



{KODE} {LOUD}

[www.kodekloud.com](http://www.kodekloud.com)

cks.kodekloud.com



KODEKLOUD





**Vijin  
Palazhi**



**Mumshad  
Mannambeth**



**CLOUD NATIVE**  
COMPUTING FOUNDATION

{KODE{CLOUD

# Disclaimer

**THE INFORMATION FOUND ON THE WEBSITE, E-LEARNING PLATFORM AND  
WITHIN THE ONLINE COURSES ARE FOR INFORMATIONAL PURPOSES  
ONLY. KODEKLOUD WILL NOT BE HELD RESPONSIBLE FOR ANY DAMAGES  
THAT MAY BE INCURRED BY YOU AS A RESULT OF YOUR USE OF SUCH  
INFORMATION. ALL INFORMATION AND CONTENT ON THE WEBSITE, E-  
LEARNING PLATFORM AND ONLINE COURSE IS COPYRIGHTED, AND MAY NOT  
BE REPUBLISHED, COPIED, SOLD OR POSTED ANYWHERE ONLINE OR IN PRINT.  
KODEKLOUD RESERVES THE RIGHT TO TAKE THE NECESSARY LEGAL ACTION  
TO PREVENT YOU FROM (RE)-PUBLISHING, COPYING, SELLING, POSTING OR  
PRINTING ANY COPYRIGHTED INFORMATION AND CONTENT AVAILABLE ON  
THE WEBSITE, E-LEARNING PLATFORM AND ONLINE COURSE.**

For the full terms & conditions visit [terms.kodekloud.com](https://terms.kodekloud.com)

For questions write to [support@kodekloud.com](mailto:support@kodekloud.com)

# Notice

- This presentation is to refer to course graphics and transcripts only.
- Some of the slides are meant to be animated. So may not be displayed correctly.
- Do not copy and paste command, code or YAML files from this file as it may not be in the right format and may contain hidden characters
- For code refer to the solutions in the lab or the Git repository associated with this course or official Kubernetes documentation pages.
- Some of the code in this deck maybe hidden for brevity

<https://github.com/kodekloudhub/certified-kubernetes-security-specialist-cks-course>

## Course Structure

Lecture

Demos

Quizzes

Hands-on Labs

Slack Channel

Q & A

# Pre-Requisites



**Certified Kubernetes  
Administrator (CKA) with Practice  
Tests**

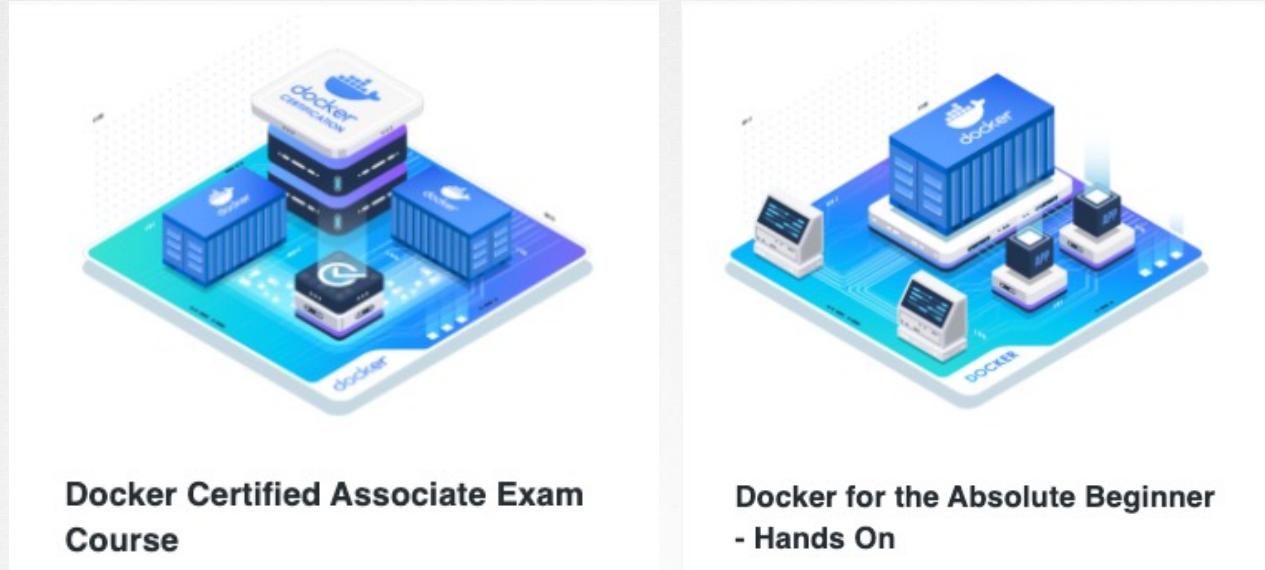
# Pre-Requisites



**Kubernetes for the Absolute  
Beginners - Hands-on**



# Pre-Requisites



# Pre-Requisites



## The Linux Basics Course

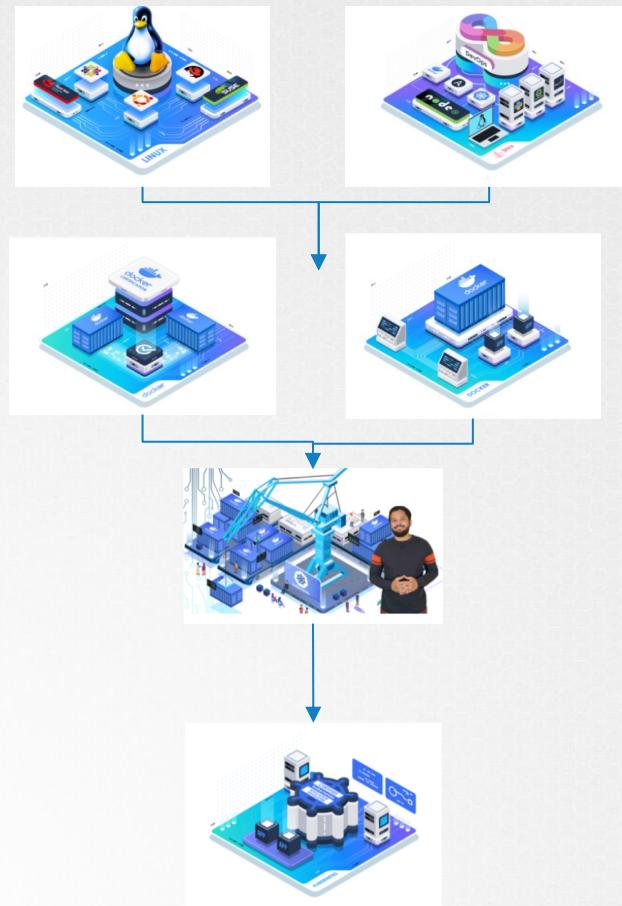
Get your Linux Basics Cleared



## DevOps Pre-Requisite Course

The course you must go through before any DevOps or Cloud Courses

# Pre-Requisites



# Course Objectives

- Understanding Kubernetes Attack Surface
- Cluster Setup & Hardening
- System Hardening
- Minimizing Microservices Vulnerabilities
- Supply Chain Security
- Monitoring, Logging & Runtime Security
- Mock Exams

# Course Objectives

- Understanding Kubernetes Attack Surface

- The Attack!

- The 4C's of Cloud Native Security

- Cluster Setup & Hardening

- System Hardening

- Minimizing Microservices Vulnerabilities

- Supply Chain Security

- Monitoring, Logging & Runtime Security

- Mock Exams

# Course Objectives

Understanding Kubernetes Attack Surface

Cluster Setup & Hardening

CIS Benchmarks

Authentication

Authorization

Service Accounts

TLS in Kubernetes

Protect node metadata and endpoints

Securing the Kubernetes Dashboard

Verifying Platform Binaries

Upgrade Kubernetes Frequently

Network Policies

Securing Ingress

System Hardening

Minimizing Microservices Vulnerabilities

Supply Chain Security

Monitoring, Logging & Runtime Security

Mock Exams

# Course Objectives

Understanding Kubernetes Attack Surface

Cluster Setup & Hardening

System Hardening

- Minimize host OS Footprint

- Limit Node Access

- SSH Hardening

- Privilege Escalation in Linux

- Remove Obsolete Packages & Services

- Restrict Kernel Modules

- Identify and disable open ports

- Minimize IAM Roles

- UFW Firewall Basics

- Kernel Hardening Tools – AppArmor

Minimizing Microservices Vulnerabilities

Supply Chain Security

Monitoring, Logging & Runtime Security

Mock Exams

# Course Objectives

- Understanding Kubernetes Attack Surface
- Cluster Setup & Hardening
- System Hardening
- Minimizing Microservices Vulnerabilities

○ Admission Controllers

○ Pod Security Policies

○ Open Policy Agent

○ Managing Kubernetes Secrets

○ Container Runtime Sandboxes

● ○ Implement Pod to Pod Encryption by use of mTLS

○ Supply Chain Security

○ Monitoring, Logging & Runtime Security

○ Mock Exams

# Course Objectives

- Understanding Kubernetes Attack Surface

- Cluster Setup & Hardening

- System Hardening

- Minimizing Microservices Vulnerabilities

- Supply Chain Security

- Minimize Base Image Footprint

- Image Security

- Secure your supply chain

- Use static analysis of workloads

- Scan images for known vulnerabilities

- Monitoring, Logging & Runtime Security

- Mock Exams

# Course Objectives

Understanding Kubernetes Attack Surface

Cluster Setup & Hardening

System Hardening

Minimizing Microservices Vulnerabilities

Supply Chain Security

Monitoring, Logging & Runtime Security

Detect malicious activities

Detect all phases of attacks

Immutability of containers

Mock Exams

Detect Threats

Perform deep analytical investigation

Use audit logs to monitor access

# Course Objectives

Understanding Kubernetes Attack Surface

Cluster Setup & Hardening

System Hardening

Minimizing Microservices Vulnerabilities

Supply Chain Security

Monitoring, Logging & Runtime Security

Mock Exams

Mock Exam 1

Mock Exam 2

Mock Exam 3

# Hands-on Labs

The screenshot shows a hands-on lab interface with a terminal window. The terminal window has tabs for 'Terminal 1' and 'Assessment Report'. The 'Terminal 1' tab is active, displaying the following command-line session:

```
root@controlplane:~# cd /root
root@controlplane:~# ls
Assessor-CLI cis-cat.zip sample-asset.yaml
root@controlplane:~# cd Assessor-CLI/
root@controlplane:~/Assessor-CLI# ls
Assessor-CLI.bat Assessor-GUI.exe config license sce      third_party_licenses
Assessor-CLI.jar README          custom misc   scripts
Assessor-CLI.sh benchmarks      lib    reports setup
root@controlplane:~/Assessor-CLI# sh ./Assessor-CLI.sh -i -rd /var/www/html/ -nts -rp index
-----
,0888880. 8888 d8888880. ,0888880. 8. 8888888888888888
8888 `88. 8888 .`888: ' `88. 8888 `88. .88. 888
,88888 `8. 8888 8.`8888. Y8 ,88888 `8. .8888. 888
8888888 8888 `8.`8888. 8888888888888888
8888888 8888 `8.`8888. 8888888888888888
8888888 8888 `8.`8888. 8888888888888888
8888888 8888 `8.`8888. 8888888888888888
`88888 .8`8888 8b `8.`8888. `88888 .8`8888. 8888888888888888
8888 ,88`8888 `8b. ;8.`8888 8888 ,88`88888888888888888888888888
`8888888P` 8888 `Y88888P ,88P` `8888888P` .8`88888888888888888888888888
```

Below the terminal output, there is a decorative graphic consisting of a grid of characters: commas, zeros, ones, and underscores.

The 'Assessment Report' tab is visible at the top right of the terminal window. The overall interface includes tabs for 'Task', 'Hint', and 'Solution' at the top left, and a timer showing '57:10' at the top center.

# Mock Exams

Task Hint Solution ⏱ 57:10

LI  
h -i -rd /var/www/html/ -nts -rp index

Benchmarks/Data-Stream Collections: CIS

Ubuntu Linux 18.04 LTS Benchmark  
v2.0.1

Profile: Level 1 - Server

Terminal 1 +

root@controlplane:~# cd /root  
root@controlplane:~# ls  
**Assessor-CLI cis-cat.zip sample-asset.yaml**  
root@controlplane:~# cd Assessor-CLI/  
root@controlplane:~/Assessor-CLI# ls  
Assessor-CLI.bat Assessor-GUI.exe config license sce third\_party\_licenses  
**Assessor-CLI.jar README custom misc scripts**  
**Assessor-CLI.sh benchmarks lib reports setup**  
root@controlplane:~/Assessor-CLI# sh ./Assessor-CLI.sh -i -rd /var/www/html/ -nts -rp index

---

```
,o888888o.    8888    d8888888o.        ,o888888o.      8.    88888888888888888888
 8888   '88.  8888  .`8888: ' `88.    8888   '88.    .88.    8888
,88888   `8. 8888  8. `8888.  Y8     ,88888   `8.    .8888.    8888
888888   8888  `8. `8888.    88888888888888888888
888888   8888  `8. `8888.    88888888888888888888
888888   8888  `8. `8888.    88888888888888888888
888888   8888  `8. `8888.    88888888888888888888
888888   8888  `8. `8888.    88888888888888888888
`88888   .8' 8888 8b.  `8. `8888.    `88888888888888888888
 8888   ,88' 8888  `8b. ;8. `8888     8888   ,88' .88888888888888888888
   `8888888P' 8888  `Y88888P ,88P'    `88888888P' .8'    `8. `88888. 8888
```

---

Welcome to CIS-CAT Pro Assessor; built on 01/28/2021 02:03 AM

---

This is the Center for Internet Security Configuration Assessment Tool, v4.3.1  
At any time during the selection process, enter 'q!' to exit.

---

Verifying application



{KODE} {LOUD}

[www.kodekloud.com](http://www.kodekloud.com)



<https://www.cncf.io/certification/cks/>

The Certified Kubernetes Security Specialist (CKS) program was created by the Cloud Native Computing Foundation (CNCF), in collaboration with The Linux Foundation, to help develop the Kubernetes ecosystem.

As one of the highest velocity projects in the history of open source, Kubernetes use is exploding. The Cloud Native Computing Foundation is committed to growing the community of Kubernetes-knowledgeable security specialists, thereby enabling continued growth across the broad set of organizations using the technology.

Certification is a key step in that process, allowing certified security specialists to quickly establish their credibility and value in the job market, and also allowing companies to more quickly hire high-quality teams to support their growth.

Cost \$300 | Online Exam

[REGISTER FOR EXAM ↗](#)

[TRAIN FOR EXAM ↗](#)

考试费 ¥2088 (含税) | 中文监考官

[中文官方考试 ↗](#)

[中国官网课程学习 ↗](#)



# Requirement



# Hands-on Exam

[Kubernetes Documentation](#) / [Home](#)

# Kubernetes Documentation

Kubernetes is an open source container orchestration engine for automating deployment, scaling, and management of containerized applications. The open source project is hosted by the Cloud Native Computing Foundation ([CNCF](#)).

## Understand the basics

Learn about Kubernetes and its fundamental concepts.

[What is Kubernetes?](#)[Kubernetes Components](#)[The Kubernetes API](#)[Understanding Kubernetes Objects](#)[Pods](#)[Learn Concepts](#)

## Try Kubernetes

Follow tutorials to learn how to deploy applications in Kubernetes.

[Hello Minikube](#)[Walkthrough the basics](#)[Stateless Example: PHP Guestbook with MongoDB](#)[Stateful Example: Wordpress with Persistent Volumes](#)[View Tutorials](#)

## Set up a K8s cluster

Get Kubernetes running based on your resources and needs.

[Install the kubeadm setup tool](#)[Learning environment](#)[Production environment](#)[Set up Kubernetes](#)

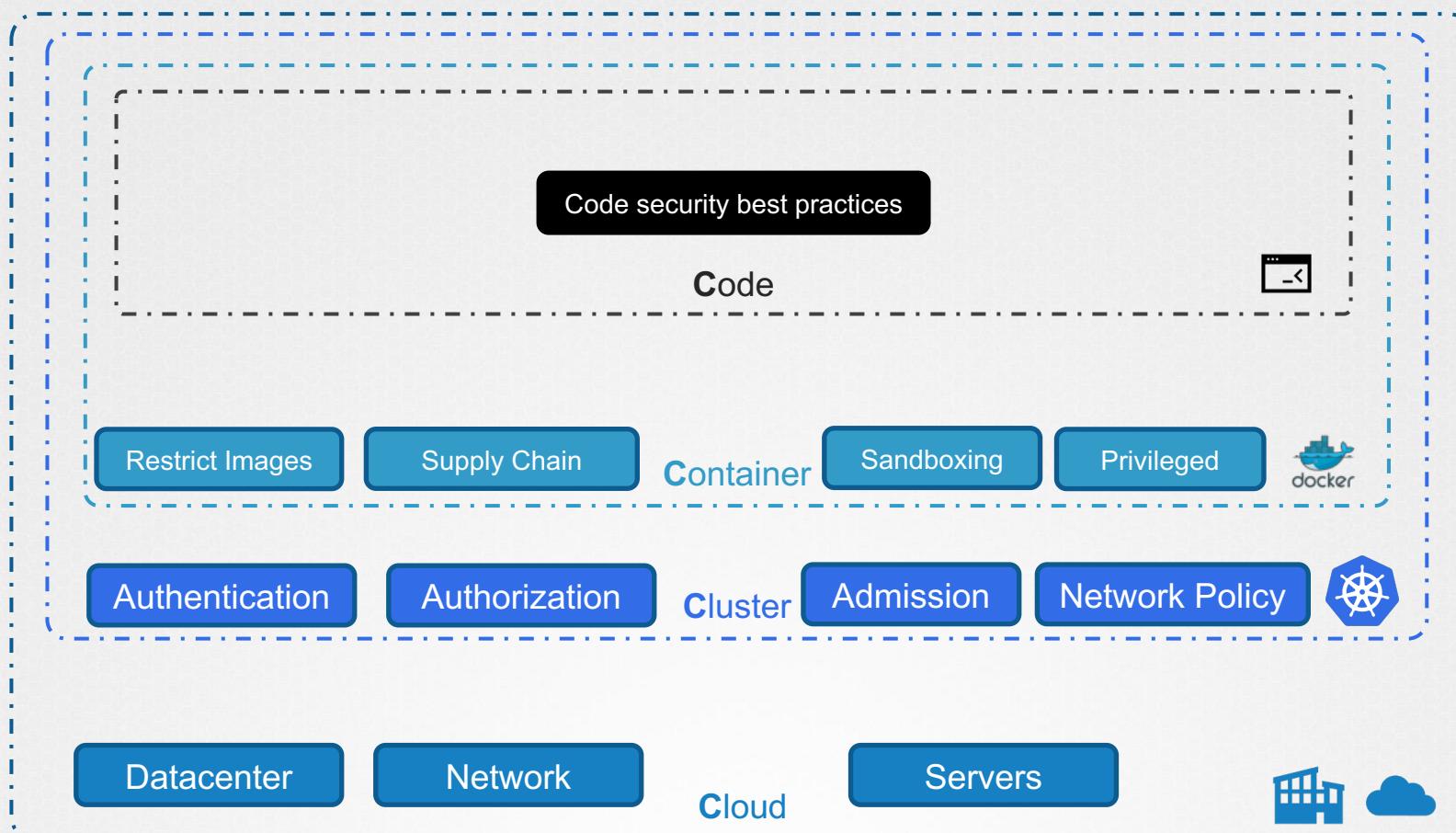


{KODE} {LOUD}

[www.kodekloud.com](http://www.kodekloud.com)

# The 4 C's of Cloud Native Security!







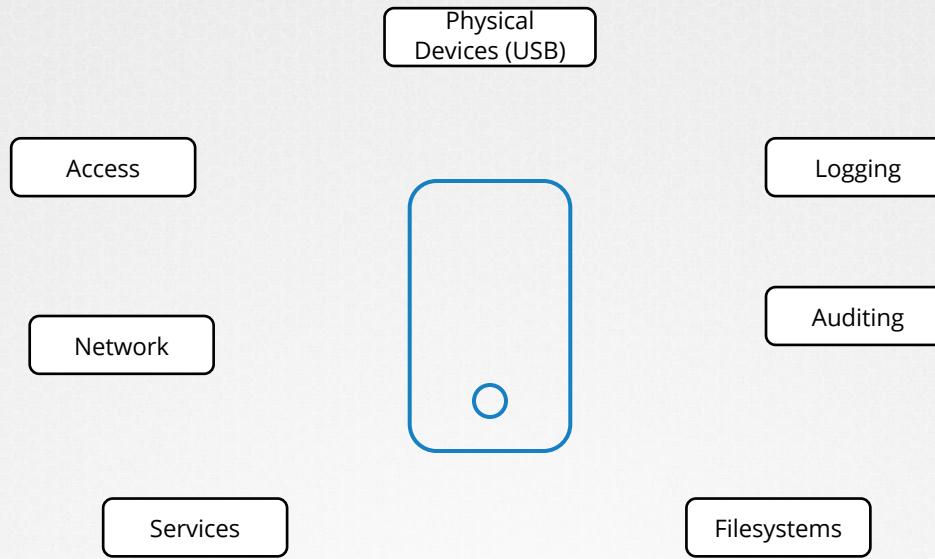
{KODE} {LOUD}

[www.kodekloud.com](http://www.kodekloud.com)

# CIS Benchmarks



# Security Benchmark





# Center for Internet Security®

*Our mission is to make the connected world a safer place by developing, validating, and promoting timely best practice solutions that help people, businesses, and governments protect themselves against pervasive cyber threats.*



OS	Cloud	Mobile	Network	Desktop	Server
Linux	Google	Apple IOS	Check Point	Web Browsers	Tomcat
Windows	Azure	Android	Cisco	MS Office	Docker
MacOS	AWS		Juniper	Zoom	Kubernetes
More..	More..		Palo Alto Networks	More..	More..



## Table of Contents

Terms of Use .....	1
Overview .....	13
Intended Audience .....	13
Consensus Guidance.....	13
Typographical Conventions .....	15
Scoring Information .....	15
Profile Definitions .....	16
Acknowledgements .....	17
Recommendations .....	19
1 Initial Setup.....	20
1.1 Filesystem Configuration .....	21
1.1.1 Disable unused filesystems.....	22
1.1.1.1 Ensure mounting of cramfs filesystems is disabled (Scored) .....	23
1.1.1.2 Ensure mounting of freevxfs filesystems is disabled (Scored).....	25
1.1.1.3 Ensure mounting of jffs2 filesystems is disabled (Scored) .....	27



### 1.1.23 Disable USB Storage (Scored)

#### Profile Applicability:

- Level 1 - Server
- Level 2 - Workstation

#### Description:

USB storage provides a means to transfer and store files insuring persistence and availability of the files independent of network connection status. Its popularity and utility has led to USB-based malware being a simple and common means for network infiltration and a first step to establishing a persistent threat within a networked environment.

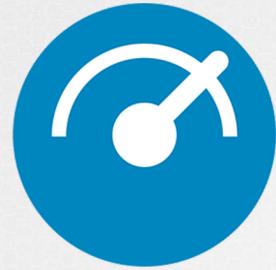
#### Rationale:

Restricting USB access on the system will decrease the physical attack surface for a device and diminish the possible vectors to introduce malware.

#### Audit:

Run the following commands and verify the output is as indicated:

```
# modprobe -n -v usb-storage
```



**CIS-CAT<sup>®</sup>Lite**



## Summary

Description	Tests					Scoring		
	Pass	Fail	Error	Unkn.	Man.	Score	Max	Percent
<b>1 Initial Setup</b>	20	23	0	0	7	20.0	43.0	47%
1.1 Filesystem Configuration	10	11	0	0	3	10.0	21.0	48%
1.1.1 Disable unused filesystems	0	7	0	0	0	0.0	7.0	0%
1.2 Configure Software Updates	0	0	0	0	2	0.0	0.0	0%
1.3 Configure sudo	1	2	0	0	0	1.0	3.0	33%
1.4 Filesystem Integrity Checking	0	2	0	0	0	0.0	2.0	0%
1.5 Secure Boot Settings	0	3	0	0	1	0.0	3.0	0%
1.6 Additional Process Hardening	2	2	0	0	0	2.0	4.0	50%
1.7 Mandatory Access Control	2	1	0	0	0	2.0	3.0	67%
1.7.1 Configure AppArmor	2	1	0	0	0	2.0	3.0	67%
1.8 Warning Banners	5	2	0	0	0	5.0	7.0	71%
1.8.1 Command Line Warning Banners	4	2	0	0	0	4.0	6.0	67%
<b>2 Services</b>	24	2	0	0	1	24.0	26.0	92%
2.1 inetd Services	2	0	0	0	0	2.0	2.0	100%
2.2 Special Purpose Services	18	1	0	0	1	18.0	19.0	95%
2.2.1 Time Synchronization	3	0	0	0	1	3.0	3.0	100%
2.3 Service Clients	4	1	0	0	0	4.0	5.0	80%



## Assessment Results

[Display Failures Only](#)

w	Benchmark Item	Result
1	<a href="#">Initial Setup</a>	
1.0	<a href="#">1.1 Filesystem Configuration</a>	
1.0	<a href="#">1.1.1 Disable unused filesystems</a>	
1.0	<a href="#">1.1.1.1 Ensure mounting of cramfs filesystems is disabled</a>	<a href="#">Fail</a>
1.0	<a href="#">1.1.1.2 Ensure mounting of freevxfs filesystems is disabled</a>	<a href="#">Fail</a>
1.0	<a href="#">1.1.1.3 Ensure mounting of jffs2 filesystems is disabled</a>	<a href="#">Fail</a>
1.0	<a href="#">1.1.1.4 Ensure mounting of hfs filesystems is disabled</a>	<a href="#">Fail</a>
1.0	<a href="#">1.1.1.5 Ensure mounting of hfsplus filesystems is disabled</a>	<a href="#">Fail</a>
1.0	<a href="#">1.1.1.6 Ensure mounting of squashfs filesystems is disabled</a>	<a href="#">Fail</a>
1.0	<a href="#">1.1.1.7 Ensure mounting of udf filesystems is disabled</a>	<a href="#">Fail</a>
1.0	<a href="#">1.1.2 Ensure /tmp is configured</a>	<a href="#">Fail</a>
1.0	<a href="#">1.1.3 Ensure nodev option set on /tmp partition</a>	<a href="#">Pass</a>
1.0	<a href="#">1.1.4 Ensure nosuid option set on /tmp partition</a>	<a href="#">Pass</a>
1.0	<a href="#">1.1.5 Ensure noexec option set on /tmp partition</a>	<a href="#">Pass</a>
1.0	<a href="#">1.1.8 Ensure nodev option set on /var/tmp partition</a>	<a href="#">Pass</a>
1.0	<a href="#">1.1.9 Ensure nosuid option set on /var/tmp partition</a>	<a href="#">Pass</a>



[cks.kodekloud.com](https://cks.kodekloud.com)



{KODE} {LOUD}

[www.kodekloud.com](http://www.kodekloud.com)

# CIS Benchmarks - Kubernetes





OS	Cloud	Mobile	Network	Desktop	Server
Linux	Google	Apple IOS	Check Point	Web Browsers	Tomcat
Windows	Azure	Android	Cisco	MS Office	Docker
MacOS	AWS		Juniper	Zoom	Kubernetes
More..	More..		Palo Alto Networks	More..	More..



## Overview

This document provides prescriptive guidance for establishing a secure configuration posture for Kubernetes 1.16 - 1.18. To obtain the latest version of this guide, please visit [www.cisecurity.org](http://www.cisecurity.org). If you have questions, comments, or have identified ways to improve this guide, please write us at [support@cisecurity.org](mailto:support@cisecurity.org).

**\*\*Special Note:** \*\*The set of configuration files mentioned anywhere throughout this benchmark document may vary according to the deployment tool and the platform. Any reference to a configuration file should be modified according to the actual configuration files used on the specific deployment.

For example, the configuration file for the Kubernetes API server installed by the `kubeadm` tool may be found in `/etc/kubernetes/manifests/kube-apiserver.yaml`, but the same file may be called `/etc/kubernetes/manifests/kube-apiserver.manifest` when installed by `kops` or `kubespray`.

## Intended Audience

This document is intended for system and application administrators, security specialists, auditors, help desk, and platform deployment personnel who plan to develop, deploy, assess, or secure solutions that incorporate Kubernetes 1.16 - 1.18.



Recommendations.....	15
1 Control Plane Components .....	15
1.1 Master Node Configuration Files .....	16
1.1.1 Ensure that the API server pod specification file permissions are set to 644 or more restrictive (Automated).....	16
1.1.2 Ensure that the API server pod specification file ownership is set to root:root (Automated) .....	18
1.1.3 Ensure that the controller manager pod specification file permissions are set to 644 or more restrictive (Automated) .....	20
1.1.4 Ensure that the controller manager pod specification file ownership is set to root:root (Automated) .....	22
1.1.5 Ensure that the scheduler pod specification file permissions are set to 644 or more restrictive (Automated) .....	24
1.1.6 Ensure that the scheduler pod specification file ownership is set to root:root (Automated) .....	26
1.1.7 Ensure that the etcd pod specification file permissions are set to 644 or more restrictive (Automated) .....	28



## 1.1 Master Node Configuration Files

1.1.1 Ensure that the API server pod specification file permissions are set to 644 or more restrictive (Automated)

### Profile Applicability:

- Level 1 - Master Node

### Description:

Ensure that the API server pod specification file has permissions of 644 or more restrictive.

### Rationale:

The API server pod specification file controls various parameters that set the behavior of the API server. You should restrict its file permissions to maintain the integrity of the file. The file should be writable by only the administrators on the system.

### Audit:

Run the below command (based on the file location on your system) on the master node.  
For example,

```
stat -c %a /etc/kubernetes/manifests/kube-apiserver.yaml
```

Verify that the permissions are 644 or more restrictive.

### Remediation:

Run the below command (based on the file location on your system) on the master node.  
For example,

```
chmod 644 /etc/kubernetes/manifests/kube-apiserver.yaml
```



1.2 API Server .....	58
1.2.1 Ensure that the --anonymous-auth argument is set to false (Manual).....	58
1.2.2 Ensure that the --basic-auth-file argument is not set (Automated).....	60
1.2.3 Ensure that the --token-auth-file parameter is not set (Automated).....	62
1.2.4 Ensure that the --kubelet-https argument is set to true (Automated) .....	64
1.2.5 Ensure that the --kubelet-client-certificate and --kubelet-client-key arguments are set as appropriate (Automated) .....	66
1.2.6 Ensure that the --kubelet-certificate-authority argument is set as appropriate (Automated) .....	68
1.2.7 Ensure that the --authorization-mode argument is not set to AlwaysAllow (Automated) .....	70



- Windows 10
- Ubuntu
- Google Chrome
- Mac OS



CIS Benchmark	CIS-CAT Pro Assessor v3:	CIS-CAT Pro Assessor v4:
Kubernetes v1.6.1		✓

<https://www.cisecurity.org/cybersecurity-tools/cis-cat-pro/cis-benchmarks-supported-by-cis-cat-pro/>



{KODE} {LOUD}

[www.kodekloud.com](http://www.kodekloud.com)

# kube-bench





Acqua  
Security



**kube-bench**



Recommendations.....	15
1 Control Plane Components .....	15
1.1 Master Node Configuration Files .....	16
1.1.1 Ensure that the API server pod specification file permissions are set to 644 or more restrictive (Automated).....	16
1.1.2 Ensure that the API server pod specification file ownership is set to root:root (Automated) .....	18
1.1.3 Ensure that the controller manager pod specification file permissions are set to 644 or more restrictive (Automated) .....	20
1.1.4 Ensure that the controller manager pod specification file ownership is set to root:root (Automated) .....	22
1.1.5 Ensure that the scheduler pod specification file permissions are set to 644 or more restrictive (Automated) .....	24
1.1.6 Ensure that the scheduler pod specification file ownership is set to root:root (Automated) .....	26
1.1.7 Ensure that the etcd pod specification file permissions are set to 644 or more restrictive (Automated) .....	28

```

[INFO] 1 Master Node Security Configuration
[INFO] 1.1 API Server
[FAIL] 1.1.1 Ensure that the --allow-privileged argument is set to false
[FAIL] 1.1.2 Ensure that the --anonymous-auth argument is set to false
[PASS] 1.1.3 Ensure that the --basic-auth-file argument is not set
[PASS] 1.1.4 Ensure that the --insecure-allow-any-token argument is not set
[FAIL] 1.1.5 Ensure that the --kubelet-https argument is set to false
[PASS] 1.1.6 Ensure that the --insecure-bind-address argument is not set
[PASS] 1.1.7 Ensure that the --insecure-port argument is set to 0
[PASS] 1.1.8 Ensure that the --secure-port argument is not set to 0
[FAIL] 1.1.9 Ensure that the --profiling argument is set to false
[FAIL] 1.1.10 Ensure that the --repair-malformed-updates argument is not set
[PASS] 1.1.11 Ensure that the admission control policy is not set to None
[FAIL] 1.1.12 Ensure that the admission control policy is set to None
[FAIL] 1.1.13 Ensure that the admission control policy is set to None
[FAIL] 1.1.14 Ensure that the admission control policy is set to None
[PASS] 1.1.15 Ensure that the admission control policy is set to None
[FAIL] 1.1.16 Ensure that the --audit-log-path argument is set to /var/log/pods
[FAIL] 1.1.17 Ensure that the --audit-log-maxage argument is set to 0
[FAIL] 1.1.18 Ensure that the --audit-log-maxbackup argument is set to 0
[FAIL] 1.1.19 Ensure that the --audit-log-maxsize argument is set to 10485760
[PASS] 1.1.20 Ensure that the --authorization-mode argument is set to 'RBAC'
[PASS] 1.1.21 Ensure that the --token-auth-file parameter is not set
[FAIL] 1.1.22 Ensure that the --kubelet-certificate-authority argument is not set
  
```

# | Deploy Kube-bench



- Deploy as a Docker Container
- Deploy as a POD in a Kubernetes cluster
- Install kube-bench binaries
- Compile from source

# Labs



- Go to kube-bench github page
- Identify latest release binary
- Install kube-bench on master node
- Run assessment and review results
- Fix issues
  - 1.1.9
  - 1.1.10
  - 1.1.19
  - 1.1.20

[cks.kodekloud.com](https://github.com/aquasecurity/kube-bench)

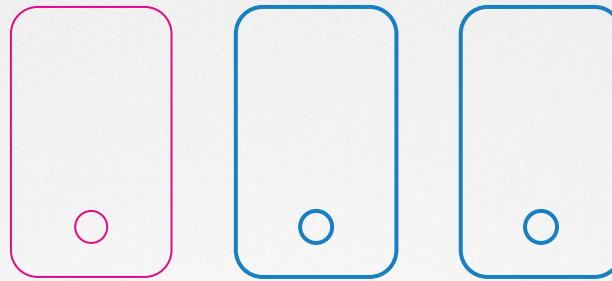


{KODE} {LOUD}

[www.kodekloud.com](http://www.kodekloud.com)

# SECURITY PRIMITIVES

# Secure Hosts



- Password based authentication disabled
- SSH Key based authentication

# Secure Kubernetes

**kube-apiserver**

# Secure Kubernetes

kube-apiserver

Who can access?

What can they do?

# Authentication

## Who can access?

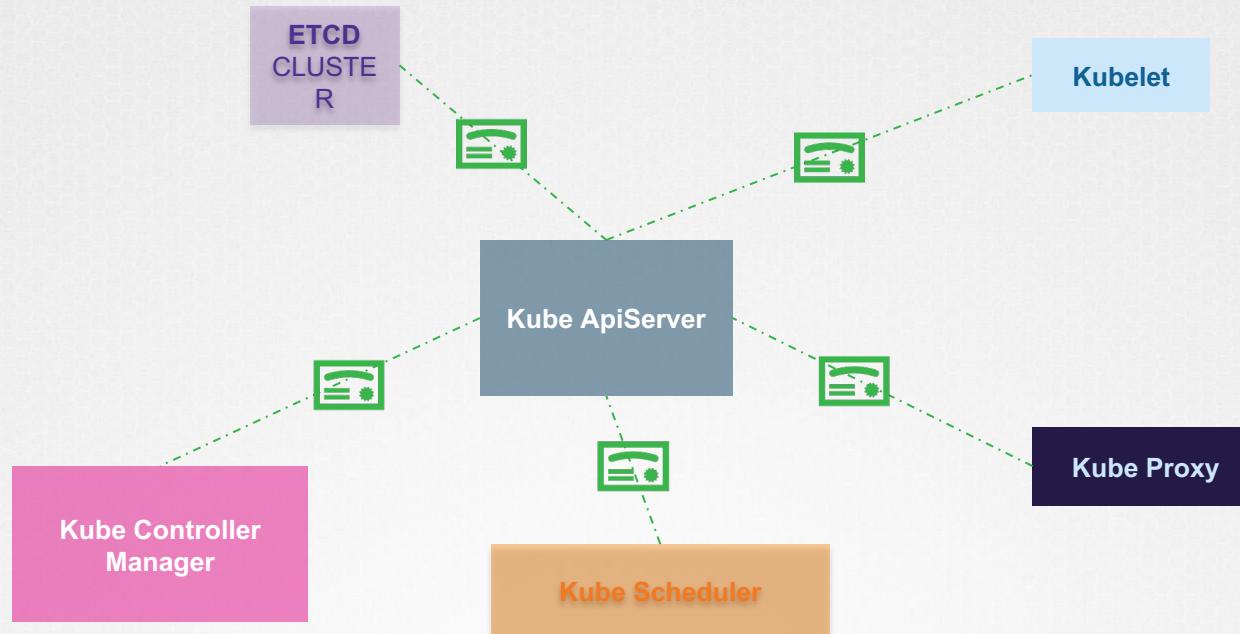
- Files – Username and Passwords
- Files – Username and Tokens
- Certificates
- External Authentication providers - LDAP
- Service Accounts

# Authorization

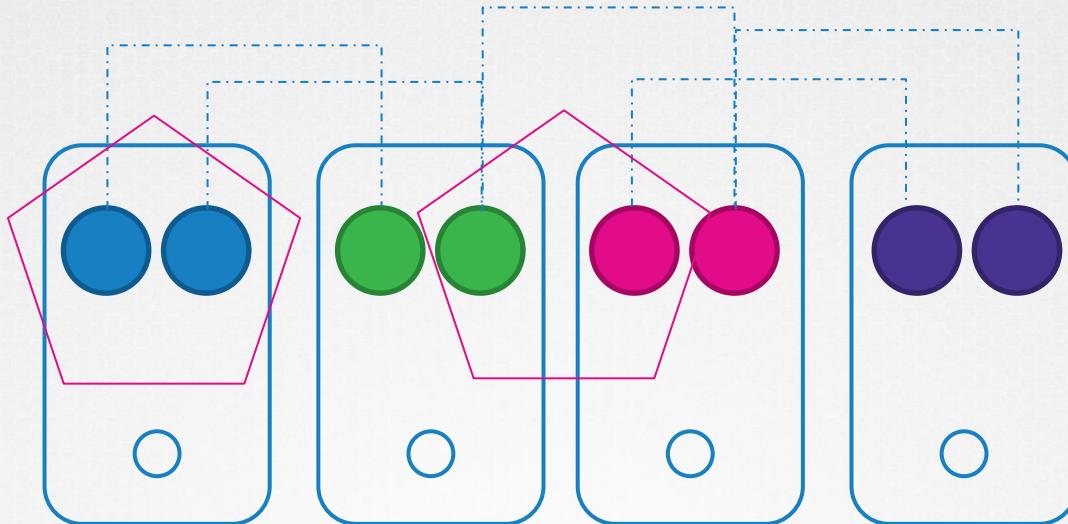
## What can they do?

