

# Kubespray

ansible 기반의 Kubernetes 설치 자동화 Open Source



## Table of Contents

1. Kubespray 소개
  - . Kubespray 설명
  - . Kubespray 동작방식 설명
2. Kubespray 구성
3. Kubernetes 배포
4. Kubespray 추가 구성의 예
  - . Kubernetes-dashboard 추가 구성

## Kubespray 설명

- Ansible 을 사용하여 Kubernetes 설치 자동화 (HA)
- CoreOS , Debian , Ubuntu , Fedora 및 CentOS / RHEL 지원
- AWS, GCE, Azure, OpenStack or Baremetal에 배포 가능
- <https://github.com/kubernetes-incubator/kubespray>

설치 Component Version  
(Default)

- ❖ kubernetes v1.6.4
- ❖ etcd v3.0.17
- ❖ flanneld v0.6.2
- ❖ calicoctl v0.23.0
- ❖ canal (given calico/flannel versions)
- ❖ weave v1.8.2
- ❖ docker v1.13.1
- ❖ rkt v1.21.0

Requirements

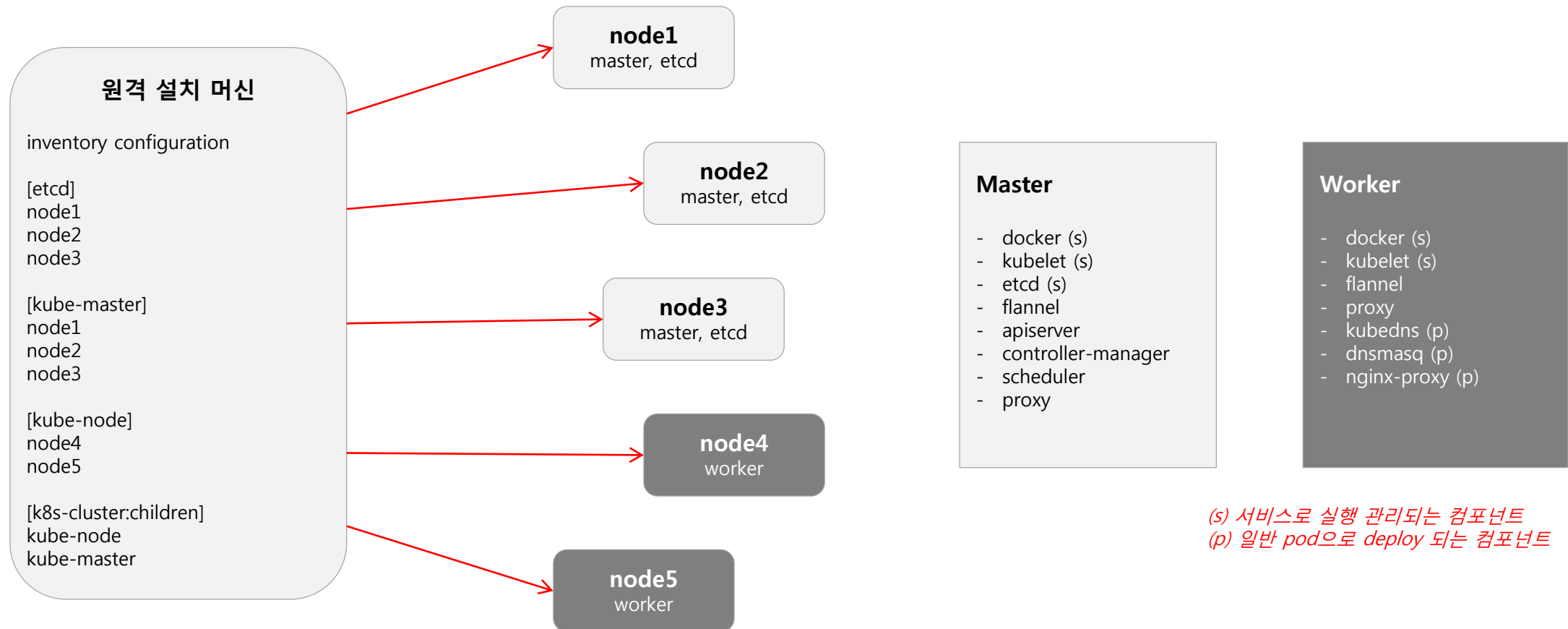
- ❖ Ansible v2.3 (or newer), python-netaddr 이 설치된 머신 필요
- ❖ Ansible Playbooks 를 실행하려면 Jinja 2.9 (or newer) 필요
- ❖ 각 설치 노드는 인터넷에 접속할 수 있어야 함
- ❖ 각 설치 노드는 방화벽을 비활성화 해야 함

