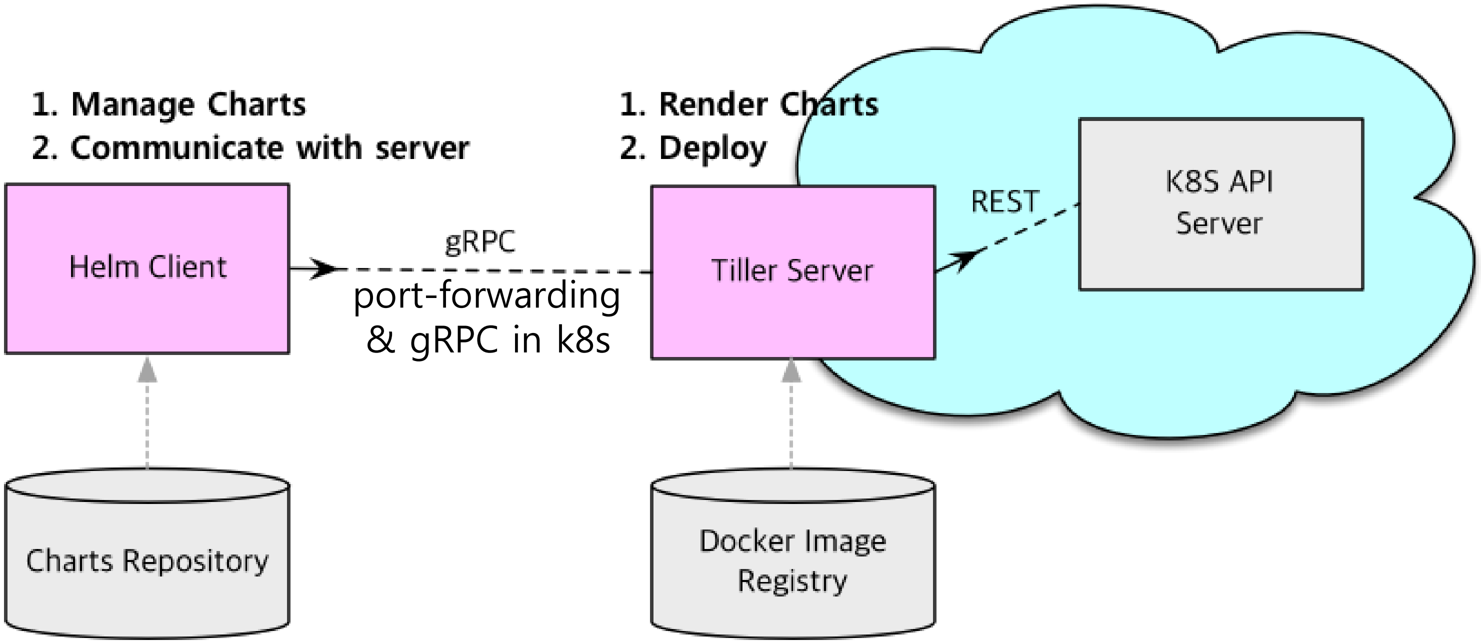
Kubernetes 설치 구성도



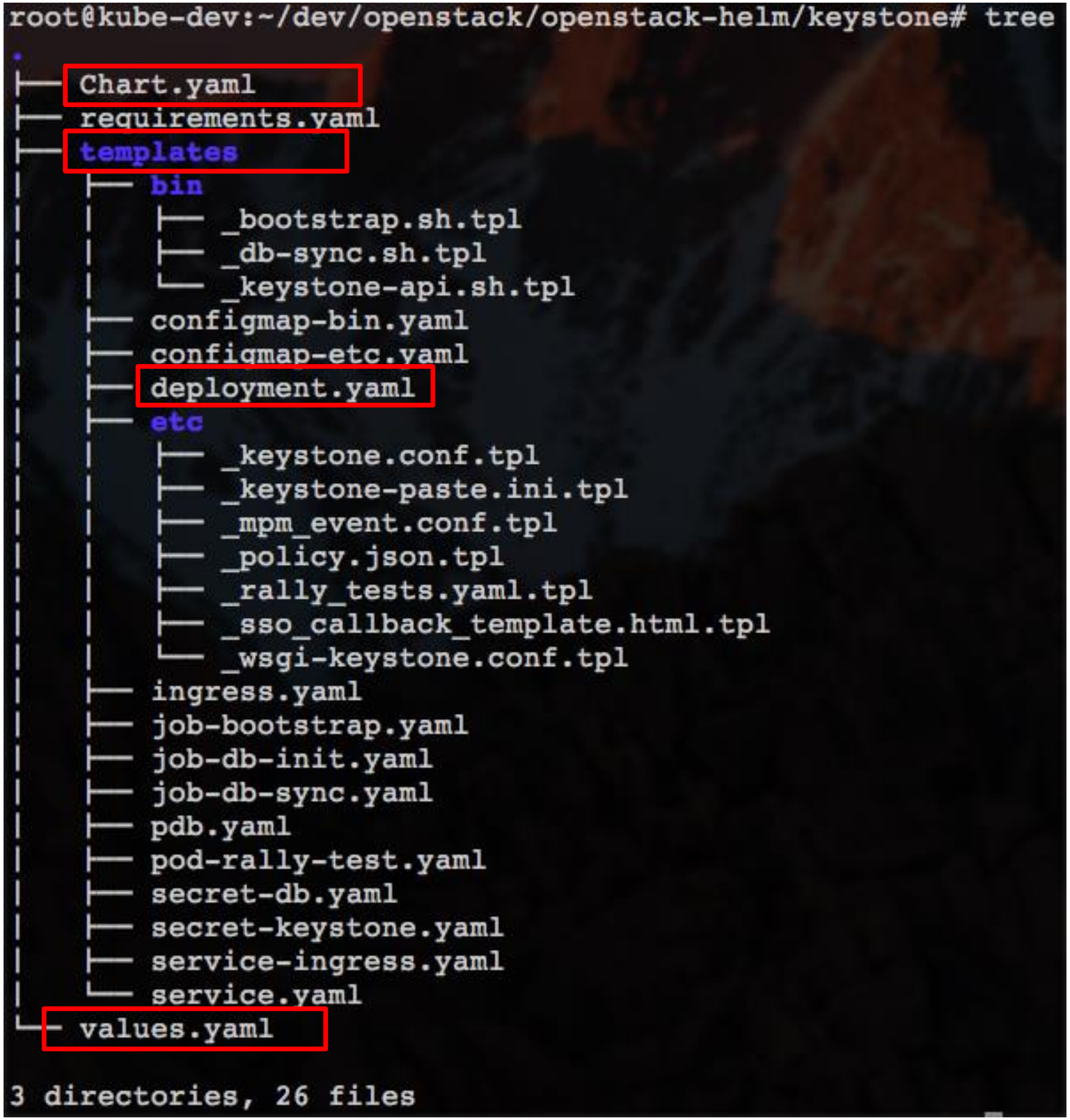
Kubernetes (v1.6.6)

Helm 이란?

- Kubernetes applications 을 Helm charts 로 관리하여 설치, 업그레이드 용이
- Client (Helm) 와 Server (Tiller) 로 구성
- Chart 는 최소한 2개의 구성요소를 가짐
  - Helm 패키지를 설명하는 Chart.yaml
  - Kubernetes manifest 파일을 가지는 Template 파일