- RBAC Authorization
- ABAC Authorization
- Node Authorization
- Webhook Mode

# TLS Certificates



# Network Policies



# References

<https://kubernetes.io/docs/tasks/administer-cluster/securing-a-cluster/>

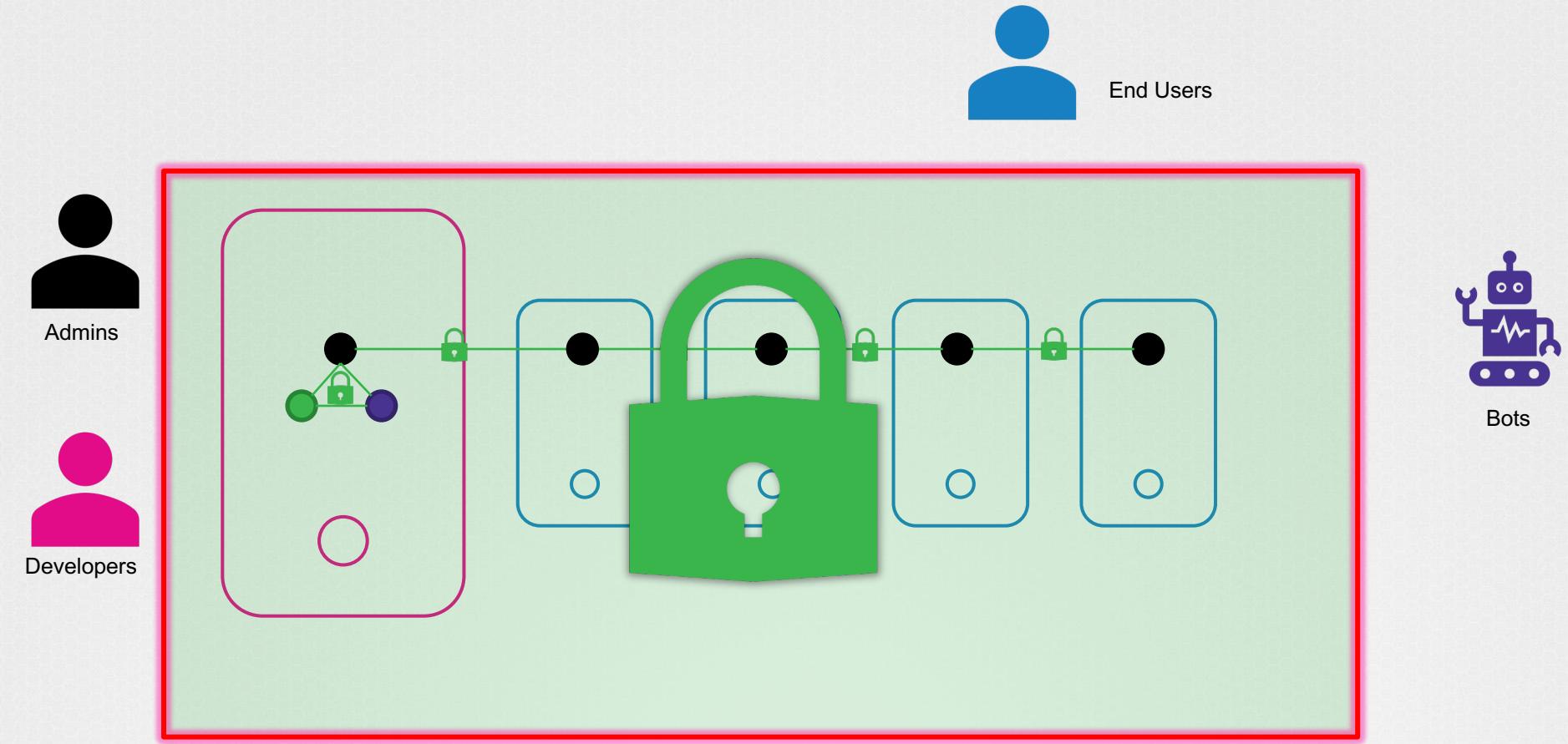


{KODE} {LOUD}

[www.kodekloud.com](http://www.kodekloud.com)

# AUTHENTICATION



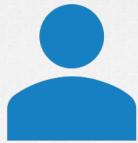




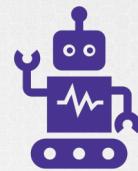
Admins



Developers



Application  
End Users



Bots

# Accounts



Admins



Developers

User



Bots

Service Accounts

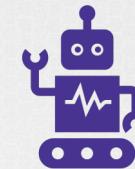
# Accounts



Admins



Developers



Bots

## User

```
▶ kubectl create user user1  
user1 Created
```



```
▶ kubectl list users  
Username  
user1  
user2
```



## Service Accounts

```
▶ kubectl create serviceaccount sa1  
Service Account sa1 Created
```



```
▶ kubectl get serviceaccount  
ServiceAccount  
sa1
```



# Accounts

User



Admins



Developers

▶ kubectl

▶ curl https://kube-server-ip:6443/

Authenticate User

1

kube-apiserver

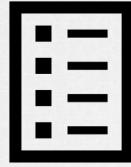
2

Process Request

# Auth Mechanisms

kube-apiserver

Static Password File



Static Token File



Certificates



Identity Services



# Auth Mechanisms

kube-apiserver

Static Password File

Static Token File



# Auth Mechanisms - Basic

kube-apiserver

--basic-auth-file=user-details.csv

user-details.csv

```
password123,user1,u0001
password123,user2,u0002
password123,user3,u0003
password123,user4,u0004
password123,user5,u0005
```

kube-apiserver.service

```
ExecStart=/usr/local/bin/kube-apiserver \\
--advertise-address=${INTERNAL_IP} \\
--allow-privileged=true \\
--apiserver-count=3 \\
--authorization-mode=Node,RBAC \\
--bind-address=0.0.0.0 \\
--enable-swagger-ui=true \\
--etcd-servers=https://127.0.0.1:2379 \\
--event-ttl=1h \\
--runtime-config=api/all \\
--service-cluster-ip-range=10.32.0.0/24 \\
--service-node-port-range=30000-32767 \\
--v=2
```

Note: Showing fewer options for simplicity

# Kube-api Server Configuration

## kube-apiserver.service

```
ExecStart=/usr/local/bin/kube-apiserver \
--advertise-address=${INTERNAL_IP} \
--allow-privileged=true \
--apiserver-count=3 \
--authorization-mode=Node,RBAC \
--bind-address=0.0.0.0 \
--enable-swagger-ui=true \
--etcd-servers=https://127.0.0.1:2379 \
--event-ttl=1h \
--runtime-config=api/all \
--service-cluster-ip-range=10.32.0.0/24 \
--service-node-port-range=30000-32767 \
--v=2
--basic-auth-file=user-details.csv
```

## /etc/kubernetes/manifests/kube-apiserver.yaml

```
apiVersion: v1
kind: Pod
metadata:
  creationTimestamp: null
  name: kube-apiserver
  namespace: kube-system
spec:
  containers:
  - command:
    - kube-apiserver
    - --authorization-mode=Node, RBAC
    - --advertise-address=172.17.0.107
    - --allow-privileged=true
    - --enable-admission-plugins=NodeRestriction
    - --enable-bootstrap-token-auth=true
    image: k8s.gcr.io/kube-apiserver-amd64:v1.11.3
    name: kube-apiserver
```

Note: Showing fewer options for simplicity

# Authenticate User

```
curl -v -k https://master-node-ip:6443/api/v1/pods -u "user1:password123"
```

```
{  
    "kind": "PodList",  
    "apiVersion": "v1",  
    "metadata": {  
        "selfLink": "/api/v1/pods",  
        "resourceVersion": "3594"  
    },  
    "items": [  
        {  
            "metadata": {  
                "name": "nginx-64f497f8fd-krkg6",  
                "generateName": "nginx-64f497f8fd-",  
                "namespace": "default",  
                "selfLink": "/api/v1/namespaces/default/pods/nginx-64f497f8fd-krkg6",  
                "uid": "77dd7dfb-2914-11e9-b468-0242ac11006b",  
                "resourceVersion": "3569",  
                "creationTimestamp": "2019-02-05T07:05:49Z",  
                "labels": {  
                    "pod-template-hash": "2090539498",  
                    "run": "nginx"  
                }  
            }  
        }  
    ]  
}
```

# Auth Mechanisms - Basic

Static Password File

user-details.csv

```
password123,user1,u0001,group1  
password123,user2,u0002,group1  
password123,user3,u0003,group2  
password123,user4,u0004,group2  
password123,user5,u0005,group2
```

Static Token File

user-token-details.csv

```
KpjCVbI7rCFAHYPkByTIzRb7gu1cUc4B,user10,u0010,group1  
rJjncHmvtxHc6M1WQddhtvNyyhgTdxSC,user11,u0011,group1  
mjpOFIEiFOKL9toiKaRNtt59ePtczZSq,user12,u0012,group2  
PG41IXhs7QjqwWkmBkvgGT9gloYUqZij,user13,u0013,group2
```

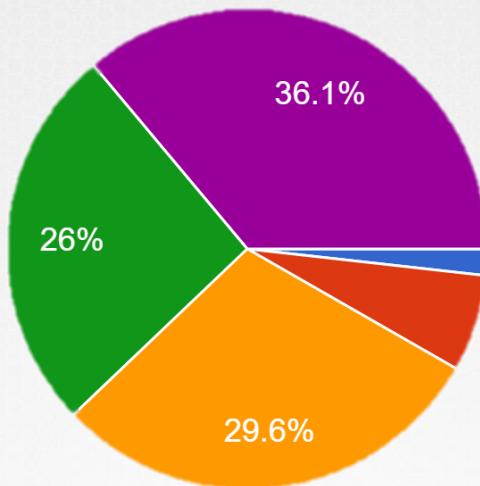
--token-auth-file=user-details.csv

```
curl -v -k https://master-node-ip:6443/api/v1/pods --header "Authorization: Bearer KpjCVbI7rCFAHYPkBzRb7gu1cUc4B"
```

Note: Showing fewer options for simplicity

# Note

- This is not a recommended authentication mechanism
- Consider volume mount while providing the auth file in a kubeadm setup
- Setup Role Based Authorization for the new users



No Clue

Not very  
Comfortable

# Goals!

- ❑ What are TLS Certificates?
- ❑ How does Kubernetes use Certificates?
- ❑ How to generate them?
- ❑ How to configure them?
- ❑ How to view them?
- ❑ How to troubleshoot issues related to Certificates

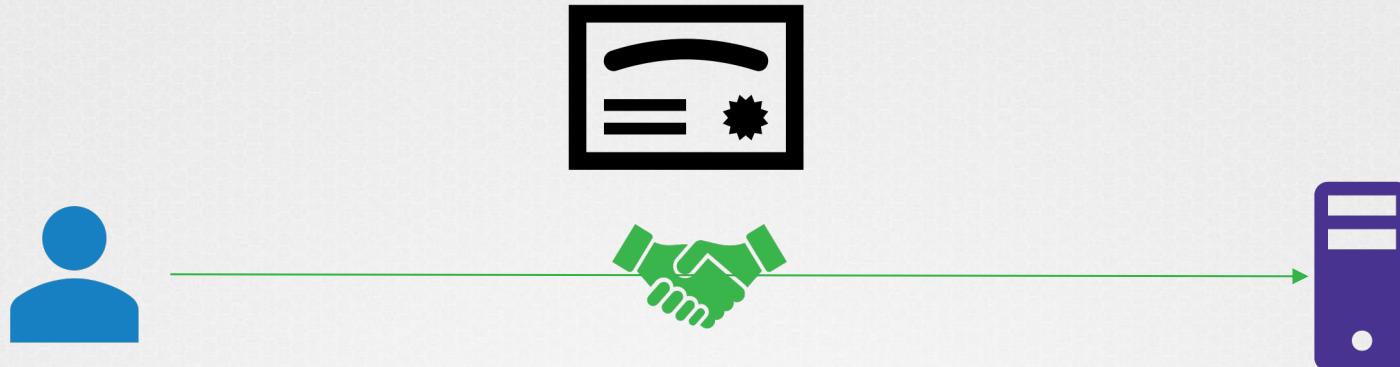


{KODE} {LOUD}

[www.kodekloud.com](http://www.kodekloud.com)

# TLS CERTIFICATES (PRE-REQ)





User: John  
Password: Pass123

**User:** John  
**Password:** Pass123



<http://my-bank.com>



XCVB: DKSJD  
User: John  
LKJSDFK  
Password: Pass123  
XZKJSDLF

XCVB: DKSJD  
LKJSDFK:  
XZKJSDLF  
XKSDJ39K34KJSD  
F0934JHSDFSDF3  
DKSDG



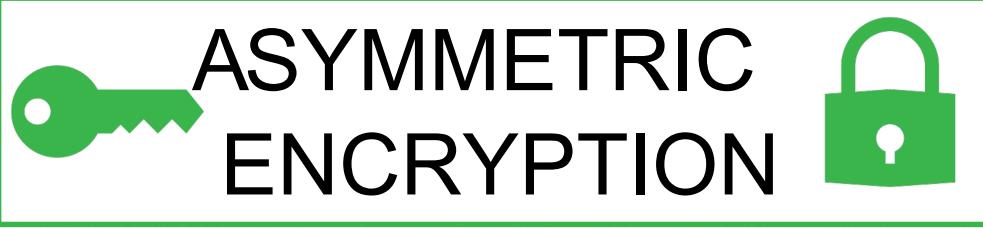
<http://my-bank.com>



**SYMMETRIC  
ENCRYPTION**



# SYMMETRIC ENCRYPTION



Private  
Key

Public Key

# ASYMMETRIC ENCRYPTION



Private  
Key



Public  
Lock



**XCVB: DKSJD  
LKJSDFK:  
XZKJSDLF**

# A SYMMETRIC ENCRYPTION SSH

```
▶ ssh-keygen
```

```
id_rsa  id_rsa.pub
```



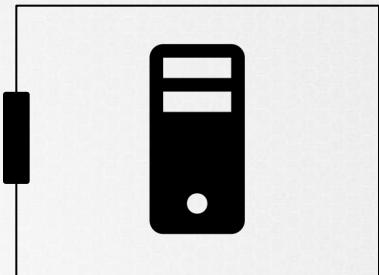
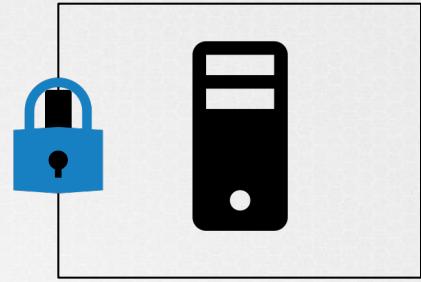
Private  
Key



Public Key

```
▶ cat ~/.ssh/authorized_keys
```

```
ssh-rsa AAAAB3NzaC1yc...KhtUBfoTzlBqR  
V1NThvOo4opzEwRQo1mWx user1
```



```
▶ ssh -i id_rsa user1@server1
```

```
Successfully Logged In!
```

# A SYMMETRIC ENCRYPTION SSH

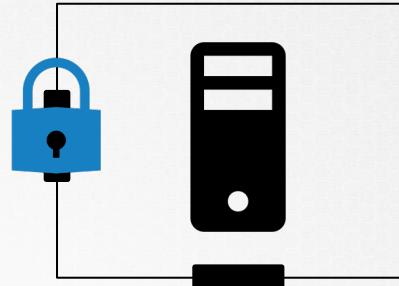
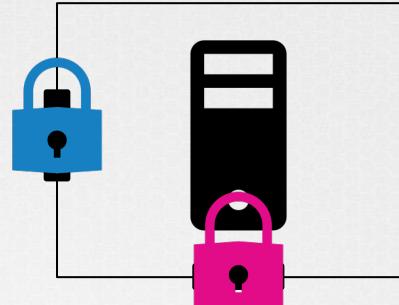


Private  
Key



Private Key

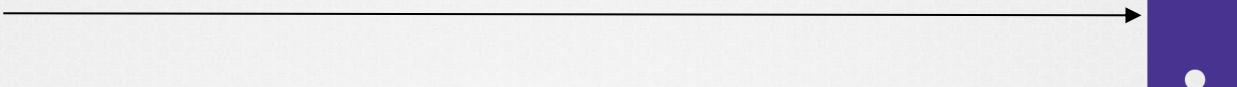
Public Lock



▶ cat ~/.ssh/authorized\_keys

```
ssh-rsa AAAAB3NzaC1yc...KhtUBfoTz1BqRV1NThv0o4opzEwRQo1mWx user1
```

```
ssh-rsa AAAXCVJSDFDF...SLKJSDLKFw23423xckjSDFDFLKJLSDFKJLx user2
```



User: John  
Password:  
Pass123

XCVB: DKSJD  
LKJSDFK:  
XZKJSDLF

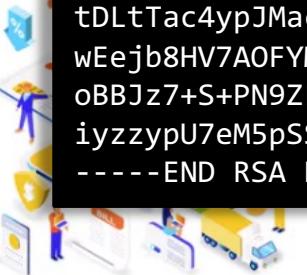


<https://my...>

Online  
banking

Get Started

Log in  
Forgot password  
Create account  
Privacy policy  
Terms & conditions  
Help



-----BEGIN RSA PRIVATE KEY-----

MIICXAIBAAKBgQDkwiLGQAgAN1HpEoLUaqKYiYJIk9wetzotW2/w4nsGhonuWGrTd7+823xd8FDH+WJLqXsTDkrpKNG3sh67dHRGGipKcEXfZnzT5yDyK/jA6uQvAzl+I4xNNqtwKDC03uoLpnMI

AoGBAOJF6VHCGrmfkUGBluhvj4MFIj5WsyIpStwal e0k1XibMPLLAXtig7ao PhXWc9n+L10ch+Fah4z! jNWp8X8sKQJBAOWECwzl h5PDUtoPAcc4Gzgz8Bzl tDLtTac4ypJMacbgs/r wEejb8HV7AOFYMA5Awln oBBJz7+S+PN9ZL9pDDE(

-----BEGIN PUBLIC KEY-----

MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCB iQKBgQDkwiLGQAgAN1HpEoLUaqKYiYJI k9wetzotW2/w4nsGhonuWGrTd7+823xd8FDH+WJLqXsTDkrpKNG3sh67dHRGGipK cEXfZnzT5yDyK/jA6uQvAzl+I4xNNqtw

KDC03uoLpnMEsayPhNtexosfScu1KXe0 L6/nTkn9Gc/YoUWzgQIDAQAB

-----END PUBLIC KEY-----

iyyzzypU7eM5pSSDoosysD5iqIcXbdh+j0LKEtGs4vdQ=

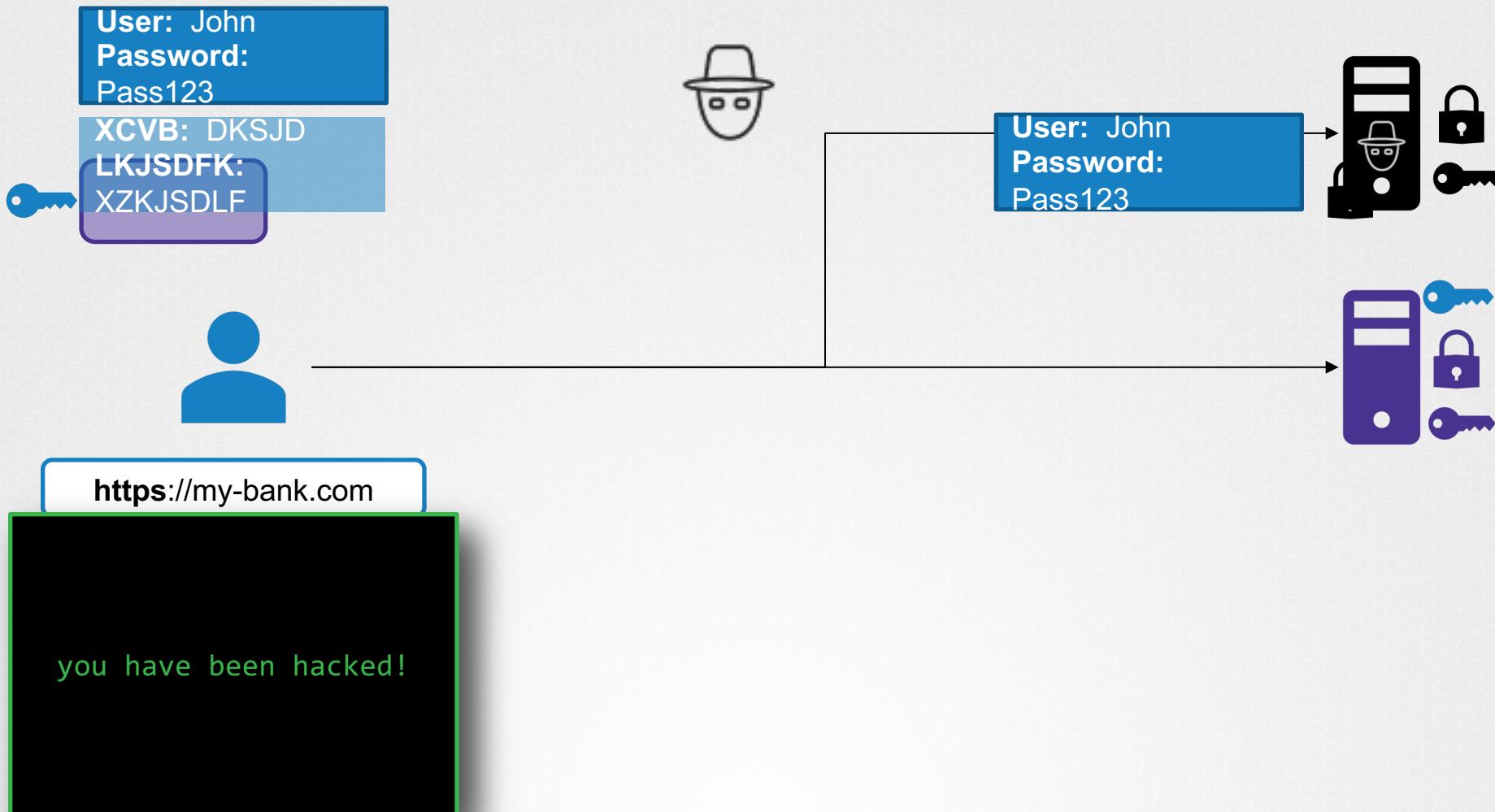
-----END RSA PRIVATE KEY-----

User: John  
Password:



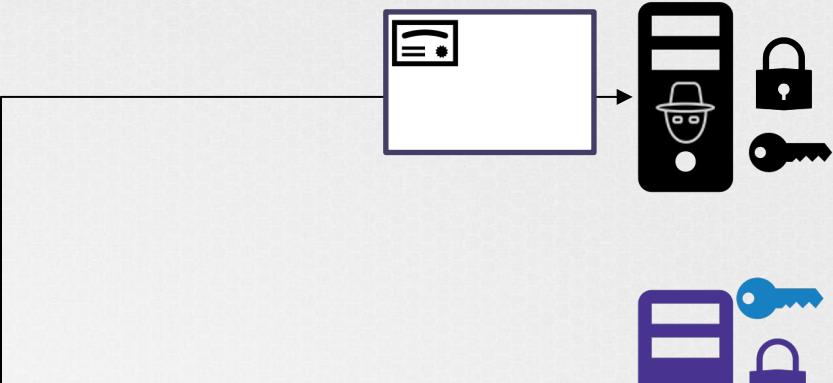
024

it > mybank.pem

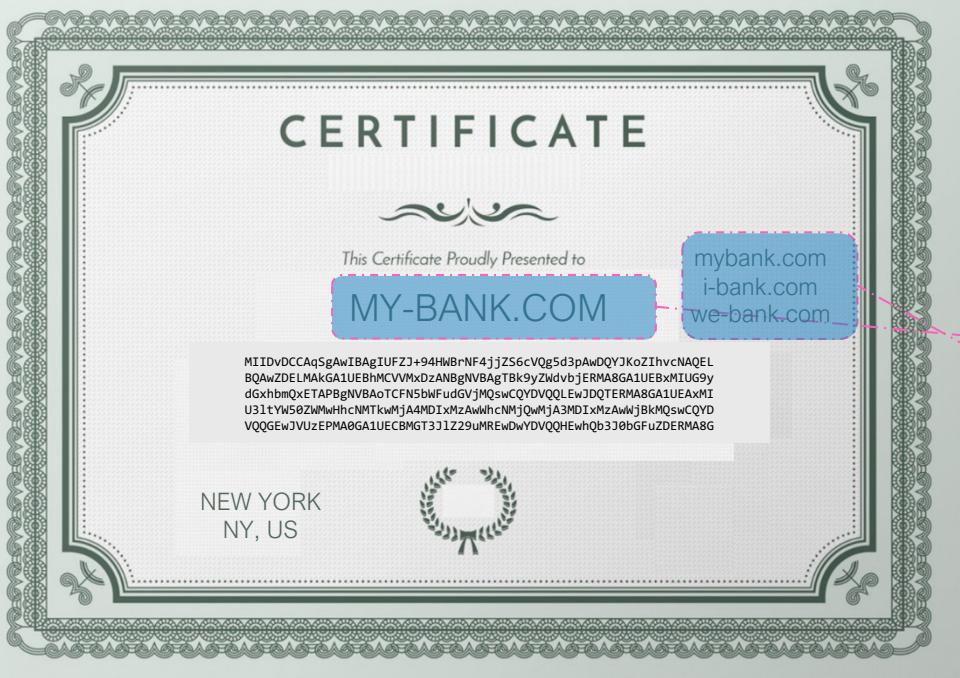




<https://my-bank.com>







Certificate:

Data:

Serial Number: 420327018966204255

Signature Algorithm: sha256WithRSAEncryption

Issuer: CN=kubernetes

Validity

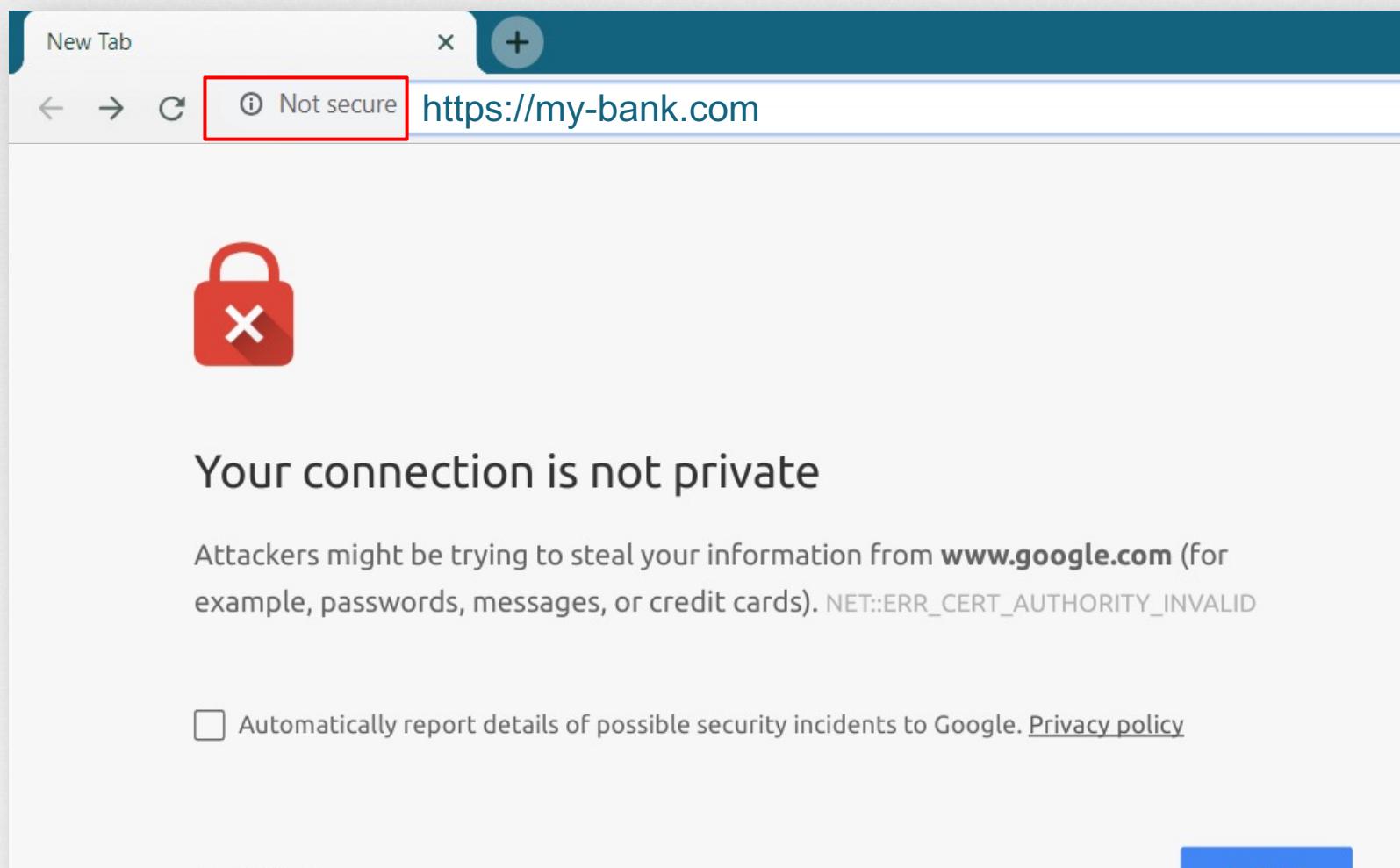
Not After : Feb 9 13:41:28 2020 GMT

Subject: CN=my-bank.com

Subject Alternative Name:

DNS:mybank.com, DNS:i-bank.com,  
DNS:we-bank.com,





New Tab x +

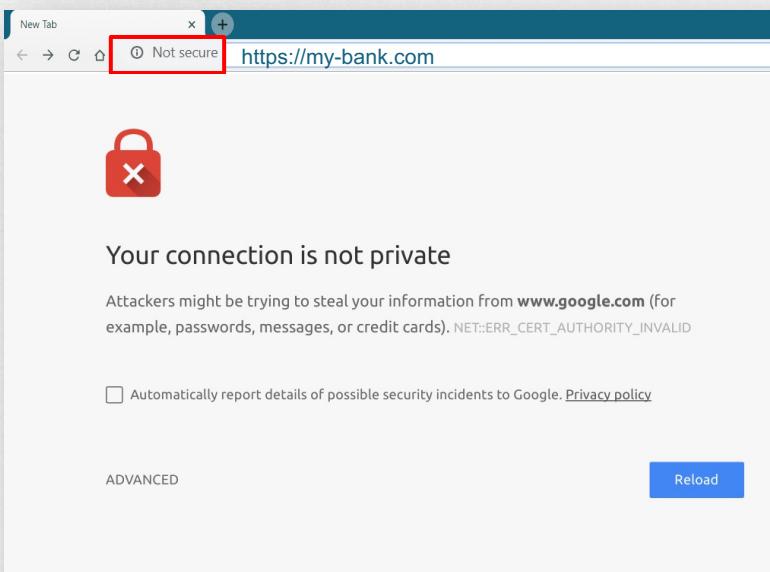
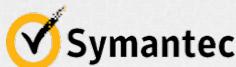
← → ⌂  ⓘ Not secure <https://my-bank.com>



## Your connection is not private

Attackers might be trying to steal your information from **www.google.com** (for example, passwords, messages, or credit cards). NET::ERR\_CERT\_AUTHORITY\_INVALID

Automatically report details of possible security incidents to Google. [Privacy policy](#)





# CERTIFICATE AUTHORITY (CA)

**Symantec**

-----BEGIN CERTIFICATE REQUEST-----

```
MIICjDCCAXQCAQAwRzELMAkGA1UEBhMCVVMxCzAJBgNVBAgMAkNBMRQwEgYDVQQK  
DAtNeU9yZywgSw5jLjEVMBMGA1UEAwwMbX1kb21haW4uY29tMIIBIjANBgkqhkiG  
9w0BAQEFAAOCAQ8AMIIBCgKCAQEAp8XohAKsHxvjs+/pRKCC2Sqx7021nuD49Kp4  
WDOnDBvxEeXNviY+SuQjpTxmuVr/orIpUC7MHk/fkbIICLT4jrXrBq4MwFfcwla1  
n8T0S9A7aLfWKL4rxJGF1U9DAdz4rqGLXFIC8obLpUWJkTerHpWg++k2UDkuPJE  
VQmQJ6Fe/3jWGaMNlnkY/eNyYn+a27NfMd1wQUzs9t5uFPpZbwG81mNjDvVIobA8  
yHNfRDNt6gKqvZtv+vGTaMOLfgjedGne2Uq7/Bbq22rSsXgfLM9wHmSpNT57Tjs9  
OQSobL4FFzoOnphhSqle1V/cGAjF1CzFIx988fH7xzduw+tRTQIDAQABoAAwDQYJ  
KoZIhvcNAQELBQADggEBABtY/tTvjFp4UlUTcI2f13TFbtYzyIwAYoB7U2sWrjzn  
uEe4k2+fosU1jXCJxk7EUT4sgGjVtoqJqrFihwQ1SLCViRgTwktLBDtvagViWNnQ  
mDJep5YY92JxtAKZZt52wsj8MeUwTUjn6eDuz5NhpoKuiWMf9LoxFYrgAGi2x1o  
Fkse6Zr6zaB/cNdm6daW8m6qVs9hKpudTiqgD3g4MEULLPK7VNxfFTMoSIfkLUui  
01F8dq2CW/ByrYMhUmONCAkKaag1FwY2Vm551HY6srcwnCPhszBCri7M5BZf70E  
rgKJPf06cAhFI7WpeuUz/0e4U12r6YF+Hhk7IDKnLeI=
```

-----END CERTIFICATE REQUEST-----

Certificate Signing Request

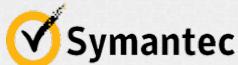
Validate Information

Sign and Send Certificate

.com"



# CERTIFICATE AUTHORITY (CA)



Certificate Signing Request (CSR)

Validate Information

Sign and Send Certificate





# CERTIFICATE AUTHORITY (CA)

## Certificates

Intended purpose: <All>

Intermediate Certification Authorities Trusted Root Certification Authorities Trusted Pub ▾

Issued To	Issued By	Expirati...	Friendly Name
COMODO RSA C...	COMODO RSA Cer...	1/19/20...	COMODO SEC...
GlobalSign	GlobalSign	3/18/20...	GlobalSign Ro...
CORP\srw-build-cd	CORP\srw-build-cd	11/7/20...	<None>
DigiCert Assure...	DigiCert Assured I...	11/10/2...	DigiCert
Symantec Enter...	Symantec Enterpri...	3/15/20...	<None>
Thawte Premiu...	Thawte Premium ...	1/1/2021	thawte
thawte Primary ...	thawte Primary Ro...	7/17/20...	thawte
thawte Primary ...	thawte Primary Ro...	12/2/20...	thawte Primar...
Thawte Timestam...	Thawte Timestamp...	1/1/2021	Thawte Time...
UTN-USFRFirst-...	UTN-USFRFirst-Oh...	7/10/20...	USFRTrust (C...

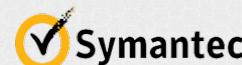
Import... Export... Remove Advanced

Certificate intended purposes

Code Signing View Close



# CERTIFICATE AUTHORITY (CA)



MIIDvZDELMAGA1UEBMCVVNxDzANBgNVBAgTBkgyZidvbTERMA8GA1UECxIu9y  
dGxhbmxQETAPBgNVBAoTCFNSWFudGVyMjQswCQYDVQQLEwJDQTERMA8GA1UEAxMI  
U3lYW50ZWlwfhcNMkWjA4ADIXMzAwhCNjQmJA3MDIxMzAwjBkMjQsNCQYD  
VQGQewJUzEPMA8GA1UECBMGT3J1Z29uMREwDwYDVQQHEwhQb3J0bGFuZDERMA8G

# CERTIFICATE AUTHORITY (CA)



CER

 Symantec

CER



# PKI

## (Public Key Infrastructure)

Symm Key



CSR



### Client Certificates



CSR



### Serving Certificates



**Certificate (Public Key)**

-----  
\*.crt \*.pem

server.crt  
server.pem  
client.crt  
client.pem

**Private Key**

-----  
\*.key \*-key.pem

server.key  
server-key.pem  
client.key  
client-key.pem



Public Key (Lock)



Private Key

**User:** John

**Password:**

Pass123

**XCVB:** DKSJD

**LKJSDFK:**

XZKJSDLF

**User:** John

**Password:**

Pass123

**XCVB:** DKSJD

**LKJSDFK:**

XZKJSDLF



{KODE} {LOUD}

[www.kodekloud.com](http://www.kodekloud.com)

# TLS CERTIFICATES

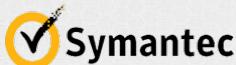
What Certificates?





# CERTIFICATE AUTHORITY (CA)

## Root Certificates



## Client Certificates

### Certificate (Public Key)

\*.crt \*.pem

server.crt  
server.pem  
client.crt  
client.pem

### Private Key

\*.key \*-key.pem

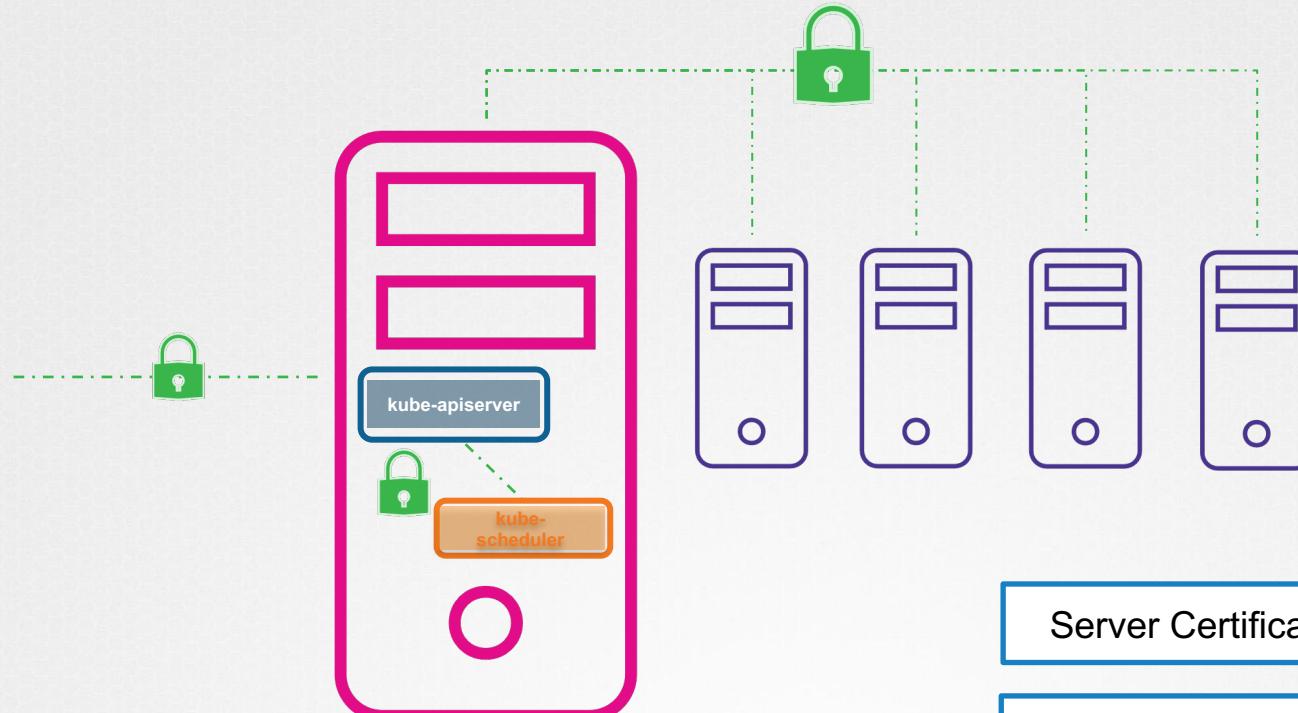
server.key  
server-key.pem  
client.key  
client-key.pem



## Certificate (Public Key)

## Private Key

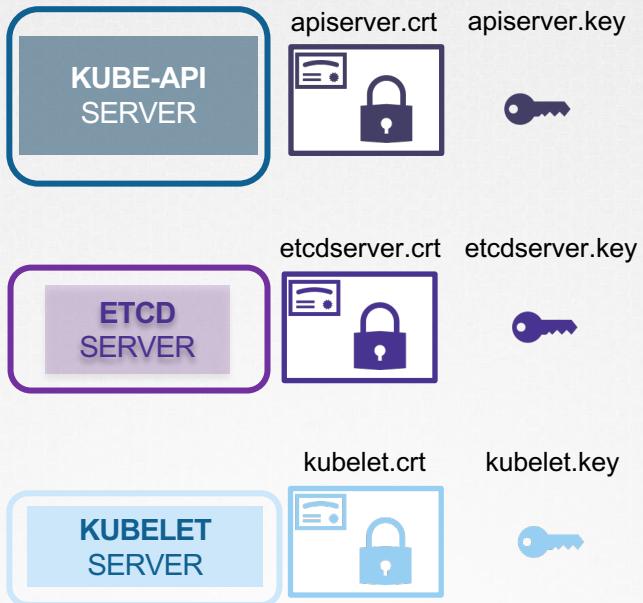
## Server Certificates



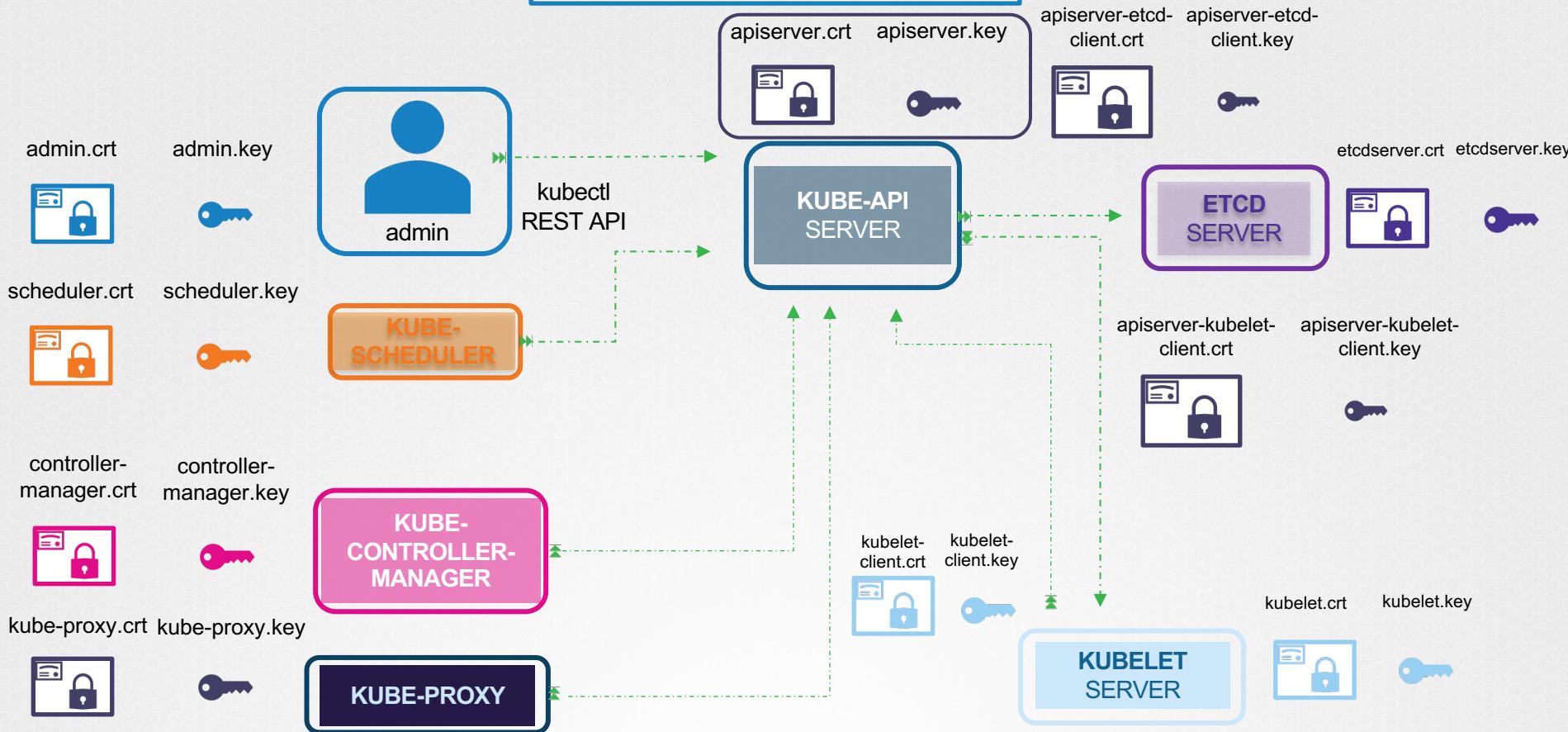
Server Certificates for Servers

Client Certificates for Clients

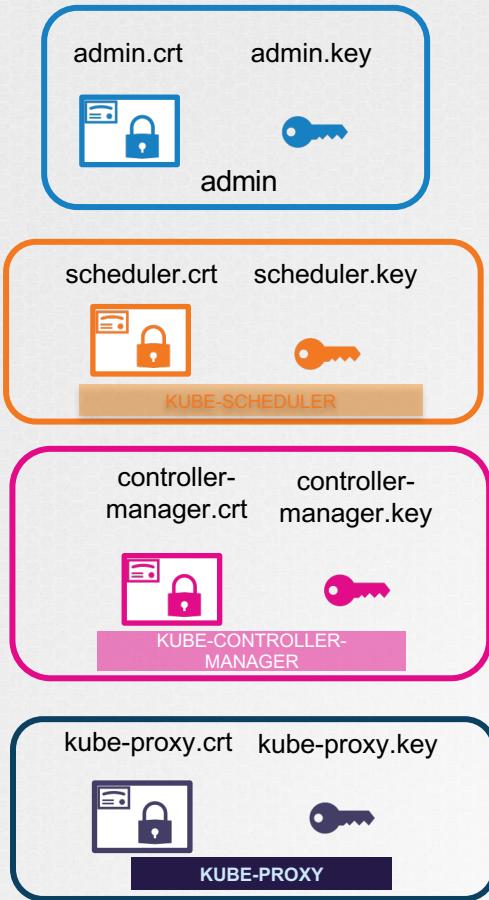
## Server Certificates for Servers



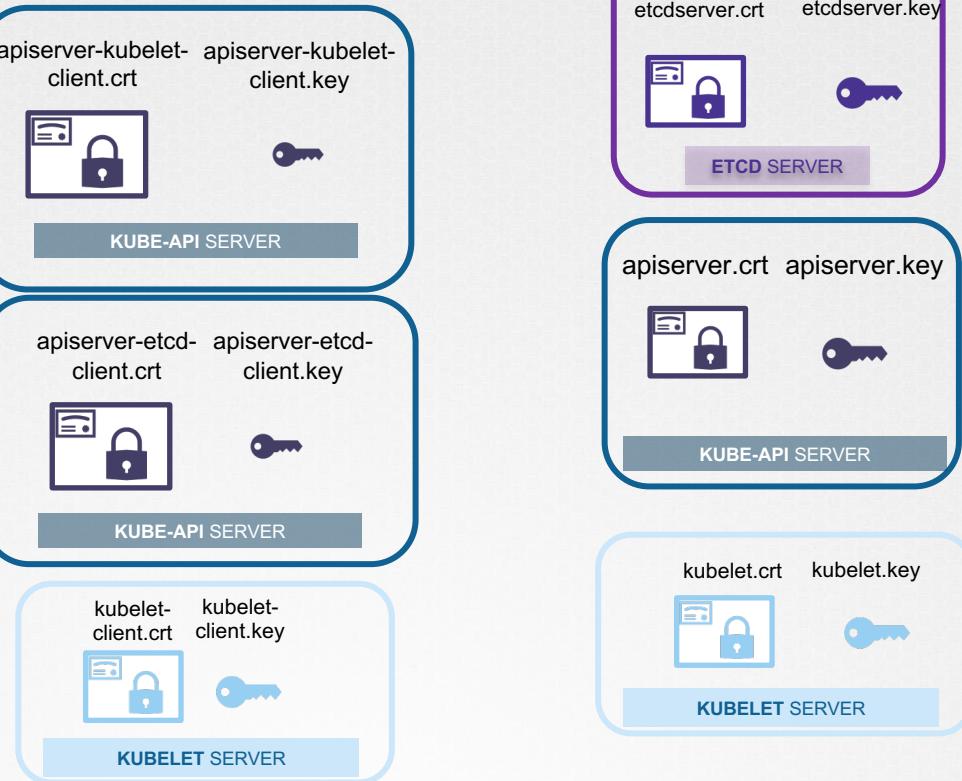
## Client Certificates for Clients



## Client Certificates for Clients



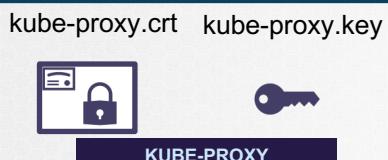
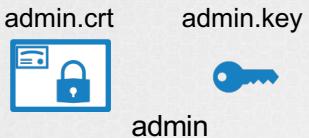
## Server Certificates for Servers