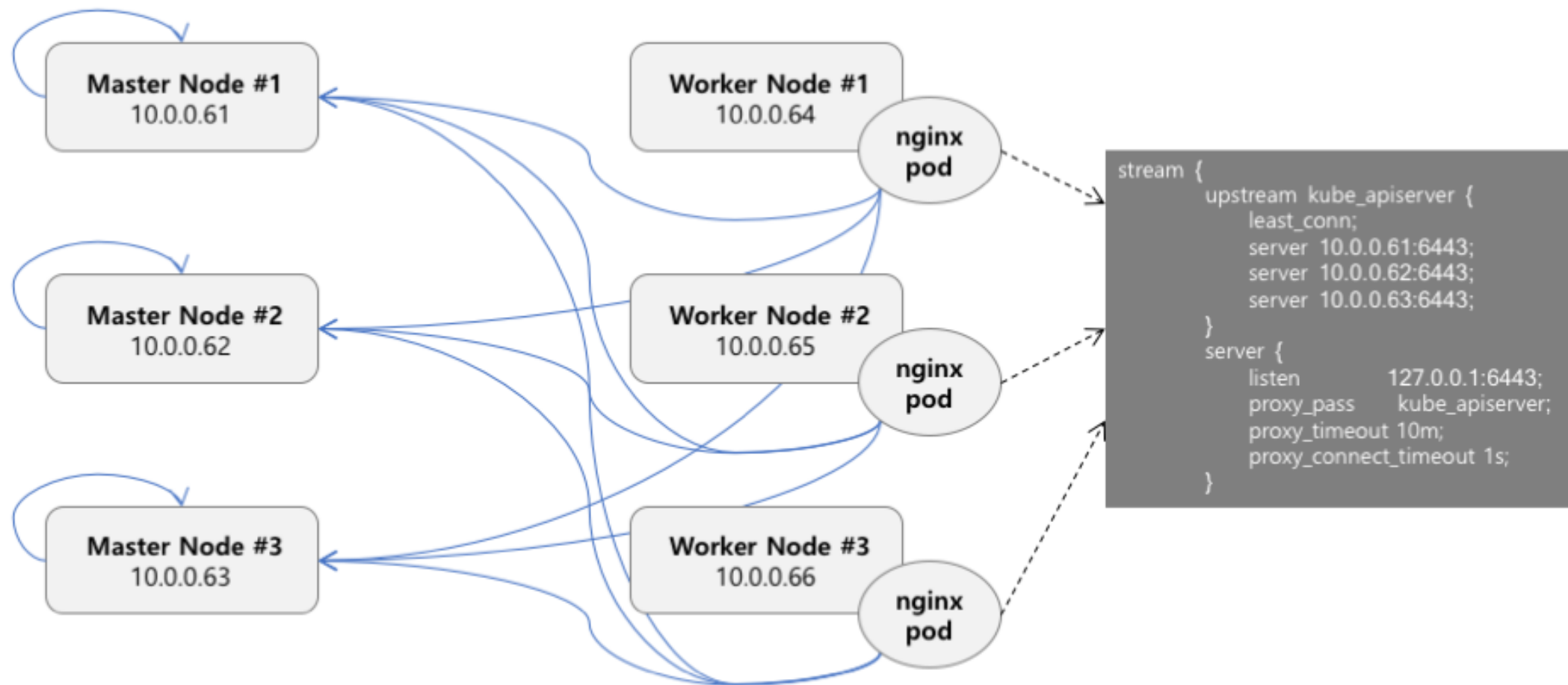
지원 Network plugin

- ❖ Calico (default)
- ❖ Flannel
- ❖ Canal
- ❖ weave

## Kubespray 동작방식 설명

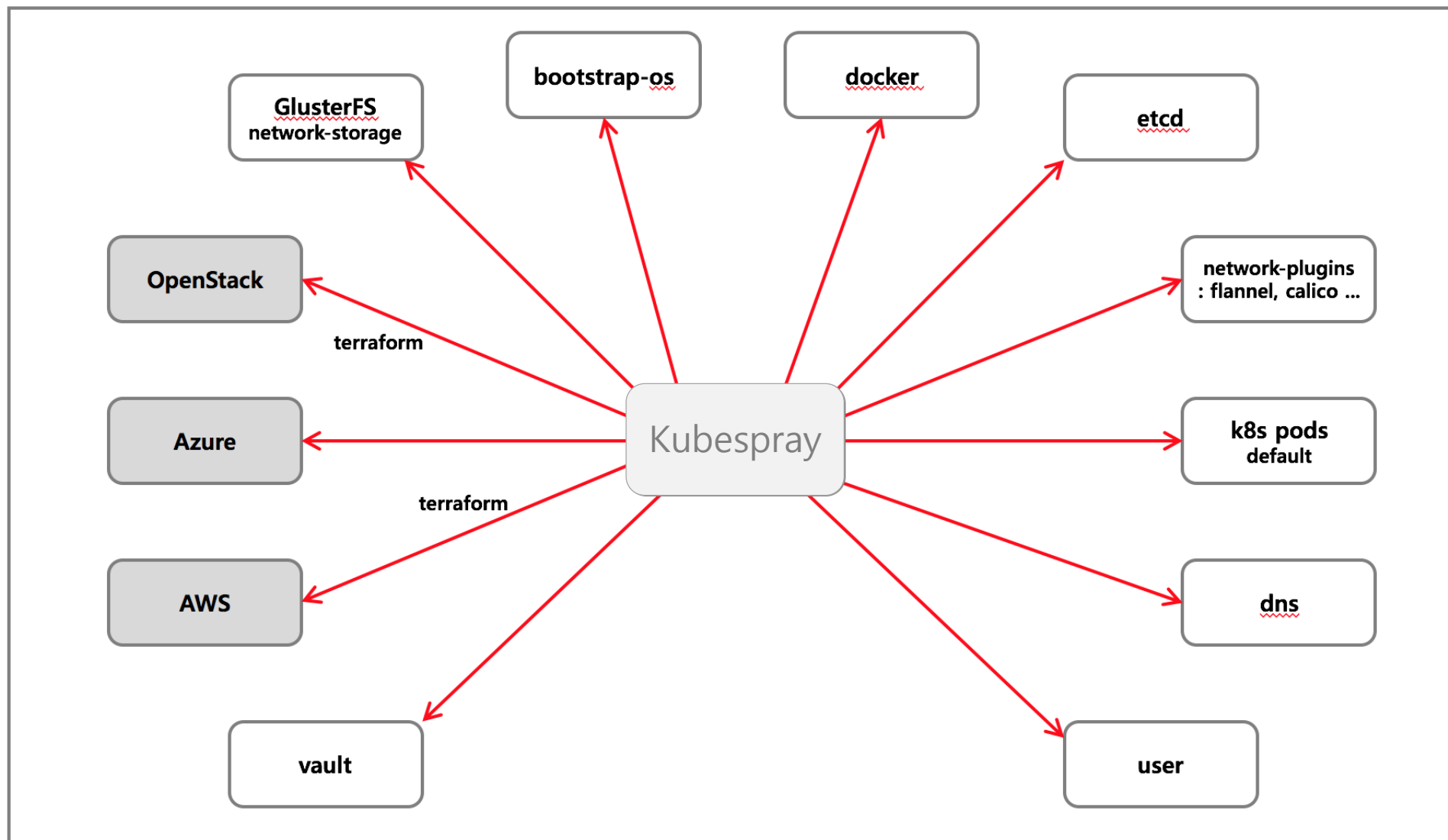
- 설치 머신(원격 또는 대상 노드) 에 Ansible 설치 후 Kubespray project를 이용하여 다중의 노드에 설치
- inventory로 설정된 노드에 역할(그룹)별로 설치
- 기설치 되어 있더라도 반복적으로 설치를 수행할 경우 변동된 사항에 대해서만 업데이트 진행





