

# 앤서블 프로젝트 중급

**랩 접속**

# 랩 접속

## 파이어폭스

파이어폭스 다운로드

[https://portableapps.com/apps/internet/firefox\\_portable](https://portableapps.com/apps/internet/firefox_portable)  
github에서는 "FirefoxPortableESR\_102.1.0\_English.paf.exe"  
다운로드 가능함.

# 랩 접속

## 파이어폭스

### Mozilla Firefox, Portable Edition

web browser



Download from PortableApps.com

Version 102.0.1 for Windows, English  
113MB download / 408MB installed  
[All Languages](#) | [Antivirus Scan](#) | [Details](#)

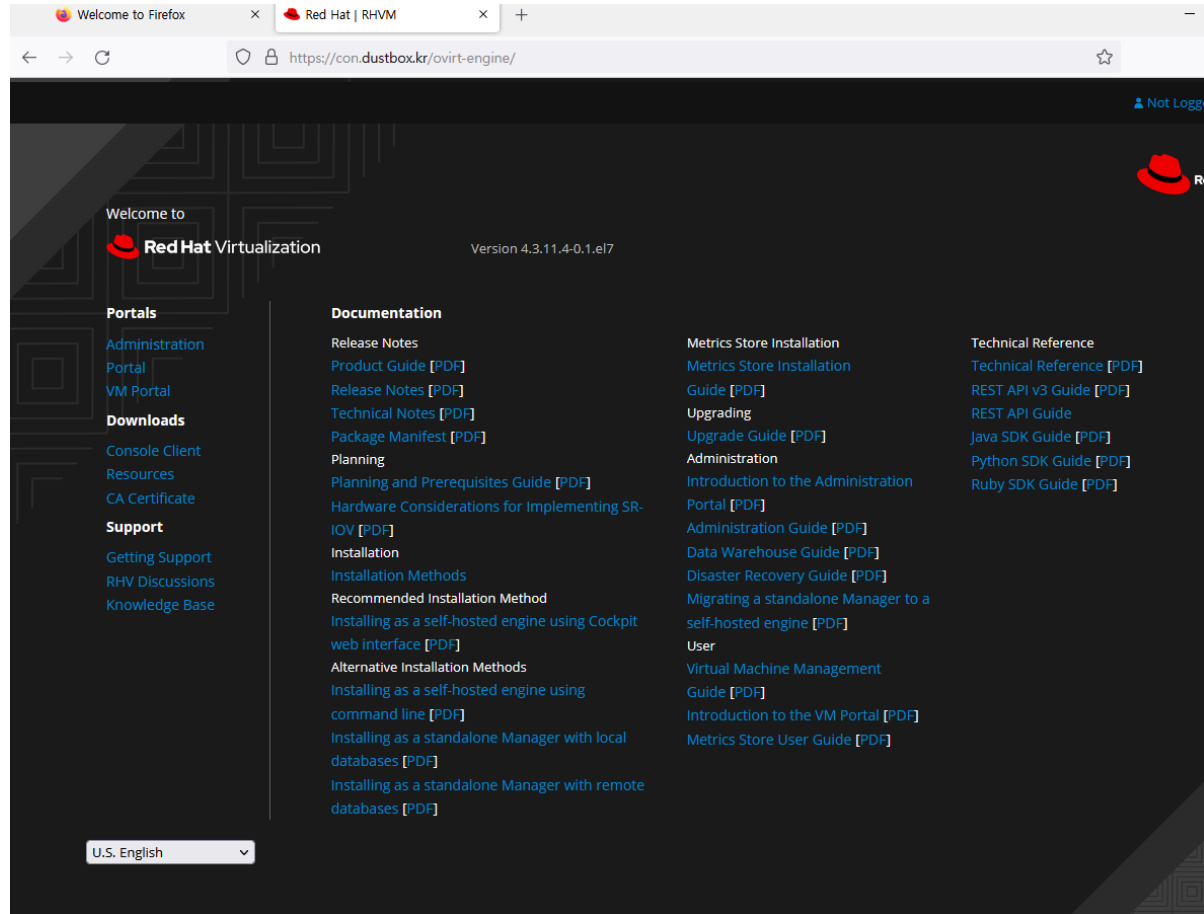


*Mozilla Firefox, Portable Edition can run from a cloud folder, external drive, or local folder without installing into Windows. It's even better with the [PortableApps.com Platform](#) for easy installs and automatic updates.*