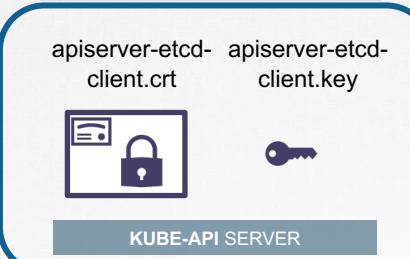
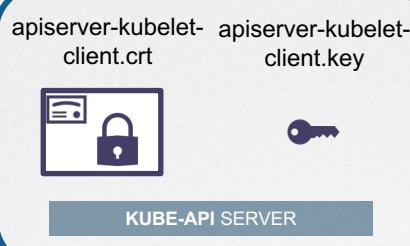


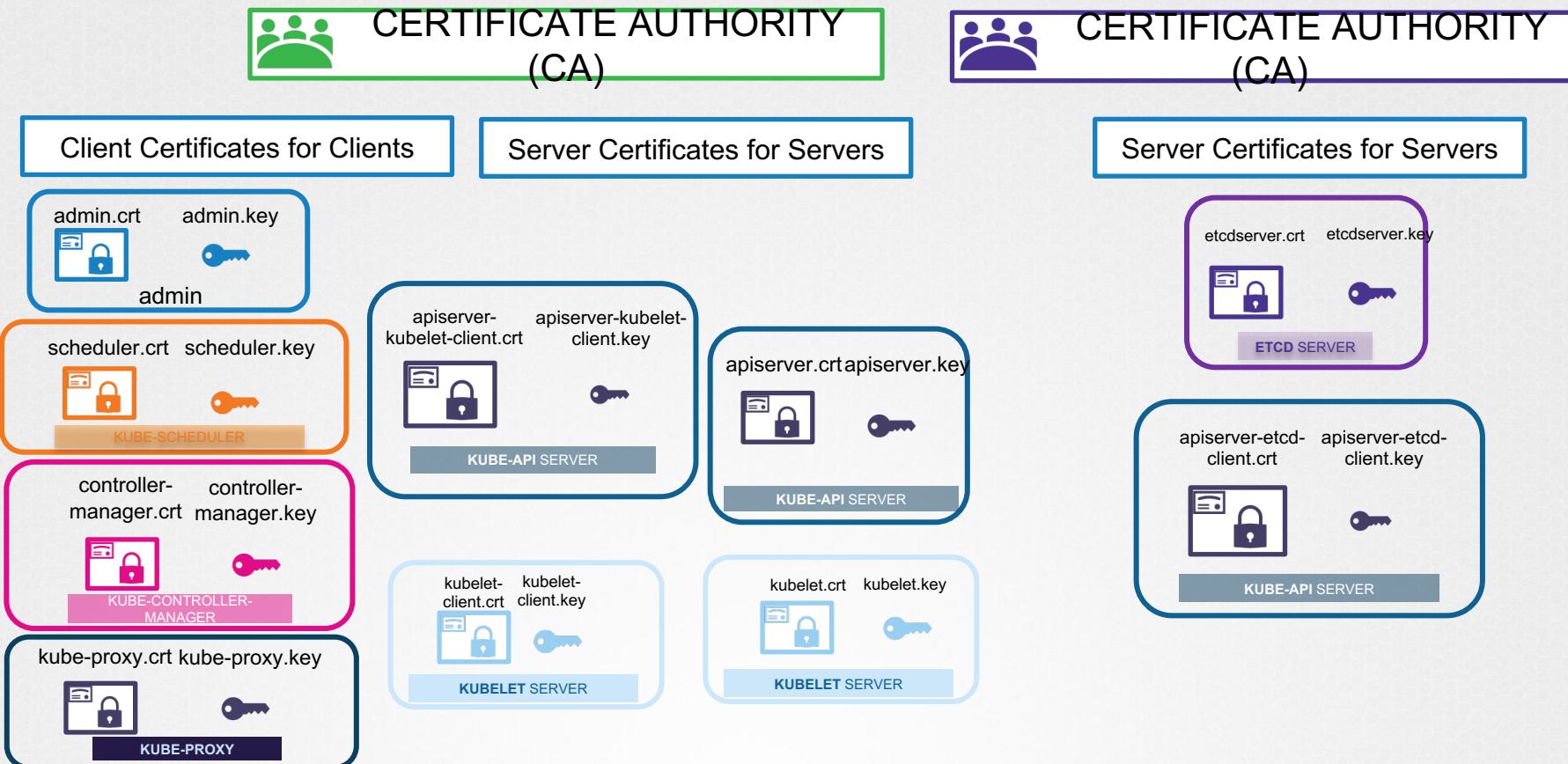
# CERTIFICATE AUTHORITY (CA)

## Client Certificates for Clients



## Server Certificates for Servers







# CERTIFICATE AUTHORITY (CA)

## Client Certificates for Clients

admin.crt

admin.key



admin

scheduler.crt

scheduler.key



KUBE-SCHEDULER

controller-  
manager.crtcontroller-  
manager.keyKUBE-CONTROLLER-  
MANAGER

kube-proxy.crt

kube-proxy.key



KUBE-PROXY

## Server Certificates for Servers

etcdserver.crt

etcdserver.key



ETCD SERVER

apiserver.crt

apiserver.key



KUBE-API SERVER

apiserver-etcd-  
client.crtapiserver-etcd-  
client.key

KUBE-API SERVER

kubelet-  
client.crtkubelet-  
client.key

KUBELET SERVER

kubelet.crt

kubelet.key



KUBELET SERVER



{KODE} {LOUD}

[www.kodekloud.com](http://www.kodekloud.com)



# TLS CERTIFICATES

Generate Certificates

EASYRSA

OPENSSL

CFSSL

OPENSSL



# CERTIFICATE AUTHORITY (CA)

## Client Certificates for Clients

admin.crt

admin.key



admin

scheduler.crt

scheduler.key



KUBE-SCHEDULER

controller-  
manager.crtcontroller-  
manager.keyKUBE-CONTROLLER-  
MANAGER

kube-proxy.crt

kube-proxy.key



KUBE-PROXY

apiserver-kubelet-  
client.crtapiserver-kubelet-  
client.keyapiserver-etcd-  
client.crtapiserver-etcd-  
client.key

KUBE-API SERVER



KUBELET SERVER



## Server Certificates for Servers

etcdserver.crt

etcdserver.key



ETCD SERVER

apiserver.crt

apiserver.key



KUBE-API SERVER

kubelet.crt

kubelet.key



KUBELET SERVER





# CERTIFICATE AUTHORITY (CA)

Generate Keys



ca.key

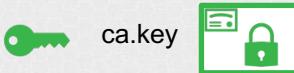
```
openssl genrsa -out ca.key 2048
```

ca.key



```
new -key ca.key -subj "/CN=KUBERNETES-CA" -out
```

```
-req -in ca.csr -signkey ca.key -out ca.crt
```



ca.crt



## ADMIN USER

admin.key



Generate Keys

Certificate  
Signing  
Request

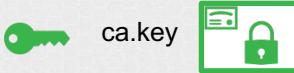
admin.csr



```
openssl req -new -key admin.key -subj \
"/CN=kube-admin/OU=system:masters" -out admin.csr
```



```
-in admin.csr -CA ca.crt -CAkey ca.key -out admin.crt
```



ca.crt

# KUBE SCHEDULER

**Generate Keys**

scheduler.key

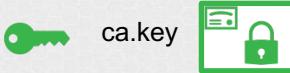
**Certificate  
Signing  
Request**

scheduler.csr

**Sign  
Certificates**

scheduler.crt





ca.crt

# KUBE CONTROLLER MANGER

**Generate Keys**

controller-manager.key

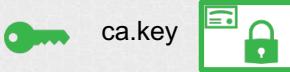
**Certificate  
Signing  
Request**

controller-manager.csr

**Sign  
Certificates**

controller-manager.crt





ca.crt

# KUBE PROXY

**Generate Keys**

kube-proxy.key

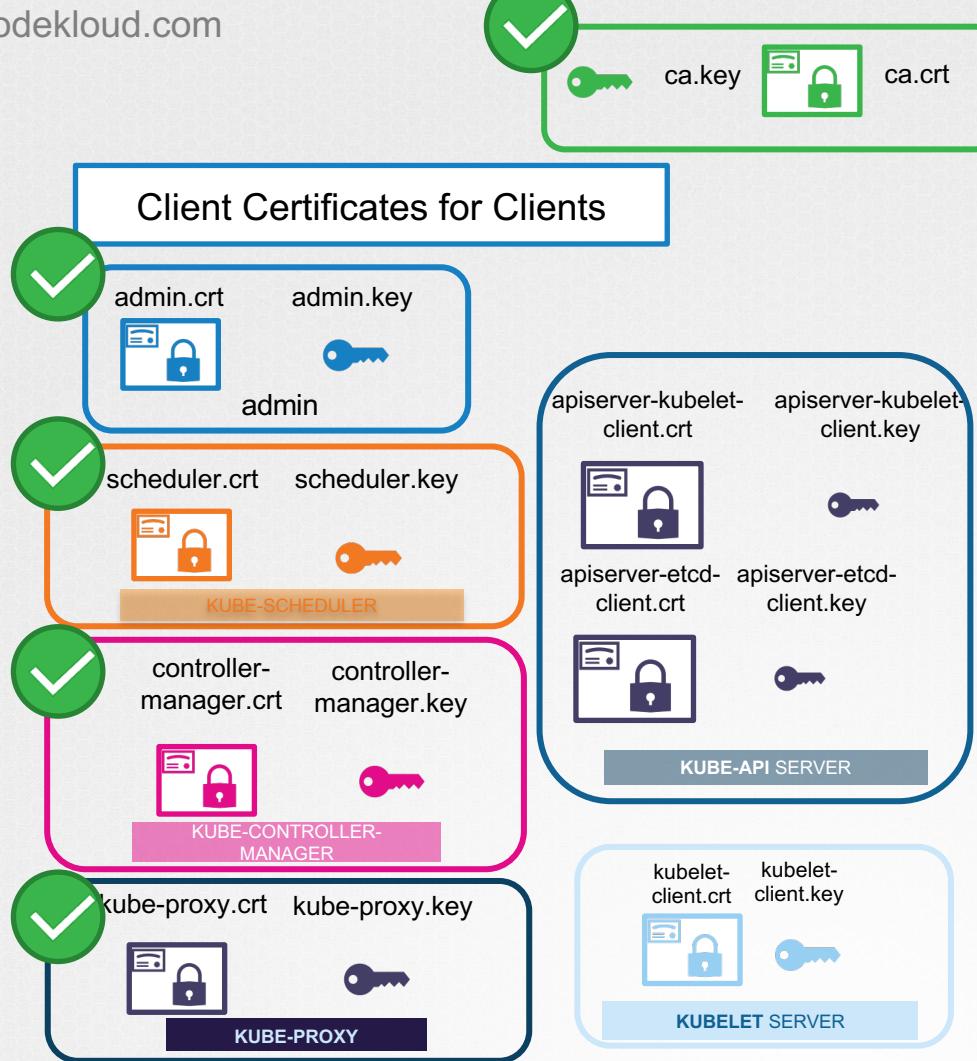
**Certificate  
Signing  
Request**

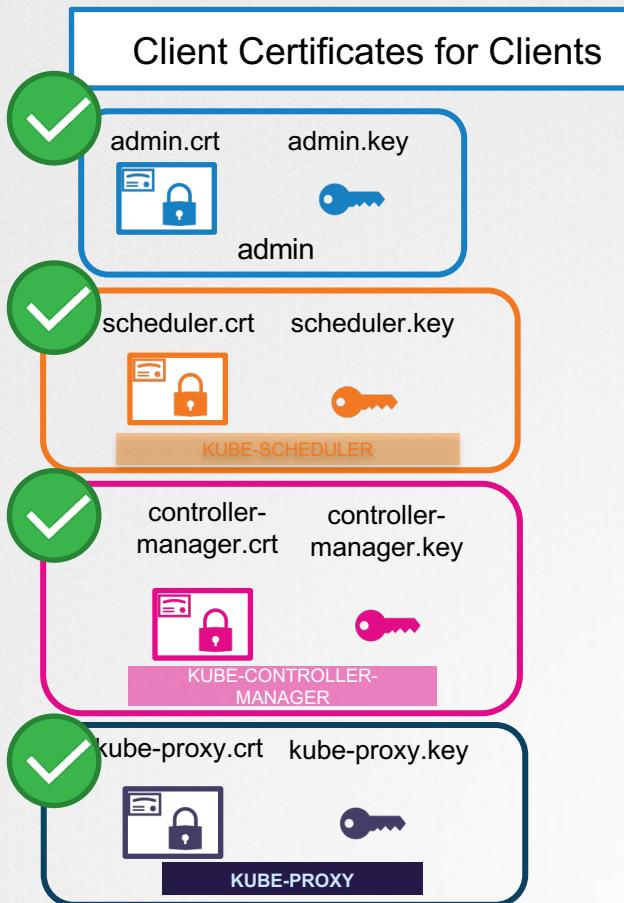
kube-proxy.csr

**Sign  
Certificates**

kube-proxy.crt





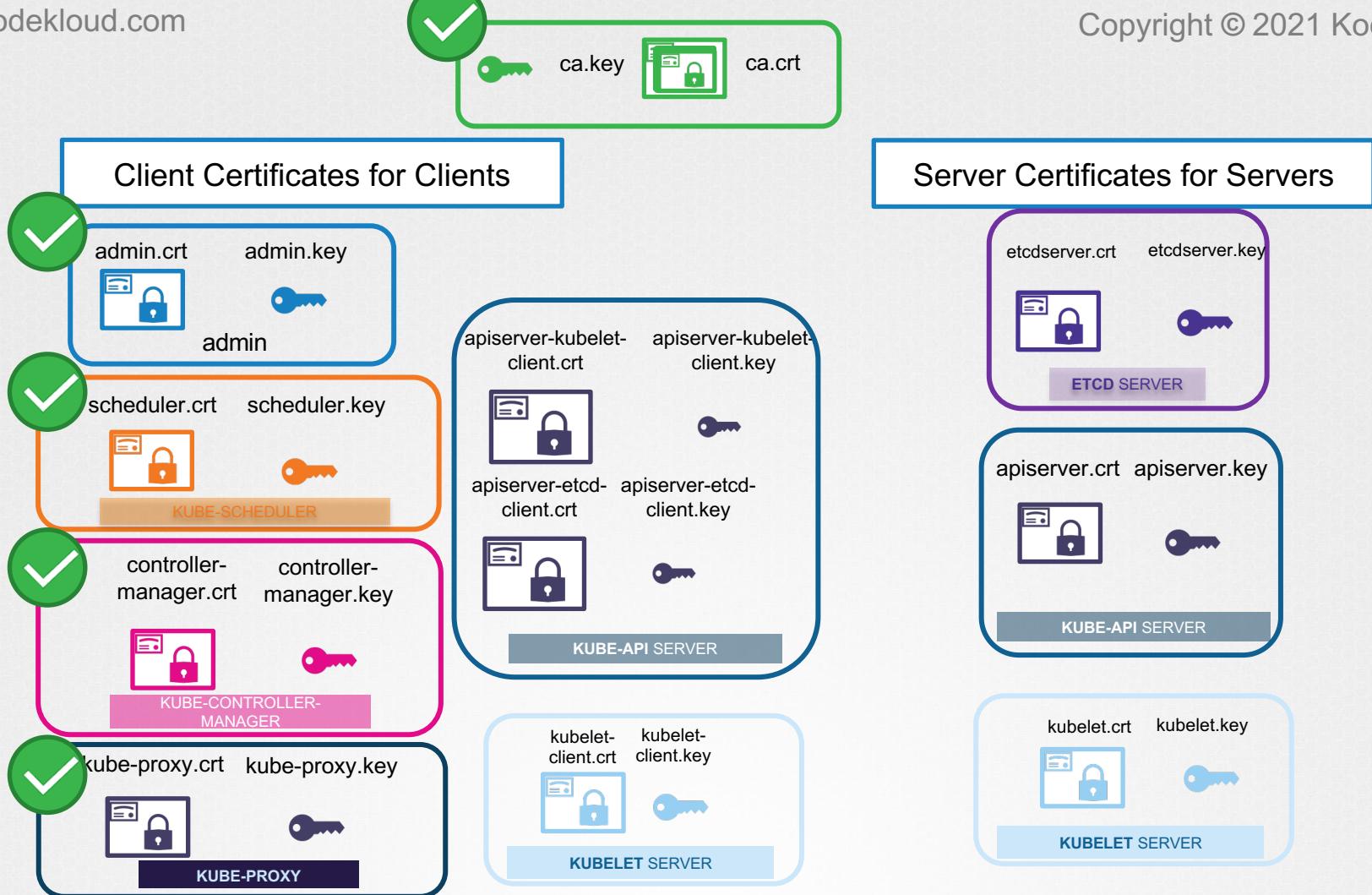


```
curl https://kube-apiserver:6443/api/v1/pods \
--key admin.key --cert admin.crt
--cacert ca.crt
```

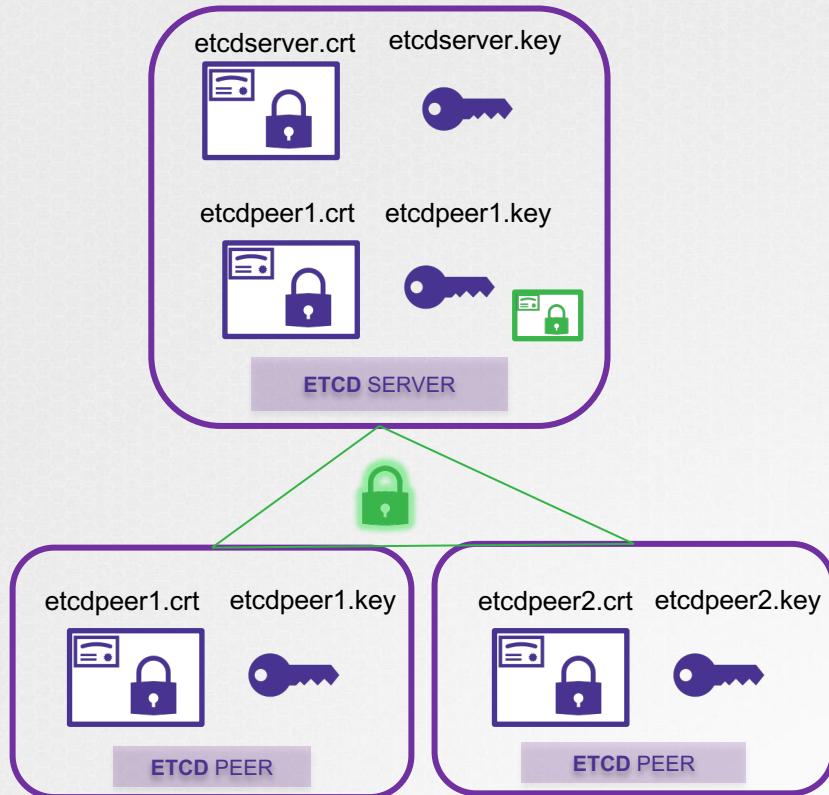
```
{
  "kind": "PodList",
  "apiVersion": "v1",
  "metadata": {
    "selfLink": "/api/v1/pods",
  },
  "items": []
}
```

**kube-config.yaml**

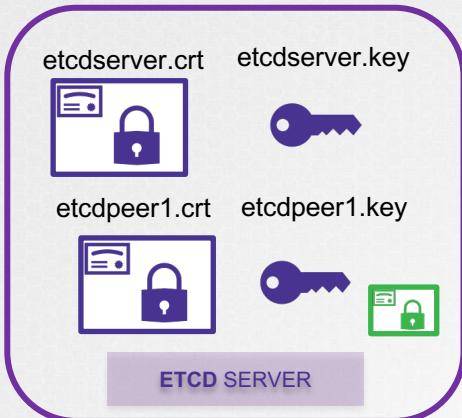
```
apiVersion: v1
clusters:
- cluster:
    certificate-authority: ca.crt
    server: https://kube-apiserver:6443
    name: kubernetes
  kind: Config
users:
- name: kubernetes-admin
  user:
    client-certificate: admin.crt
    client-key: admin.key
```



## ETCD SERVERS

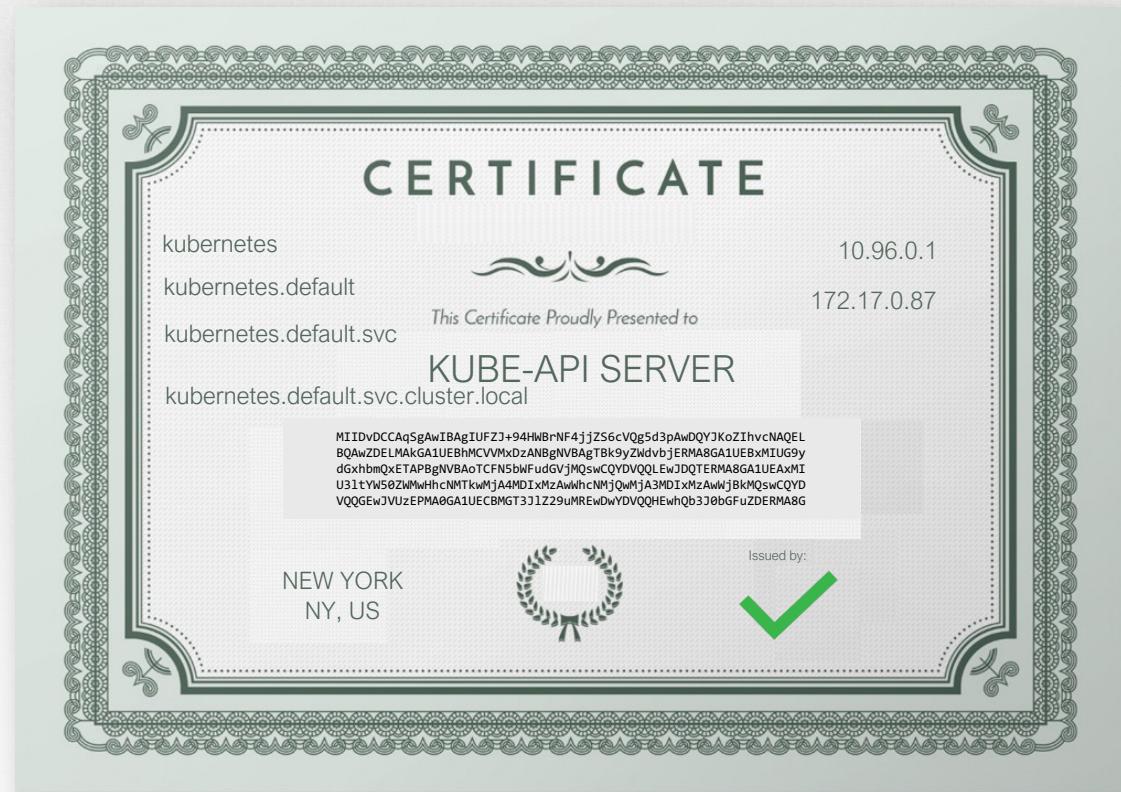


## ETCD SERVERS



```
cat etcd.yaml
- etcd
  - --advertise-client-urls=https://127.0.0.1:2379
  - --key-file=/path-to-certs/etcdserver.key
  - --cert-file=/path-to-certs/etcdserver.crt
  - --client-cert-auth=true
  - --data-dir=/var/lib/etcd
  - --initial-advertise-peer-urls=https://127.0.0.1:2380
  - --initial-cluster=master=https://127.0.0.1:2380
  - --listen-client-urls=https://127.0.0.1:2379
  - --listen-peer-urls=https://127.0.0.1:2380
  - --name=master
  - --peer-cert-file=/path-to-certs/etcdpeer1.crt
  - --peer-client-cert-auth=true
  - --peer-key-file=/etc/kubernetes/pki/etcd/peer.key
  - --peer-trusted-ca-file=/etc/kubernetes/pki/etcd/ca.crt
  - --snapshot-count=10000
  - --trusted-ca-file=/etc/kubernetes/pki/etcd/ca.crt
```

## KUBE API SERVER



apiserver.crt apiserver.key

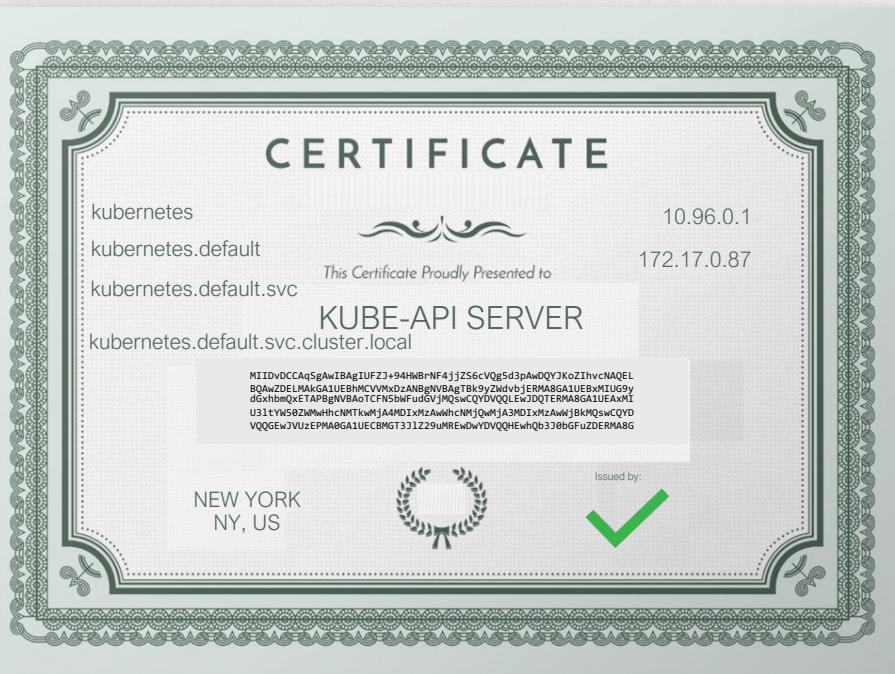


KUBE-API SERVER

## KUBE API SERVER

```
openssl genrsa -out apiserver.key 2048  
apiserver.key
```

```
openssl req -new -key apiserver.key -subj \  
"/CN=kube-apiserver" -out apiserver.csr  
apiserver.csr
```



apiserver.crt apiserver.key



KUBE-API SERVER

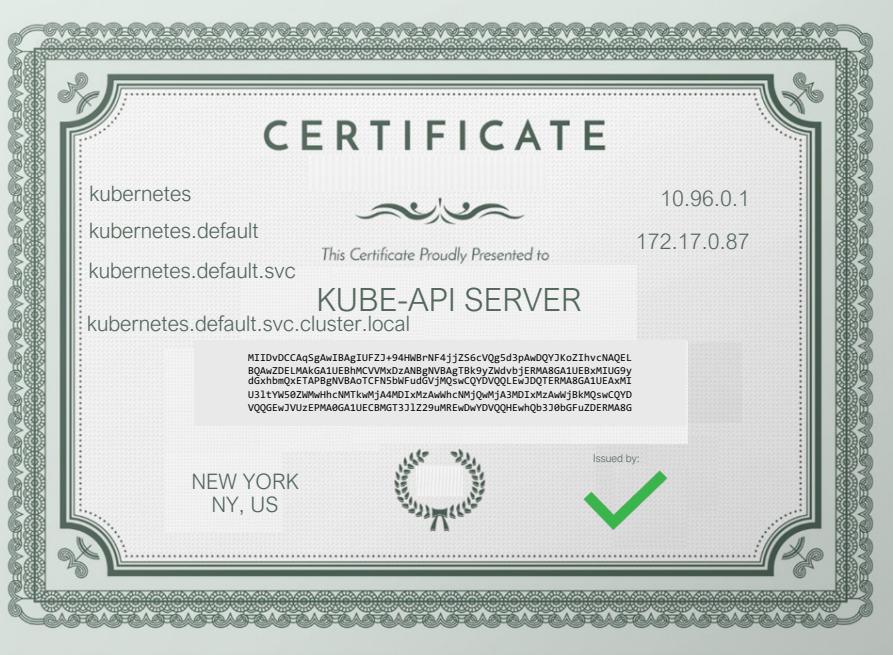
## KUBE API SERVER

```
openssl req -new -key apiserver.key -subj \
"/CN=kube-apiserver" -out apiserver.csr -config openssl.cnf
```

apiserver.csr

openssl.cnf

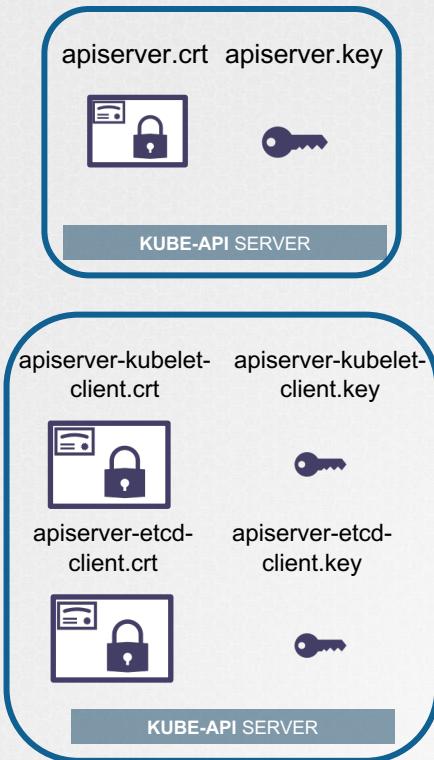
```
[req]
req_extensions = v3_req
[ v3_req ]
basicConstraints = CA:FALSE
keyUsage = nonRepudiation,
subjectAltName = @alt_names
[alt_names]
DNS.1 = kubernetes
DNS.2 = kubernetes.default
DNS.3 = kubernetes.default.svc
DNS.4 = kubernetes.default.svc.cluster.local
IP.1 = 10.96.0.1
IP.2 = 172.17.0.87
```



```
openssl x509 -req -in apiserver.csr \
-CA ca.crt -CAkey ca.key -out apiserver.crt
```

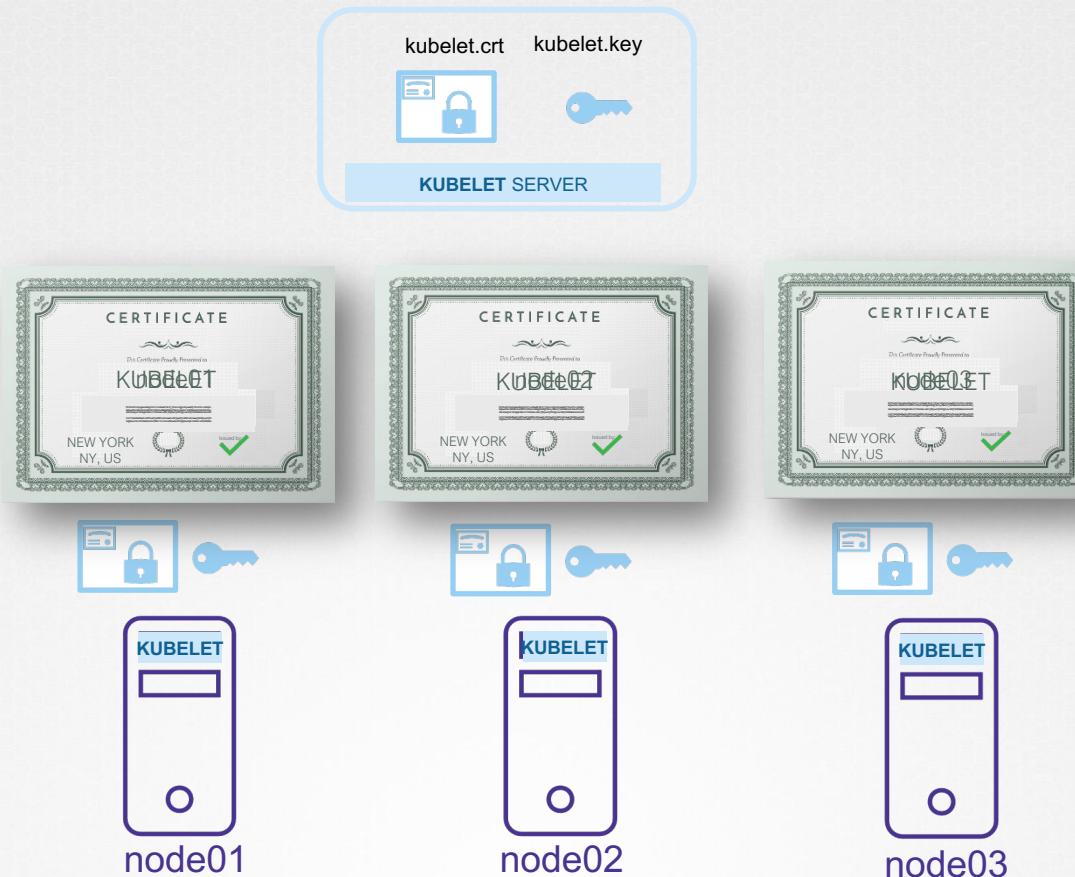
apiserver.crt

## KUBE API SERVER



```
ExecStart=/usr/local/bin/kube-apiserver \
--advertise-address=${INTERNAL_IP} \
--allow-privileged=true \
--apiserver-count=3 \
--authorization-mode=Node,RBAC \
--bind-address=0.0.0.0 \
--enable-swagger-ui=true \
--etcd-cafile=/var/lib/kubernetes/ca.pem \
--etcd-certfile=/var/lib/kubernetes/apiserver-etcd-client.crt \
--etcd-keyfile=/var/lib/kubernetes/apiserver-etcd-client.key \
--etcd-servers=https://127.0.0.1:2379 \
--event-ttl=1h \
--kubelet-certificate-authority=/var/lib/kubernetes/ca.pem \
--kubelet-client-certificate=/var/lib/kubernetes/apiserver-client.crt \
--kubelet-client-key=/var/lib/kubernetes/apiserver-client.key \
--kubelet-https=true \
--runtime-config=api/all \
--service-account-key-file=/var/lib/kubernetes/service-account.pem \
--service-cluster-ip-range=10.32.0.0/24 \
--service-node-port-range=30000-32767 \
--client-ca-file=/var/lib/kubernetes/ca.pem \
--tls-cert-file=/var/lib/kubernetes/apiserver.crt \
--tls-private-key-file=/var/lib/kubernetes/apiserver.key \
--v=2
```

## KUBECTL NODES (SERVER CERT)



openssl.cnf

```
[req]
req_extensions =
[ v3_req ]
basicConstraints
keyUsage = nonRepudiation, digitalSignature, keyEncipherment
subjectAltName = [alt_names]
DNS.1 = kubernetes
DNS.2 = kubernetes
DNS.3 = kubernetes
DNS.4 = kubernetes
IP.1 = 10.96.0.1
IP.2 = 172.17.0.1
```

## KUBECTL NODES (SERVER CERT)



`kubelet-config.yaml (node01)`

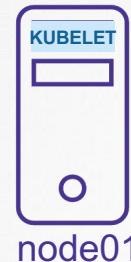
```
kind: KubeletConfiguration
apiVersion: kubelet.config.k8s.io/v1beta1
authentication:
  x509:
    clientCAFile: "/var/lib/kubernetes/ca.pem"
authorization:
  mode: Webhook
clusterDomain: "cluster.local"
clusterDNS:
  - "10.32.0.10"
podCIDR: "${POD_CIDR}"
resolvConf: "/run/systemd/resolve/resolv.conf"
runtimeRequestTimeout: "15m"
tlsCertFile: "/var/lib/kubelet/kubelet-node01.crt"
tlsPrivateKeyFile: "/var/lib/kubelet/kubelet-
node01.key"
```

## KUBECTL NODES (CLIENT CERT)

Kubelet-client.crt Kubelet-client.key



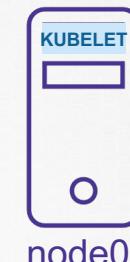
KUBELET SERVER



node01



node02



node03



{KODE} {LOUD}

[www.kodekloud.com](http://www.kodekloud.com)



# TLS CERTIFICATES

[View Certificate Details](#)

“The Hard Way”

kubeadm

## “The Hard Way”

```
▶ cat /etc/systemd/system/kube-apiserver.service
```

```
[Service]
ExecStart=/usr/local/bin/kube-apiserver \
--advertise-address=172.17.0.32 \
--allow-privileged=true \
--apiserver-count=3 \
--authorization-mode=Node,RBAC \
--bind-address=0.0.0.0 \
--client-ca-file=/var/lib/kubernetes/ca.pem \
--enable-swagger-ui=true \
--etcd-cafile=/var/lib/kubernetes/ca.pem \
--etcd-certfile=/var/lib/kubernetes/kubernetes.pem \
--etcd-keyfile=/var/lib/kubernetes/kubernetes-key.pem \
--event-ttl=1h \
--kubelet-certificate-authority=/var/lib/kubernetes/ca.pem \
--kubelet-client-key=/var/lib/kubernetes/kubernetes-key.pem \
--kubelet-https=true \
--service-node-port-range=30000-32767 \
--tls-cert-file=/var/lib/kubernetes/kubernetes.pem \
--tls-private-key-file=/var/lib/kubernetes/kubernetes-key.pem
--v=2
```

## kubeadm

```
▶ cat /etc/kubernetes/manifests/kube-apiserver.yaml
```

```
spec:
  containers:
    - command:
        - kube-apiserver
        - --authorization-mode=Node,RBAC
        - --advertise-address=172.17.0.32
        - --allow-privileged=true
        - --client-ca-file=/etc/kubernetes/pki/ca.crt
        - --disable-admission-plugins=PersistentVolumeLabel
        - --enable-admission-plugins=NodeRestriction
        - --enable-bootstrap-token-auth=true
        - --etcd-cafile=/etc/kubernetes/pki/etcd/ca.crt
        - --etcd-certfile=/etc/kubernetes/pki/apiserver-etcd-client.crt
        - --etcd-keyfile=/etc/kubernetes/pki/apiserver-etcd-client.key
        - --etcd-servers=https://127.0.0.1:2379
        - --insecure-port=0
        - --kubelet-client-certificate=/etc/kubernetes/pki/apiserver-kubelet-client.crt
        - --kubelet-client-key=/etc/kubernetes/pki/apiserver-kubelet-client.key
        - --kubelet-preferred-address-types=InternalIP,ExternalIP,HostId
        - --proxy-client-cert-file=/etc/kubernetes/pki/front-proxy-client.crt
        - --proxy-client-key-file=/etc/kubernetes/pki/front-proxy-client.key
        - --requestheader-allowed-names=front-proxy-client
        - --requestheader-treat-unknown-headers-as-admission=true
        - --service-node-port-range=30000-32767
        - --tls-cert-file=/var/lib/kubernetes/kubernetes.pem
        - --tls-private-key-file=/var/lib/kubernetes/kubernetes-key.pem
      image: k8s.gcr.io/kube-apiserver:v1.21.2
      ports:
        - containerPort: 443
      resources:
        limits:
          memory: 2Gi
        requests:
          memory: 1Gi
    securityContext:
      privileged: true
```

Component	Type	Certificate Path	CN Name	ALT Names	Organization	Issuer	Expiration
kube-apiserver	Server						
kube-apiserver	Server						
kube-apiserver	Server						
kube-apiserver	Client (Kubelet)						
kube-apiserver	Client (Kubelet)						
kube-apiserver	Client (Etcd)						
kube-apiserver	Client (Etcd)						
kube-apiserver	Client (Etcd)						

```
▶ cat /etc/kubernetes/manifests/kube-apiserver.yaml
```

```
spec:  
  containers:  
    - command:  
        - kube-apiserver  
        - --authorization-mode=Node,RBAC  
        - --advertise-address=172.17.0.32  
        - --allow-privileged=true  
        - --client-ca-file=/etc/kubernetes/pki/ca.crt  
        - --disable-admission-plugins=PersistentVolumeLabel  
        - --enable-admission-plugins=NodeRestriction  
        - --enable-bootstrap-token-auth=true  
        - --etcd-cafile=/etc/kubernetes/pki/etcd/ca.crt  
        - --etcd-certfile=/etc/kubernetes/pki/apiserver-etcd-client.crt  
        - --etcd-keyfile=/etc/kubernetes/pki/apiserver-etcd-client.key  
        - --etcd-servers=https://127.0.0.1:2379  
        - --insecure-port=0  
        - --kubelet-client-certificate=/etc/kubernetes/pki/apiserver-kubelet-client.crt  
        - --kubelet-client-key=/etc/kubernetes/pki/apiserver-kubelet-client.key  
        - --kubelet-preferred-address-types=InternalIP,ExternalIP,Hostname  
        - --proxy-client-cert-file=/etc/kubernetes/pki/front-proxy-client.crt  
        - --proxy-client-key-file=/etc/kubernetes/pki/front-proxy-client.key  
        - --secure-port=6443  
        - --service-account-key-file=/etc/kubernetes/pki/sa.pub  
        - --service-cluster-ip-range=10.96.0.0/12  
        - --tls-cert-file=/etc/kubernetes/pki/apiserver.crt  
        - --tls-private-key-file=/etc/kubernetes/pki/apiserver.key
```

`/etc/kubernetes/pki/apiserver.crt`

```
▶ openssl x509 -in /etc/kubernetes/pki/apiserver.crt -text -noout
```

Certificate:

  Data:

    Version: 3 (0x2)

    Serial Number: 3147495682089747350 (0x2bae26a58f090396)

    Signature Algorithm: sha256WithRSAEncryption

      Issuer: CN=kubernetes

      Validity

        Not Before: Feb 11 05:39:19 2019 GMT

        Not After : Feb 11 05:39:20 2020 GMT

      Subject: CN=kube-apiserver

      Subject Public Key Info:

        Public Key Algorithm: rsaEncryption

        Public-Key: (2048 bit)

          Modulus:

            00:d9:69:38:80:68:3b:b7:2e:9e:25:00:e8:fd:01:

            Exponent: 65537 (0x10001)

      X509v3 extensions:

        X509v3 Key Usage: critical

          Digital Signature, Key Encipherment

        X509v3 Extended Key Usage:

          TLS Web Server Authentication

        X509v3 Subject Alternative Name:

          DNS:master, DNS:kubernetes, DNS:kubernetes.default,

DNS:kubernetes.default.svc, DNS:kubernetes.default.svc.cluster.local, IP

Address:10.96.0.1, IP Address:172.17.0.27

Certificate Path	CN Name	ALT Names	Organization	Issuer	Expiration
/etc/kubernetes/pki/apiserver.crt	kube-apiserver	DNS:master DNS:kubernetes DNS:kubernetes.default DNS:kubernetes.default.svc IP Address:10.96.0.1 IP Address:172.17.0.27		kubernetes	Feb 11 05:39:20 2020
/etc/kubernetes/pki/apiserver.key					
/etc/kubernetes/pki/ca.crt	kubernetes			kubernetes	Feb 8 05:39:19 2029
/etc/kubernetes/pki/apiserver-kubelet-client.crt	kube-apiserver-kubelet-client		system:masters	kubernetes	Feb 11 05:39:20 2020
/etc/kubernetes/pki/apiserver-kubelet-client.key					
/etc/kubernetes/pki/apiserver-etcd-client.crt	kube-apiserver-etcd-client		system:masters	self	Feb 11 05:39:22 2020
/etc/kubernetes/pki/apiserver-etcd-client.key					
/etc/kubernetes/pki/etcd/ca.crt	kubernetes			kubernetes	Feb 8 05:39:21 2017