*kubernetes 설치 node 어디에서든지 127.0.0.1:6443 (localhost) 로 apiserver 접근 가능*

## Kubespray 지원 범위



## Kubespray root 구성

<b>inventory/inventory.cfg</b> <b>inventory/group_vars/</b>	<ul style="list-style-type: none"> <li>❖ 설치 대상 node 정보(ssh host, port, user, password 등) 및 node 그룹 정보</li> <li>❖ common한 설정으로 external LB 지정이나 access ip, protocol proxy, kpm 등 정보</li> <li>❖ k8s cluster 관련 설정인 cert, network plugin, log level, user 등 정보</li> </ul>
<b>roles/</b>	<ul style="list-style-type: none"> <li>❖ cluster.yml 에서 정의한 role 의 대상</li> <li>❖ role 의 path 정보 대로 디렉토리 구조가 구성되어 있음</li> </ul>
<b>cluster.yml</b>	<ul style="list-style-type: none"> <li>❖ 설치 순서를 기술</li> <li>❖ 설치 대상 node 별 실행할 role 정의</li> </ul>
<b>upgrade-cluster.yml</b>	<ul style="list-style-type: none"> <li>❖ cluster.yml 로 설치 후 버전 업그레이드 등의 이유로 수정이 필요할 경우 수행</li> <li>❖ 대부분의 설치 순서는 cluster.yml 과 같으나 설치시 pod의 배포를 못 하도록 설정하고 진행하는 점이 다름</li> </ul>
<b>scale.yml</b>	<ul style="list-style-type: none"> <li>❖ node 의 scale 추가 시 수행</li> <li>❖ 각 node의 세팅(etcd, kubelet 등)까지만 진행하도록 되어 있음</li> </ul>
<b>reset.yml</b>	<ul style="list-style-type: none"> <li>❖ 설치 초기화 role을 지정</li> <li>❖ 설치 초기화 대상은 다음과 같음 <ul style="list-style-type: none"> <li>. etcd, kubelet 과 같은 system service 에 대한 stop &amp; remove</li> <li>. docker 의 config (dns, options) remove</li> <li>. docker 의 모든 containers remove</li> <li>. kubelet 에 mount된 directory 모두 unmount</li> <li>. iptables flushfile과 directory (etcd, cni, dns, nginx, kubelet, kubernetes, calico, flannel) remove</li> <li>. dns 세팅 remove</li> <li>. hosts 정리 및 network restart</li> </ul> </li> </ul>



## Kubespray inventory 구성

```
# vi inventory/inventory.cfg
node1 ansible_port=22 ansible_host=10.0.0.240 ip=10.0.0.240
node2 ansible_port=22 ansible_host=10.0.0.241 ip=10.0.0.241
node3 ansible_port=22 ansible_host=10.0.0.242 ip=10.0.0.242
node4 ansible_port=22 ansible_host=10.0.0.243 ip=10.0.0.243
node5 ansible_port=22 ansible_host=10.0.0.244 ip=10.0.0.244

[etcd]
node1
node2
node3

[kube-master]
node1
node2
node3

[kube-node]
node4
node5

[k8s-cluster:children]
kube-node
kube-master
```



## Kubespray Role 구성

자동 인식 및 실행을 위해서는 모든 하위 directory 에 main.yml 이 있어야 함

### kargo/roles/

defaults

handlers

meta

tasks

templates

vars

Role directory	Description
defaults	role에서 쓰일 기본 변수 선언 모음
handlers	task 맨 뒤 notify에 정의된 handler name 을 기입하는 방법으로 사용한다. 주로 restart 용으로 쓰임
meta	dependency role 을 정의
tasks	위부터 아래 방향 순차적으로 수행할 일에 대해 정의 include 를 이용하여 다른 yml 을 지정 가능
templates	jinja2를 이용하여 설정 파일의 개요를 디자인하고 변수 처리할 수 있음
vars	role에서 쓰일 변수 모음 (다만, Kubespray에서는 OS별 변수 지정용으로 사용 중)
files	shell 을 비롯한 기타 필요한 파일이 위치함

## 설치 대상에 배포

cluster 설치

```
$ cd /app/kargo# ansible-playbook -u centos -b -i inventory/inventory.cfg cluster.yml --private-key=/home/centos/fabric8.pem
```

설치 로그를 별도의 파일로 남길 경우

```
$ ansible-playbook -u centos -b -i inventory/inventory.cfg cluster.yml --private-key=/home/centos/fabric8.pem |& tee log.txt
```