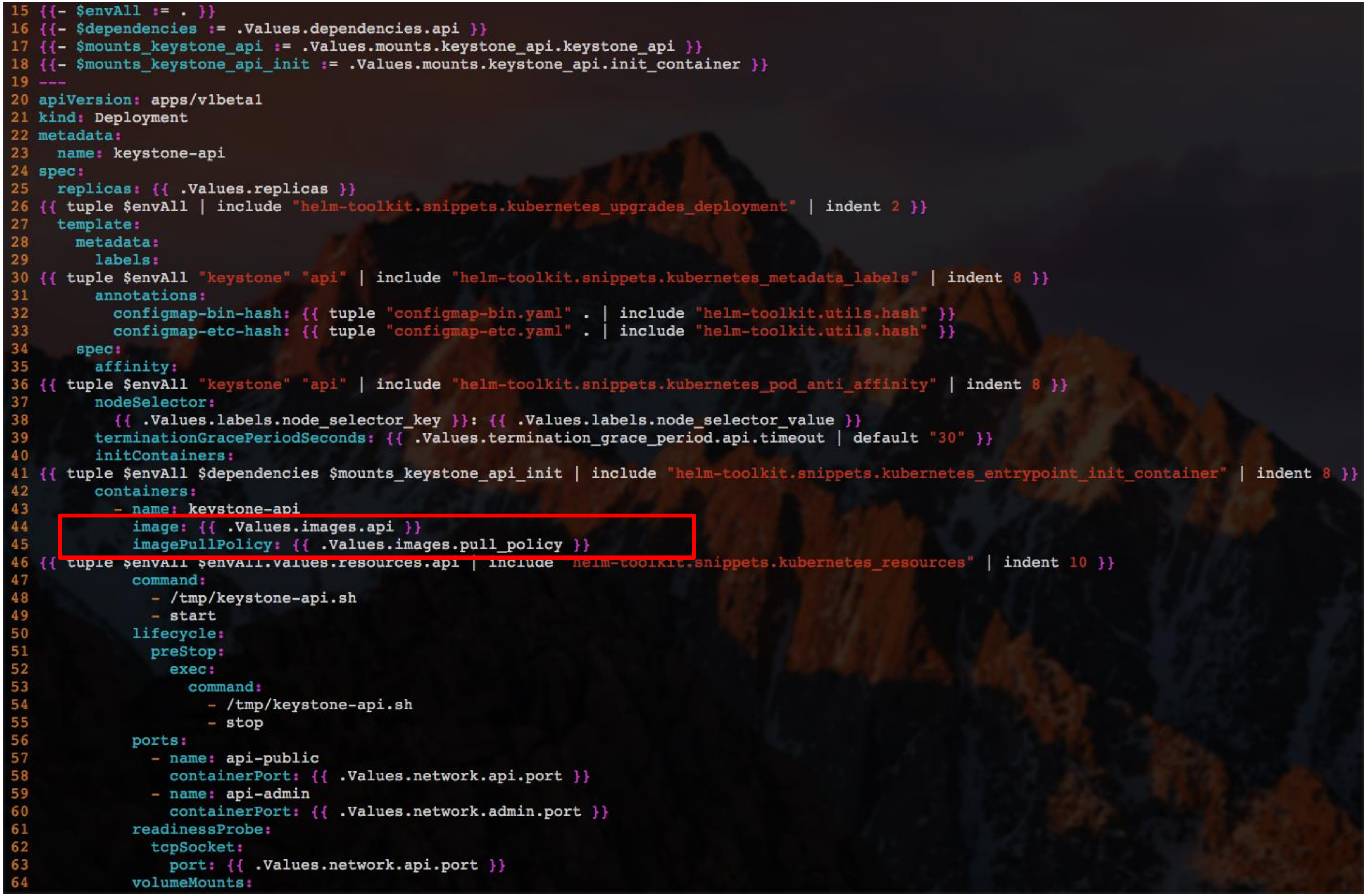


Helm Chart Example – Keystone Chart 구조



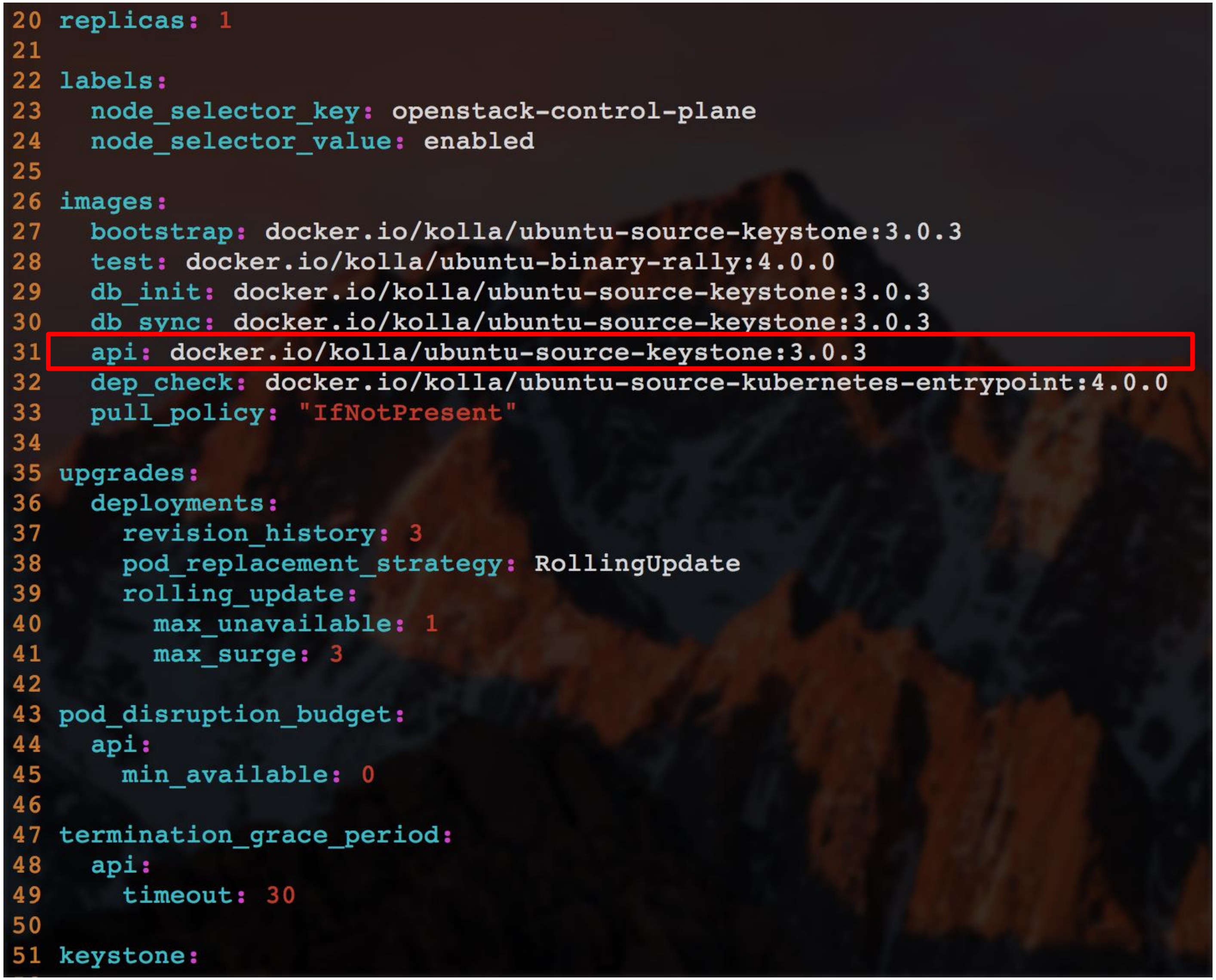
```
root@kube-dev:~/dev/openstack/openstack-helm/keystone# tree
.
├── Chart.yaml
├── requirements.yaml
├── templates
│   ├── bin
│   │   ├── _bootstrap.sh.tpl
│   │   ├── _db-sync.sh.tpl
│   │   └── _keystone-api.sh.tpl
│   ├── configmap-bin.yaml
│   ├── configmap-etc.yaml
│   ├── deployment.yaml
│   ├── etc
│   │   ├── _keystone.conf.tpl
│   │   ├── _keystone-paste.ini.tpl
│   │   ├── _mpm_event.conf.tpl
│   │   ├── _policy.json.tpl
│   │   ├── _rally_tests.yaml.tpl
│   │   ├── _sso_callback_template.html.tpl
│   │   └── _wsgi-keystone.conf.tpl
│   ├── ingress.yaml
│   ├── job-bootstrap.yaml
│   ├── job-db-init.yaml
│   ├── job-db-sync.yaml
│   ├── pdb.yaml
│   ├── pod-rally-test.yaml
│   ├── secret-db.yaml
│   ├── secret-keystone.yaml
│   ├── service-ingress.yaml
│   └── service.yaml
└── values.yaml

3 directories, 26 files
```