Default CN	Parent CA	O (in Subject)	kind	hosts (SAN)
kube-etcd	etcd-ca		server, client [1][etcdbug]	localhost , 127.0.0.1
kube-etcd-peer	etcd-ca		server, client	<hostname> , <Host_IP> , localhost , 127.0.0.1
kube-etcd-healthcheck-client	etcd-ca		client	
kube-apiserver-etcd-client	etcd-ca	system:masters	client	
kube-apiserver	kubernetes-ca		server	<hostname> , <Host_IP> , <advertise_IP> , [1]
kube-apiserver-kubelet-client	kubernetes-ca	system:masters	client	
front-proxy-client	kubernetes-front-proxy-ca		client	

Default CN	recommend key path	recommended cert path	command	key argument	cert argument
etcd-ca		etcd/ca.crt	kube-apiserver		-etcd-cafile
etcd-client	apiserver-etcd-client.key	apiserver-etcd-client.crt	kube-apiserver	-etcd-keyfile	-etcd-certfile
kubernetes-ca		ca.crt	kube-apiserver		-client-ca-file
kube-apiserver	apiserver.key	apiserver.crt	kube-apiserver	-tls-private-key-file	-tls-cert-file
apiserver-kubelet-client		apiserver-kubelet-client.crt	kube-apiserver		-kubelet-client-certificate
front-proxy-ca		front-proxy-ca.crt	kube-apiserver		-requestheader-client-ca-file
front-proxy-client	front-proxy-client.key	front-proxy-client.crt	kube-apiserver	-proxy-client-key-file	-proxy-client-cert-file
etcd-ca		etcd/ca.crt	etcd		-trusted-ca-file, -peer-trusted-ca-file
kube-etcd	etcd/server.key	etcd/server.crt	etcd	-key-file	-cert-file
kube-etcd-peer	etcd/peer.key	etcd/peer.crt	etcd	-peer-key-file	-peer-cert-file
etcd-ca		etcd/ca.crt	etcdctl[2]		-cacert
kube-etcd-healthcheck-client	etcd/healthcheck-client.key	etcd/healthcheck-client.crt	etcdctl[2]	-key	-cert

# Inspect Service Logs

```
▶ journalctl -u etcd.service -l
```

```
2019-02-13 02:53:28.144631 I | etcdmain: etcd Version: 3.2.18
2019-02-13 02:53:28.144680 I | etcdmain: Git SHA: eddf599c6
2019-02-13 02:53:28.144684 I | etcdmain: Go Version: go1.8.7
2019-02-13 02:53:28.144688 I | etcdmain: Go OS/Arch: linux/amd64
2019-02-13 02:53:28.144692 I | etcdmain: setting maximum number of CPUs to 4, total number of available CPUs is 4
2019-02-13 02:53:28.144734 N | etcdmain: the server is already initialized as member before, starting as etcd
member...
2019-02-13 02:53:28.146625 I | etcdserver: name = master
2019-02-13 02:53:28.146637 I | etcdserver: data dir = /var/lib/etcd
2019-02-13 02:53:28.146642 I | etcdserver: member dir = /var/lib/etcd/member
2019-02-13 02:53:28.146645 I | etcdserver: heartbeat = 100ms
2019-02-13 02:53:28.146648 I | etcdserver: election = 1000ms
2019-02-13 02:53:28.146651 I | etcdserver: snapshot count = 10000
2019-02-13 02:53:28.146677 I | etcdserver: advertise client URLs = 2019-02-13 02:53:28.185353 I | etcdserver/api:
enabled capabilities for version 3.2
2019-02-13 02:53:28.185588 I | embed: ClientTLS: cert = /etc/kubernetes/pki/etcd/server.crt, key =
/etc/kubernetes/pki/etcd/server.key, ca = , trusted-ca = /etc/kubernetes/pki/etcd/old-ca.crt, client-cert-auth =
true
2019-02-13 02:53:30.080017 I | embed: ready to serve client requests
2019-02-13 02:53:30.080130 I | etcdserver: published {Name:master ClientURLs:[https://127.0.0.1:2379]} to cluster
c9be114fc2da2776
2019-02-13 02:53:30.080281 I | embed: serving client requests on 127.0.0.1:2379
WARNING: 2019/02/13 02:53:30 Failed to dial 127.0.0.1:2379: connection error: desc = "transport: authentication
handshake failed: remote error: tls: bad certificate"; please retry.
```

# View Logs

▶ kubectl logs etcd-master

```
2019-02-13 02:53:28.144631 I | etcdmain: etcd Version: 3.2.18
2019-02-13 02:53:28.144680 I | etcdmain: Git SHA: eddf599c6
2019-02-13 02:53:28.144684 I | etcdmain: Go Version: go1.8.7
2019-02-13 02:53:28.144688 I | etcdmain: Go OS/Arch: linux/amd64
2019-02-13 02:53:28.144692 I | etcdmain: setting maximum number of CPUs to 4, total number of available CPUs is 4
2019-02-13 02:53:28.144734 N | etcdmain: the server is already initialized as member before, starting as etcd
member...
2019-02-13 02:53:28.146625 I | etcdserver: name = master
2019-02-13 02:53:28.146637 I | etcdserver: data dir = /var/lib/etcd
2019-02-13 02:53:28.146642 I | etcdserver: member dir = /var/lib/etcd/member
2019-02-13 02:53:28.146645 I | etcdserver: heartbeat = 100ms
2019-02-13 02:53:28.146648 I | etcdserver: election = 1000ms
2019-02-13 02:53:28.146651 I | etcdserver: snapshot count = 10000
2019-02-13 02:53:28.146677 I | etcdserver: advertise client URLs = 2019-02-13 02:53:28.185353 I | etcdserver/api:
enabled capabilities for version 3.2
2019-02-13 02:53:28.185588 I | embed: ClientTLS: cert = /etc/kubernetes/pki/etcd/server.crt, key =
/etc/kubernetes/pki/etcd/server.key, ca = , trusted-ca = /etc/kubernetes/pki/etcd/old-ca.crt, client-cert-auth =
true
2019-02-13 02:53:30.080017 I | embed: ready to serve client requests
2019-02-13 02:53:30.080130 I | etcdserver: published {Name:master ClientURLs:[https://127.0.0.1:2379]} to cluster
c9be114fc2da2776
2019-02-13 02:53:30.080281 I | embed: serving client requests on 127.0.0.1:2379
WARNING: 2019/02/13 02:53:30 Failed to dial 127.0.0.1:2379: connection error: desc = "transport: authentication
handshake failed: remote error: tls: bad certificate"; please retry.
```

# View Logs

▶ docker ps -a

CONTAINER ID	STATUS
23482a09f25b	Up 12 minutes
b9bf77348c96	Up 18 minutes
87fc69913973	Up 18 minutes
fda322157b86	Exited (255) 18 minutes ago
0794bdf5d7d8	Up 40 minutes
00f3f95d2102	Up 40 minutes
b8e6a0e173dd	Up About an hour
18e47bad320e	Up About an hour
4d087daf0380	Exited (1) About an hour ago
e923140101a3	Up About an hour
e0db7e63d18e	Up About an hour
74c257366f65	Up About an hour
8f514eac9d04	Exited (255) 40 minutes ago
b39c5c594913	Exited (1) 40 minutes ago
3aefcb20ed30	Up 2 hours
576c8a273b50	Up 2 hours
4b3c5f34efde	Up 2 hours

CONTAINER ID	STATUS	NAMES
23482a09f25b	Up 12 minutes	k8s_kube-apiserver_kube-apiserver-master_kube-system_8758a3d10776bb527e043
b9bf77348c96	Up 18 minutes	k8s_etcd_etcd-master_kube-system_2cc1c8a24b68ab9b46bca47e153e74c6_0
87fc69913973	Up 18 minutes	k8s_POD_etcd-master_kube-system_2cc1c8a24b68ab9b46bca47e153e74c6_0
fda322157b86	Exited (255) 18 minutes ago	k8s_kube-apiserver_kube-apiserver-master_kube-system_8758a3d10776bb527e043
0794bdf5d7d8	Up 40 minutes	k8s_kube-scheduler_kube-scheduler-master_kube-system_009228e74aef4d7babd790
00f3f95d2102	Up 40 minutes	k8s_kube-controller-manager_kube-controller-manager-master_kube-system_ac1
b8e6a0e173dd	Up About an hour	k8s_weave_weave-net-8dzwb_kube-system_22cd7993-2f2d-11e9-a2a6-0242ac110021
18e47bad320e	Up About an hour	k8s_weave-npc_weave-net-8dzwb_kube-system_22cd7993-2f2d-11e9-a2a6-0242ac110
4d087daf0380	Exited (1) About an hour ago	k8s_weave_weave-net-8dzwb_kube-system_22cd7993-2f2d-11e9-a2a6-0242ac110021
e923140101a3	Up About an hour	k8s_kube-proxy_kube-proxy-cdmlz_kube-system_22cd267f-2f2d-11e9-a2a6-0242ac1
e0db7e63d18e	Up About an hour	k8s_POD_weave-net-8dzwb_kube-system_22cd7993-2f2d-11e9-a2a6-0242ac110021_0
74c257366f65	Up About an hour	k8s_POD_kube-proxy-cdmlz_kube-system_22cd267f-2f2d-11e9-a2a6-0242ac110021_0
8f514eac9d04	Exited (255) 40 minutes ago	k8s_kube-controller-manager_kube-controller-manager-master_kube-system_ac1
b39c5c594913	Exited (1) 40 minutes ago	k8s_kube-scheduler_kube-scheduler-master_kube-system_009228e74aef4d7babd790
3aefcb20ed30	Up 2 hours	k8s_POD_kube-apiserver-master_kube-system_8758a3d10776bb527e043fccfc835986
576c8a273b50	Up 2 hours	k8s_POD_kube-controller-manager-master_kube-system_ac1d4c5ae0fbe553b664a6c
4b3c5f34efde	Up 2 hours	k8s_POD_kube-scheduler-master_kube-system_009228e74aef4d7babd7968782118d5e

# View Logs

▶ docker logs 87fc

```
2019-02-13 02:53:28.144631 I | etcdmain: etcd Version: 3.2.18
2019-02-13 02:53:28.144680 I | etcdmain: Git SHA: eddf599c6
2019-02-13 02:53:28.144684 I | etcdmain: Go Version: go1.8.7
2019-02-13 02:53:28.144688 I | etcdmain: Go OS/Arch: linux/amd64
2019-02-13 02:53:28.144692 I | etcdmain: setting maximum number of CPUs to 4, total number of available CPUs is 4
2019-02-13 02:53:28.144734 N | etcdmain: the server is already initialized as member before, starting as etcd
member...
2019-02-13 02:53:28.146625 I | etcdserver: name = master
2019-02-13 02:53:28.146637 I | etcdserver: data dir = /var/lib/etcd
2019-02-13 02:53:28.146642 I | etcdserver: member dir = /var/lib/etcd/member
2019-02-13 02:53:28.146645 I | etcdserver: heartbeat = 100ms
2019-02-13 02:53:28.146648 I | etcdserver: election = 1000ms
2019-02-13 02:53:28.146651 I | etcdserver: snapshot count = 10000
2019-02-13 02:53:28.146677 I | etcdserver: advertise client URLs = 2019-02-13 02:53:28.185353 I | etcdserver/api: enabled capabilities for version 3.2
2019-02-13 02:53:28.185588 I | embed: ClientTLS: cert = /etc/kubernetes/pki/etcd/server.crt, key = /etc/kubernetes/pki/etcd/server.key, ca = , trusted-ca = /etc/kubernetes/pki/etcd/old-ca.crt, client-cert-auth = true
2019-02-13 02:53:30.080017 I | embed: ready to serve client requests
2019-02-13 02:53:30.080130 I | etcdserver: published {Name:master ClientURLs:[https://127.0.0.1:2379]} to cluster c9be114fc2da2776
2019-02-13 02:53:30.080281 I | embed: serving client requests on 127.0.0.1:2379
WARNING: 2019/02/13 02:53:30 Failed to dial 127.0.0.1:2379: connection error: desc = "transport: authentication handshake failed: remote error: tls: bad certificate"; please retry.
```

# Practice Test



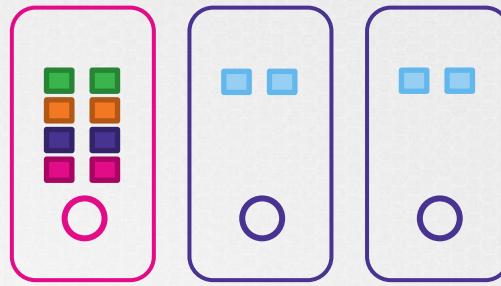
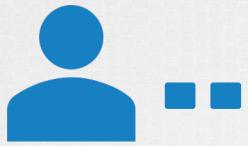
{KODE} {LOUD}

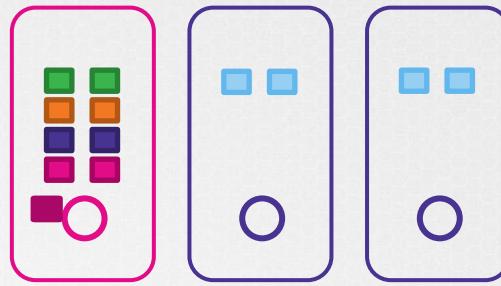
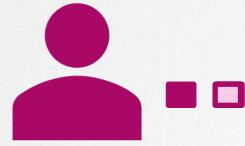
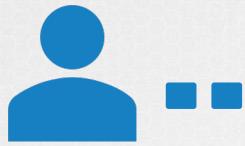
[www.kodekloud.com](http://www.kodekloud.com)



# TLS CERTIFICATES

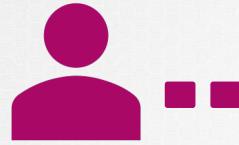
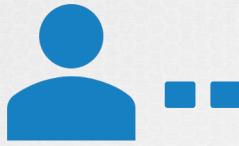
Certificate Workflow & API





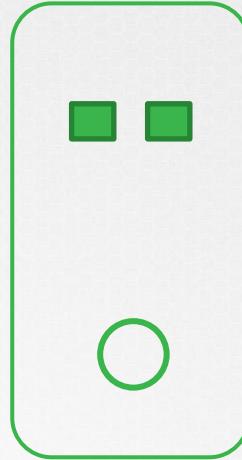
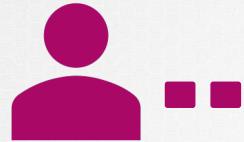
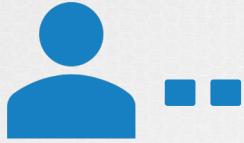


# CERTIFICATE AUTHORITY (CA)





# CERTIFICATE AUTHORITY (CA)

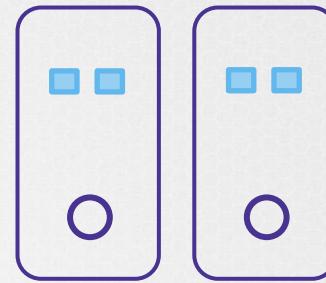
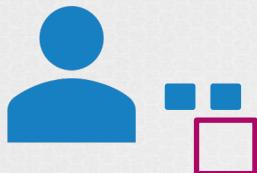


1. Create CertificateSigningRequest Object

2. Review Requests

3. Approve Requests

4. Share Certs to Users





```
▶ openssl genrsa -out jane.key 2048
jane.key

▶ openssl req -new -key jane.key -subj "/CN=jane" -out jane.csr
jane.csr

-----BEGIN CERTIFICATE REQUEST-----
MIICWDCCAUACQAwEzERMA8GA1UEAwIBmV3LXVzZXIwggEiMA0GCSqGSIb3DQEB
AQAA4IBDwAwggEKAoIBAQD00WJW+DXsAJSIrjpNo5vRIBp1nzg+6xc9+UVwkKi0
LFC27t+1eEn0N5Muq99NevmMEOnrDUO/thyVqP2w2XNIDRXjYyF40fbmD+5zlyCK
9w0BAQsFAAOCAQEAS9iS6C1uxTuf5BBYSU7QFQHUza1NxAdYsaORRQNwHZwHqGi4
hOK4a2zyNyj4400ijyaD6tUW8DSxkr8BLK8Kg3srREtJql5rLZy9LRVrsJghD4gY
P9NL+aDRSxROVSqBaB2nWeYpM5cJ5TF531esNSNMLQ2++RMnjDQJ7juPEic8/dhk
Wr2EUM6UawzykrdHImwTv2mIMY0R+DntV1Yie+0H9/YE1t+FSGjh5LYUvI1Dqiy
413E/y3qL71WFAcuH3OsVpUUnQISMdQs0qWCsbE56CC5DhPGZIpUbnKUpAwka+8E
vwQ07jG+hpknxmuFAeXxgUwodALaJ7ju/TDIcw==
-----END CERTIFICATE REQUEST-----
```



## jane.csr

```
-----BEGIN CERTIFICATE REQUEST-----
MIICWDCCAUAQAwEzERMA8GA1UEAwIBmV3LXVzZXIwggEiMA0GCSqGSIb3DQE
B
AQUAA4IBDwAwggEKAoIBAQD00WJWxDxsAJSIrjpNo5vRIBp1n zg+6x9+UVwK
Ki0
Lfc27t+1eEnON5Muq99NevmMEOnrDUO/thyVqp2w2XNIDRXjYyF40FbmD+5zW
yCK
9w0BAQsFAAOCAQEAS9iS6C1uxTu5BBYSU7QFQHuzalNxAdYsaORRQNwHZwHqG
i4
hOK4a2zyNyia4400ijyaD6tUW8DSxkr8BLK8Kg3srREtJq15rLZy9LRVrsJgh
D4gY
P9NL+aDRSxROVSqBaB2nWeYpM5cJ5TF531esNSNMLQ2++RMnjDQJ7juPEic8/dhk
Wr2EUM6uawzykrdHIwTv2m1MY0R+DntV1Yie+0H9/YElt+FSGjh5L5YUvI1Dqiy
413E/y3ql71WfAcuH3OsVpuUnQISMdQs0qNCsbE56CC5DhPGZIpUbnKUpAwka+8E
vwQ07jG+hpknxmuFAeXxgUwodALaJ7ju/TDIcw==
-----END CERTIFICATE REQUEST-----
```



cat jane.csr | base64

```
LS0tLS1CRUdJTiBDRVJUSUZJQ0FEURSBURFVRVNULS0
EL5OKTUI5Q1dEQ0NBVFQVFBD0V6RVJNQTHAQTFVRU
F3d0TRibVYQZTFHWeTpYSXdnZ0VptUEwRONTudTSWIzR
FFSgpbUWBQTRJQKR3QXdndZ0VLQW9JQKFRRB8wV0px
K0RYC0FKU01yaB0bzV2UkQCcGxuemcNnhj0StVVnd
p82kwCkxmQzIBdCsxZUVuT04ITXVxOT10ZXztTUVPbn
Jnhj0StVVndrS2kwCkxmQzI3dCsxZUVuT0
41TXVxOT10ZXztTUVPbnJ
```



## jane-csr.yaml

```
apiVersion: certificates.k8s.io/v1beta1
kind: CertificateSigningRequest
metadata:
  name: jane
spec:
  groups:
  - system:authenticated
  usages:
  - digital signature
  - key encipherment
  - server auth
  request:
```

```
▶ kubectl get csr
```

NAME	AGE	REQUESTOR	CONDITION
jane	10m	admin@example.com	
Pending			

```
▶ kubectl certificate approve jane
```

```
jane approved!
```

```
kubectl get csr jane -o yaml
```

```
apiVersion: certificates.k8s.io/v1beta1
kind: CertificateSigningRequest
metadata:
  creationTimestamp: 2019-02-13T16:36:43Z
  name: new-user
spec:
  groups:
    - system:masters
    - system:authenticated
  usages:
    - digital signature
    - key encipherment
    - server auth
  username: kubernetes-admin
status:
```

```
  certificate:
```

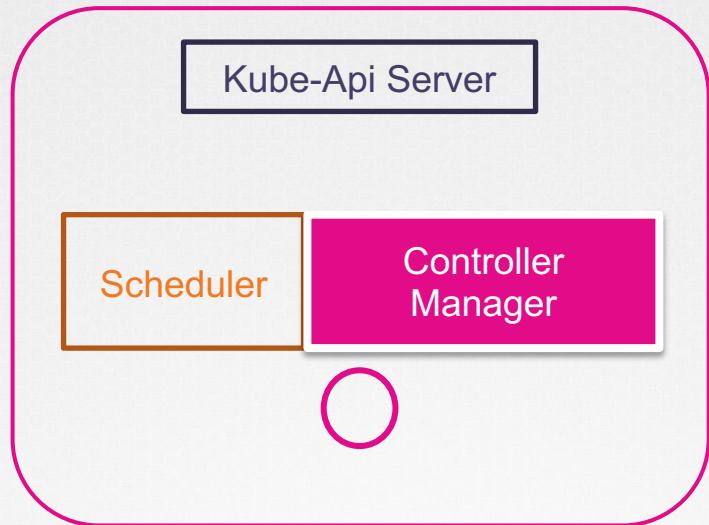
```
LS0tLS1CRUdjTiBDRVJUSUZJQ0FURS0tLS0tCk1JSURDakNDQWZLZ0F3SUJBZ01VRmwy
Q2wxYXoxaWl5M3JNVisreFRYQUowU3ndn0RRWUpLb1pJaHZjTkFRRUwKQ1FBd0ZURVRN
QKVHQTFVRUF4TUthM1ZpWlhKdVpYUmxfekf1RncweE9UQX1NVE14TmpNeU1EQmFGd1dn
Y0ZFeDl2ajNuSXY3eFdDS1NIRm5sU041c0t5Z0VxUkwzTFM5V29GelhHZDdwCmlEZ2FO
MVVRMF BXTVhjN09FVnVjSwc1Yk4weEVHTkVwRU5tdU1BN1ZWeHVjS1h6aG91dDY0MEd1
MGU0YXFKWVIKwmVmBjBvRTFCY3dod2xic0I1ND0KLS0tLS1FTkQgQ0VSVE1GSUNBVUEut
LS0tLQo=
```

```
  conditions:
```

- lastUpdateTime: 2019-02-13T16:37:21Z
- message: This CSR was approved by kubectl certificate approve.
- reason: KubectlApprove
- type: Approved

```
echo "LS0...Qo=" | base64 --decode
```

```
-----BEGIN CERTIFICATE-----
MIICWDCCAUACAQAwEzERMA8GA1UEAwIBmV3LXVzZXIwgg
AQUAA4IBDwAwggEKAoIBAQD00WJW+DXsAJSIrjpNo5vRIB
Lfc27t+1eEnON5Muq99NevmMEOnrDUO/thyVqP2w2XNIDR
y3BihhB93MJ70ql3UTvZ8TELqyaDknR1/jv/SxgXkok0AB
IF5nxAttMvkDPQ7NbeZRG43b+QW1VGR/z6DW0fJnbfez0t
EcCXAwqChjBLkz2BHP R4J89D6Xb8k39pu6jpyngV6uP0tI
j2qEL+hZEWkkFz801NNtyT5LxMqENDCnIgwC4GZiRGbrAg
9w0BAQsFAAOCAQEAS9iS6C1uxTuf5BBYSU7QFQHUza1NxA
hOK4a2zyNy i4400ijyaD6tUW8DSxkr8BLK8Kg3srREtJql
P9NL+aDRSxROVSqBaB2nWeYpM5cJ5TF53lesNSNMLQ2++R
Wr2EUM6UawzykrdHImwTv2m1MY0R+DntV1Yie+0H9/YE1t
413E/y3qL71WfAcuH3OsVpUUnQISMdQs0qWCsbE56CC5Dh
vwQ07jG+hpknxmuFAeXxgUwodALaJ7ju/TDIcw==
-----END CERTIFICATE-----
```



Controller Manager

CSR-  
APPROVING

CSR-SIGNING

```
▶ cat /etc/kubernetes/manifests/kube-controller-manager.yaml
```

```
spec:  
  containers:  
    - command:  
        - kube-controller-manager  
        - --address=127.0.0.1  
        - --cluster-signing-cert-file=/etc/kubernetes/pki/ca.crt  
        - --cluster-signing-key-file=/etc/kubernetes/pki/ca.key  
        - --controllers=*,bootstrapsigner,tokencleaner  
        - --kubeconfig=/etc/kubernetes/controller-manager.conf  
        - --leader-elect=true  
        - --root-ca-file=/etc/kubernetes/pki/ca.crt  
        - --service-account-private-key-file=/etc/kubernetes/pki/sa.key  
        - --use-service-account-credentials=true
```

## Client Certificates for Clients



kube-scheduler

kube-controller-  
manager

kubelet

kube-proxy

## Server Certificates for Servers

kube-apiserver

kubelet

ETCD  
CLUSTE  
R

- Kube-api.service
- CA CERT for ETCD
  - CERT for ETCD
  - KEY for ETCD
  - CA CERT for KUBEL
  - CERT for KUBELET C
  - KEY for KUBELET C
  - CERT for Service Acc
  - CERT for TLS
  - KEY for TLS

- Kubelet-config.yaml
- tlsCertFile
  - tlsPrivateKeyFile



# References

<https://kubernetes.io/docs/tasks/administer-cluster/securing-a-cluster/>

<https://kubernetes.io/docs/setup/certificates/>

<https://kubernetes.io/docs/tasks/administer-cluster/kubeadm/kubeadm-certs/>

<https://github.com/kubernetes/kubernetes/blob/495ee5ea40becadaba0babbfe7808a507e872eefb/cmd/kubeadm/app/constants/constants.go>

<https://github.com/coreos/coreos-kubernetes/blob/master/Documentation/openssl.md>

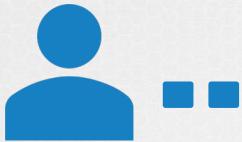


{KODE} {LOUD}

[www.kodekloud.com](http://www.kodekloud.com)

# Security KUBECONFIG





```
▶ curl https://my-kube-playground:6443/api/v1/pods \
  --key admin.key
  --cert admin.crt
  --cacert ca.crt
```

```
{
  "kind": "PodList",
  "apiVersion": "v1",
  "metadata": {
    "selfLink": "/api/v1/pods",
  },
  "items": []
}
```

```
▶ kubectl get pods
  --server my-kube-playground:6443
  --client-key admin.key
  --client-certificate admin.crt
  --certificate-authority ca.crt
```

No resources found.

```
SHOME/.kube/config
```

## KubeConfig File

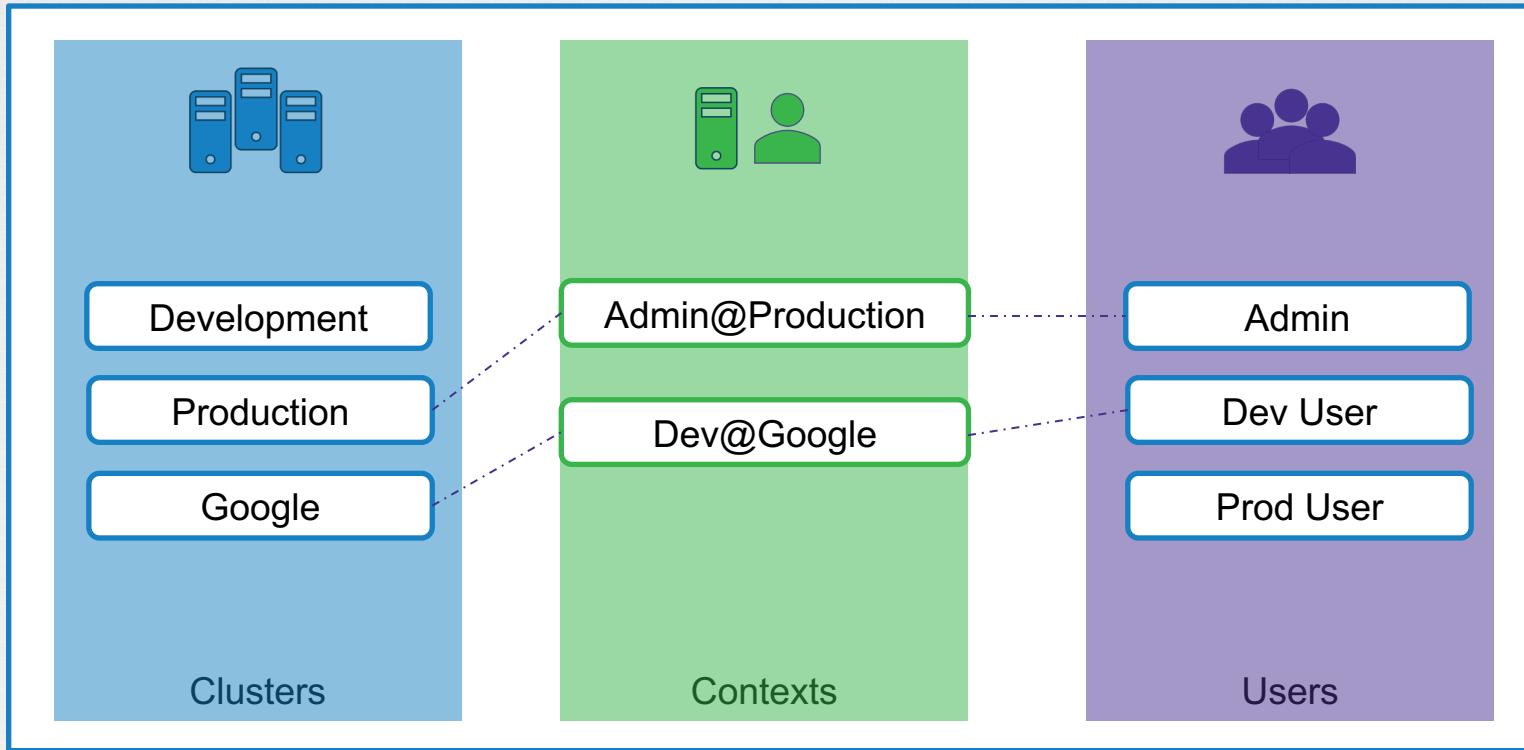
```
--server my-kube-playground:6443  
--client-key admin.key  
--client-certificate admin.crt  
--certificate-authority ca.crt
```

```
▶ kubectl get pods  
--kubeconfig config
```

```
No resources found.
```

# KubeConfig File

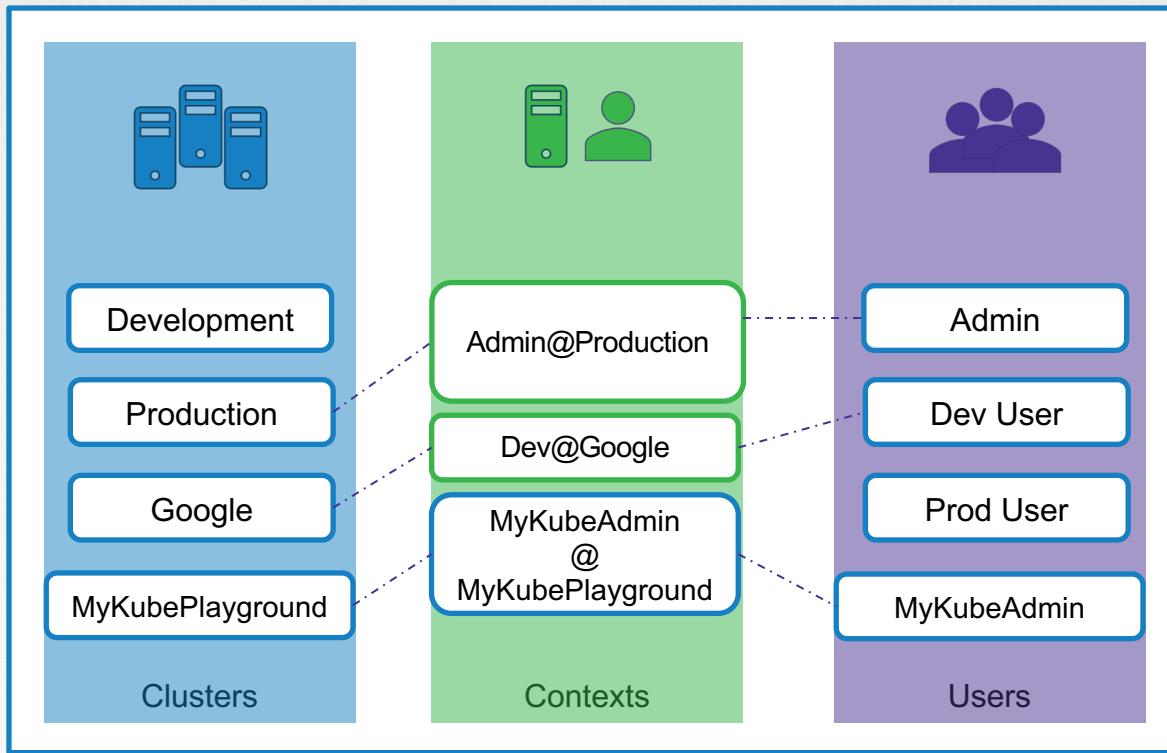
\$HOME/.kube/config



# KubeConfig File

\$HOME/.kube/config

```
--server my-kube-playground:6443  
--client-key admin.key  
--client-certificate admin.crt  
--certificate-authority ca.crt
```



# KubeConfig File

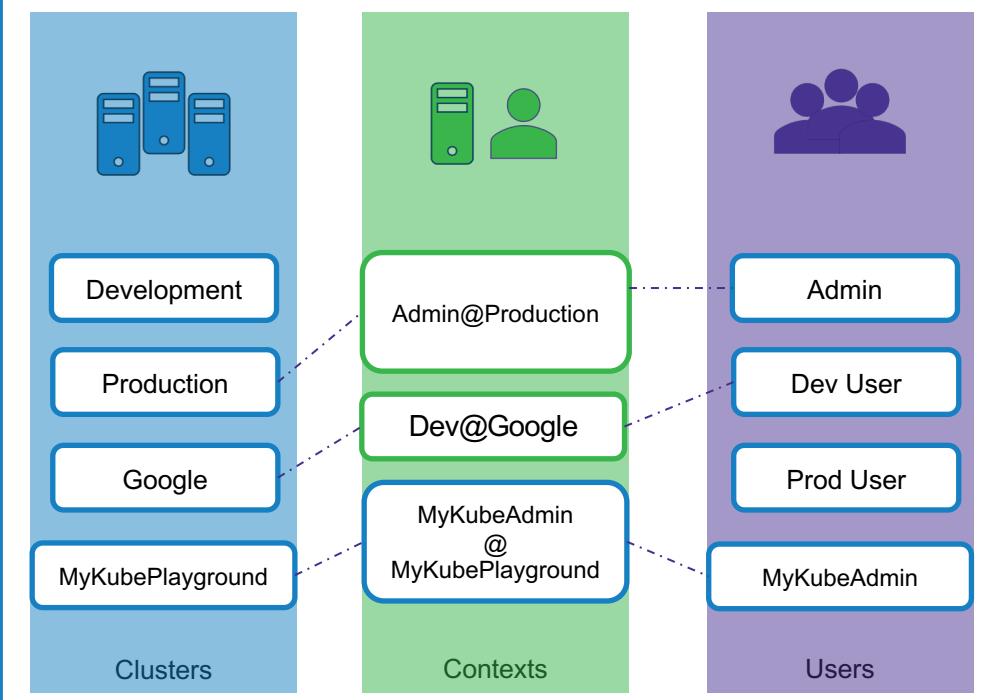
```
apiVersion: v1
kind: Config

clusters:
- name: my-kube-playground
  cluster:
    certificate-authority: ca.crt
    server: https://my-kube-playground:6443

contexts:
- name: my-kube-admin@my-kube-playground
  context:
    cluster:
      user:
        name: my-kube-admin

users:
- name: my-kube-admin
  user:
    client-certificate: admin.crt
    client-key: admin.key
```

\$HOME/.kube/config



# KubeConfig File

```
apiVersion: v1
kind: Config

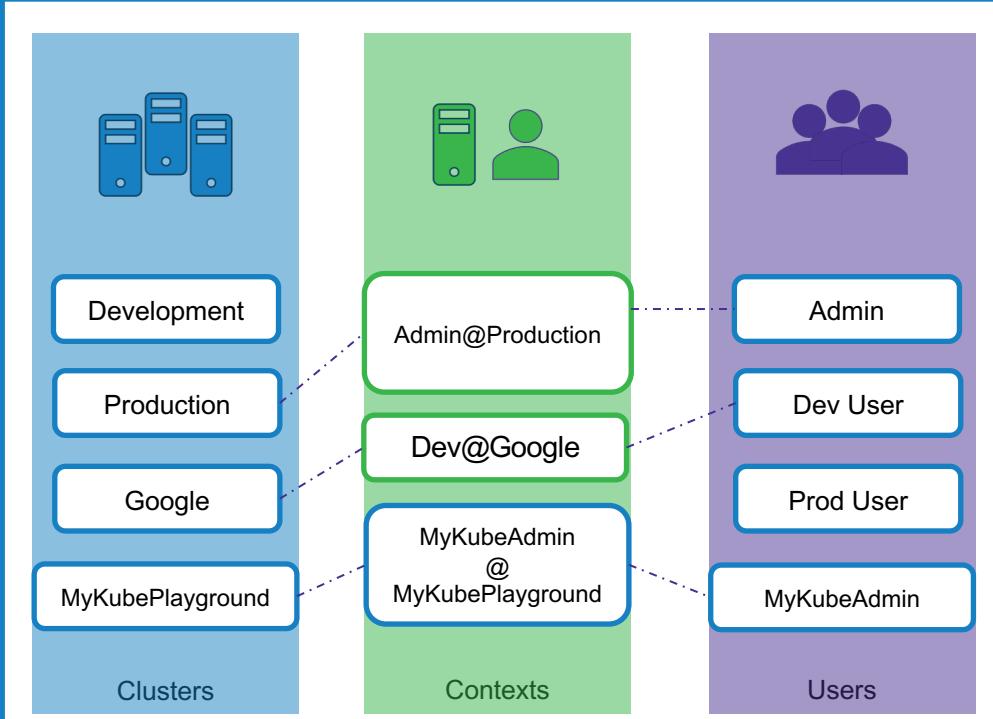
current-context: dev-user@google

clusters:
- name: my-kube-playground  (values hidden...)
- name: development
- name: production
- name: google

contexts:
- name: my-kube-admin@my-kube-playground
- name: dev-user@google
- name: prod-user@production

users:
- name: my-kube-admin
- name: admin
- name: dev-user
- name: prod-user
```

\$HOME/.kube/config



# Kubectl config

```
kubectl config view
```

```
apiVersion: v1

kind: Config
current-context: kubernetes-admin@kubernetes

clusters:
- cluster:
  certificate-authority-data: REDACTED
  server: https://172.17.0.5:6443
  name: kubernetes

contexts:
- context:
  cluster: kubernetes
  user: kubernetes-admin
  name: kubernetes-admin@kubernetes

users:
- name: kubernetes-admin
  user:
    client-certificate-data: REDACTED
    client-key-data: REDACTED
```

```
kubectl config view -kubeconfig=my-custom-config
```

```
apiVersion: v1

kind: Config
current-context: my-kube-admin@my-kube-playground

clusters:
- name: my-kube-playground
- name: development
- name: production

contexts:
  name: my-kube-admin@my-kube-playground
  Name: prod-user@production

users:
- name: my-kube-admin
- name: prod-user
```

# Kubectl config

```
▶ kubectl config view
```

```
apiVersion: v1

kind: Config
current-context: my-kube-admin@my-kube-playground

clusters:
- name: my-kube-playground
- name: development
- name: production

contexts:
  name: my-kube-admin@my-kube-playground
  Name: prod-user@production

users:
- name: my-kube-admin
- name: prod-user
```

```
▶ kubectl config use-context prod-user@production
```

```
apiVersion: v1

kind: Config
current-context: prod-user@production

clusters:
- name: my-kube-playground
- name: development
- name: production

contexts:
  name: my-kube-admin@my-kube-playground
  Name: prod-user@production

users:
- name: my-kube-admin
- name: prod-user
```

# Kubectl config

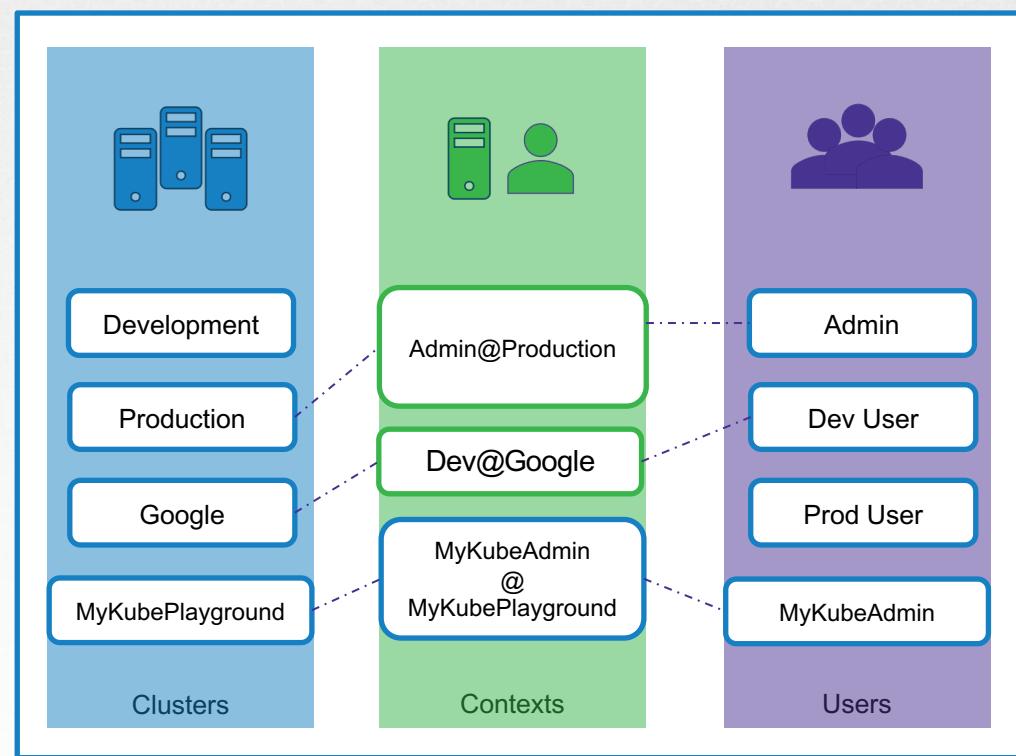
```
kubectl config -h
```

Available Commands:

current-context	Displays the current-context
delete-cluster	Delete the specified cluster from the kubeconfig
delete-context	Delete the specified context from the kubeconfig
get-clusters	Display clusters defined in the kubeconfig
get-contexts	Describe one or many contexts
rename-context	Renames a context from the kubeconfig file.
set	Sets an individual value in a kubeconfig file
set-cluster	Sets a cluster entry in kubeconfig
set-context	Sets a context entry in kubeconfig
set-credentials	Sets a user entry in kubeconfig
unset	Unsets an individual value in a kubeconfig file
use-context	Sets the current-context in a kubeconfig file
view	Display merged kubeconfig settings or a specified kubeconfig file