Also Available: [Firefox ESR](#) (Extended Support Release), [Firefox Developer Edition](#), [Firefox Beta](#), [Firefox Nightly](#), [Legacy versions](#)

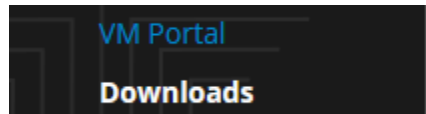
# 랩 접속

## 파이어폭스



# 랩 접속

## 파이어폭스

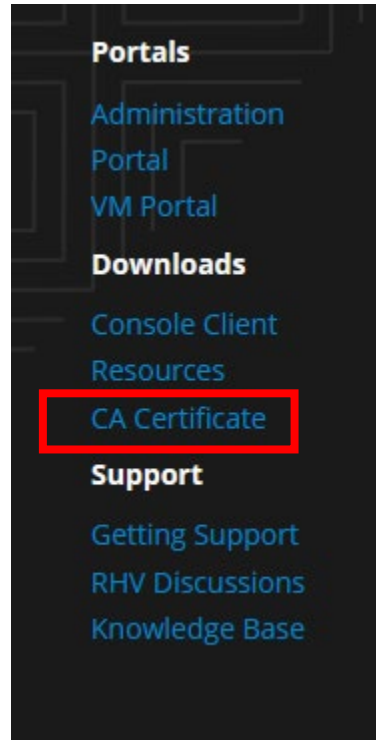


PKI파일은 github에서 다운로드 가능함.

"pki-resource"

# 랩 접속

## 파이어폭스



# 랩 접속

## 파이어폭스





# 랩 접속

## 파이어폭스

 Privacy & Security



### Certificates

- ☒ Query OCSP responder servers to confirm the current validity of certificates

**View Certificates...**

Security DeVICES...

# 랩 접속

## 파이어폭스

Your Certificates

Authentication Decisions

People

Servers

Authorities

You have certificates on file that identify these certificate authorities

Certificate Name	Security Device
▼ AC Camerfirma S.A.	
Chambers of Commerce Root - 2008	Builtin Object Token
Global Chambersign Root - 2008	Builtin Object Token
▼ AC Camerfirma SA CIF A82743287	
Camerfirma Chambers of Commerce Ro...	Builtin Object Token
Camerfirma Global Chambersign Root	Builtin Object Token

View...

Edit Trust...

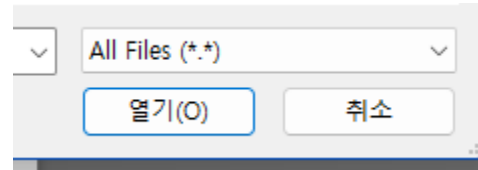
Import...

Export...

Delete or Distrust...