Helm Chart Example – Keystone deployment.yaml



```
{{- $envAll := . }}
{{- $dependencies := .Values.dependencies.api }}
{{- $mounts_keystone_api := .Values.mounts.keystone_api.keystone_api }}
{{- $mounts_keystone_api_init := .Values.mounts.keystone_api.init_container }}
---
apiVersion: apps/v1beta1
kind: Deployment
metadata:
  name: keystone-api
spec:
  replicas: {{ .Values.replicas }}
{{ tuple $envAll | include "helm-toolkit.snippets.kubernetes_upgrades_deployment" | indent 2 }}
  template:
    metadata:
      labels:
{{ tuple $envAll "keystone" "api" | include "helm-toolkit.snippets.kubernetes_metadata_labels" | indent 8 }}
      annotations:
        configmap-bin-hash: {{ tuple "configmap-bin.yaml" . | include "helm-toolkit.utils.hash" }}
        configmap-etc-hash: {{ tuple "configmap-etc.yaml" . | include "helm-toolkit.utils.hash" }}
    spec:
      affinity:
{{ tuple $envAll "keystone" "api" | include "helm-toolkit.snippets.kubernetes_pod_anti_affinity" | indent 8 }}
      nodeSelector:
        {{ .Values.labels.node_selector_key }}: {{ .Values.labels.node_selector_value }}
      terminationGracePeriodSeconds: {{ .Values.termination_grace_period.api.timeout | default "30" }}
      initContainers:
{{ tuple $envAll $dependencies $mounts_keystone_api_init | include "helm-toolkit.snippets.kubernetes_entrypoint_init_container" | indent 8 }}
      containers:
        - name: keystone-api
          image: {{ .Values.images.api }}
          imagePullPolicy: {{ .Values.images.pull_policy }}
{{ tuple $envAll $envAll.Values.resources.api | include "helm-toolkit.snippets.kubernetes_resources" | indent 10 }}
          command:
            - /tmp/keystone-api.sh
            - start
          lifecycle:
            preStop:
              exec:
                command:
                  - /tmp/keystone-api.sh
                  - stop
          ports:
            - name: api-public
              containerPort: {{ .Values.network.api.port }}
            - name: api-admin
              containerPort: {{ .Values.network.admin.port }}
          readinessProbe:
            tcpSocket:
              port: {{ .Values.network.api.port }}
          volumeMounts:
```