# Namespaces

\$HOME/.kube/config



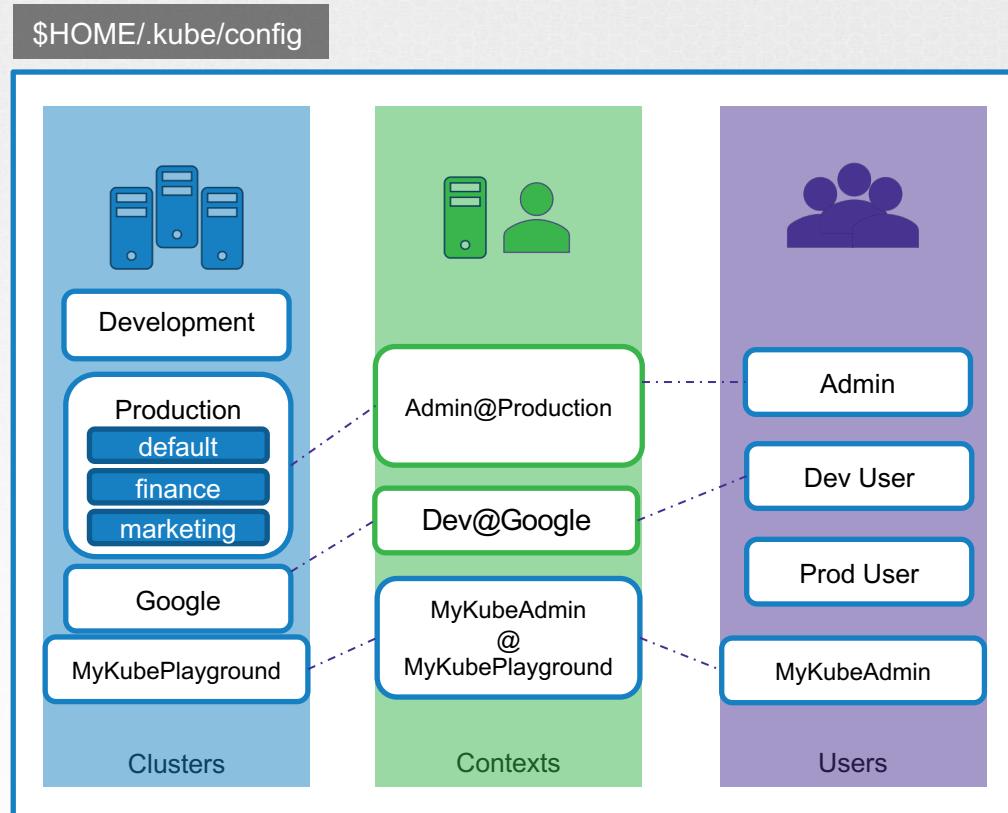
# Namespaces

```
apiVersion: v1
kind: Config

clusters:
- name: production
  cluster:
    certificate-authority: ca.crt
    server: https://172.17.0.51:6443

contexts:
- name: admin@production
  context:
    cluster: production
    user: admin
    namespace: finance

users:
- name: admin
  user:
    client-certificate: admin.crt
    client-key: admin.key
```



# Certificates in KubeConfig

```
apiVersion: v1
kind: Config

clusters:
- name: production
  cluster:
    certificate-authority: /etc/kubernetes/pki/ca.crt
    server: https://172.17.0.51:6443

contexts:
- name: admin@production
  context:
    cluster: production
    user: admin
    namespace: finance

users:
- name: admin
  user:
    client-certificate: /etc/kubernetes/pki/users/admin.crt
    client-key: /etc/kubernetes/pki/users/admin.key
```

# Certificates in KubeConfig

```
apiVersion: v1
kind: Config

clusters:
- name: production
  cluster:
    certificate-authority: /etc/kubernetes/pki/ca.crt

  certificate-authority-data:
```

```
-----BEGIN CERTIFICATE-----
MIICWDCCAUACAQAwEzERMA8GA1UEAwIbmV3LXVzZXIwggEiMA0G
AQUAA4IBDwAwggEKAoIBAQDO0WJW+DXsAJSIjpNo5vRIBplnzb+e
Lfc27t+1eEnON5Muq99NevmMEOnrDUO/thyVqP2w2XNIDRXjYyF4G
y3BihhB93MJ70ql3UTvZ8TELqyaDknR1/jv/SxgXkok0ABUTpWMx4
IF5nxAttMvkDPQ7NbeZRG43b+QW1VGR/z6DWOfJnbfezOtaAydGL
EcCXAwqChjBLkz2BHP4J89D6Xb8k39pu6jpyngV6uP0tIbOzpqn
j2qEL+hZEWkkFz801NNTyT5LxMqENDCnIgwC4GZiRGbrAgMBAAGG
9w0BAQsFAAOCAQEAS9iS6C1uxTuf5BBYSU7QFQHUza1NxAdYsaORR
hOK4a2zyNyij4400ijyaD6tUW8DSxkr8BLK8Kg3srRETJql15rLz
P9NL+aDRSxROVSqBaB2nWeYpM5cJ5TF531esNSNMLQ2++RMnjDQJ
Wr2EUM6UawzykrdHImwTv2m1MY0R+DntV1Yie+0H9/YEl+fSGjh5
413E/y3qL71WfAcuH3OsVpUUnQISMdQs0qWCsbE56CC5DhPGZIpUb
vwQ07jG+hpknxmuFAeXgxUwodAlaJ7ju/TDIcw==
-----END CERTIFICATE-----
```

```
▶ cat ca.crt | base64
LS0tLS1CRUdJTlBDRVJUSUZJQ0FURSBDRVJUSUZJQ0FURSB
tLS0tKUlJQ1dEQQNBVUFDoVFBd0V6RVJNOThHOT
F3d0libWVztFhw1pYSxdnZ0VptUEwR0NTcUDTS
FFFQgbPUW/BQTJQkR3QXdnZ0VQW9JQkFRRE8w
K0RYC0FKU01yanBObzV2Uk1CcGxuemcrNnhjOST
rS2kwCkxmQzI3dCsxZUVuT041TXVx0t10zxztTU
J U01yanBObzV2Uk1CcGxuemcrNnhjOSTV
VndrS2kwCkxmQzI3dCsxZUVuT041TXVx0t10zxztTU
```

# Certificates in KubeConfig

```
apiVersion: v1
kind: Config

clusters:
- name: production
  cluster:
    certificate-authority: /etc/kubernetes/pki/ca.crt

  certificate-authority-data: LS0tLS1CRUdJTiBDRVJU
    SUZJQ0FURSBSRVFVRVNULS0tLS0KTUlJ
    Q1dEQ0NBVUFDQVFb0V6RVJNQThHQTFV
    RUF3d0libVYzTFhWelpYSXdnZ0VpTUEw
    R0NTcUdT SWIzRFFFQgpBUVVBQTRJQkR3
    QXd nZ0VLQW9JQkFRRE8wV0pXK0RYc0FK
    U0lyanBObzV2Uk1CcGxuemcrNnhj0StV
    VndrS2kwCkxmQzI3dCsxZUVuT041TXVx
    OTl0ZXZtTUVPbnJ
```

```
echo "LS0...bnJ" | base64 --decode
-----BEGIN CERTIFICATE-----
MIICWDCCAUACAQAwEzERMA8GA1UEAwIBmV3LXVzZXIwggEiMA0GCSqAQUAA4IBDwAwggEKAoIBAQDO0WJW+DXsAJSIRjpNo5vRIBplnzg+6xcLfc27t+1eEn0N5Muq99NevmMEOnrDU0/thyVqP2w2XNIDRXjYf40Fby3BiHb93MJ70ql3UTvZ8TELqyaDknR1/jv/SxgXkok0ABUTpwMx4BpIF5nxAttMvkDPQ7NbeZRG43b+QNLVGR/z6DWOfJnbfezOtaAydGLTZFEcCXAwqChjBLkz2BHPR4J89D6xb8k39pu6jpyngV6uP0tIbOzpqnV0Yj2qEL+hZEWkkFz801NNtyT5LxMqENDCnIgwC4GZiRGbrAgMBAAGgADAn9w0BAQsFAAOCAQEAS9iS6C1uxTuf5BBYSU7QFQHUzalNxAdYsaORRQNh0K4a2zyNy i4400ijyaD6tUW8DSxkr8BLK8Kg3srREtJq15rLZy9LRVP9NL+aDRSxROVSqBaB2nWeYpM5cJ5TF531esNSNMLQ2++RMnjDQJ7juWr2EUM6UawzykrdHImwTv2m1MY0R+DNtV1Yie+0H9/YE1t+FSGjh5L5413E/y3qL71WfAcuH30sVpUUnQISMdQs0qWCsbE56CC5DhPGZIpUbnnvwQ07jG+hpknxmuFAeXxgUwodALaJ7ju/TDIcW==
-----END CERTIFICATE-----
```

# Reference

<https://kubernetes.io/docs/concepts/configuration/organize-cluster-access-kubeconfig/>



{KODE} {LOUD}

[www.kodekloud.com](http://www.kodekloud.com)

# API Groups

Pre-Requisite



```
curl https://kube-master:6443/version
```

```
{  
    "major": "1",  
    "minor": "13",  
    "gitVersion": "v1.13.0",  
    "gitCommit": "ddf47ac13c1a9483ea035a79cd7c10005ff21a6d",  
    "gitTreeState": "clean",  
    "buildDate": "2018-12-03T20:56:12Z",  
    "goVersion": "go1.11.2",  
    "compiler": "gc",  
    "platform": "linux/amd64"  
}
```

```
curl https://kube-master:6443/api/v1/pods
```

```
{  
    "kind": "PodList",  
    "apiVersion": "v1",  
    "metadata": {  
        "selfLink": "/api/v1/pods",  
        "resourceVersion": "153068"  
    },  
    "items": [  
        {  
            "metadata": {  
                "name": "nginx-5c7588df-ghsb",  
                "generateName": "nginx-5c7588df-",  
                "namespace": "default",  
                "creationTimestamp": "2019-03-20T10:57:48Z",  
                "labels": {  
                    "app": "nginx",  
                    "pod-template-hash": "5c7588df"  
                },  
                "ownerReferences": [  
                    {  
                        "apiVersion": "apps/v1",  
                        "kind": "ReplicaSet",  
                        "name": "nginx-5c7588df",  
                        "uid": "398ce179-4af9-11e9-beb6-020d3114c7a7",  
                        "controller": true,  
                        "blockOwnerDeletion": true  
                    }  
                ]  
            },  
            "status": {  
                "phase": "Running",  
                "conditions": [  
                    {  
                        "type": "Ready",  
                        "status": "True",  
                        "lastProbeTime": null,  
                        "lastTransitionTime": "2019-03-20T10:57:48Z"  
                    }  
                ],  
                "podIP": "10.244.1.11",  
                "ip": "10.244.1.11",  
                "containerStatuses": [  
                    {  
                        "name": "nginx",  
                        "state": {  
                            "running": {  
                                "startedAt": "2019-03-20T10:57:48Z"  
                            }  
                        },  
                        "lastState": {},  
                        "ready": true,  
                        "restartCount": 0,  
                        "image": "nginx:1.14.2",  
                        "imageID": "sha256:3f3a2a2a2a2a2a2a2a2a2a2a2a2a2a2a",  
                        "containerID": "ContainerID-REDACTED",  
                        "containerIDList": ["ContainerID-REDACTED"],  
                        "livenessProbe": {  
                            "exec": {  
                                "command": ["curl", "-f", "http://localhost:80"]  
                            },  
                            "initialDelaySeconds": 0,  
                            "periodSeconds": 10,  
                            "timeoutSeconds": 5,  
                            "failureThreshold": 3  
                        },  
                        "readinessProbe": {  
                            "httpGet": {  
                                "port": 80, "path": "/"  
                            },  
                            "initialDelaySeconds": 0,  
                            "periodSeconds": 10,  
                            "timeoutSeconds": 5,  
                            "failureThreshold": 3  
                        },  
                        "restartWebhook": {  
                            "webhook": "http://127.0.0.1:10250/restart/  
                        }  
                    }  
                ]  
            }  
        }  
    ]  
}
```

/metrics

/healthz

/version

/api

/apis

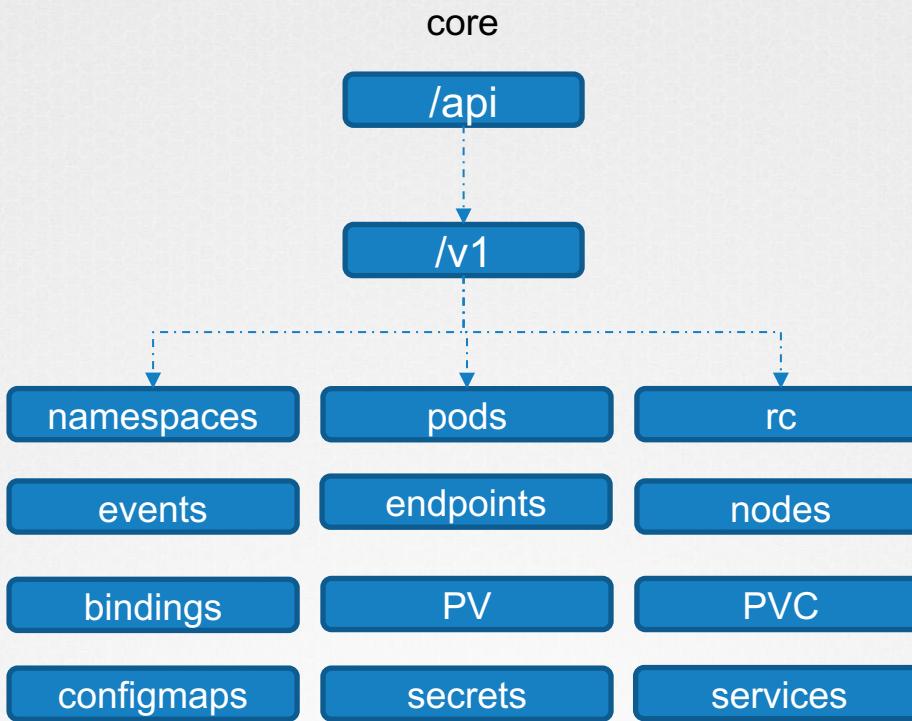
/logs

core

named

/api

/apis



named

`/apis`

### API Groups

`/apps`    `/extensions`    `/networking.k8s.io`    `/storage.k8s.io`    `/authentication.k8s.io`    `/certificates.k8s.io`

`/v1``/v1``/v1``/deployments``/replicasets``/statefulsets``/networkpolicies``/certificatesigningrequests`

### Resources

`list``get``create``delete``update``watch`

### Verbs

# Pod v1 core

kubectl example

curl example

## Group

## Version

core

v1

### ⚠ Warning:

It is recommended that users create Pods only through a Controller, and not directly. See Controllers: Deploy

### ⓘ Appears In:

- PodList [core/v1]

## Field

## Description

apiVersion

string

APIVersion defines the versioned schema of this representation of an object. Servers should c

<https://git.k8s.io/community/contributors/devel/api-conventions.md#resources>

## Overview

## WORKLOADS APIs

Container v1 core

CronJob v1beta1 batch

DaemonSet v1 apps

Deployment v1 apps

Job v1 batch

## Pod v1 core

Write Operations

Read Operations

Status Operations

Proxy Operations

Misc Operations

ReplicaSet v1 apps

ReplicationController v1 core

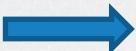
StatefulSet v1 apps

```
▶ curl http://localhost:6443 -k
```

```
{  
  "paths": [  
    "/api",  
    "/api/v1",  
    "/apis",  
    "/apis/",  
    "/healthz",  
    "/logs",  
    "/metrics",  
    "/openapi/v2",  
    "/swagger-2.0.0.json",
```

```
▶ curl http://localhost:6443/apis -k | grep "name"
```

```
"name": "extensions",  
"name": "apps",  
"name": "events.k8s.io",  
"name": "authentication.k8s.io",  
"name": "authorization.k8s.io",  
"name": "autoscaling",  
"name": "batch",  
"name": "certificates.k8s.io",  
"name": "networking.k8s.io",  
"name": "policy",  
"name": "rbac.authorization.k8s.io",  
"name": "storage.k8s.io",  
"name": "admissionregistration.k8s.io",  
"name": "apiextensions.k8s.io",  
"name": "scheduling.k8s.io",
```



Kube ApiServer

```
▶ curl http://localhost:6443 -k
{
  "kind": "Status",
  "apiVersion": "v1",
  "metadata": {

  },
  "status": "Failure",
  "message": "forbidden: User \\"system:anonymous\\" cannot get path \"/\"",
  "reason": "Forbidden",
  "details": {

  },
  "code": 403
}
```



```
▶ curl http://localhost:6443 -k
```

```
--key admin.key
--cert admin.crt
--cacert ca.crt
```

```
{
  "paths": [
    "/api",
    "/api/v1",
    "/apis",
    "/apis/",
    "/healthz",
    "/logs",
    "/metrics"
  ]
}
```

# kubectl proxy



```
> kubectl proxy
Starting to serve on 127.0.0.1:8001
```

```
> curl http://localhost:8001 -k
{
  "paths": [
    "/api",
    "/api/v1",
    "/apis",
    "/apis/",
    "/healthz",
    "/logs",
    "/metrics",
    "/openapi/v2",
    "/swagger-2.0.0.json",
```

Kube proxy

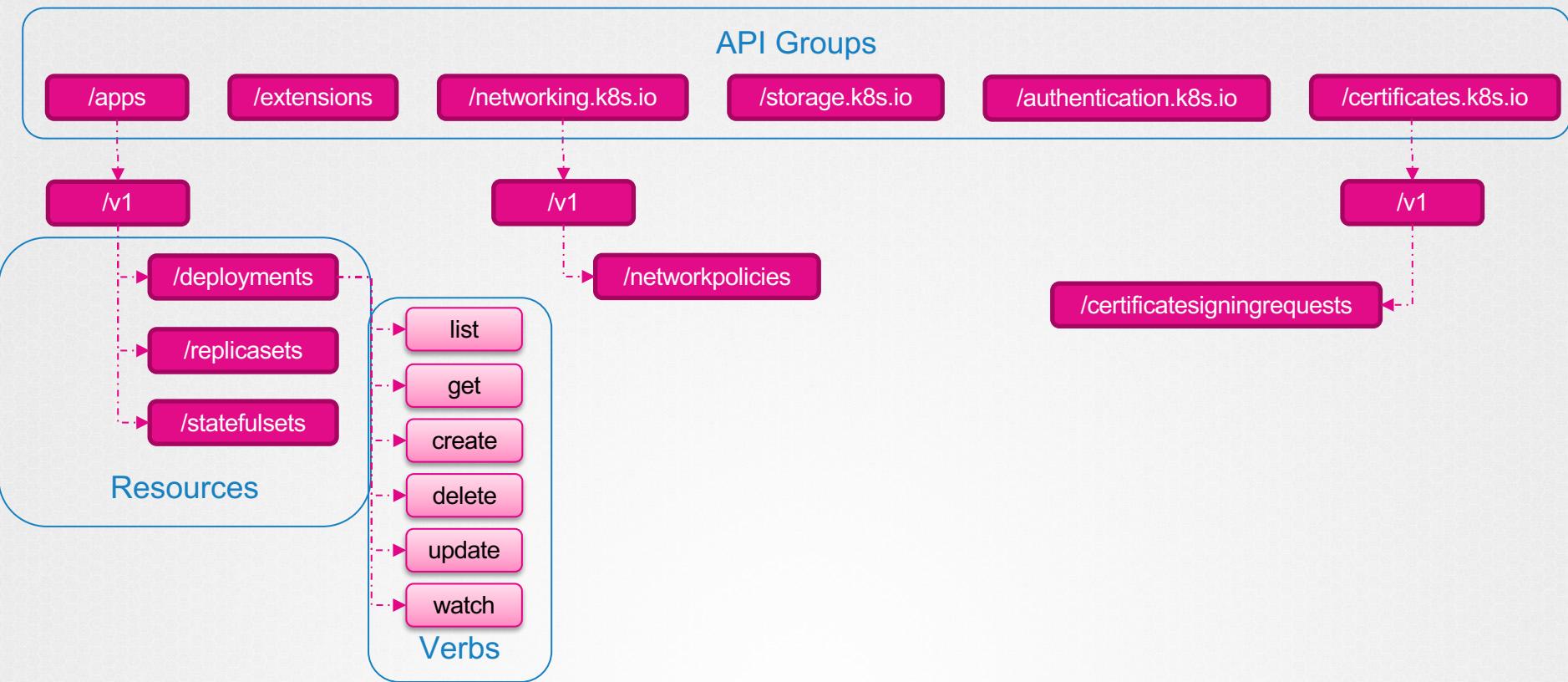


Kubectl proxy

# Key Takeaways

named

/apis



<https://github.com/kubernetes/community/blob/master/contributors/design-proposals/api-machinery/api-group.md>

<https://kubernetes.io/docs/concepts/overview/kubernetes-api/#api-groups>



{KODE} {LOUD}

[www.kodekloud.com](http://www.kodekloud.com)

# AUTHORIZATION



# Why Authorization?



Admins



Developers



Bots

```
▶ kubectl get pods
```

NAME	READY	STATUS	RESTARTS
nginx	1/1	Running	0

53s

```
▶ kubectl get nodes
```

NAME	STATUS	ROLES	AGE	VERSION
worker-1	Ready	<none>	5d21h	v1.13.0
worker-2	Ready	<none>	5d21h	v1.13.0

```
▶ kubectl delete node worker-2
```

Node worker-2 Deleted!

```
▶ kubectl get pods
```

NAME	READY	STATUS	RESTARTS
nginx	1/1	Running	0

53s

```
▶ kubectl get nodes
```

NAME	STATUS	ROLES	AGE	VERSION
worker-1	Ready	<none>	5d21h	v1.13.0
worker-2	Ready	<none>	5d21h	v1.13.0

```
▶ kubectl delete node worker-2
```

Error from server (Forbidden): nodes "worker-1" is forbidden: User "developer" cannot delete resource "nodes"

```
▶ kubectl get pods
```

Error from server (Forbidden): pods is forbidden: User "Bot-1" cannot list "pods"

```
▶ kubectl get nodes
```

Error from server (Forbidden): pods is forbidden: User "Bot-1" cannot get "nodes"

```
▶ kubectl delete node worker-2
```

Error from server (Forbidden): nodes "worker-1" is forbidden: User "Bot-1" cannot delete resource "nodes"

# Authorization Mechanisms

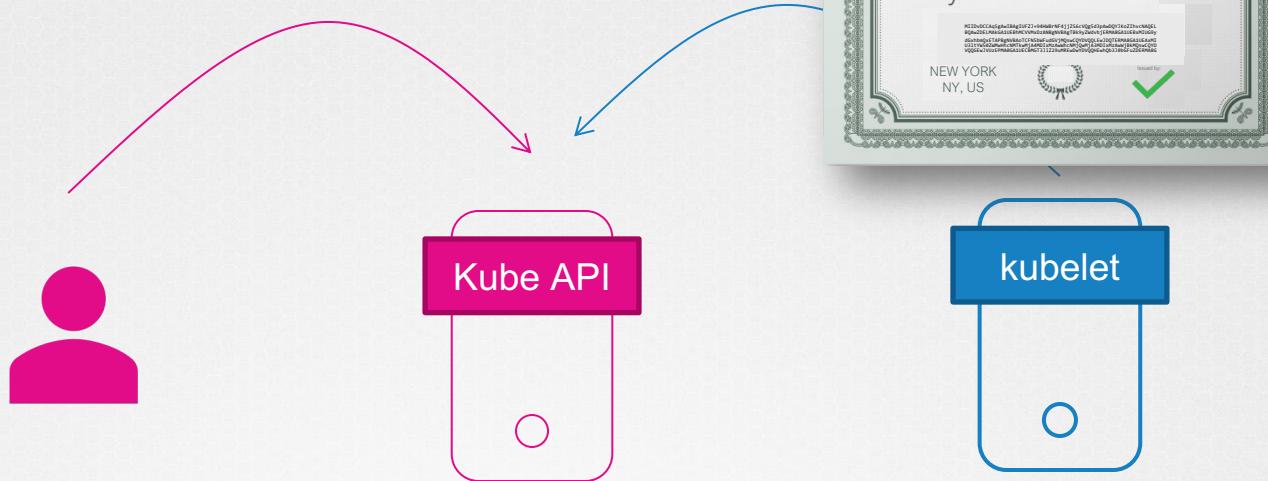
Node

ABAC

RBAC

Webhook

# Node Authorizer



- Read
  - Services
  - Endpoints
  - Nodes
  - Pods
- Write
  - Node status
  - Pod status
  - events

# ABAC



dev-user



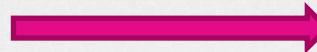
- ✓ Can view PODs
- ✓ Can create PODs
- ✓ Can Delete PODs

```
{"kind": "Policy", "spec": {"user": "dev-user", "namespace": "*", "resource": "pods", "apiGroup": "*"}}
```

# ABAC



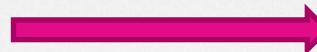
dev-user



- ✓ Can view PODs
- ✓ Can create PODs
- ✓ Can Delete PODs



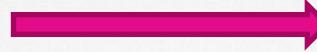
dev-user-2



- ✓ Can view PODs
- ✓ Can create PODs
- ✓ Can Delete PODs



dev-users



- ✓ Can view PODs
- ✓ Can create PODs
- ✓ Can Delete PODs



security-1



- ✓ Can view CSR
- ✓ Can approve CSR

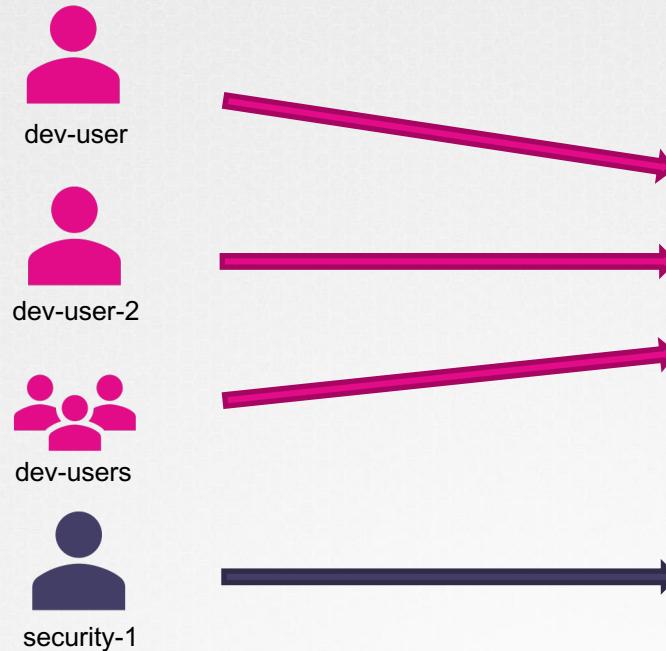
```
{"kind": "Policy", "spec": { "user": "dev-user", "namespace": "*", "resource": "pods", "apiGroup": "*" } }

{"kind": "Policy", "spec": { "user": "dev-user-2", "namespace": "*", "resource": "pods", "apiGroup": "*" } }

{"kind": "Policy", "spec": { "group": "dev-users", "namespace": "*", "resource": "pods", "apiGroup": "*" } }

{"kind": "Policy", "spec": { "user": "security-1", "namespace": "*", "resource": "csr", "apiGroup": "*" } }
```

# RBAC



# Webhook



User **dev-user**  
requested read  
access to **Pods**.  
Should I allow?



Open Policy Agent

I checked. Yes!

# Authorization Mode

NODE

ABAC

RBAC

WEBHOOK

AlwaysAllow

AlwaysDeny

# Authorization Mode

AlwaysAllow

NODE

ABAC

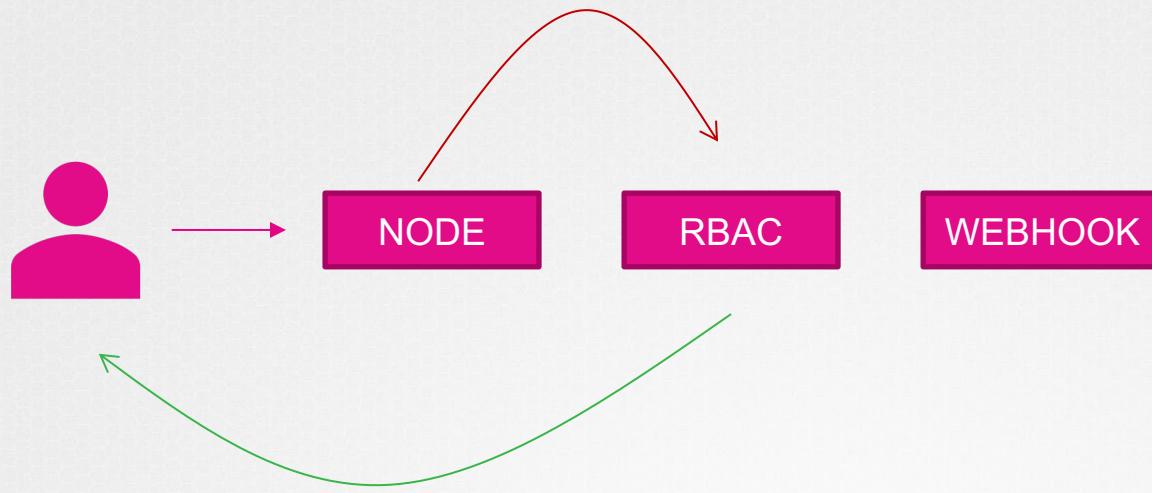
RBAC

WEBHOOK

AlwaysDeny

```
ExecStart=/usr/local/bin/kube-apiserver \
--advertise-address=${INTERNAL_IP} \
--allow-privileged=true \
--apiserver-count=3 \
--authorization-mode=Node,RBAC,Webhook \
--bind-address=0.0.0.0 \
--enable-swagger-ui=true \
--etcd-cafile=/var/lib/kubernetes/ca.pem \
--etcd-certfile=/var/lib/kubernetes/apiserver-etcd-client.crt \
--etcd-keyfile=/var/lib/kubernetes/apiserver-etcd-client.key \
--etcd-servers=https://127.0.0.1:2379 \
--event-ttl=1h \
--kubelet-certificate-authority=/var/lib/kubernetes/ca.pem \
--kubelet-client-certificate=/var/lib/kubernetes/apiserver-etcd-client.crt \
--kubelet-client-key=/var/lib/kubernetes/apiserver-etcd-client.key \
--service-node-port-range=30000-32767 \
--client-ca-file=/var/lib/kubernetes/ca.pem \
--tls-cert-file=/var/lib/kubernetes/apiserver.crt \
--tls-private-key-file=/var/lib/kubernetes/apiserver.key \
--v=2
```

# Authorization Mode



```
ExecStart=/usr/local/bin/kube-apiserver \
    --advertise-address=${INTERNAL_IP} \
    --allow-privileged=true \
    --apiserver-count=3 \
    --authorization-mode=Node,RBAC,Webhook \
    --bind-address=0.0.0.0 \\\
```



{KODE} {LOUD}

[www.kodekloud.com](http://www.kodekloud.com)



# RBAC



# RBAC



- ✓ Can view PODs
- ✓ Can create PODs
- ✓ Can Delete PODs
- ✓ Can Create ConfigMaps

Developer

## developer-role.yaml

```
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: developer
rules:
- apiGroups: []
  resources: ["pods"]
  verbs: ["list", "get", "create", "update", "delete"]
- apiGroups: []
  resources: ["ConfigMap"]
  verbs: ["create"]
```

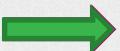


```
kubectl create -f developer-role.yaml
```

# RBAC



dev-user



- ✓ Can view PODs
- ✓ Can create PODs
- ✓ Can Delete PODs
- ✓ Can Create ConfigMaps

**Developer**

Namespace: default

▶ `kubectl create -f devuser-developer-binding.yaml`

developer-role.yaml

Copyright © 2021 KodeKloud

```
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: developer
rules:
- apiGroups: [ "" ]
  resources: [ "pods" ]
  verbs: [ "list", "get", "create", "update", "delete", "patch" ]
- apiGroups: [ "" ]
  resources: [ "ConfigMap" ]
  verbs: [ "create" ]
```

devuser-developer-binding.yaml

```
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: devuser-developer-binding
subjects:
- kind: User
  name: dev-user
  apiGroup: rbac.authorization.k8s.io
roleRef:
  kind: Role
  name: developer
  apiGroup: rbac.authorization.k8s.io
```

# View RBAC

```
▶ kubectl get roles
```

NAME	AGE
developer	4s

```
▶ kubectl get rolebindings
```

NAME	AGE
devuser-developer-binding	24s

```
▶ kubectl describe role developer
```

Name:	developer		
Labels:	<none>		
Annotations:	<none>		
PolicyRule:			
Resources	Non-Resource URLs	Resource Names	Verbs
-----	-----	-----	-----
ConfigMap	[ ]	[ ]	[create]
pods	[ ]	[ ]	[get watch list create delete]

# View RBAC

```
▶ kubectl describe rolebinding devuser-developer-binding  
Name:      devuser-developer-binding  
Labels:    <none>  
Annotations: <none>  
Role:  
  Kind:  Role  
  Name:  developer  
Subjects:  
  Kind  Name      Namespace  
  ----  --  -----  
  User  dev-user
```

# Check Access

```
▶ kubectl auth can-i create deployments
```

```
yes
```

```
▶ kubectl auth can-i delete nodes
```

```
no
```

```
▶ kubectl auth can-i create deployments --as dev-user
```

```
no
```

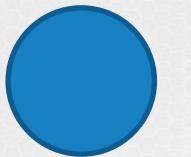
```
▶ kubectl auth can-i create pods --as dev-user
```

```
yes
```

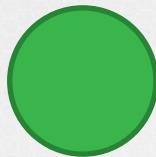
```
▶ kubectl auth can-i create pods --as dev-user --namespace test
```

```
no
```

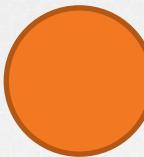
# Resource Names



blue



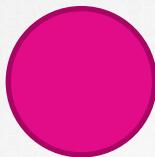
green



orange



purple



pink

developer-role.yaml

```
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: developer
rules:
- apiGroups: []
  resources: ["pods"]
  verbs: ["get", "create", "update"]
  resourceNames: ["blue", "orange"]
```

# References

<https://kubernetes.io/docs/reference/access-authn-authz/authorization/#checking-api-access>

<https://kubernetes.io/docs/reference/access-authn-authz/rbac>

<https://kubernetes.io/docs/reference/access-authn-authz/rbac/#kubectl-create-role>

<https://kubernetes.io/docs/reference/access-authn-authz/rbac/#kubectl-create-rolebinding>



{KODE} {LOUD}

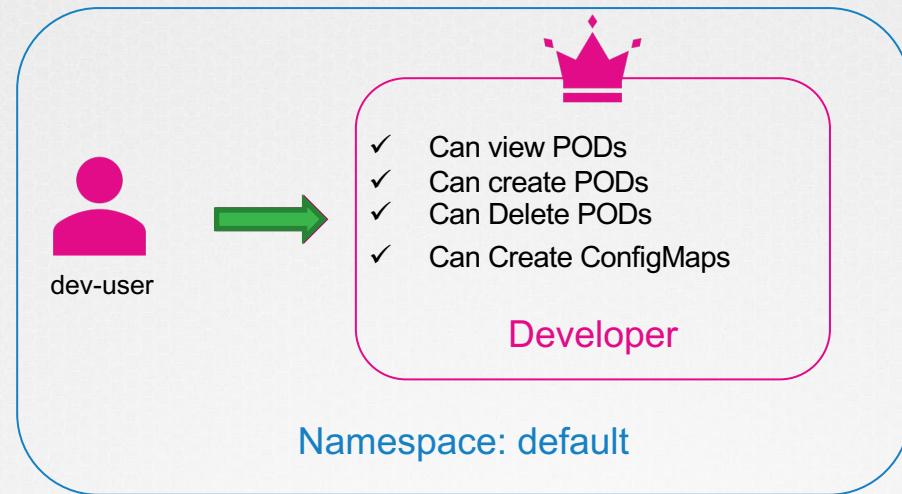
[www.kodekloud.com](http://www.kodekloud.com)



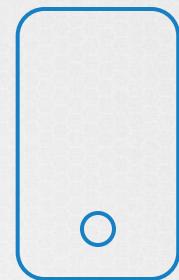
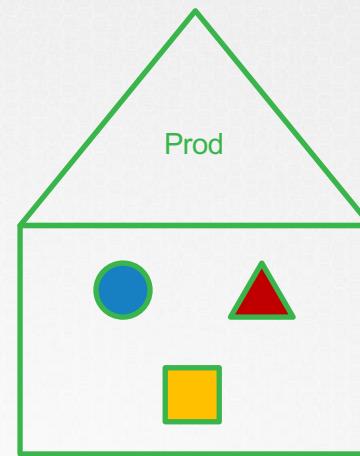
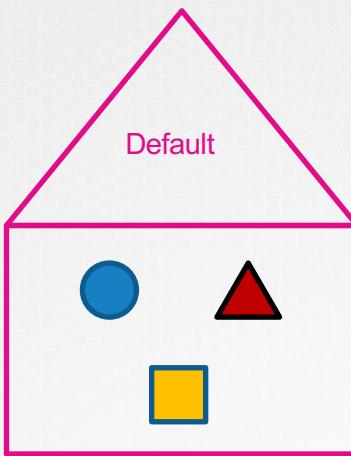
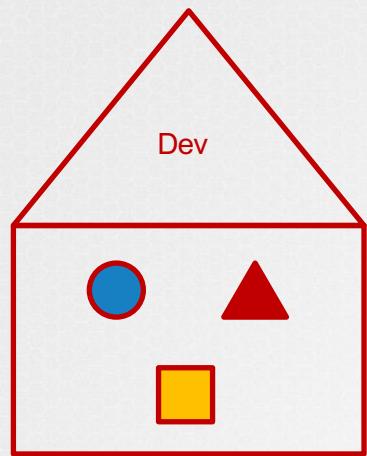
# Cluster Roles



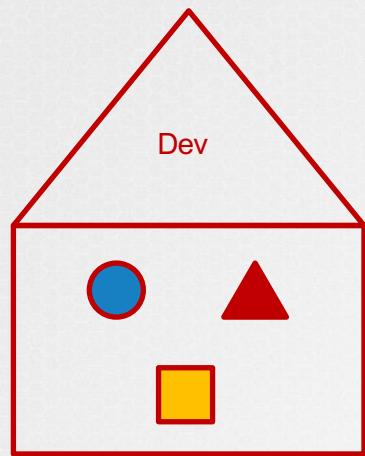
# Roles



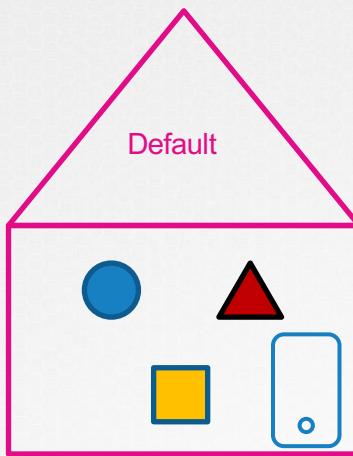
# Namespace



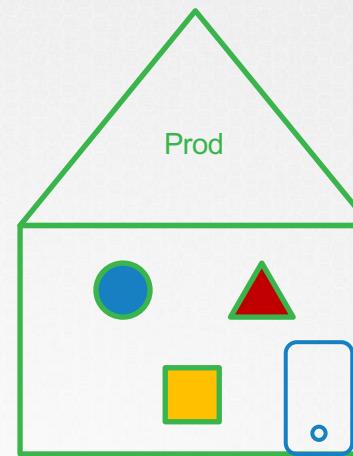
# Namespace



Dev

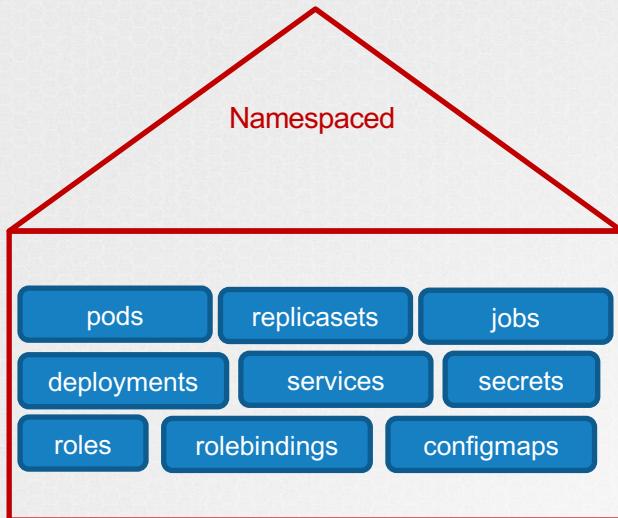


Default

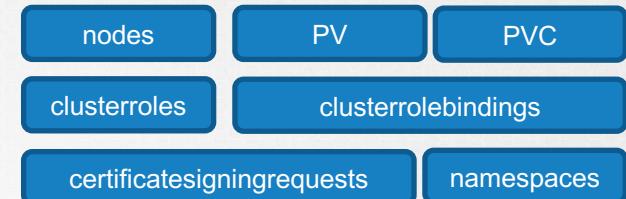


Prod

# Namespace



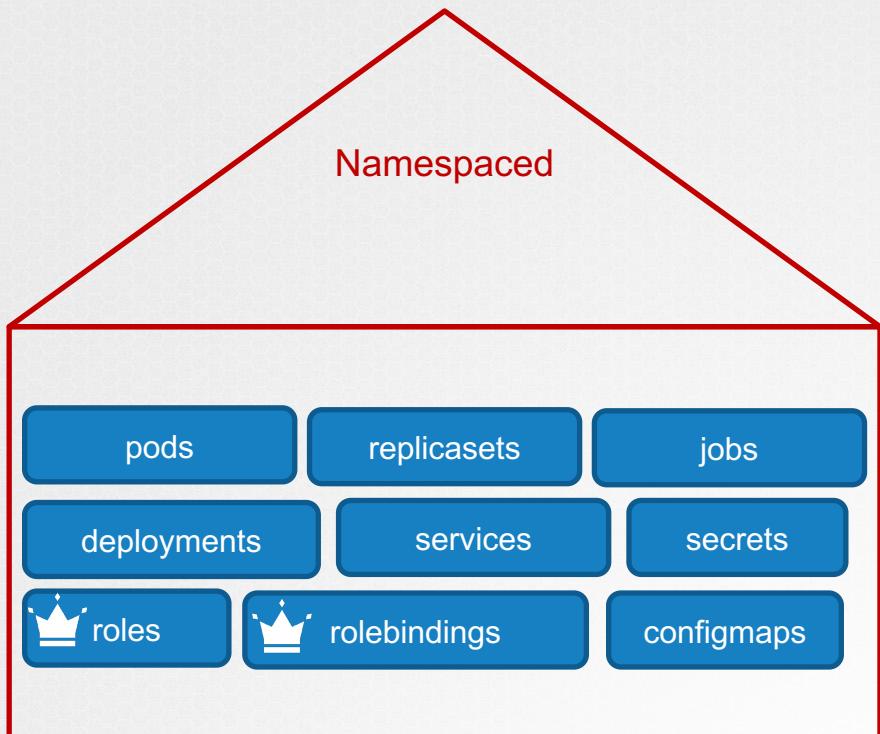
Cluster Scoped



```
▶ kubectl api-resources --namespaced=true
```

```
▶ kubectl api-resources --namespaced=false
```