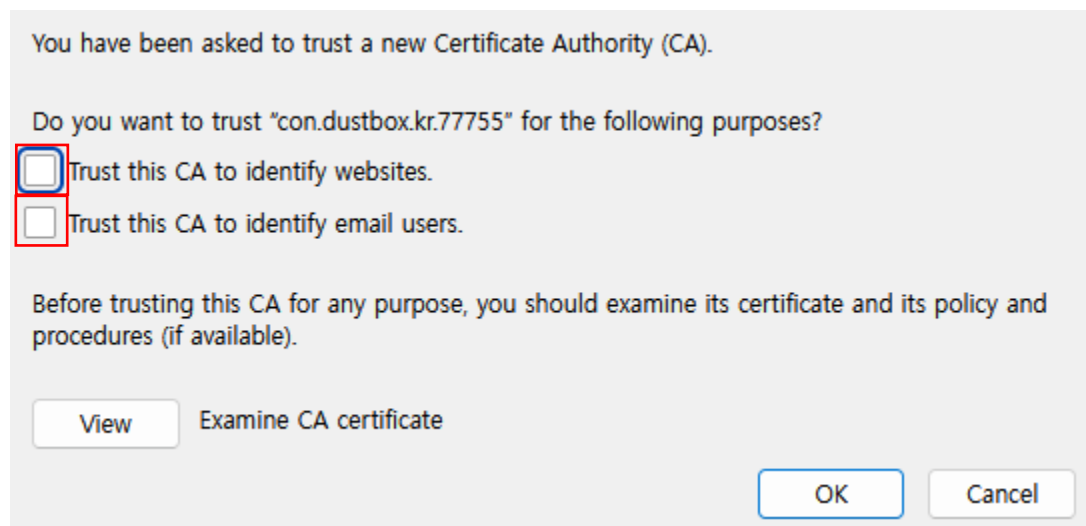
# 랩 접속

## 파이어폭스



# 랩 접속

## 파이어폭스



두개다 체크

# 랩 접속

## 랩 계정

Linux CentOS-Stream-9

ID: root

PW: centos

VM Dashboard Account

ID: skt1~27

PW: stkuser

초기 로그인 시 자동적으로 비밀번호 변경 요구, 가상머신은 변경할 필요는 없음.

# 랩 접속

## 랩 계정

외부에서 접근 가능한 콘솔 서버 정보

주소

ssh://console.dustbox.kr

포트

7722/TCP

ID: skt

PW: sktuser

초기 로그인 시 자동적으로 비밀번호 변경 요구

# 랩 접속

## 랩 계정

만약 웹 VNC콘솔의 글자가 작은 경우 다음처럼 옵션을 설정한다. 해상도는 원하시는 크기로 설정 후 재-부팅 해주시면 화면 크기가 다시 재조정 됩니다.

```
# vi /etc/default/grub
```

```
...
```

```
GRUB_CMDLINE_LINUX="crashkernel=auto resume=/dev/mapper/cl-swap rd.lvm.  
lv=cl/root rd.lvm.lv=cl/swap rhgb quiet video=640x480"
```

```
...
```

```
# grub2-mkconfig -o /etc/grub2.cfg
```

```
# reboot
```

# 랩 접속

## 랩 계정

가급적이면 타이핑 편하게 ssh를 통해서 가상머신으로 접근 부탁드립니다.

가상머신의 아이피 주소는 대시보드에 접근하면 화면에 출력이 됩니다.

이 부분은 강사가 따로 설명드릴 예정 입니다.



# 시작하기 앞서서

이번 과정은 이틀동안 쿠버네티스 설치를 위한 플레이북 작성 및 구성.

이 과정을 진행하기 위해서 다음과 같은 조건이 필요.

# 시작 전 설명

# 시작하기 앞서서

이번 과정은 이틀동안 쿠버네티스 설치를 위한 플레이북 작성 및 구성.

이 과정을 진행하기 위해서 다음과 같은 조건이 필요.

# 시작하기 앞서서

- 앤서블 모듈 사용 방법 및 Jinja2, Template 이해
- roles기반으로 플레이북 다중 플레이북 실행 및 구성
- inventory 및 ansible magic variable 이해
- 확장성을 고려한 role 구성

# 시작하기 앞서서

**위의 기술에 대한 이해도가 없는 경우 진행이 어려움**

# 시작하기 앞서서

비 기술적인 부분에 대해서는 다음과 같은 기술이 필요하다.

# 시작하기 앞서서

- 절차에 대한 정리 및 과정 성립
- 기능을 role 혹은 tasks별 분리
- roles기반 playbook 묶음 디자인

# 시작하기 앞서서

참고용 플레이북은 다음 주소에 있음.

<https://github.com/tangt64/duststack-k8s-auto>

스켈레톤(skeleton) 파일은 아래 파일 사용이 가능.