Helm Chart Example – Keystone values.yaml



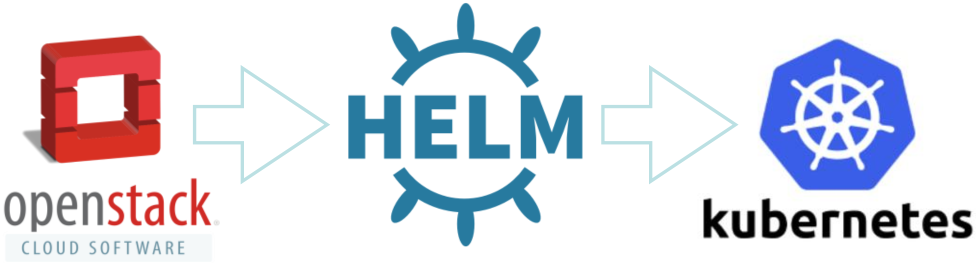
```yaml
replicas: 1

labels:
  node_selector_key: openstack-control-plane
  node_selector_value: enabled

images:
  bootstrap: docker.io/kolla/ubuntu-source-keystone:3.0.3
  test: docker.io/kolla/ubuntu-binary-rally:4.0.0
  db_init: docker.io/kolla/ubuntu-source-keystone:3.0.3
  db_sync: docker.io/kolla/ubuntu-source-keystone:3.0.3
  api: docker.io/kolla/ubuntu-source-keystone:3.0.3
  dep_check: docker.io/kolla/ubuntu-source-kubernetes-entrypoint:4.0.0
  pull_policy: "IfNotPresent"

upgrades:
  deployments:
    revision_history: 3
    pod_replacement_strategy: RollingUpdate
    rolling_update:
      max_unavailable: 1
      max_surge: 3

pod_disruption_budget:
  api:
    min_available: 0

termination_grace_period:
  api:
    timeout: 30

keystone:
```

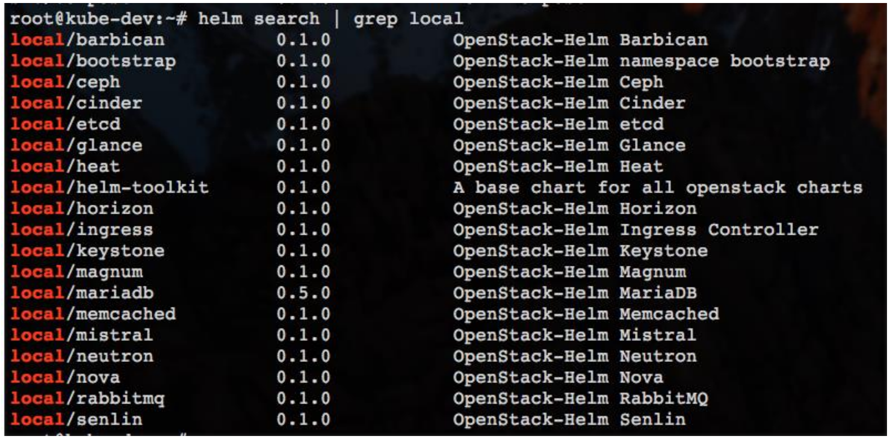
OpenStack-Helm Project

https://github.com/openstack/openstack-helm

- **The goal of OpenStack-Helm is to enable deployment, maintenance, and upgrading of loosely coupled OpenStack services and their dependencies individually or as part of complex environments.**

- AT&T가 2016년 11월 시작한 project
- openstack kolla image들을 k8s helm chart로 관리하는 프로젝트
- 2017년 4월 11일에 openstack 정식 project로 합류