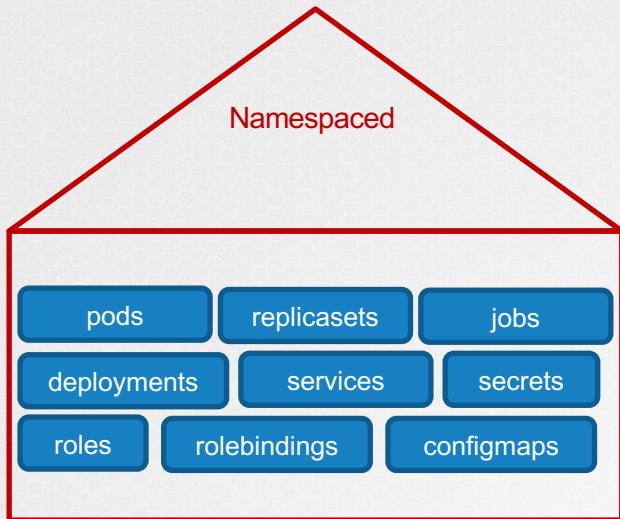
# Namespace



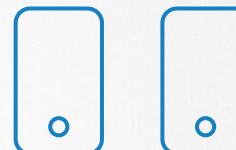
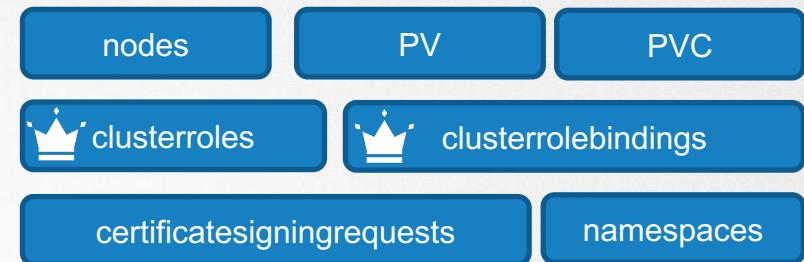
Cluster Scoped



# Namespace



Cluster Scoped



 clusterroles

- ✓ Can view Nodes
- ✓ Can create Nodes
- ✓ Can delete Nodes

Cluster Admin



- ✓ Can view PVs
- ✓ Can create PVs
- ✓ Can delete PVCs

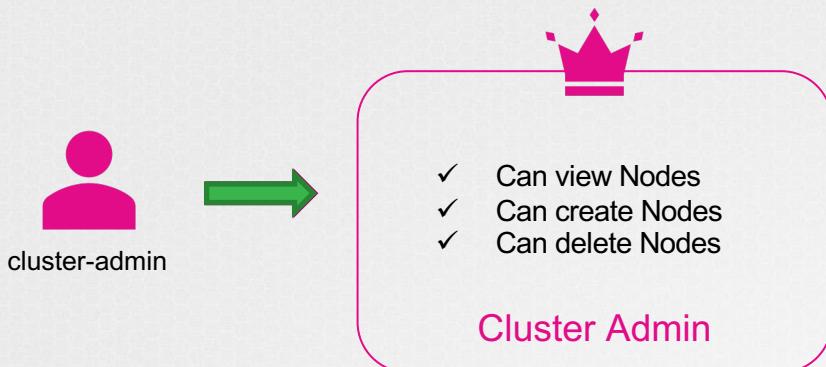
Storage Admin

cluster-admin-role.yaml

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: cluster-administrator
rules:
- apiGroups: []
  resources: ["nodes"]
  verbs: ["list", "get", "create", "delete"]
```

▶ kubectl create -f cluster-admin-role.yaml

# clusterrolebinding



cluster-admin-role.yaml

Copyright © 2021 KodeKloud

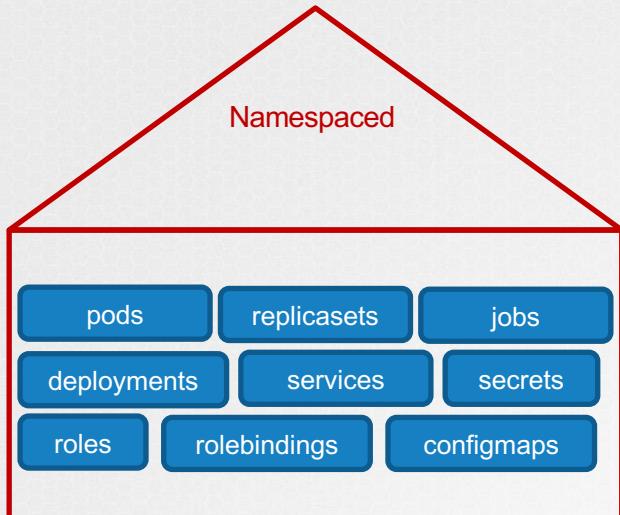
```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: cluster-administrator
rules:
- apiGroups: [ "" ]
  resources: [ "nodes" ]
  verbs: [ "list", "get", "create", "delete" ]
```

cluster-admin-role-binding.yaml

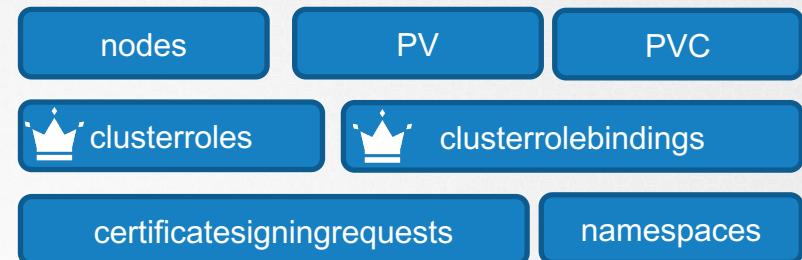
```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: cluster-admin-role-binding
subjects:
- kind: User
  name: cluster-admin
  apiGroup: rbac.authorization.k8s.io
roleRef:
  kind: ClusterRole
  name: cluster-administrator
  apiGroup: rbac.authorization.k8s.io
```

```
kubectl create -f cluster-admin-role-binding.yaml
```

# Cluster Roles



Cluster Scoped





{KODE} {LOUD}

[www.kodekloud.com](http://www.kodekloud.com)



# Image Security



# Image

nginx-pod.yaml

```
apiVersion: v1
kind: Pod
metadata:
  name: nginx-pod
spec:
  containers:
  - name: nginx
    image: nginx
```

# Image

**image:** nginx



Image/  
Repository

# Image

**image:** nginx/nginx



User/  
Account      Image/  
Repository

# Image

**image:** docker.io/nginx/nginx



Registry      User/  
Account      Image/  
                  Repository

gcr.io/ kubernetes-e2e-test-images/dnsutils

# Private Repository

```
▶ docker login private-registry.io
```

Login with your Docker ID to push and pull images from Docker Hub. If you don't have a Docker ID, head over to <https://hub.docker.com> to create one.

**Username:** registry-user

**Password:**

**WARNING!** Your password will be stored unencrypted in /home/vagrant/.docker/config.json.

Login Succeeded

```
▶ docker run private-registry.io/apps/internal-app
```

# Private Repository

```
▶ docker login private-registry.io
```

```
▶ docker run private-registry.io/apps/internal-app
```

nginx-pod.yaml

```
apiVersion: v1
kind: Pod
metadata:
  name: nginx-pod
spec:
  containers:
  - name: nginx
    image:
  imagePullSecrets:
  - name: regcred
```

```
▶ kubectl create secret docker-registry regcred \
  --docker-server= private-registry.io \
  --docker-username= registry-user \
  --docker-password= registry-password \
  --docker-email= registry-user@org.com
```

# References

<https://kubernetes.io/docs/tasks/configure-pod-container/pull-image-private-registry/>



{KODE} {LOUD}

[www.kodekloud.com](http://www.kodekloud.com)



# Securing the Kubelet



**kube-apiserver**

**Master**

Manage, Plan, Schedule, Monitor  
Nodes



**Worker Nodes**

Host Application as Containers



**kubelet**

**Controller-  
Manager**

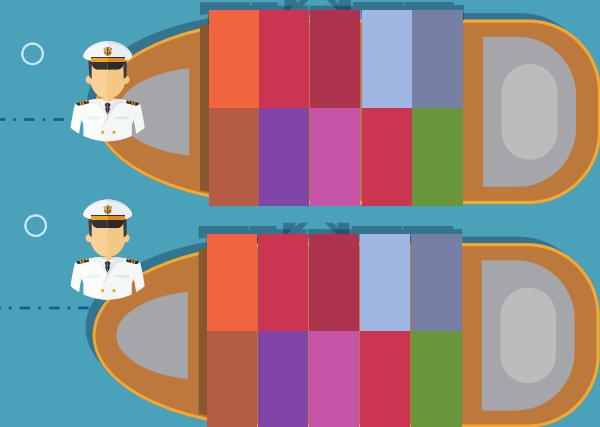
**CLUSTER**



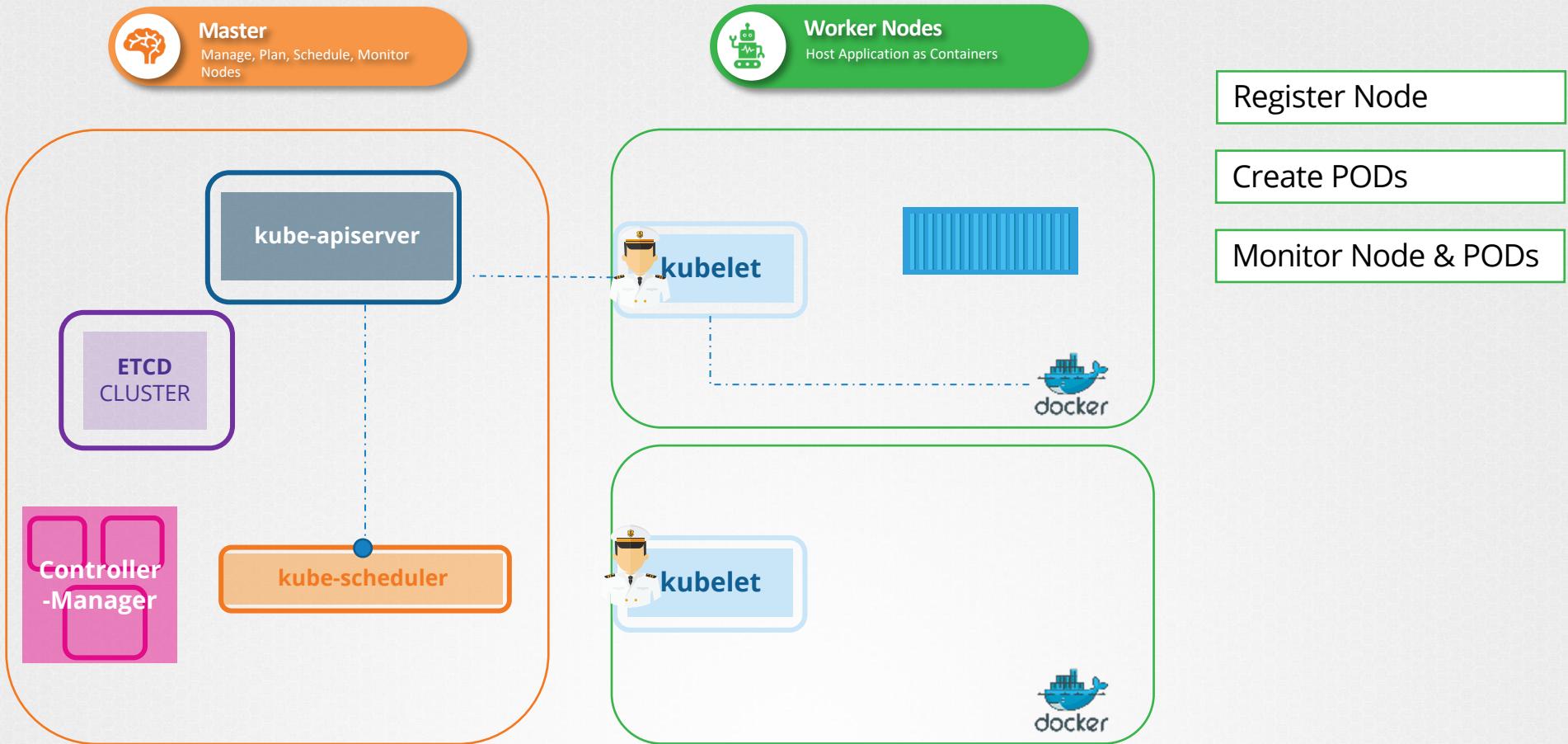
**e-scheduler**



**docker**



# Kubernetes Architecture



# Installing kubelet

```
▶ wget https://storage.googleapis.com/kubernetes-release/release/v1.20.0/bin/linux/amd64/kubelet
```

## kubelet.service

```
ExecStart=/usr/local/bin/kubelet \
--container-runtime=docker \
--image-pull-progress-deadline=2m \
--kubeconfig=/var/lib/kubelet/kubeconfig \
--network-plugin=cni \
--register-node=true \
--v=2 \
--cluster-domain=cluster.local \
--file-check-frequency=0s \
--healthz-port=10248 \
--cluster-dns=10.96.0.10 \
--http-check-frequency=0s \
--sync-frequency=0s
```



Kubeadm does not  
deploy Kubelets

# Installing kubelet

```
wget https://storage.googleapis.com/kubernetes-release/release/v1.20.0/bin/linux/amd64/kubelet
```

## kubelet.service

```
ExecStart=/usr/local/bin/kubelet \
--container-runtime=remote \
--image-pull-progress-deadline=2m \
--kubeconfig=/var/lib/kubelet/kubeconfig \
--network-plugin=cni \
--register-node=true \
--v=2 \
--config=/var/lib/kubelet/kubelet-config.yaml \
--file-check-frequency=0s \
--healthz-port=10248 \
--cluster-dns=10.96.0.10 \
--http-check-frequency=0s \
--sync-frequency=0s
```

## kubelet-config.yaml

```
apiVersion: kubelet.config.k8s.io/v1beta1
kind: KubeletConfiguration
clusterDomain: cluster.local
fileCheckFrequency: 0s
healthzPort: 10248
clusterDNS:
- 10.96.0.10
httpCheckFrequency: 0s
syncFrequency: 0s
```

# View kubelet options

```
▶ ps -aux | grep kubelet
```

```
root      2095  1.8  2.4 960676 98788 ?          Ssl  02:32   0:36 /usr/bin/kubelet --bootstrap-kubeconfig=/etc/kubernetes/bootstrap-kubelet.conf --kubeconfig=/etc/kubernetes/kubelet.conf --config=/var/lib/kubelet/config.yaml --cgroup-driver=cgroupfs --cni-bin-dir=/opt/cni/bin --cni-conf-dir=/etc/cni/net.d --network-plugin=cni
```

```
▶ cat /var/lib/kubelet/config.yaml
```

```
apiVersion: kubelet.config.k8s.io/v1beta1
clusterDNS:
- 10.96.0.10
clusterDomain: cluster.local
cpuManagerReconcilePeriod: 0s
evictionPressureTransitionPeriod: 0s
fileCheckFrequency: 0s
healthzBindAddress: 127.0.0.1
healthzPort: 10248
httpCheckFrequency: 0s
imageMinimumGCAge: 0s
kind: KubeletConfiguration
nodeStatusReportFrequency: 0s
nodeStatusUpdateFrequency: 0s
rotateCertificates: true
runtimeRequestTimeout: 0s
staticPodPath: /etc/kubernetes/manifests
streamingConnectionIdleTimeout: 0s
```

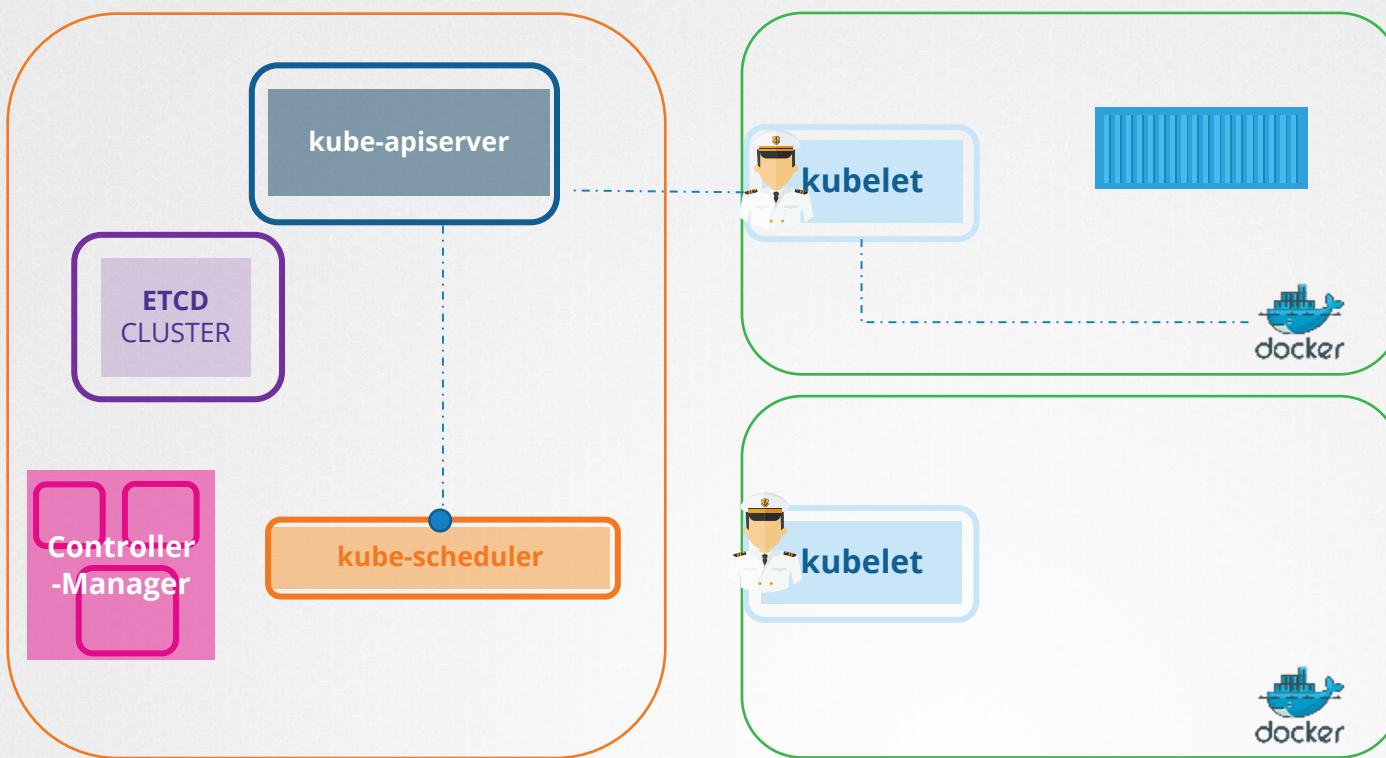
# Kubelet - Security

**Master**

Manage, Plan, Schedule, Monitor  
Nodes

**Worker Nodes**

Host Application as Containers



Register Node

Create PODs

Monitor Node & PODs

# Kubelet

Port	Description
10250	Serves API that allows full access
10255	Serves API that allows unauthenticated read-only access

# Kubelet

Port	Description
10250	Serves API that allows full access
10255	Serves API that allows unauthenticated access

```
▶ curl -sk https://localhost:10250/pods/
```

```
{"kind": "PodList", "apiVersion": "v1", "metadata": {}, "items": [{"metadata": {"name": "kube-proxy-nngzb", "generateName": "kube-proxy-", "namespace": "kube-system", "selfLink": "/apis/.../namespaces/kube-system/pods/kube-proxy-nngzb", "uid": "3c5e5745-f67e-4f91-beb9-f11d4da7a0d0", "resourceVersion": "1", "creationTimestamp": "2018-07-17T15:39:52Z", "labels": {"app": "kube-proxy", "component": "kube-proxy", "kubernetes.io/hostname": "host01", "kubernetes.io/os": "linux", "kubernetes.io/arch": "amd64", "kubernetes.io/role": "node", "kubernetes.io/cluster": "cluster"}, "status": {"phase": "Running", "conditions": [{"type": "Ready", "status": "True", "lastProbeTime": null, "lastTransitionTime": "2018-07-17T15:39:52Z"}, {"type": "PodScheduled", "status": "True", "lastProbeTime": null, "lastTransitionTime": "2018-07-17T15:39:52Z"}], "podIP": "10.250.0.10", "podIPV6": null}, {"metadata": {"name": "kube-proxy", "generateName": "kube-proxy-", "namespace": "kube-system", "selfLink": "/apis/.../namespaces/kube-system/pods/kube-proxy", "uid": "3c5e5745-f67e-4f91-beb9-f11d4da7a0d0", "resourceVersion": "1", "creationTimestamp": "2018-07-17T15:39:52Z", "labels": {"app": "kube-proxy", "component": "kube-proxy", "kubernetes.io/hostname": "host01", "kubernetes.io/os": "linux", "kubernetes.io/arch": "amd64", "kubernetes.io/role": "node", "kubernetes.io/cluster": "cluster"}, "status": {"phase": "Running", "conditions": [{"type": "Ready", "status": "True", "lastProbeTime": null, "lastTransitionTime": "2018-07-17T15:39:52Z"}, {"type": "PodScheduled", "status": "True", "lastProbeTime": null, "lastTransitionTime": "2018-07-17T15:39:52Z"}], "podIP": "10.250.0.10", "podIPV6": null}], "status": {"availableCapacity": {"cpu": 1, "memory": "383Mi"}, "unavailableCapacity": {"cpu": 0, "memory": "0Mi"}, "totalCapacity": {"cpu": 1, "memory": "383Mi"}, "totalNodes": 1, "totalPods": 2}}
```

```
▶ curl -sk https://localhost:10250/logs/syslog
```

```
Nov 10 11:26:59 host01 kernel: [    0.000000] Linux version 4.15.0-7.3.0 (Ubuntu 4.15.0-29.31~1~ubu...#31-Ubuntu SMP Tue Jul 17 15:39:52 UTC 2018 (Ubuntu 4.15.0-29.31~1~ubu...Nov 10 11:26:59 host01 kernel: [    0.000000] Command line: BOOT\_...root=/dev/mapper/host01--vg-root ro quiet splashNov 10 11:26:59 host01 kernel: [    0.000000] KERNEL supported cpusNov 10 11:26:59 host01 kernel: [    0.000000] Intel GenuineIntelNov 10 11:26:59 host01 kernel: [    0.000000] AMD AuthenticAMDNov 10 11:26:59 host01 kernel: [    0.000000] Centaur CentaurHalNov 10 11:26:59 host01 kernel: [    0.000000] x86/fpu: x87 FPU wi...Nov 10 11:26:59 host01 kernel: [    0.000000] e820: BIOS-provided
```

```
"/attach/{podNamespace}/{podID}/{containerName}": "proxy",
"/attach/{podNamespace}/{podID}/{uid}/{containerName}": "proxy",
"/configz": "proxy",
"/containerLogs/{podNamespace}/{podID}/{containerName}": "log",
"/cri/": "proxy",
"/cri/foo": "proxy",
"/debug/flags/v": "proxy",
"/debug/pprof/{subpath: *}": "proxy",
"/exec/{podNamespace}/{podID}/{containerName}": "proxy",
"/exec/{podNamespace}/{podID}/{uid}/{containerName}": "proxy",
"/healthz": "proxy",
"/healthz/log": "proxy",
"/healthz/ping": "proxy",
"/healthz/syncloop": "proxy",
"/logs/": "log",
"/logs/{logpath: *}": "log",
"/metrics": "metrics",
"/metrics/cadvisor": "metrics",
"/metrics/probes": "metrics",
"/metrics/resource/v1alpha1": "metrics",
"/pods/": "proxy",
"/portForward/{podNamespace}/{podID}": "proxy",
"/portForward/{podNamespace}/{podID}/{uid}": "proxy",
"/run/{podNamespace}/{podID}/{containerName}": "proxy",
"/run/{podNamespace}/{podID}/{uid}/{containerName}": "proxy",
"/runningpods/": "proxy",
"/spec/": "spec",
"/stats/": "stats",
"/stats/container": "stats",
"/stats/summary": "stats".
```

# Kubelet

Port	Description
10250	Serves API that allows full access
10255	Serves API that allows unauthenticated read-only access

```
▶ curl -sk http://localhost:10255/metrics
```

```
# HELP apiserver_audit_event_total [ALPHA] Counter of audit events generated and sent to the audit backend.
# TYPE apiserver_audit_event_total counter
apiserver_audit_event_total 0
# HELP apiserver_audit_requests_rejected_total [ALPHA] Counter of apiserver requests rejected due to an error in audit log processing.
# TYPE apiserver_audit_requests_rejected_total counter
apiserver_audit_requests_rejected_total 0
# HELP apiserver_client_certificate_expiration_seconds [ALPHA] Distribution of the remaining lifetime on the certificate used by clients.
# TYPE apiserver_client_certificate_expiration_seconds histogram
apiserver_client_certificate_expiration_seconds_bucket{le="0"} 0
apiserver_client_certificate_expiration_seconds_bucket{le="1800"} 0
apiserver_client_certificate_expiration_seconds_bucket{le="3600"} 0
apiserver_client_certificate_expiration_seconds_bucket{le="7200"} 0
apiserver_client_certificate_expiration_seconds_bucket{le="21600"} 0
apiserver_client_certificate_expiration_seconds_bucket{le="43200"} 0
apiserver_client_certificate_expiration_seconds_bucket{le="86400"} 0
apiserver_client_certificate_expiration_seconds_bucket{le="172800"} 0
apiserver_client_certificate_expiration_seconds_bucket{le="345600"} 0
```

# Kubelet Security

Authentication

Authorization

# Kubelet Security

## Authentication

```
▶ curl -sk https://localhost:10250/pods/
```

```
{"kind":"PodList","apiVersion":"v1","metadata":{},"items":[{"metadata":{"name":"kube-proxy-nngzb","generateName":"kube-proxy-", "namespace":"kube-system","selfLink":"/api/v1/namespaces/kube-system/pods/kube-proxy-nngzb","uid":"3c5e5745-f67e-4f91-beb9-f11d4da7a0d0","resourceVersion":"631","creationTimestamp":"2021-03-11T11:44:21Z"}]}
```

### kubelet.service

```
ExecStart=/usr/local/bin/kubelet \\  
...  
--anonymous-auth=false  
...
```

### kubelet-config.yaml

```
apiVersion: kubelet.config.k8s.io/v1beta1  
kind: KubeletConfiguration  
authentication:  
  anonymous:  
    enabled: false
```

# Kubelet Security

Authentication

Certificates (X509)

API Bearer Tokens

kubelet.service

```
ExecStart=/usr/local/bin/kubelet \\  
...  
--client-ca-file=/path/to/ca.crt \\  
...
```

kubelet-config.yaml

```
apiVersion: kubelet.config.k8s.io/v1beta1  
kind: KubeletConfiguration  
authentication:  
  x509:  
    clientCAFile: /path/to/ca.crt
```

```
▶ curl -sk https://localhost:10250/pods/ -key kubelet-key.pem -cert kubelet-cert.pem
```

```
▶ cat /etc/systemd/system/kube-apiserver.service
```

```
[Service]  
ExecStart=/usr/local/bin/kube-apiserver \\  
...  
--kubelet-client-certificate=/path/to/kubelet-cert.pem \\  
--kubelet-client-key=/path/to/kubelet-key.pem \\
```

# Kubelet Security

kubelet.service

```
ExecStart=/usr/local/bin/kubelet \\  
...  
--authorization-mode=Webhook  
...
```

kubelet-config.yaml

```
apiVersion: kubelet.config.k8s.io/v1beta1  
kind: KubeletConfiguration  
authorization:  
  mode: Webhook
```

kube-apiserver

Authorization

# Kubelet Security

```
curl -sk https://localhost:10255/metrics
```

```
# HELP apiserver_audit_event_total [ALPHA] Counter of audit events generated and sent to the audit backend.
# TYPE apiserver_audit_event_total counter
apiserver_audit_event_total 0
# TYPE apiserver_client_certificate_expiration_seconds histogram
apiserver_client_certificate_expiration_seconds_bucket{le="0"} 0
apiserver_client_certificate_expiration_seconds_bucket{le="1800"} 0
apiserver_client_certificate_expiration_seconds_bucket{le="86400"} 0
apiserver_client_certificate_expiration_seconds_bucket{le="172800"} 0
apiserver_client_certificate_expiration_seconds_bucket{le="345600"} 0
apiserver_client_certificate_expiration_seconds_bucket{le="604800"} 0
apiserver_client_certificate_expiration_seconds_bucket{le="2.592e+06"} 0
apiserver_client_certificate_expiration_seconds_bucket{le="7.776e+06"} 0
apiserver_client_certificate_expiration_seconds_bucket{le="1.5552e+07"} 0
```

kubelet.service

```
ExecStart=/usr/local/bin/kubelet \\
...
--read-only-port=10255
...
```

kubelet-config.yaml

```
apiVersion: kubelet.config.k8s.io/v1beta1
kind: KubeletConfiguration
readOnlyPort: 0
```

# Kubelet Security

## kubelet.service

```
ExecStart=/usr/local/bin/kubelet \\  
    ...  
    --anonymous-auth=false \\  
    --client-ca-file=/path/to/ca.crt \\  
    --authorization-mode=Webhook  
  
    --read-only-port=0
```

## kubelet-config.yaml

```
apiVersion: kubelet.config.k8s.io/v1beta1  
kind: KubeletConfiguration  
authentication:  
  anonymous:  
    enabled: false  
  x509:  
    clientCAFile: /path/to/ca.crt  
authorization:  
  mode: Webhook  
readOnlyPort: 0
```

# Labs – Kubelet Security

- Explore kubelet security
- Enable Authentication/Authorization on Kubelet

[cks.kodekloud.com](https://cks.kodekloud.com)

# References

<https://kubernetes.io/docs/reference/command-line-tools-reference/kubelet/>

<https://kubernetes.io/docs/tasks/administer-cluster/kubelet-config-file/>

<https://kubernetes.io/docs/setup/production-environment/tools/kubeadm/kubelet-integration/#configure-kubelets-using-kubeadm>

<https://kubernetes.io/docs/reference/command-line-tools-reference/kubelet-authentication-authorization/>

<https://gist.github.com/lizrice/c32740fac51db2a5518f06c3dae4944f>



{KODE} {LOUD}

[www.kodekloud.com](http://www.kodekloud.com)

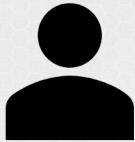


# Kubectl Proxy



```
▶ kubectl get nodes
```

NAME	STATUS	ROLES	AGE	VERSION
master	Ready	control-plane,master	25h	v1.20.1
worker	Ready	<none>	25h	v1.20.1



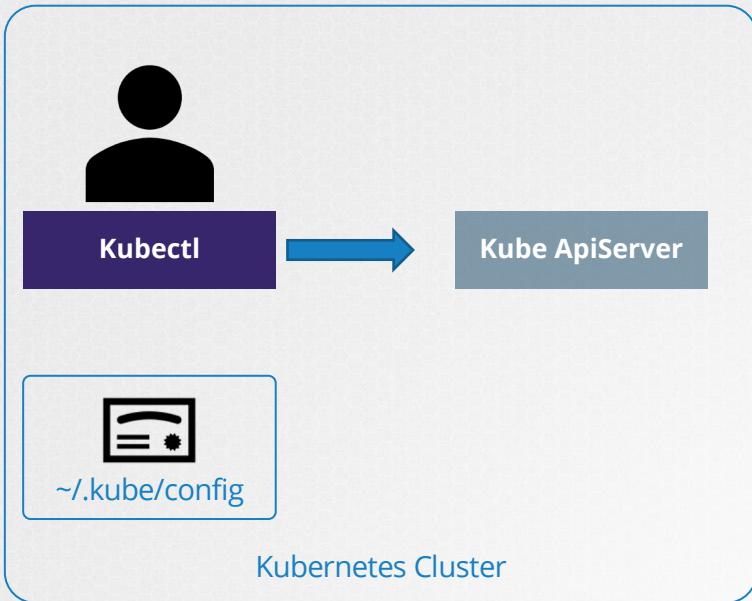
Kubectl



Kube ApiServer

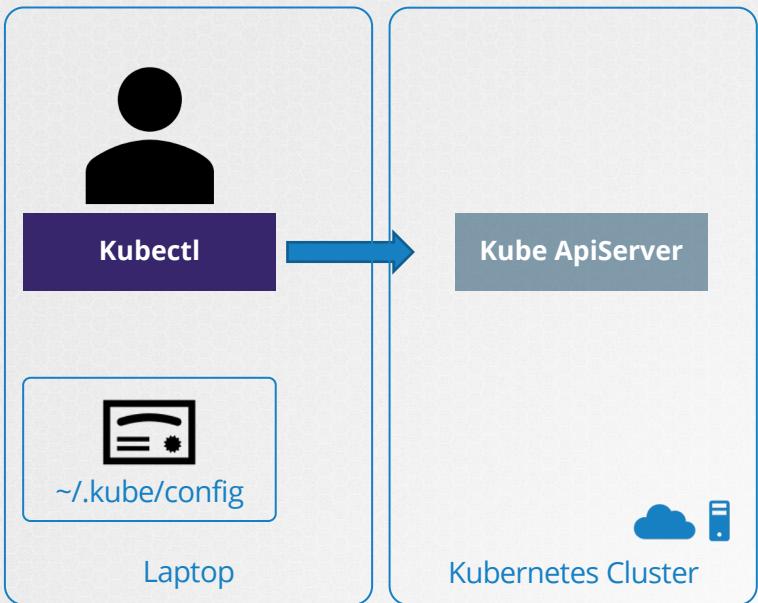


~/.kube/config



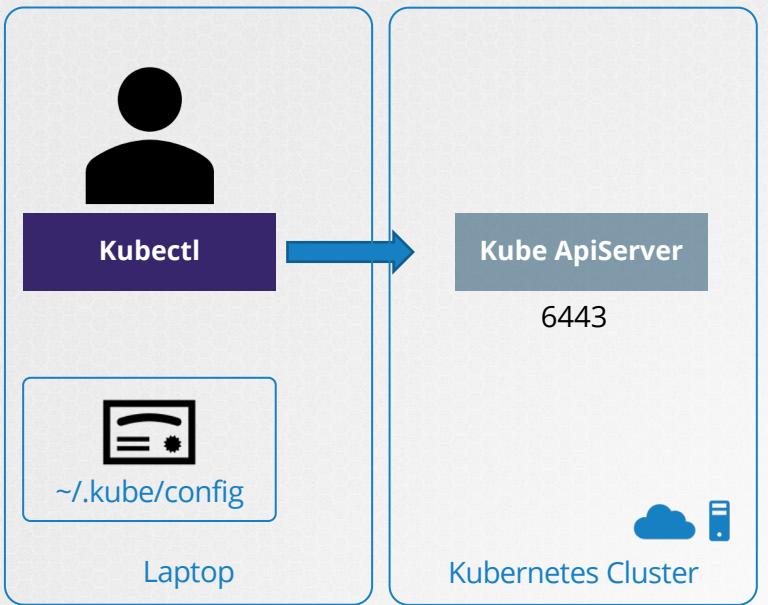
```
▶ kubectl get nodes
```

NAME	STATUS	ROLES	AGE	VERSION
master	Ready	control-plane,master	25h	v1.20.1
worker	Ready	<none>	25h	v1.20.1



```
▶ kubectl get nodes
```

NAME	STATUS	ROLES	AGE	VERSION
master	Ready	control-plane,master	25h	v1.20.1
worker	Ready	<none>	25h	v1.20.1



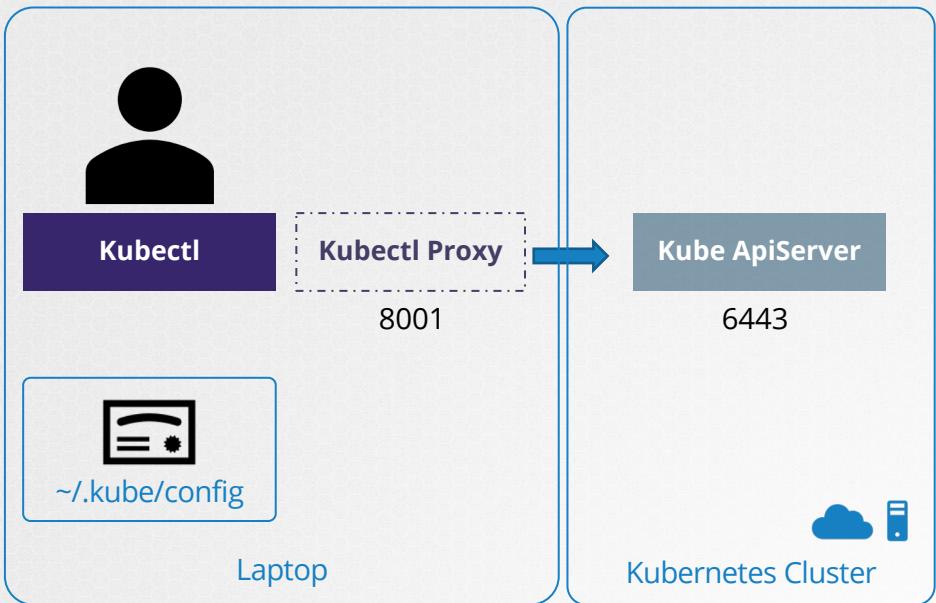
```
▶ curl http://<kube-api-server-ip>:6443 -k
{
  "kind": "Status",
  "apiVersion": "v1",
  "metadata": {

  },
  "status": "Failure",
  "message": "forbidden: User \"system:anonymous\" cannot get path \"/\"",
  "reason": "Forbidden",
  "details": {

  },
  "code": 403
}
```

```
▶ curl http://<kube-api-server-ip>:6443 -k
--key admin.key
--cert admin.crt
--cacert ca.crt
{
  "paths": [
    "/api",
    "/api/v1",
    "/apis",
    "/apis/",
    "/healthz",
    "/logs",
    "/metrics"
  ]
}
```

# kubectl proxy

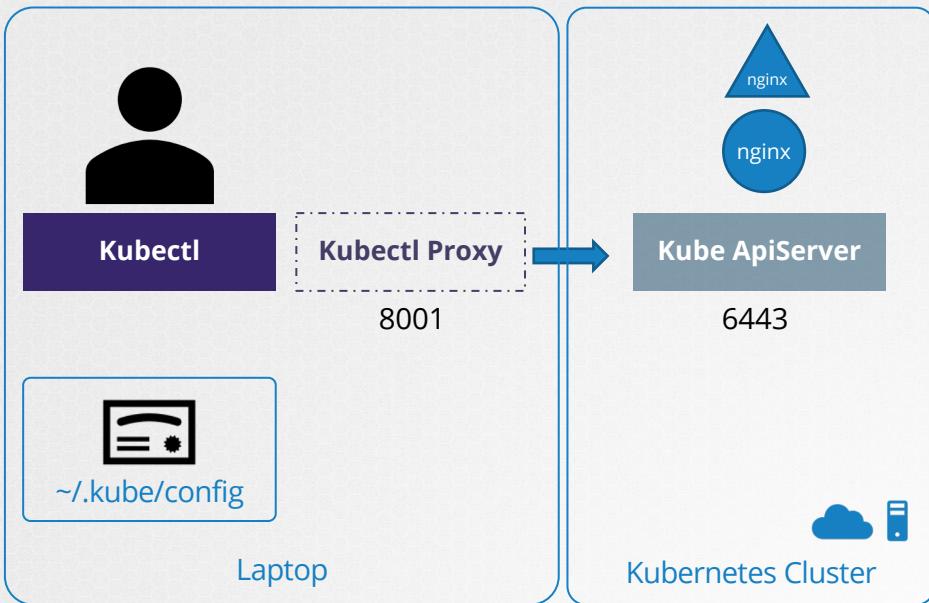


```
▶ kubectl proxy
Starting to serve on 127.0.0.1:8001
```

```
▶ curl http://localhost:8001 -k
{
  "paths": [
    "/api",
    "/api/v1",
    "/apis",
    "/apis/",
    "/healthz",
    "/logs",
    "/metrics",
    "/openapi/v2",
    "/swagger-2.0.0.json",
```

# Kubectl Proxy

```
▶ curl http://localhost:8001/api/v1/namespaces/default/services/nginx/proxy/
```

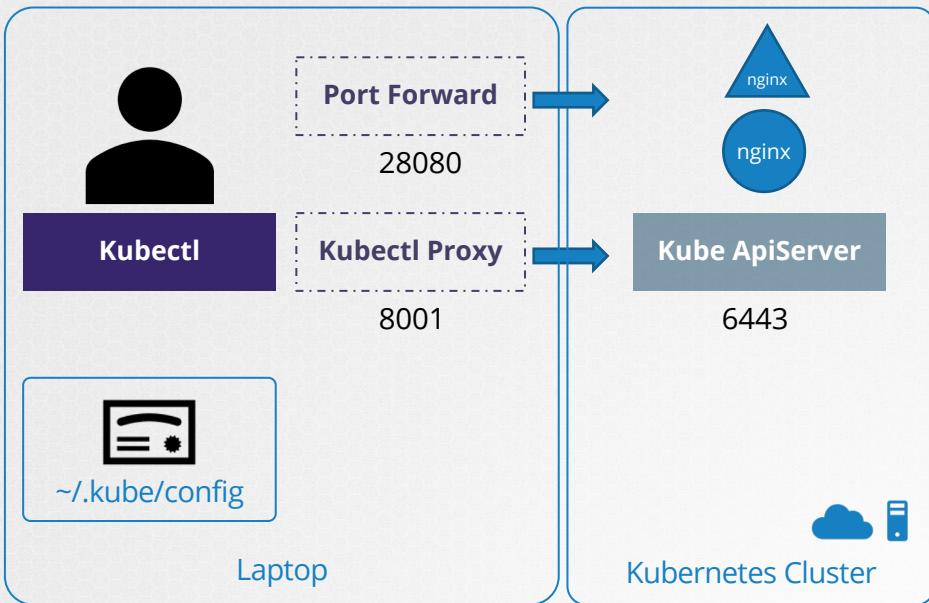


```
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
body {
    width: 35em;
    margin: 0 auto;
    font-family: Tahoma, Verdana, Arial, sans-serif;
}
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
```

# Kubectl Port Forward



```
▶ kubectl port-forward service/nginx 28080:80
```

```
▶ curl http://localhost:28080/
```

```
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
  body {
    width: 35em;
    margin: 0 auto;
    font-family: Tahoma, Verdana, Arial, sans-serif;
  }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
```

Hands-on Labs  
[cks.kodekloud.com](https://cks.kodekloud.com)



{KODE} {LOUD}

[www.kodekloud.com](http://www.kodekloud.com)

# Kubernetes Dashboard

# Kubernetes Dashboard

The screenshot shows the Kubernetes Dashboard running in a web browser. The URL is `localhost:8001/api/v1/namespaces/kubernetes-dashboard/services/https:kubernetes-dashboard:/proxy/#/overview?namespace=default`. The dashboard has a blue header bar with the title "Kubernetes Dashboard". Below the header is a navigation bar with tabs: "Overview" (selected), "Workloads", "Deployments", "Pods", and "Replica Sets". On the left, there's a sidebar with links for Cluster (Cluster Roles, Namespaces, Nodes, Persistent Volumes, Storage Classes), Namespace (default selected), and Workloads (Cron Jobs, Daemon Sets, Deployments, Jobs, Pods, Replica Sets, Replication Controllers, Stateful Sets). The main content area has three large green circles under the heading "Workload Status": "Deployments", "Pods", and "Replica Sets". Below this is a table titled "Deployments" with one row for "nginx-deployment". The table columns are Name, Namespace, Labels, Pods, Created, and Images. The "nginx-deployment" row shows 2 / 2 pods created 21 minutes ago with the image nginx:1.14.2. At the bottom, there are "Previous" and "Next" buttons. The footer of the dashboard shows "Memory Usage" and other metrics.

Name	Namespace	Labels	Pods	Created	Images
nginx-deployment	default	-	2 / 2	21 minutes ago	nginx:1.14.2

<https://kubernetes.io/docs/tasks/access-application-cluster/web-ui-dashboard/>

# Kubernetes Dashboard

## Secrets

Name	Namespace
calico-kube-controllers-token-tdmx9	calico-system
calico-node-token-vppvb	calico-system
calico-typha-token-hlshc	calico-system
default-token-wvftn	calico-system
node-certs	calico-system
typha-certs	calico-system
default-token-pkg4p	default
kubernetes-dashboard-token-fc2fq	default
default-token-qp79r	kube-node-lease
default-token-6mxqh	kube-public

## Metadata

Name	Namespace	Created
kubernetes-dashboard-token-fc2fq	default	Mar 5, 2021
<b>Annotations</b>		
kubernetes.io/service-account.name: kubernetes-dashboard		kubernetes.io/service-ac

## Data

ca.crt

**1066 bytes**

namespace

**7 bytes**

token