## 설치 확인

```
$ kubectl get pods -n kube-system -o wide
```

NAME	READY	STATUS	RESTARTS	AGE	IP	NODE
dnsmasq-395172919-whmh7	1/1	Running	1	1h	10.233.92.5	kargo-w2
dnsmasq-autoscaler-1155841093-kmxms	1/1	Running	1	1h	10.233.84.2	kargo-w1
flannel-kargo-m1	1/1	Running	0	6m	10.0.0.240	kargo-m1
flannel-kargo-m2	1/1	Running	0	6m	10.0.0.241	kargo-m2
flannel-kargo-m3	1/1	Running	0	6m	10.0.0.242	kargo-m3
flannel-kargo-w1	1/1	Running	0	6m	10.0.0.243	kargo-w1
flannel-kargo-w2	1/1	Running	0	6m	10.0.0.244	kargo-w2
kube-apiserver-kargo-m1	1/1	Running	0	6m	10.0.0.240	kargo-m1
kube-apiserver-kargo-m2	1/1	Running	0	6m	10.0.0.241	kargo-m2
kube-apiserver-kargo-m3	1/1	Running	0	6m	10.0.0.242	kargo-m3
kube-controller-manager-kargo-m1	1/1	Running	0	6m	10.0.0.240	kargo-m1
kube-controller-manager-kargo-m2	1/1	Running	0	6m	10.0.0.241	kargo-m2
kube-controller-manager-kargo-m3	1/1	Running	0	6m	10.0.0.242	kargo-m3
kube-proxy-kargo-m1	1/1	Running	0	6m	10.0.0.240	kargo-m1
kube-proxy-kargo-m2	1/1	Running	0	6m	10.0.0.241	kargo-m2
kube-proxy-kargo-m3	1/1	Running	0	6m	10.0.0.242	kargo-m3
kube-proxy-kargo-w1	1/1	Running	0	6m	10.0.0.243	kargo-w1
kube-proxy-kargo-w2	1/1	Running	0	6m	10.0.0.244	kargo-w2
kube-scheduler-kargo-m1	1/1	Running	0	6m	10.0.0.240	kargo-m1
kube-scheduler-kargo-m2	1/1	Running	0	6m	10.0.0.241	kargo-m2
kube-scheduler-kargo-m3	1/1	Running	1	6m	10.0.0.242	kargo-m3
kubedns-3830354952-zph03	3/3	Running	3	1h	10.233.92.4	kargo-w2
kubedns-autoscaler-54374881-2r648	1/1	Running	1	1h	10.233.84.4	kargo-w1
nginx-proxy-kargo-w1	1/1	Running	0	6m	10.0.0.243	kargo-w1
nginx-proxy-kargo-w2	1/1	Running	0	6m	10.0.0.244	kargo-w2

## Kubernetes-dashboard 추가 구성

경로 : cluster.yml

작업 : 수정

내용 : 가장 마지막에 추가

```
- hosts: kube-master[0]
  any_errors_fatal: "{{ any_errors_fatal | default(true) }}"
  roles:
    - { role: dashboard, when: "use_dashboard == true", tags: dashboard }
```

경로 : inventory/group\_vars/k8s-cluster.yml

작업 : 수정

내용 : kubernetes dashboard 사용여부

```
# Kubernetes dashboard
use_dashboard: true
```

경로 : roles/dashboard/defaults/main.yml

작업 : 신규

내용 : 대시보드 설정 정보 (현재는 없음)

```
---
# kubernetes dashboard configurations
```

## Kubernetes-dashboard 추가 구성

경로 : roles/dashboard/templates/dashboard.yaml.j2

작업 : 신규

내용 : 대시보드 배포 템플릿

```
kind: Deployment
apiVersion: extensions/v1beta1
metadata:
  labels:
    app: kubernetes-dashboard
  name: kubernetes-dashboard
  namespace: kube-system
spec:
  replicas: 1
  revisionHistoryLimit: 10
  selector:
    matchLabels:
      app: kubernetes-dashboard
  template:
    metadata:
      labels:
        app: kubernetes-dashboard
      annotations:
        scheduler.alpha.kubernetes.io/tolerations: |
          [
            {
              "key": "dedicated",
              "operator": "Equal",
              "value": "master",
              "effect": "NoSchedule"
            }
          ]
    spec:
      containers:
        - name: kubernetes-dashboard
          image: gcr.io/google_containers/kubernetes-dashboard-amd64:v1.6.0
          imagePullPolicy: Always
          ports:
            - containerPort: 9090
              protocol: TCP
          args:
            livenessProbe:
              httpGet:
                path: /
                port: 9090
              initialDelaySeconds: 30
              timeoutSeconds: 30
---
kind: Service
apiVersion: v1
metadata:
  labels:
    app: kubernetes-dashboard
  name: kubernetes-dashboard
  namespace: kube-system
spec:
  type: NodePort
  ports:
    - port: 8800
      targetPort: 9090
      protocol: TCP
  selector:
    app: kubernetes-dashboard
```

## Kubernetes-dashboard 추가 구성

경로 : roles/dashboard/tasks/main.yml

작업 : 신규

내용 : 대시보드 배포 작업

```
---
- name: Kubernetes dashboard | copy yaml
  template:
    src: dashboard.yaml.j2
    dest: "{{ kube_config_dir }}/dashboard.yaml"
    force: true
    owner: root
    mode: 0755

- name: Kubernetes dashboard | create
  command: "{{ bin_dir }}/kubectl create -f {{ kube_config_dir }}/dashboard.yaml"
  ignore_errors: yes
```

The background is a dark blue-grey gradient. It is populated with numerous small, interconnected geometric shapes, primarily triangles and quadrilaterals, in various colors including yellow, green, blue, purple, red, and orange. These shapes are scattered across the frame, creating a complex, abstract pattern that resembles a network or a molecular structure. The text "Thank you for Reading" is centered in the middle of the image in a white, sans-serif font.

Thank you for Reading