[https://github.com/tangt64/training\\_memos/tree/main/skt-ansible-project](https://github.com/tangt64/training_memos/tree/main/skt-ansible-project)



# 목적 및 고려사항

# 목적

기존에 kubeadm 명령어로 수동으로 설치를 앤서블 전환  
설치 전/후로 필요한 작업을 앤서블 자동화  
운영에 필요한 추가적인 기능을 앤서블 기반으로 추가 및 활성화

# 추후 고려사항

프로비저닝 및 디폴로이먼트 완료 후 관리 부분  
스케일 업/다운  
특정 기능 제거 그리고 업그레이드

이번 교육에서는 **마스터+워크노드** 구성만 합니다.  
**스케일 업/다운** 그리고 **업그레이드**는 구성하지 않습니다.

# 쿠버네티스 설치 명령어

# 쿠버네티스 설치 명령어

```
# kubeadm init --apiserver-advertise-  
address=192.168.100.100 --cri-  
socket=/var/run/crio/crio.sock
```

```
# mkdir -p -m 0700 ~/.kube/
```

```
# cp /etc/kubernetes/admin.conf ~/.kube/config
```

# 쿠버네티스 설치 명령어

```
# chown -c root:root ~/.kube/config  
# kubectl get nodes  
# kubeadm join 192.168.68.122:6443 --token  
gcd426.2itdtcs7olp7ds6r --discovery-token-ca-  
cert-hash  
sha256:bfad2126496a0779ccbc2cf9b841834cc930384eb  
0158cce40332f74a7b544d7  
# kubectl top nodes
```

**설치 전 OS에서 해야 될 필수 작업**

# OS작업

```
master/node]# cat <<EOF>> /etc/hosts
```

```
192.168.100.100 master.example.com master
```

```
192.168.100.110 node1.example.com node1
```

```
192.168.100.120 node2.example.com node2
```



# OS작업

```
master/node]# cat <<EOF>> /etc/modules.d/k8s-modules.conf
```

```
br_netfilter
```

```
overlay
```

```
EOF
```

# OS작업

```
master/node]# cat <<EOF>> /etc/sysctl.d/99-k8s.conf
```

```
net.bridge.bridge-nf-call-iptables = 1
```

```
net.ipv4.ip_forward = 1
```

```
net.bridge.bridge-nf-call-ip6tables = 1
```

```
EOF
```

```
master/node]# sysctl -f /etc/sysctl.d/99-k8s.conf -p --system
```

# OS작업

```
master/node]# cat <<EOF>> /etc/yum.repos.d/kubernetes.repo
```

```
[kubernetes]
```

```
name=kubernetes repository
```

```
baseurl=https://packages.cloud.google.com/yum/repos/kubernetes-el7-x86_64
```

```
enabled=1
```

```
gpgcheck=1
```

```
repo_gpgcheck=1
```

```
gpgkey=https://packages.cloud.google.com/yum/doc/yum-key.gpg
```

```
https://packages.cloud.google.com/yum/doc/rpm-package-key.gpg
```

```
EOF
```

# OS작업

```
master/node]# yum install kubeadm
```

# OS작업

```
master/node]# cat <<EOF>> /etc/yum.repos.d/devel_kubic_libcontainers_stable_CRIO_1.18_1.18.3.repo

[devel_kubic_libcontainers_stable_CRIO_1.18_1.18.3]

name=Release 1.18.3 (CentOS_7)

type=rpm-md

baseurl=https://download.opensuse.org/repositories/devel:/kubic:/libcontainers:/stable:/CRIO:/1.18
8:/1.18.3/CentOS_7/

gpgcheck=1

gpgkey=https://download.opensuse.org/repositories/devel:/kubic:/libcontainers:/stable:/CRIO:/1.18
:/1.18.3/CentOS_7/repodata/repomd.xml.key

enabled=1

EOF
```

# OS작업

```
master/node]# cat <<EOF>> /etc/yum.repos.d/devel_kubic_libcontainers_stable.repo
```

```
[devel_kubic_libcontainers_stable]
```

```
name=Stable Releases of Upstream github.com/containers packages (CentOS_7)
```

```
type=rpm-md
```

```
baseurl=https://download.opensuse.org/repositories/devel:/kubic:/libcontainers:/stable/CentOS_7/
```

```
gpgcheck=1
```

```
gpgkey=https://download.opensuse.org/repositories/devel:/kubic:/libcontainers:/stable/CentOS_7/repodata/repomd.xml.key
```

```
enabled=1
```

```
EOF
```