```
eyJhbGciOiJSUzI1NiIsImtpZCI6ImlCbilxSjgwM1AtMxdmbHkybEd0dXVhcjJMS2hY2UiOjkZWzhWx0Iiwiia3ViZXJuZXRLcy5pb9zZXJ2aWN1YWNgjb3VudC9zZWNy2mFtZSI6Imt1YmVybmv0ZXMtZGFzaGJvYXJkIiwiia3ViZXJuZXRLcy5pb9zZXJ2aWNhY2Nvdw50OmRlZmFlbHQ6a3ViZXJuZXRLcy1kYXNoYm9hcmQifQ.otsa0gIdh20k3j1vCJ15ci0iqDQqdK1SyE_PuGX05TNbbIJby2m6JgEJQZCdvmMMWnFbXraajIb1x44W]GnkimPwQqr9HyeSkIlhlrw
```

**Update**

**Cancel**

# Kubernetes Dashboard

The screenshot shows the 'Create' page of the Kubernetes Dashboard. On the left, there's a sidebar with navigation links like Overview, Workloads, Cron Jobs, Daemon Sets, Deployments, Jobs, Pods, Replica Sets, Replication Controllers, Stateful Sets, Discovery and Load Balancing, Ingresses, and Services. The main area has tabs for 'Create from input', 'Create from file', and 'Create from form'. A large text area contains YAML code for a deployment:

```
117 type: RollingUpdate
118 rollingUpdate:
119   maxUnavailable: 25%
120   maxSurge: 25%
121   revisionHistoryLimit: 10
122   progressDeadlineSeconds: 600
123 status:
124   observedGeneration: 1
125   replicas: 2
126   updatedReplicas: 2
127   readyReplicas: 2
128   availableReplicas: 2
129   conditions:
130     - type: Progressing
131       status: 'True'
132       lastUpdateTime: '2021-03-05T05:55:37Z'
133       lastTransitionTime: '2021-03-05T05:55:19Z'
134       reason: NewReplicaSetAvailable
135       message: ReplicaSet "nginx-deployment-66b6c48dd5" has successfully progressed.
136     - type: Available
137       status: 'True'
138       lastUpdateTime: '2021-03-06T02:38:00Z'
139       lastTransitionTime: '2021-03-06T02:38:00Z'
140       reason: MinimumReplicasAvailable
141       message: Deployment has minimum availability.
```

At the bottom, there are 'Upload' and 'Cancel' buttons.

# Kubernetes Dashboard

Research

## Lessons from the Cryptojacking Attack at Tesla

by RedLock CSI Team | 02.20.18, 6:00 AM

### The Cryptojacking Epidemic

A few months ago, the RedLock Cloud Security Intelligence (CSI) team found hundreds of Kubernetes administration consoles accessible over the internet without any password protection.

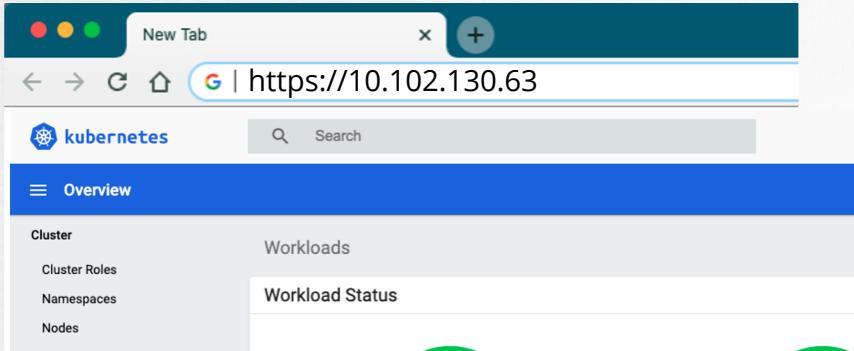
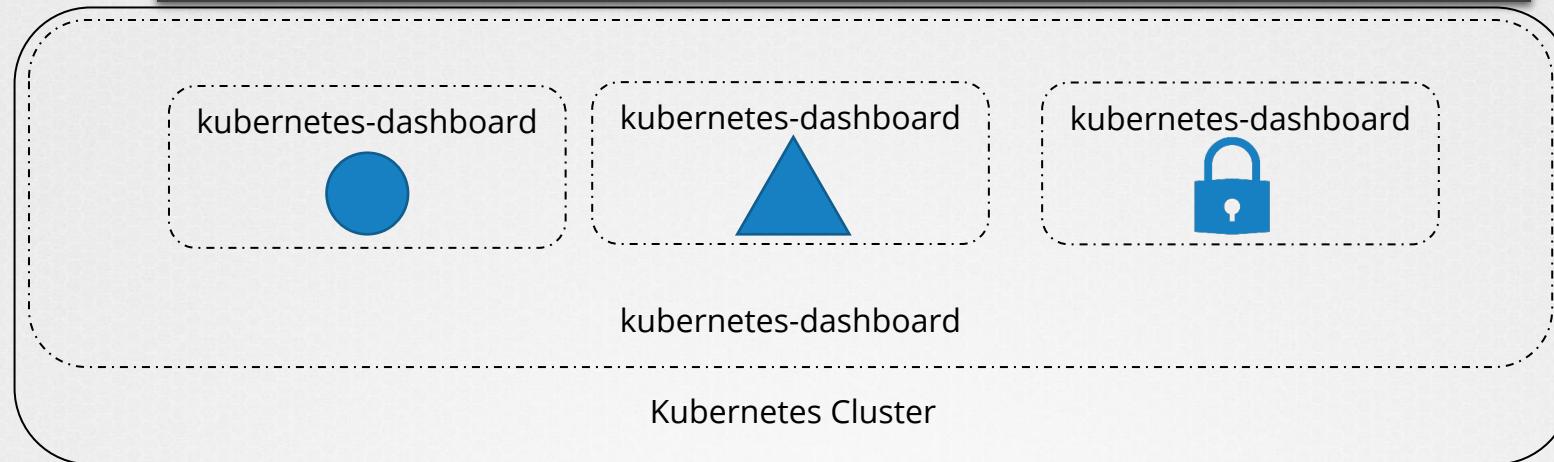
A couple of the instances belonged to [Aviva](#), a British multinational insurance company, and [Gemalto](#), the world's largest manufacturer of SIM cards. Within these consoles, access credentials to these organizations' Amazon Web Services (AWS) and Microsoft Azure environments were exposed. Upon further investigation, the team determined that hackers had secretly infiltrated these organizations' public cloud environments and were using the compute instances to [mine cryptocurrencies](#) (refer to [Cloud Security Trends - October 2017](#) report).

Since then, a number of other cryptojacking incidents have been uncovered and there are notable differences in the attacks. In cases involving the [WannaMine](#) malware, a tool called Mimikatz is used to pull credentials from a computer's memory to infect other computers on the network. The malware then uses the infected computers' compute to mine a cryptocurrency called Monero quietly in the background. The use of Mimikatz ensures that the malware does not have to rely on the EternalBlue exploit and enables it to evade detection on fully patched systems.

<https://redlock.io/blog/cryptojacking->

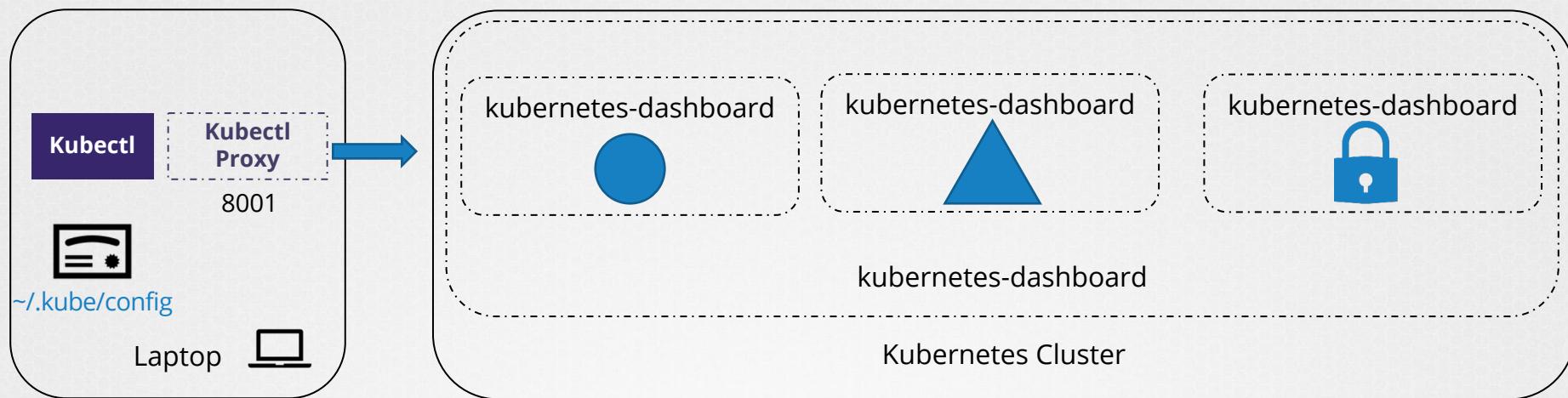
# Deploying Kubernetes Dashboard

```
kubectl apply -f https://<path-to-Kubernetes-dashboard>/recommended.yaml
```

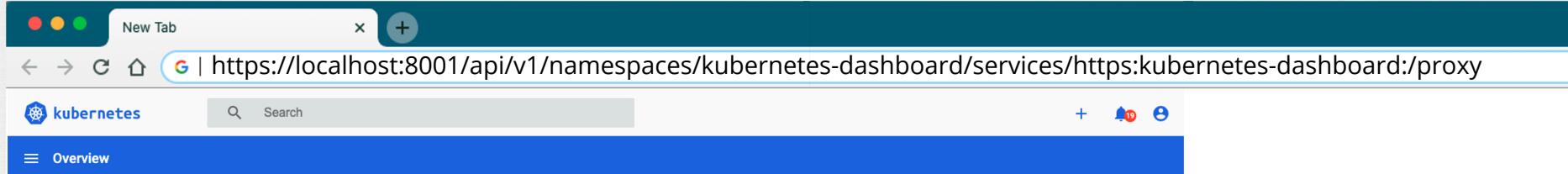


```
* ➔ kubectl describe service kubernetes-dashboard -n
Name:          kubernetes-dashboard
Namespace:     kubernetes-dashboard
Labels:        k8s-app=kubernetes-dashboard
Annotations:  
Type:          ClusterIP
IP:            10.102.130.63
Port:          <unset>  443/TCP
ServicePort:   8443/TCP
Endpoints:    10.102.130.63:8443
```

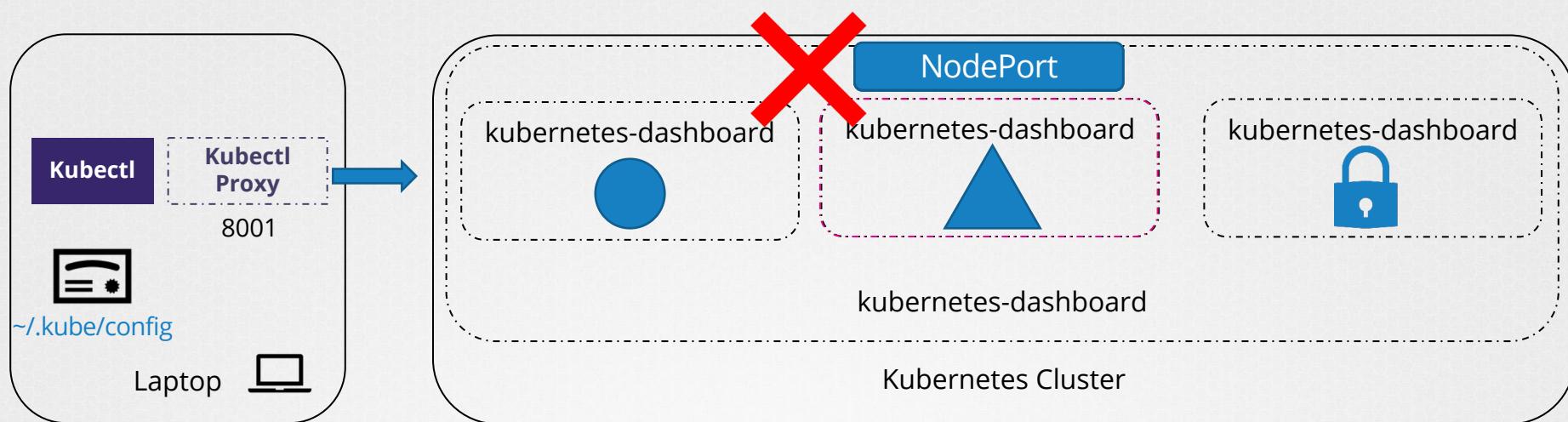
# Accessing Kubernetes Dashboard



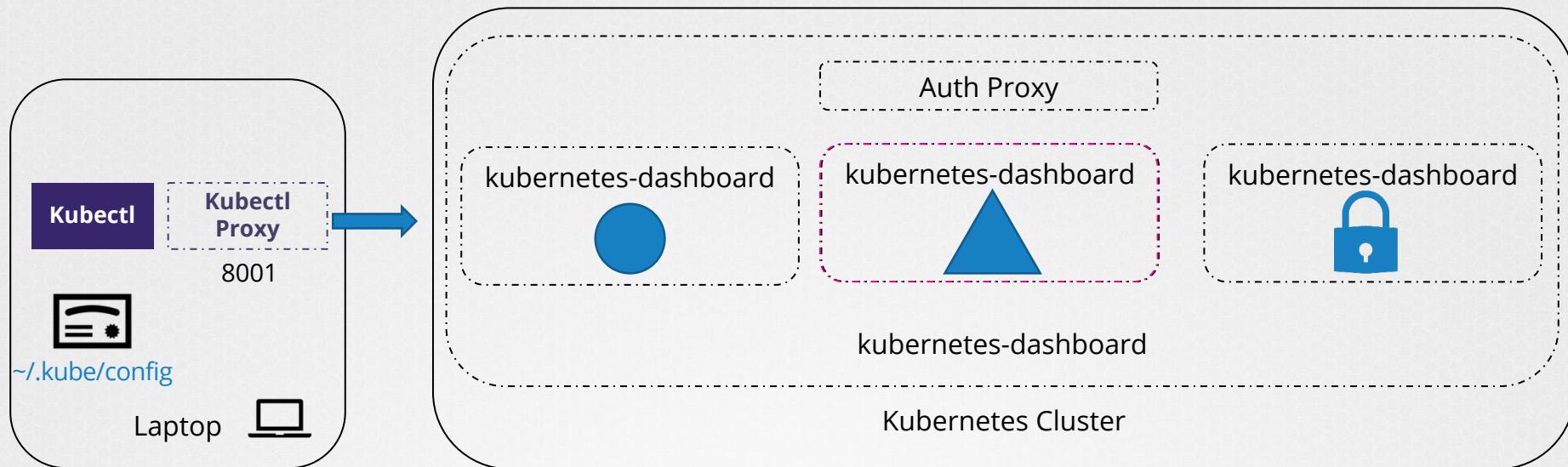
```
▶ kubectl proxy
Starting to serve on 127.0.0.1:8001
```



# Accessing Kubernetes Dashboard



# Accessing Kubernetes Dashboard



# References

<https://kubernetes.io/docs/tasks/access-application-cluster/web-ui-dashboard/>

<https://github.com/kubernetes/dashboard>

## **Good Watch:**

<https://www.youtube.com/watch?v=od8TnlvuADg>

<https://blog.heptio.com/on-securing-the-kubernetes-dashboard-16b09b1b7aca>



{KODE} {LOUD}

[www.kodekloud.com](http://www.kodekloud.com)



# Kubernetes Dashboard - Authentication



# Login

## Kubernetes Dashboard

Token

Every Service Account has a Secret with valid Bearer Token that can be used to log in to Dashboard. To find out more about how to configure and use Bearer Tokens, please refer to the [Authentication](#) section.

Kubeconfig

Please select the kubeconfig file that you have created to configure access to the cluster. To find out more about how to configure and use kubeconfig file, please refer to the [Configure Access to Multiple Clusters](#) section.

Enter token \*

**Sign in**

# Creating sample user

## Creating a Service Account

We are creating Service Account with name `admin-user` in namespace `kubernetes-dashboard` first.

```
cat <<EOF | kubectl apply -f -
apiVersion: v1
kind: ServiceAccount
metadata:
  name: admin-user
  namespace: kubernetes-dashboard
EOF
```

```
cat <<EOF | kubectl apply -f -
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: admin-user
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: cluster-admin
subjects:
- kind: ServiceAccount
  name: admin-user
  namespace: kubernetes-dashboard
EOF
```

# Retrieve the Token

```
▶ kubectl describe secret kubernetes-dashboard-token-fc2fq
```

```
Name:      kubernetes-dashboard-token-fc2fq
Namespace:  default
Labels:    <none>
Annotations: kubernetes.io/service-account.name: kubernetes-dashboard
             kubernetes.io/service-account.uid: 635c208a-6752-43f7-9ca4-e43665df1353
Type:      kubernetes.io/service-account-token
```

```
Data
```

```
====
```

```
ca.crt:  1066 bytes
namespace: 7 bytes
token:   eyJhbGciOiJSUzI1NiIsImtpZCI6ImICbi1xSjgwMIAtMXdmbHkybEdOdXVhcjJMS2FFR3hqOVpkR0NEYlpzUG8ifQ.eyJpc3MiOiJrdWJlcm5ldGVzL3NlcnZpY2VhY2NvdW50Iiwia3ViZXJuZXRIcy5pb9y9zZXJ2aWNlYWNjb3VudC9uYW1lc3BhY2UiOjkZWZhdWx0Iiwia3ViZXJuZXRIcy5pb9y9zZXJ2aWNlYWNjb3VudC9zZWNyZXQubmFtZSI6Imt1YmVybmV0ZXMtZGFzaGJvYXJkLXRva2VuLWZjMmZxiIwia3ViZXJuZXRIcy5pb9y9zZXJ2aWNlYWNjb3VudC9zZXJ2aWNlWFjY291bnQubmFtZSI6Imt1YmVybmV0ZXMtZGFzaGJvYXJkIiwia3ViZXJuZXRIcy5pb9y9zZXJ2aWNlYWNjb3VudC9zZXJ2aWNlWFjY291bnQubmFtZSI6Imt1YmVybmV0ZXMtZGFzaGJvYXJkIiwia3ViZXJuZXRIcy5pb9y9zZXJ2aWNlYWNjb3VudC9zZXJ2aWNlWFjY291bnQubmFtZSI6Imt1YmVybmV0ZXMtZGFzaGJvYXJkIiwic3Viljoic3IzdGVtOnNlcnZpY2VhY2NvdW50OmRIZmF1bHQ6a3ViZXJuZXRIcy1kYXNoYm9hcmQifQ.otsaOgldh20k3iuQZEJjGLGDdIpNvn3z4B4iTbra8CzembC2VdCYoKCx5n0-at2CS3G7QVAnHU0mRn2kVs1u2L0NVLYxyvaGpvW-HVffmrmt4gzQv8NIBenBV58NYi0G-9ZWlplvCJl5ci0iqDjqdK1SyE_PuGX05TNbbIJby2m6JgEJQZCdvMMWnFbXraaJlb1X44WluLhmbP8_5afzfVywvlpt1H0XPdYK_jPJyuptL6r7NQ4HCU-T_mZvFZ_ZjT8YGZzv6p4VU6iUxUZswHytLOVeCsiluAqZrGB_xuFjhX9iYbwPEvp2YZz-wiL7GnkimPwQQr9HyeSkIhlrw
```

## Kubernetes Dashboard

Token

Every Service Account has a Secret with valid Bearer Token that can be used to log in to Dashboard. To find out more about how to configure and use Bearer Tokens, please refer to the [Authentication](#) section.

Kubeconfig

Please select the kubeconfig file that you have created to configure access to the cluster. To find out more about how to configure and use kubeconfig file, please refer to the [Configure Access to Multiple Clusters](#) section.

Enter token \*

**Sign in**

Hands-on Labs  
[cks.kodekloud.com](https://cks.kodekloud.com)



{KODE} {LOUD}

[www.kodekloud.com](http://www.kodekloud.com)

# Verify Platform Binaries Before Deploying

# v1.20 Release Notes

## v1.20.0

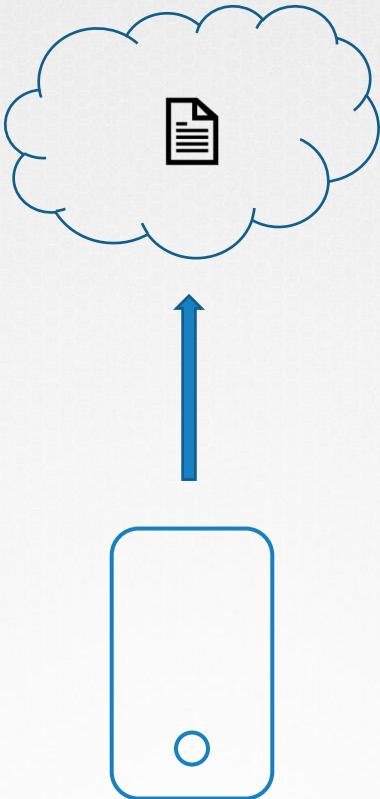
[Documentation](#)

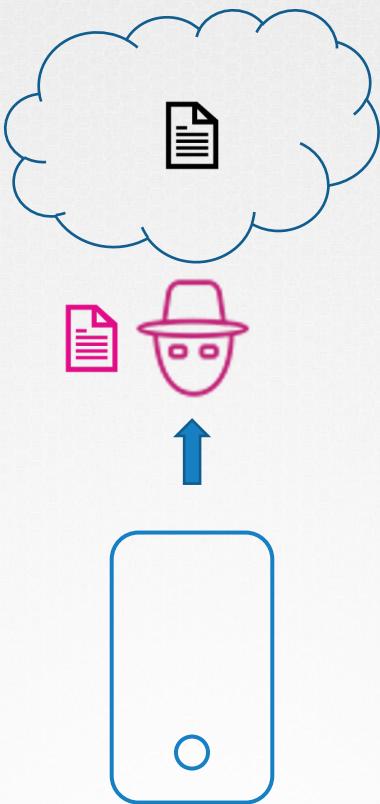
### Downloads for v1.20.0 [🔗](#)

filename	sha512 hash
<a href="#">kubernetes.tar.gz</a>	ebfe49552bbda02807034488967b3b62bf9e3e507d56245e298c4c19090387136572c1fca789e772a5e8a19535531d01dcedb61980e42ca7b0461d3864 df2c14
<a href="#">kubernetes-src.tar.gz</a>	bcbd67ed0bb77840828c08c6118ad0c9bf2bcd16763afaaf8731fd6ce735be654feef61e554bcc34c77c65b02a25dae565adc5e1dc49a2daaa0d115b f1efef6

### Client Binaries

filename	sha512 hash
<a href="#">kubernetes-client-darwin-amd64.tar.gz</a>	3609f6483f4244676162232b3294d7a2dc40ae5bdd86a842a05aa768f5223b8f50e1d6420fd8afb2d0ce19de06e1d38e5e5b10154ba0c b71a74233e6dc94d5a0
<a href="#">kubernetes-client-linux-386.tar.gz</a>	e06c08016a08137d39804383fdc33a40bb2567aa77d88a5c3fd5b9d93f5b581c635b2c4faaa718ed3bb2d120cb14fe91649ed4469ba72 c3a3dda1e343db545ed







5c97e0a6cd3e7a4fa0c189037e85fe9adff27c345e0398ccb791d2c16881  
466f



37b2ecd287c9a5e4aeaaa567d3a573329d2a576f8ff4605202c0eabd94d2  
9234

Kubernetes Documentation / Getting started / Release notes and version skew / v1.20 Release Notes

## v1.20 Release Notes

### v1.20.0

[Documentation](#)

### Downloads for v1.20.0

filename	sha512 hash
<a href="#">kubernetes.tar.gz</a>	ebfe49552bbda02807034488967b3b62bf9e3e507d56245e298c4c19090387136572c1fca789e772a5e8a19535531d01dcedb61980e42ca7b0461d3864 df2c14

```
curl https://dl.k8s.io/v1.20.0/kubernetes.tar.gz -L -o kubernetes.tar.gz
```

```
shasum -a 512 kubernetes.tar.gz
```

```
ebfe49552bbda02807034488967b3b62bf9e3e507d56245e298c4c19090387136572c1fca789e772a5e8a19535531d01dcedb61980e42ca7b0461d3864  
df2c14
```

<https://kubernetes.io/docs/setup/release/notes/>

# Generate SHA

MacOS

```
▶ shasum -a 512 kubernetes.tar.gz
```

Linux

```
▶ sha512sum kubernetes.tar.gz
```

Hands-on Labs  
[cks.kodekloud.com](https://cks.kodekloud.com)



{KODE} {LOUD}

[www.kodekloud.com](http://www.kodekloud.com)

# Kubernetes Releases



```
kubectl get nodes
```

NAME	STATUS	ROLES	AGE	VERSION
master	Ready	master	1d	v1.11.3
node-1	Ready	<none>	1d	v1.11.3
node-2	Ready	<none>	1d	v1.11.3

v1.11.3

The diagram shows the version number 'v1.11.3' in large blue letters. Below each of the three digits '1', '11', and '3' is a small blue curly brace. Underneath these three groups of braces are three blue rectangular boxes. The first box contains the word 'MAJOR', the second contains 'MINOR', and the third contains 'PATCH'. This visualizes how each part of the version number corresponds to a specific level of change in a software update.

MAJOR    MINOR    PATCH

-Features    -Bug Fixes  
-Functionalities

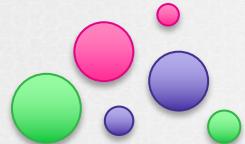


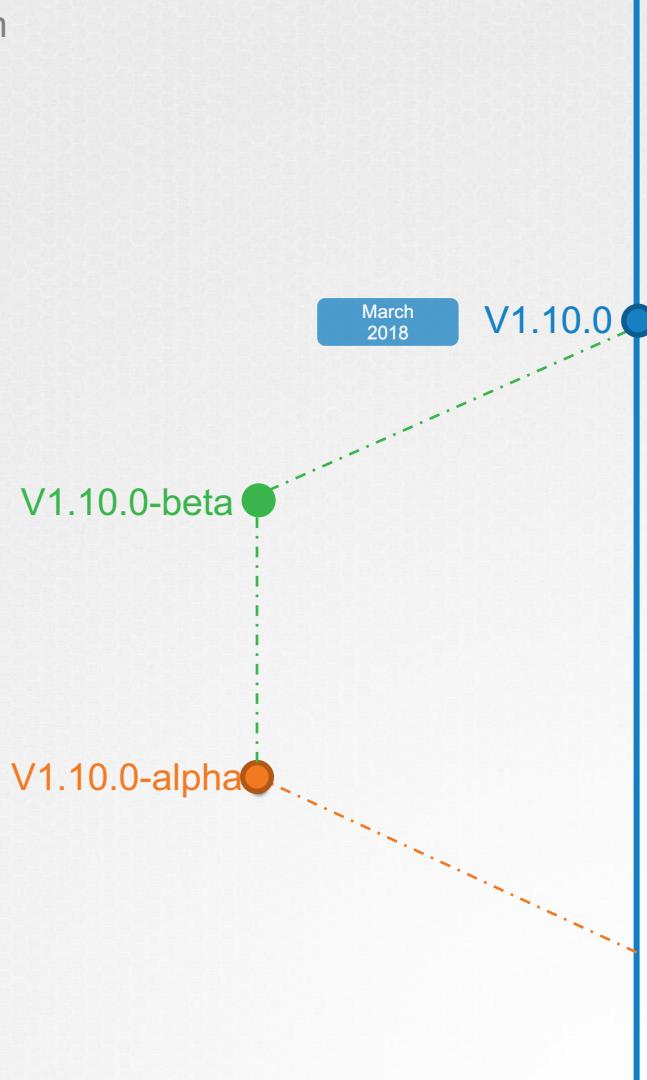


March  
2018

V1.10.0

V1.10.0-alpha





www.kodekloud.com

kubernetes / kubernetes

Watch 2,918 Unstar 49,224 Fork 17,023

Code Issues 2,151 Pull requests 992 Projects 11 Insights

Releases Tags

8 days ago v1.13.5-beta.0 ...  
9cb83c5 zip tar.gz

v1.13.4  
c27b913

k8s-release-robot released this 8 days ago · 8 commits to release-1.13 since this release

See [kubernetes-announce@](#) and [CHANGELOG-1.13.md](#) for details.

SHA512 for [kubernetes.tar.gz](#) :  
591cd3f4f479744a1d47544902817350321c63f8c37ad771d559e293bcdcb421e89d62663300a6739c667d34e1e24bb080dd735  
62dc29713381db079ba6e9223

Additional binary downloads are linked in the [CHANGELOG-1.13.md](#).

Assets 3

<a href="#">kubernetes.tar.gz</a>	1.85 MB
<a href="#">Source code (zip)</a>	
<a href="#">Source code (tar.gz)</a>	

<https://github.com/kubernetes/kubernetes/releases>



v1.13.4



# References

<https://github.com/kubernetes/community/blob/master/contributors/design-proposals/release/versioning.md>

<https://github.com/kubernetes/community/blob/master/contributors/design-proposals/api-machinery/api-group.md>

<https://blog.risingstack.com/the-history-of-kubernetes/>

<https://kubernetes.io/docs/setup/version-skew-policy/>

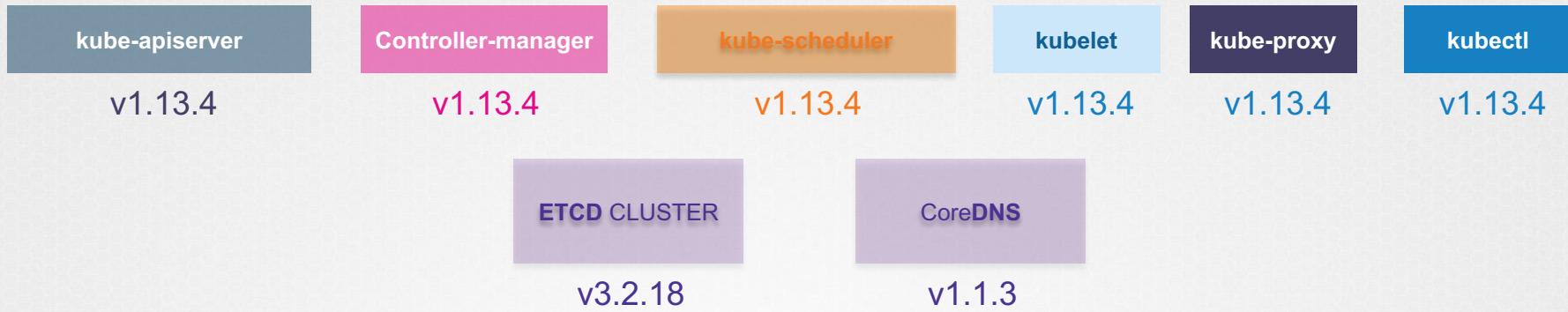


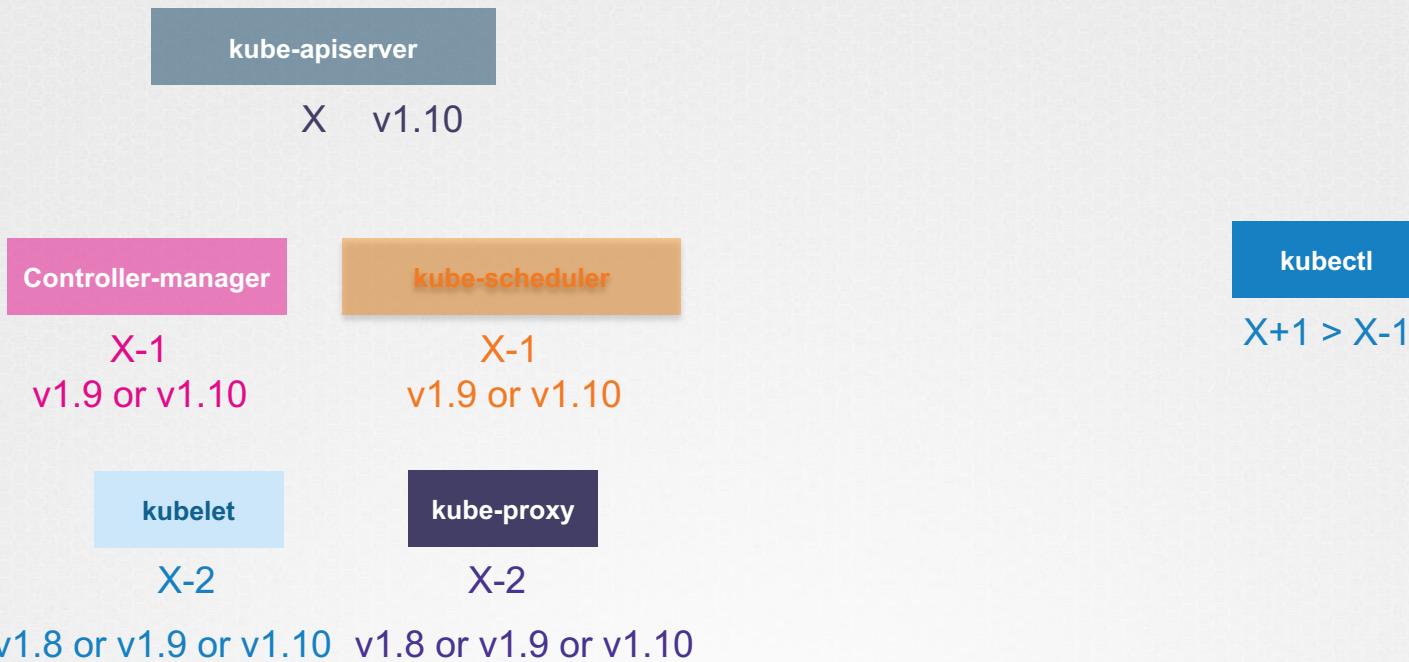
{KODE} {LOUD}

[www.kodekloud.com](http://www.kodekloud.com)

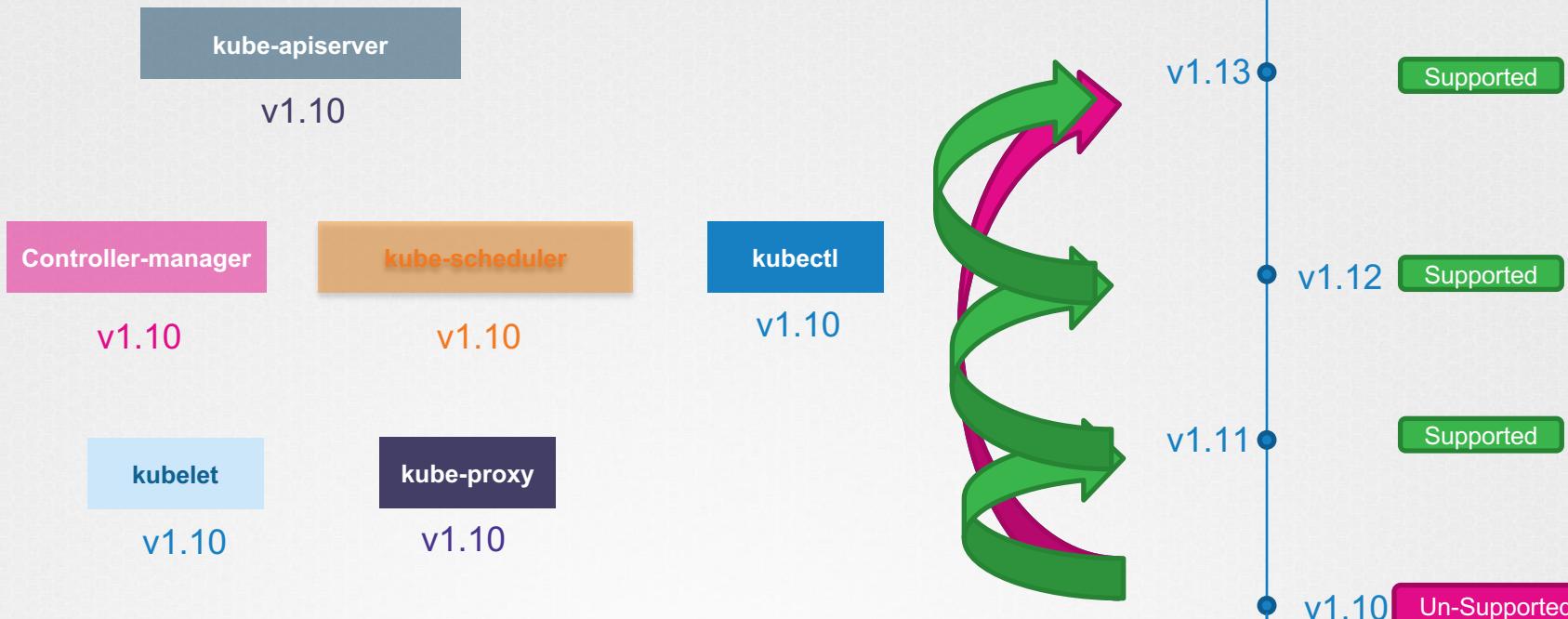
# Cluster Upgrade Process













✓ standard-cluster-1

Details Storage Nodes

#### Cluster

Master version	1.10.12-gke.7	<a href="#">Upgrade available</a>
Endpoint	35.238.15.143	<a href="#">Show credentials</a>
Client certificate	Enabled	
Binary authorisation	Disabled	
Kubernetes alpha features	Disabled	
Total size	3	
Master zone	us-central1-a	
Node zones	us-central1-a	
Network	default	

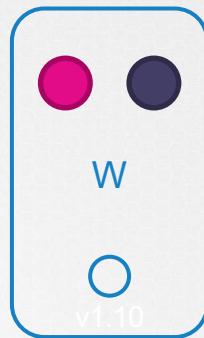
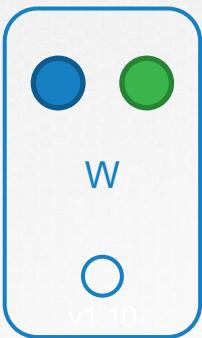
kubeadm

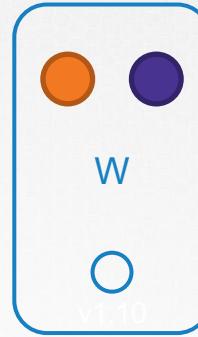
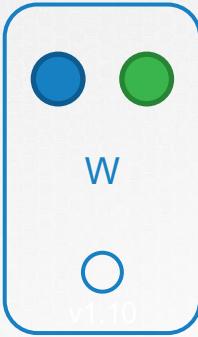
“The hard way”

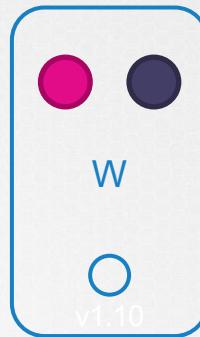
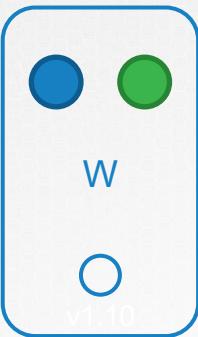
▶ kubeadm upgrade plan

▶ kubeadm upgrade apply





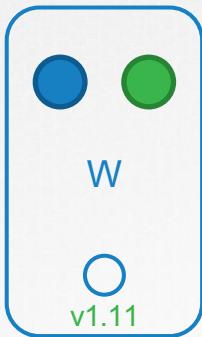




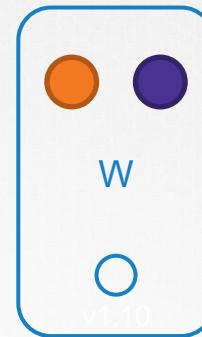
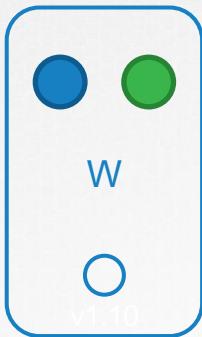
# Strategy - 1



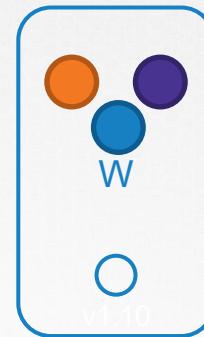
# Strategy - 1



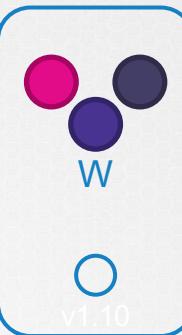
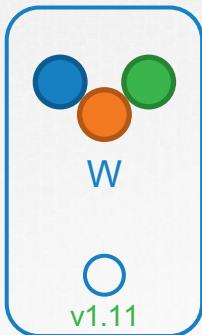
# Strategy - 2



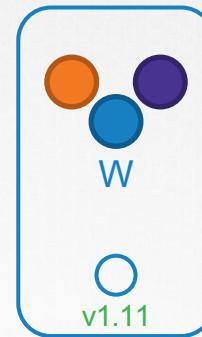
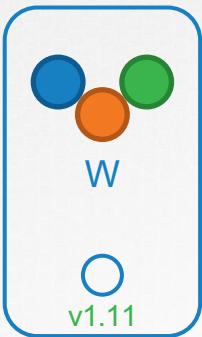
# Strategy - 2



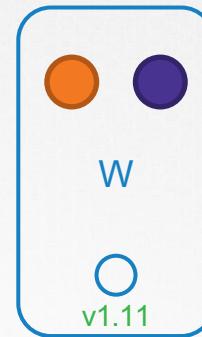
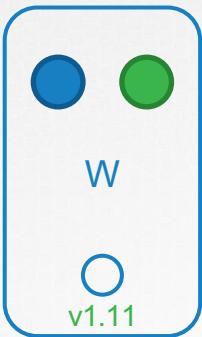
# Strategy - 2



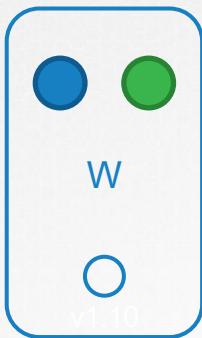
# Strategy - 2



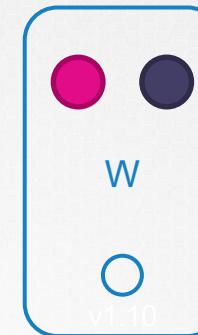
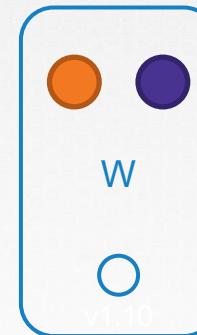
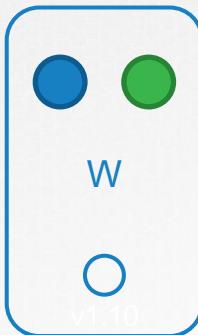
# Strategy - 2



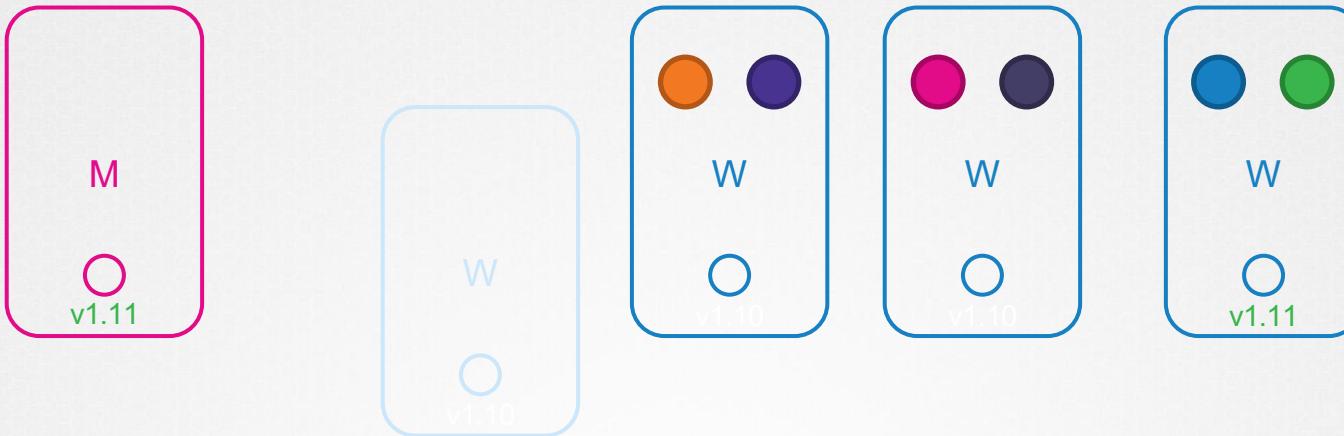
# Strategy - 3



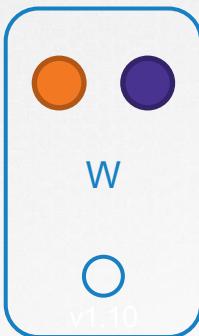
# Strategy - 3



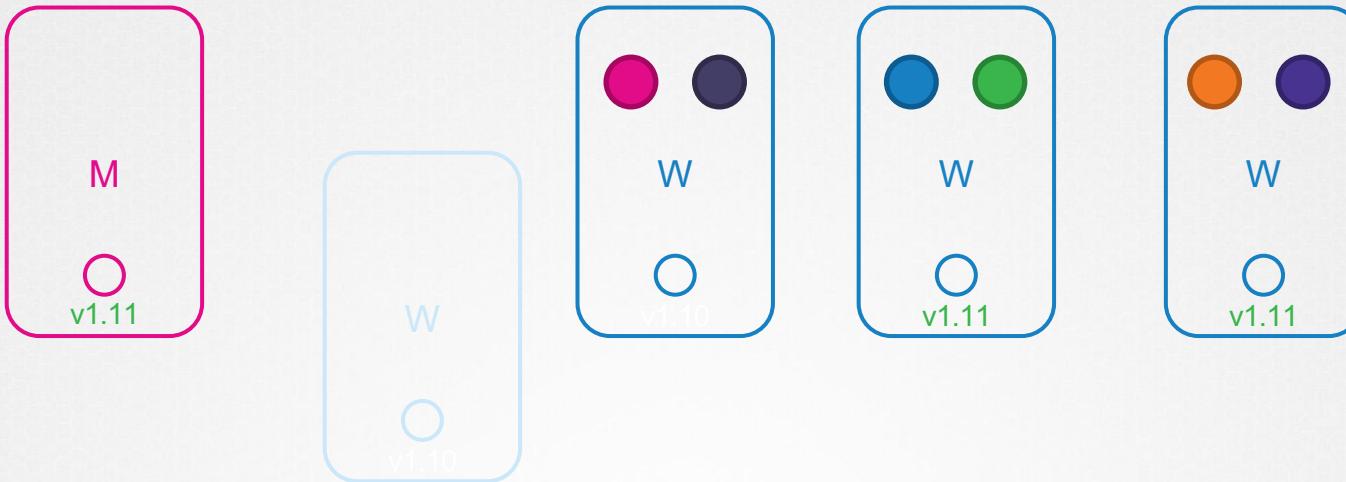
# Strategy - 3



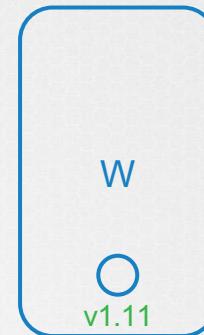
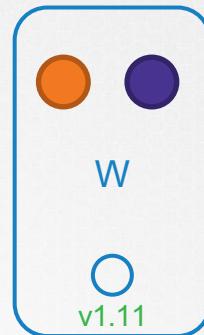
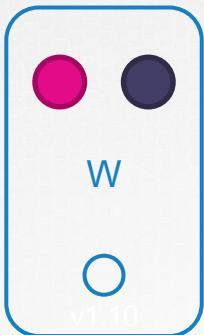
# Strategy - 3



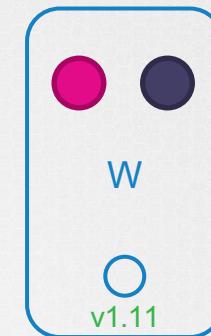
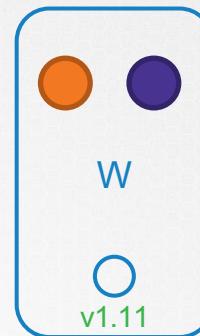
# Strategy - 3



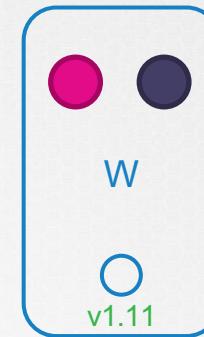
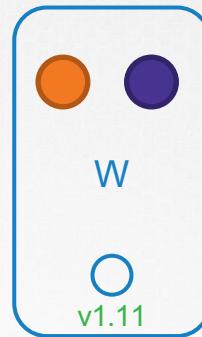
# Strategy - 3



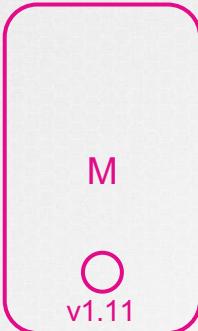
# Strategy - 3



# Strategy - 3



# kubeadm - upgrade