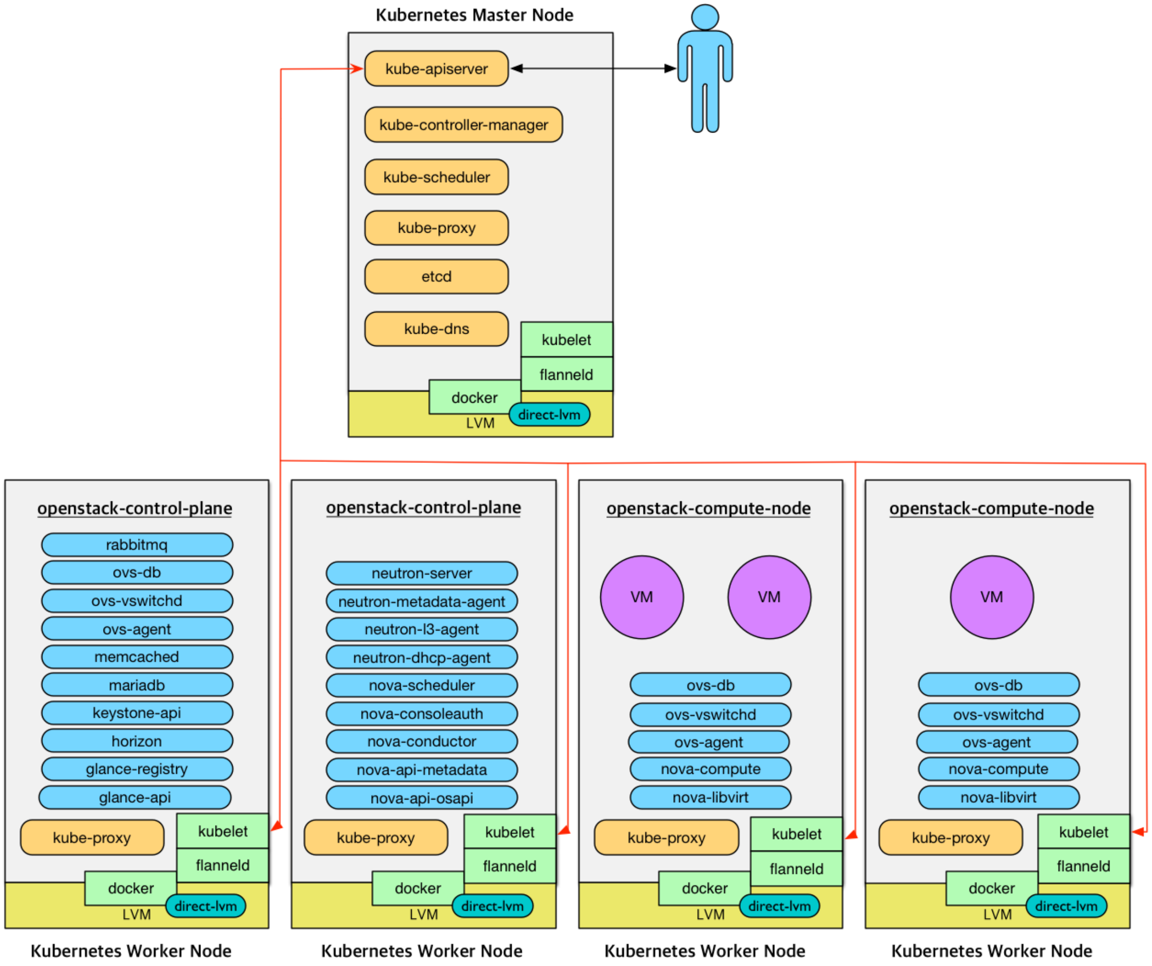




```
root@kube-dev:~# helm search | grep local
local/barbican          0.1.0           OpenStack-Helm Barbican
local/bootstrap         0.1.0           OpenStack-Helm namespace bootstrap
local/ceph              0.1.0           OpenStack-Helm Ceph
local/cinder            0.1.0           OpenStack-Helm Cinder
local/etcd              0.1.0           OpenStack-Helm etcd
local/glance            0.1.0           OpenStack-Helm Glance
local/heat              0.1.0           OpenStack-Helm Heat
local/helm-toolkit      0.1.0           A base chart for all openstack charts
local/horizon           0.1.0           OpenStack-Helm Horizon
local/ingress           0.1.0           OpenStack-Helm Ingress Controller
local/keystone          0.1.0           OpenStack-Helm Keystone
local/magnum            0.1.0           OpenStack-Helm Magnum
local/mariadb           0.5.0           OpenStack-Helm MariaDB
local/memcached         0.1.0           OpenStack-Helm Memcached
local/mistral           0.1.0           OpenStack-Helm Mistral
local/neutron           0.1.0           OpenStack-Helm Neutron
local/nova              0.1.0           OpenStack-Helm Nova
local/rabbitmq          0.1.0           OpenStack-Helm RabbitMQ
local/senlin            0.1.0           OpenStack-Helm Senlin
```

OpenStack on Kubernetes

- openstack-control-plane으로 labeling된 node에 controller component들 배포
- openstack-compute-node로 labeling된 node에 compute, ovs 관련 component들 배포



Q&A