# OS작업

```
master/node]# yum install CRIO
```

# OS작업

```
master/node]# cat <<EOF>> /etc/sysconfig/kubelet
```

```
KUBELET_EXTRA_ARGS=--cgroup-driver=systemd --container-runtime-  
endpoint="unix:///var/run/crio/crio.sock"
```

```
EOF
```



# OS작업

```
master/node]# for i in 6443/tcp 23279-2380/tcp 10250/tcp 10251/tcp  
10252/tcp 10255/tcp 300000-32767/tcp 179/tcp ; do firewall-cmd --permanent  
--add-port $i ; done
```

# OS작업

```
master/node]# systemctl stop firewalld
```

```
master/node]# systemctl disable firewalld
```

```
node]# yum install tc
```

```
master/node]# swapoff -a
```

# OS작업

```
master/node]# sed -i '/ swap / s/^/#/' /etc/fstab
```

```
master/node]# sed -i s/^SELINUX=.*$/SELINUX=permissive/ /etc/selinux/config
```

# OS작업

```
# cat <<EOF>> /etc/yum.conf
```

```
> exclude=kubelet*
```

```
> EOF
```

**설치 후 OS에서 해야 될 필수 작업**

# OS작업

리 부팅 이후에도 올바르게 서비스가 동작하는지 확인 필요  
서비스 들어가기 전, 저장소 기반으로 패키지 업데이트

# 작업 순서

# 작업순서

앤서블 작성하기 전 먼저 **role**기반으로 기능을 디렉터리로 구성  
후 다시 역할별로 묶는다.



**작성 시작**

# 공통목표

## CNI

calico

flannel

## RUNTIME

cri-o

공통목표

**extension**

metrics

helm

# 공통목표

**master**

single master

**worker node**

at least two worker nodes

공통목표

**inventory**

[master], [node]

# 참고용

<https://github.com/tangt64/duststack-k8s-auto>

# DAY 1 목표

# DAY 1 목표

**host\_vars, group\_vars**를 통해서 쿠버네티스에서 사용할 변수 구성

인벤토리에서 사용할 서버 대상 분류

쿠버네티스 설치 전에 필요한 OS설정 및 구성



## DAY 2 목표

# DAY 2 목표

쿠버네티스 주요 컴포넌트 설치 및 구성

필요한 기능이 있으면 해당 기능을 roles기반으로 추가





MASTER\_HOSTS\_COUNT: "{{ groups['k8s\_master'] | length }}"



- name: install kubernetes master for the first master node

hosts: k8s\_master[0]

tags: k8s-master-single

roles:

- { role: k8s-master-single, when: k8s\_proxy\_mode == "multi" }

- name: install kubernetes master for the rest of master nodes

hosts: k8s\_master[1:{{ MASTER\_HOSTS\_COUNT }}]

tags: k8s-master-multi

roles:

- { role: k8s-master-ha, when: k8s\_proxy\_mode == "multi" }



- name: install and enable to cri-o,docker enviroment

hosts: k8s\_master:k8s\_node

tags: k8s-prepare

roles:

- { role: k8s-prepare }
- { role: core-crio, when: k8s\_runtime\_environment == "crio" }
- { role: core-containerd, when: k8s\_runtime\_environment == "containerd" }

# 팁

## **YUM vs DNF vs PACKAGE**

특별한 기능이 필요하지 않으면 package명령어 사용 권장.

package 모듈은 배포판에 맞추어서 패키지 관리 명령어를 실행함.

# 팁

## **templates vs files**

자주 바뀌는 설명은 templates기반으로 구성.

고정적이며 내용이 자주 바뀌지 않는 경우 files기반으로 배포



# 팁

**import, include, \_playbook, \_roles**

재활용을 위해서 import, include를 사용해서 최대한 독립성 유지