```
▶ kubeadm upgrade plan
```

```
[preflight] Running pre-flight checks.
[upgrade] Making sure the cluster is healthy:
[upgrade/config] Making sure the configuration is correct:
[upgrade] Fetching available versions to upgrade to
[upgrade/versions] Cluster version: v1.11.8
[upgrade/versions] kubeadm version: v1.11.3
[upgrade/versions] Latest stable version: v1.13.4
[upgrade/versions] Latest version in the v1.11 series: v1.11.8
```

```
Components that must be upgraded manually after you have
upgraded the control plane with 'kubeadm upgrade apply':
```

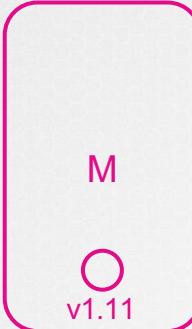
COMPONENT	CURRENT	AVAILABLE
Kubelet	3 x v1.11.3	v1.13.4

```
Upgrade to the latest stable version:
```

COMPONENT	CURRENT	AVAILABLE
API Server	v1.11.8	v1.13.4
Controller Manager	v1.11.8	v1.13.4
Scheduler	v1.11.8	v1.13.4
Kube Proxy	v1.11.8	v1.13.4
CoreDNS	1.1.3	1.1.3
Etcd	3.2.18	N/A

```
You can now apply the upgrade by executing the following command:
```

# kubeadm - upgrade



```
▶ kubeadm upgrade plan  
[preflight] Running pre-flight checks.  
[upgrade] Making sure the cluster is healthy:  
[upgrade/config] Making sure the configuration is correct:  
[upgrade] Fetching available versions to upgrade to  
[upgrade/versions] Cluster version: v1.11.8  
[upgrade/versions] kubeadm version: v1.11.3  
[upgrade/versions] Latest stable version: v1.13.4  
[upgrade/versions] Latest version in the v1.11 series: v1.11.8
```

Components that must be **upgraded manually** after you have upgraded the control plane with 'kubeadm upgrade apply':

COMPONENT	CURRENT	AVAILABLE
Kubelet	3 x v1.11.3	v1.13.4

Upgrade to the latest stable version:

COMPONENT	CURRENT	AVAILABLE
API Server	v1.11.8	v1.13.4
Controller Manager	v1.11.8	v1.13.4
Scheduler	v1.11.8	v1.13.4
Kube Proxy	v1.11.8	v1.13.4
CoreDNS	1.1.3	1.1.3
Etcd	3.2.18	N/A

You can now apply the upgrade by executing the following command:

```
kubeadm upgrade apply v1.13.4
```

Note: Before you can perform this upgrade, you have to update kubeadm to v1.13.4.

# kubeadm - upgrade

```
▶ apt-get upgrade -y kubeadm=1.12.0-00
```

```
▶ kubeadm upgrade apply v1.12.0
```

...

[upgrade/successful] SUCCESS! Your cluster was upgraded to "v1.12.0". Enjoy!

[upgrade/kubelet] Now that your control plane is upgraded, please proceed with upgrading your kubelets if you haven't already done so.

```
▶ kubectl get nodes
```

NAME	STATUS	ROLES	AGE	VERSION
master	Ready	master	1d	v1.11.3
node-1	Ready	<none>	1d	v1.11.3
node-2	Ready	<none>	1d	v1.11.3



```
▶ apt-get upgrade -y kubelet=1.12.0-00
```

```
▶ systemctl restart kubelet
```

# kubeadm - upgrade

```
▶ kubectl get nodes
```

NAME	STATUS	ROLES	AGE	VERSION
master	Ready	master	1d	v1.11.3
node-1	Ready	<none>	1d	v1.11.3
node-2	Ready	<none>	1d	v1.11.3



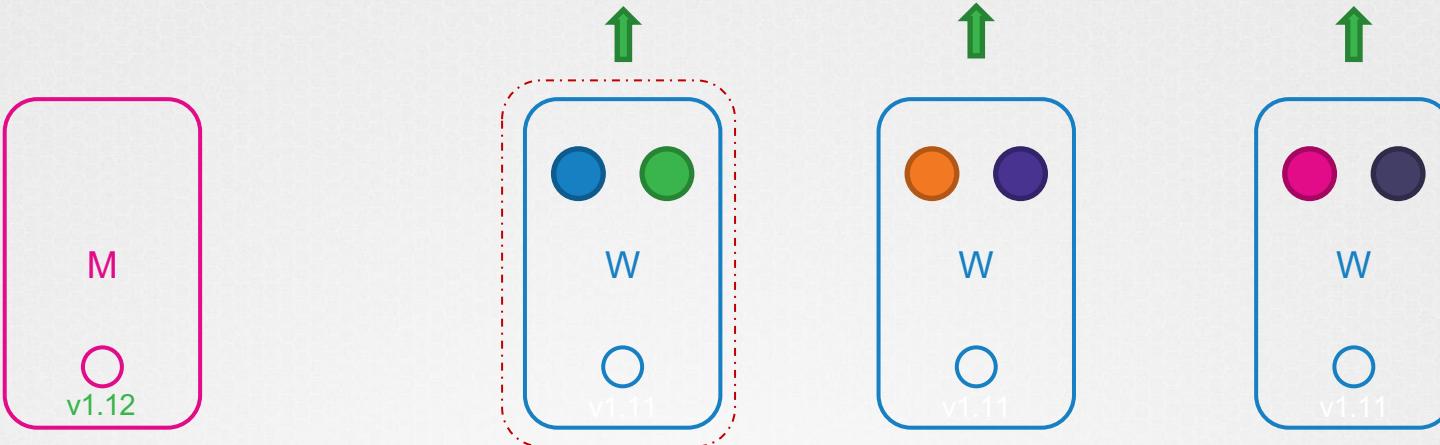
```
▶ apt-get upgrade -y kubelet=1.12.0-00
```

```
▶ systemctl restart kubelet
```

```
▶ kubectl get nodes
```

NAME	STATUS	ROLES	AGE	VERSION
master	Ready	master	1d	v1.12.0
node-1	Ready	<none>	1d	v1.11.3
node-2	Ready	<none>	1d	v1.11.3

# kubeadm - upgrade



```
▶ kubectl drain node-1
```

# kubeadm - upgrade



```
▶ kubectl drain node-1
```

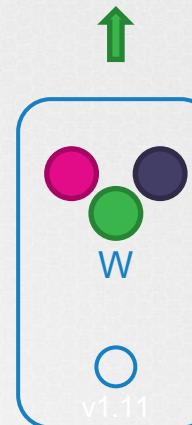
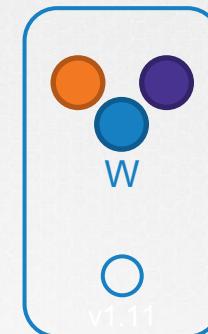


```
▶ apt-get upgrade -y kubeadm=1.12.0-00
```

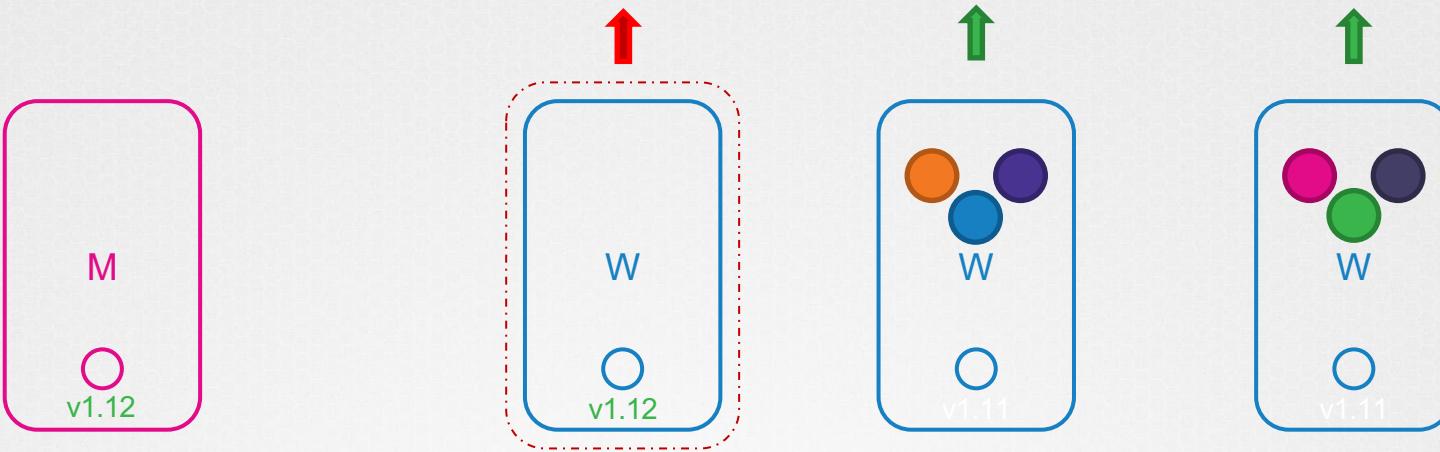
```
▶ apt-get upgrade -y kubelet=1.12.0-00
```

```
▶ kubeadm upgrade node config --kubelet-version v1.12.0
```

```
▶ systemctl restart kubelet
```



# kubeadm - upgrade



```
▶ kubectl drain node-1
```

```
▶ kubectl uncordon node-1
```

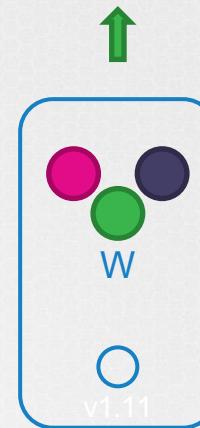
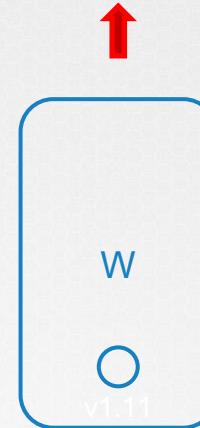
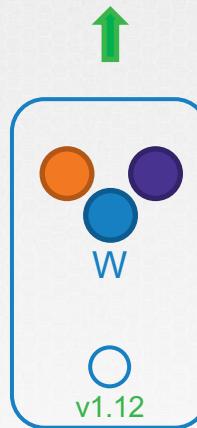
```
▶ apt-get upgrade -y kubeadm=1.12.0-00
```

```
▶ apt-get upgrade -y kubelet=1.12.0-00
```

```
▶ kubeadm upgrade node config --kubelet-version v1.12.0
```

```
▶ systemctl restart kubelet
```

# kubeadm - upgrade



```
▶ kubectl drain node-1
```

```
▶ kubectl uncordon node-1
```

```
▶ kubectl drain node-2
```

```
▶ kubectl uncordon node-2
```

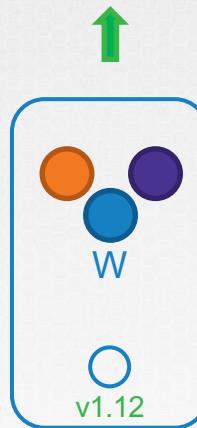
```
▶ apt-get upgrade -y kubeadm=1.12.0-00
```

```
▶ apt-get upgrade -y kubelet=1.12.0-00
```

```
▶ kubeadm upgrade node config --kubelet-vers
```

```
▶ systemctl restart kubelet
```

# kubeadm - upgrade



```
▶ kubectl drain node-1
```

```
▶ kubectl uncordon node-1
```

```
▶ kubectl drain node-2
```

```
▶ kubectl uncordon node-2
```

```
▶ kubectl drain node-3
```

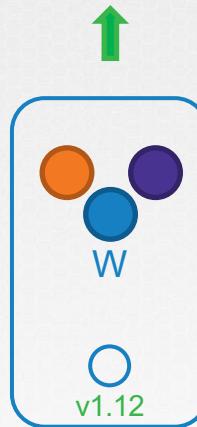
```
▶ apt-get upgrade -y l
```

```
▶ apt-get upgrade -y l
```

```
▶ kubeadm upgrade node
```

```
▶ systemctl restart k
```

# kubeadm - upgrade



```
▶ kubectl drain node-1  
▶ kubectl uncordon node-1  
▶ kubectl drain node-2  
▶ kubectl uncordon node-2  
▶ kubectl drain node-3  
▶ kubectl uncordon node-3
```



{KODE} {LOUD}

[www.kodekloud.com](http://www.kodekloud.com)



# Backup and Restore



# Backup Candidates



Resource Configuration



ETCD Cluster



Persistent Volumes

# Imperative



Resource Configuration

```
▶ kubectl create namespace new-namespace
```

```
▶ kubectl create secret
```

```
▶ kubectl create configmap
```

# Declarative



Resource Configuration

```
pod-definition.yml
```

```
apiVersion: v1
```

```
kind: Pod
```

```
metadata:
```

```
  name: myapp-pod
```

```
  labels:
```

```
    app: myapp
```

```
    type: front-end
```

```
spec:
```

```
  containers:
```

```
  - name: nginx-container
```

```
    image: nginx
```

```
▶ kubectl apply -f pod-definition.yml
```

# Backup – Resource Configs

kube-apiserver



Resource Configuration

```
▶ kubectl get all --all-namespaces -o yaml > all-deploy-services.yaml
```



VELERO

Formerly called ARK by HeptIO

# Backup - ETCD

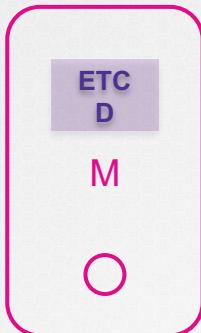
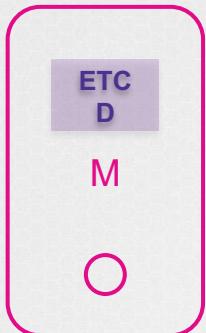


ETCD Cluster

# Backup - ETCD



ETCD Cluster



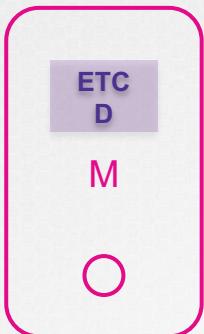
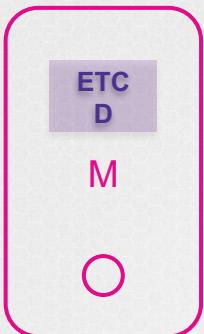
## etcd.service

```
ExecStart=/usr/local/bin/etcd \
--name ${ETCD_NAME} \
--cert-file=/etc/etcd/kubernetes.pem \
--key-file=/etc/etcd/kubernetes-key.pem \
--peer-cert-file=/etc/etcd/kubernetes.pem \
--peer-key-file=/etc/etcd/kubernetes-key.pem \
--trusted-ca-file=/etc/etcd/ca.pem \
--peer-trusted-ca-file=/etc/etcd/ca.pem \
--peer-client-cert-auth \
--client-cert-auth \
--initial-advertise-peer-urls https://$INTERNAL_IP:2380 \
--listen-peer-urls https://$INTERNAL_IP:2380 \
--listen-client-urls https://$INTERNAL_IP:2379,https://$INTERNAL_IP:2379 \
--advertise-client-urls https://$INTERNAL_IP:2379 \
--initial-cluster-token etcd-cluster-0 \
--initial-cluster controller-0=https://$CONTROLLER0_IP:2380 \
--initial-cluster-state new \
--data-dir=/var/lib/etcd
```

# Backup - ETCD



ETCD Cluster



```
▶ ETCCTL_API=3 etcdctl \
    snapshot save snapshot.db
```

```
▶ ls
snapshot.db
```

```
▶ ETCCTL_API=3 etcdctl \
    snapshot status snapshot.db
```

HASH	REVISION	TOTAL KEYS	TOTAL SIZE
e63b3fc5	473353	875	4.1 MB

# Restore - ETCD



ETCD Cluster

```
▶ ETCDCTL_API=3 etcdctl \
    snapshot save snapshot.db
```

```
▶ ls
snapshot.db
```

```
▶ service kube-apiserver stop
Service kube-apiserver stopped
```

```
▶ ETCDCTL_API=3 etcdctl \
    snapshot restore snapshot.db \
    --data-dir /var/lib/etcd-from-backup
```

```
I | mvcc: restore compact to 475629
```

# Restore - ETCD



ETCD Cluster

```
▶ ETCDCTL_API=3 etcdctl \
    snapshot restore snapshot.db \
    --data-dir /var/lib/etcd-from-backup
```

I | mvcc: restore compact to 475629

```
▶ systemctl daemon-reload
```

```
▶ service etcd restart
```

Service etcd **restarted**

```
▶ ETCDCTL_API=3 etcdctl \
    snapshot save snapshot.db
```

```
▶ ls
snapshot.db
```

```
▶ service kube-apiserver stop
Service kube-apiserver stopped
```

```
etcd.service
```

```
ExecStart=/usr/local/bin/etcd \
--name ${ETCD_NAME} \
--cert-file=/etc/etcd/kubernetes.pem \
--key-file=/etc/etcd/kubernetes-key.pem \
--peer-cert-file=/etc/etcd/kubernetes.pem \
--peer-key-file=/etc/etcd/kubernetes-key.pem \
--trusted-ca-file=/etc/etcd/ca.pem \
--peer-trusted-ca-file=/etc/etcd/ca.pem \
--peer-client-cert-auth \
--client-cert-auth \
--initial-advertise-peer-urls https://${INTERNAL_IP}:2380 \
--listen-peer-urls https://${INTERNAL_IP}:2380 \
--listen-client-urls https://${INTERNAL_IP}:2379,http://${INTERNAL_IP}:2379 \
--advertise-client-urls https://${INTERNAL_IP}:2379 \
--initial-cluster-token etcd-cluster-0 \
--initial-cluster controller-0=https://${CONTROLLER0}:2380 \
--initial-cluster-state new \
--data-dir=/var/lib/etcd-from-backup
```

# Restore - ETCD



ETCD Cluster

```
▶ ETCDCTL_API=3 etcdctl \
    snapshot save snapshot.db
```

```
▶ ls
snapshot.db
```

```
▶ service kube-apiserver stop
Service kube-apiserver stopped
```

```
▶ ETCDCTL_API=3 etcdctl \
    snapshot restore snapshot.db \
    --data-dir /var/lib/etcd-from-backup
```

```
I | mvcc: restore compact to 475629
```

```
▶ systemctl daemon-reload
```

```
▶ service etcd restart
```

```
Service etcd restarted
```

```
▶ service kube-apiserver start
```

```
Service kube-apiserver started
```

```
▶ ETCDCTL_API=3 etcdctl \
    snapshot save snapshot.db \
    --endpoints=https://127.0.0.1:2379 \
    --cacert=/etc/etcd/ca.crt \
    --cert=/etc/etcd/etcd-server.crt \
    --key=/etc/etcd/etcd-server.key
```

# References

<https://kubernetes.io/docs/tasks/administer-cluster/configure-upgrade-etcd/>

<https://github.com/etcd-io/etcd/blob/master/Documentation/op-guide/recovery.md>

<https://www.youtube.com/watch?v=qRPNuT080Hk>



{KODE} {LOUD}

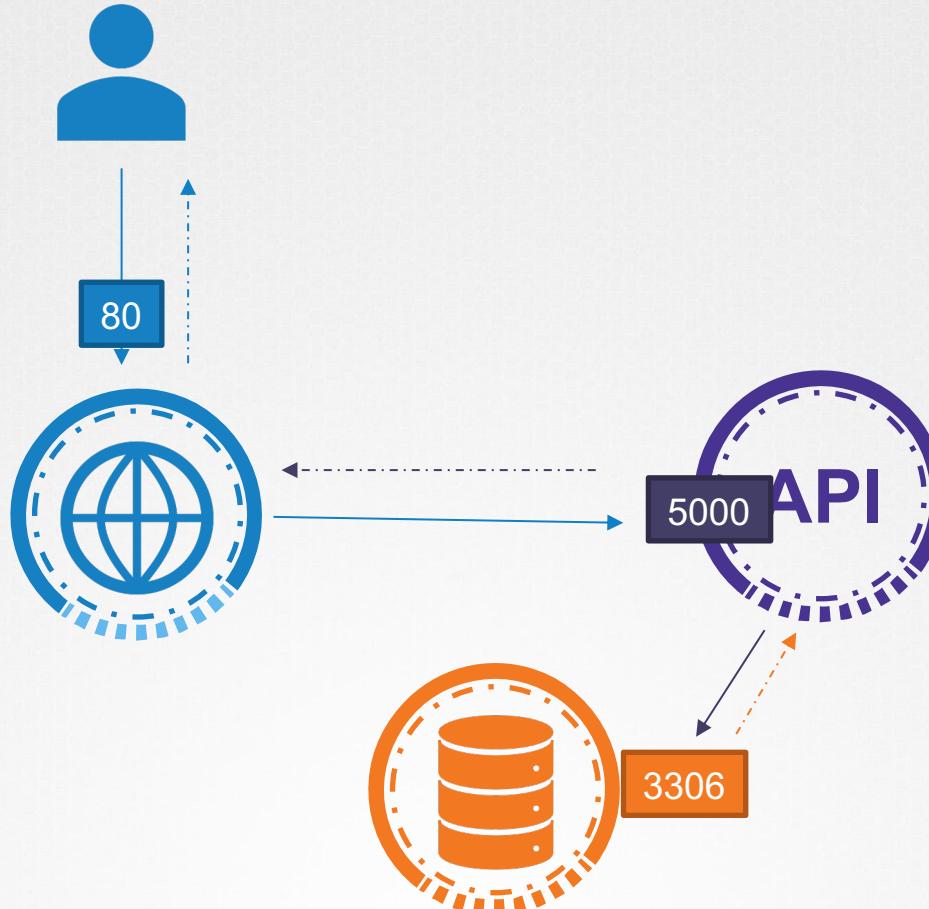
[www.kodekloud.com](http://www.kodekloud.com)



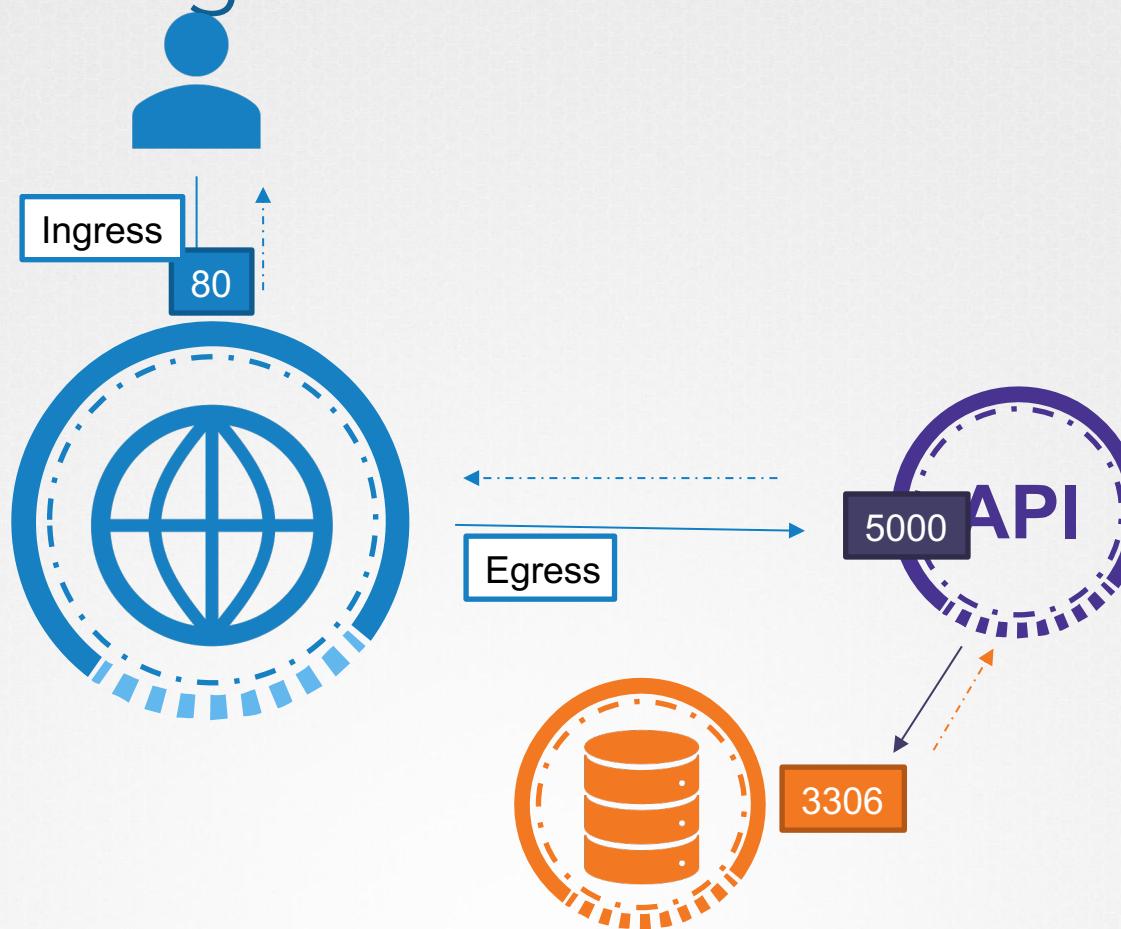
# Network Policies



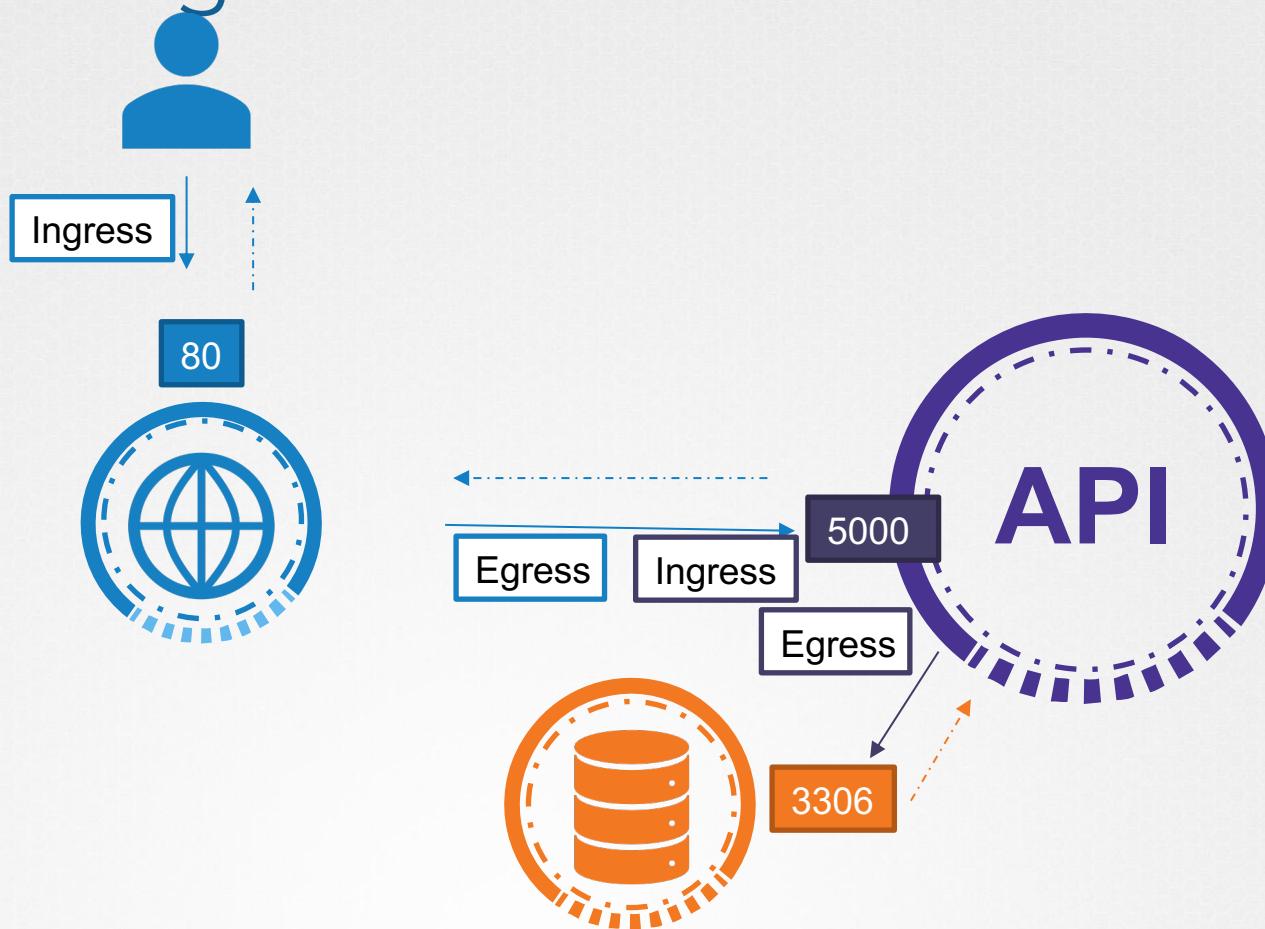
# Traffic



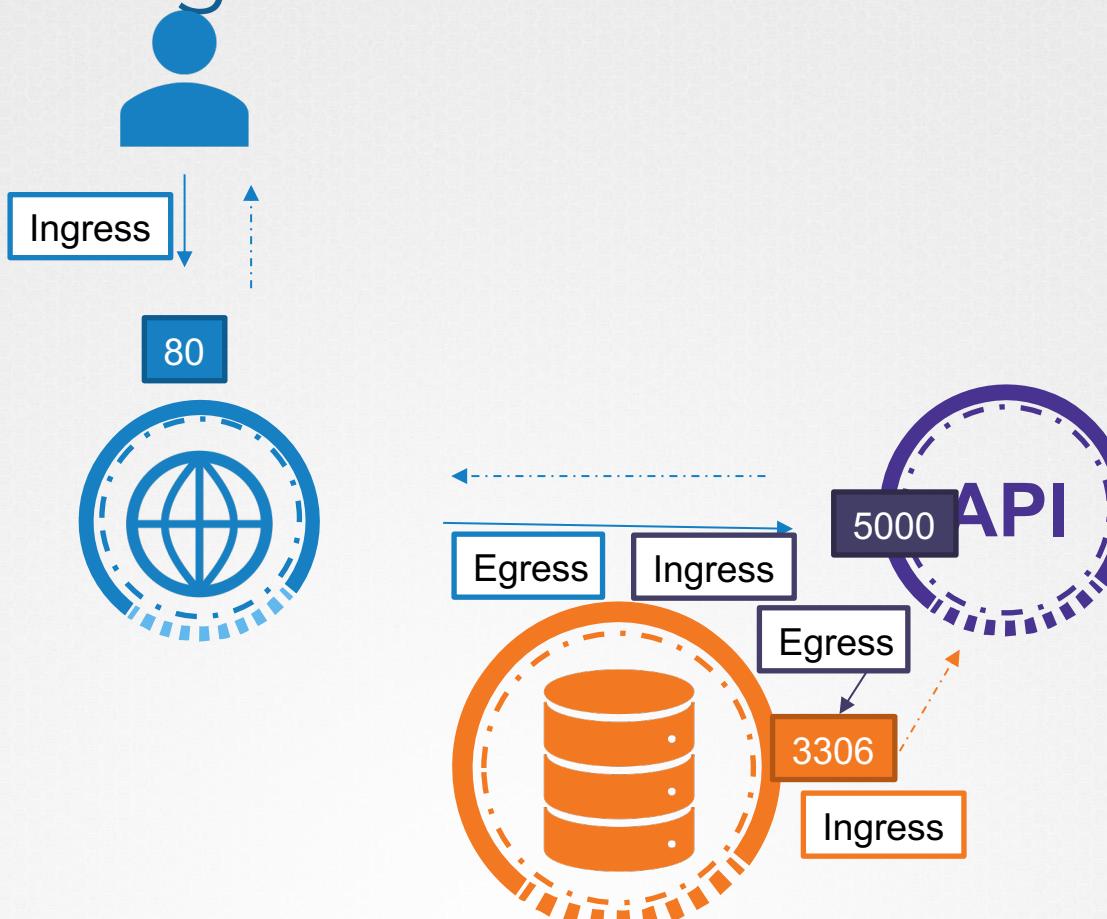
# Ingress & Egress



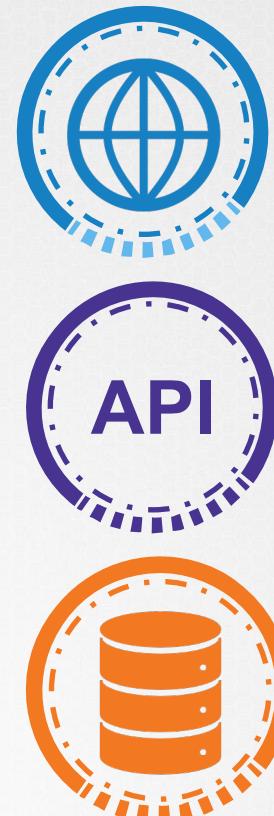
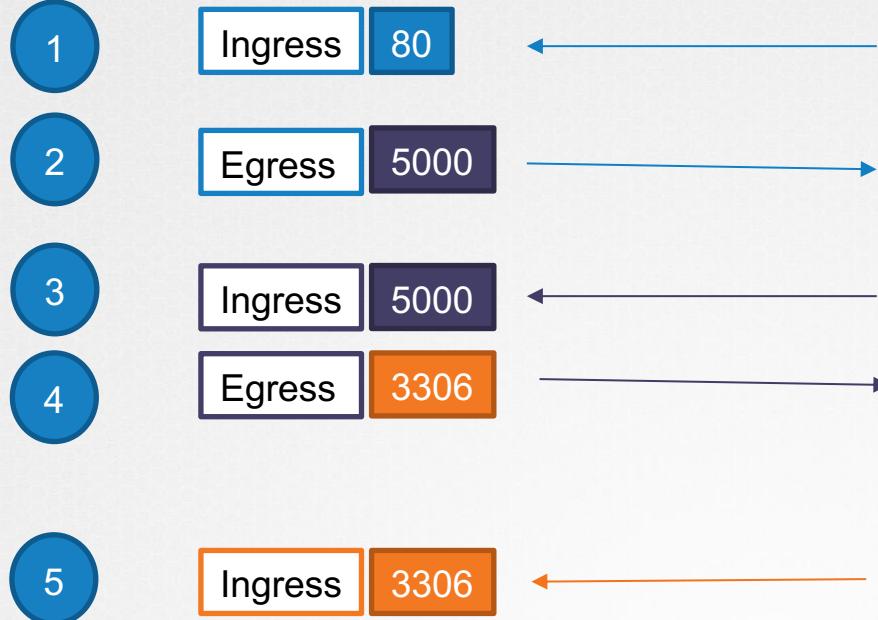
# Ingress & Egress



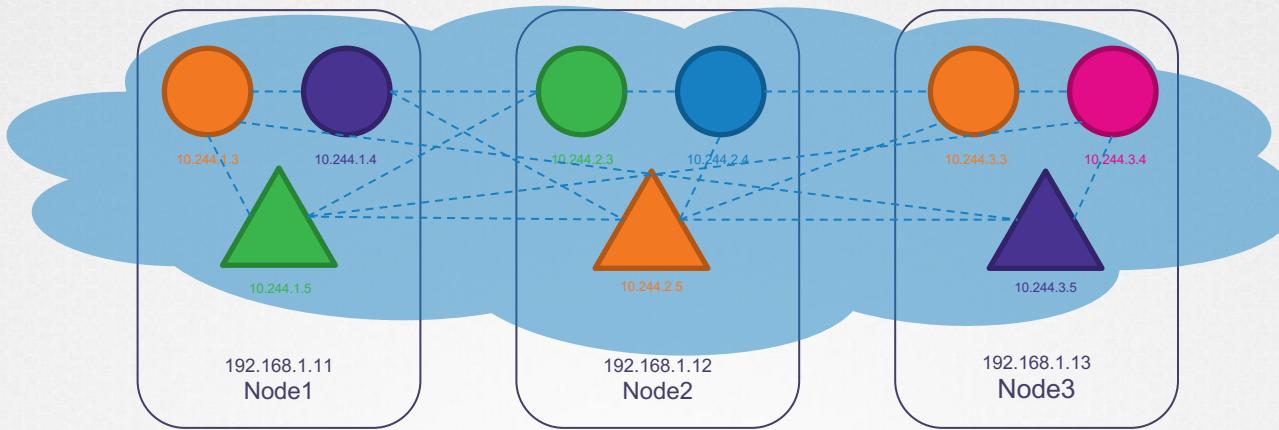
# Ingress & Egress



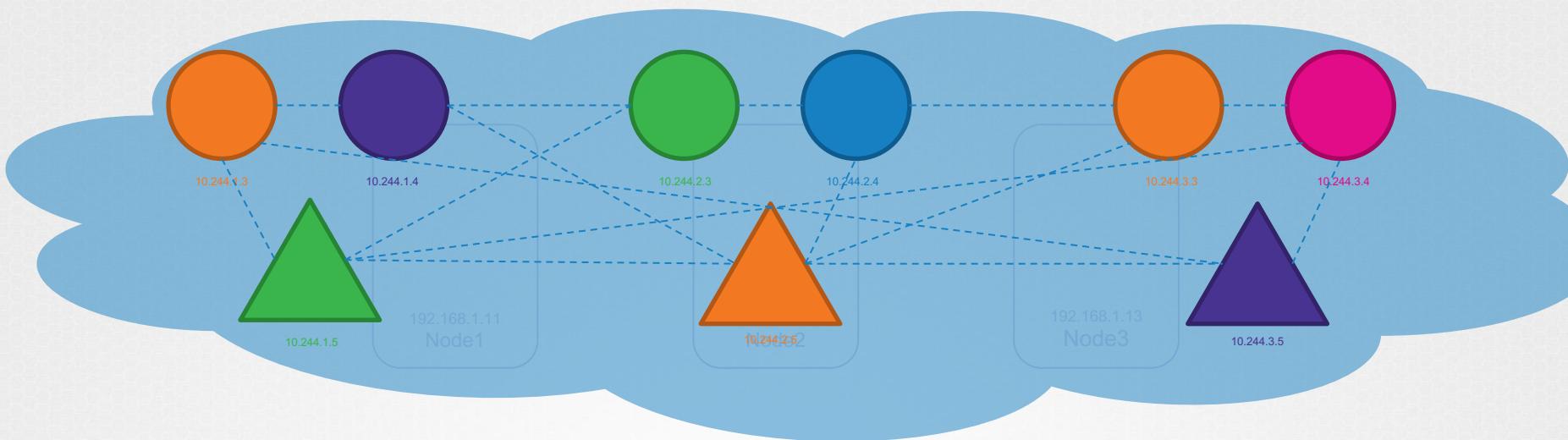
# Traffic



# Network Security

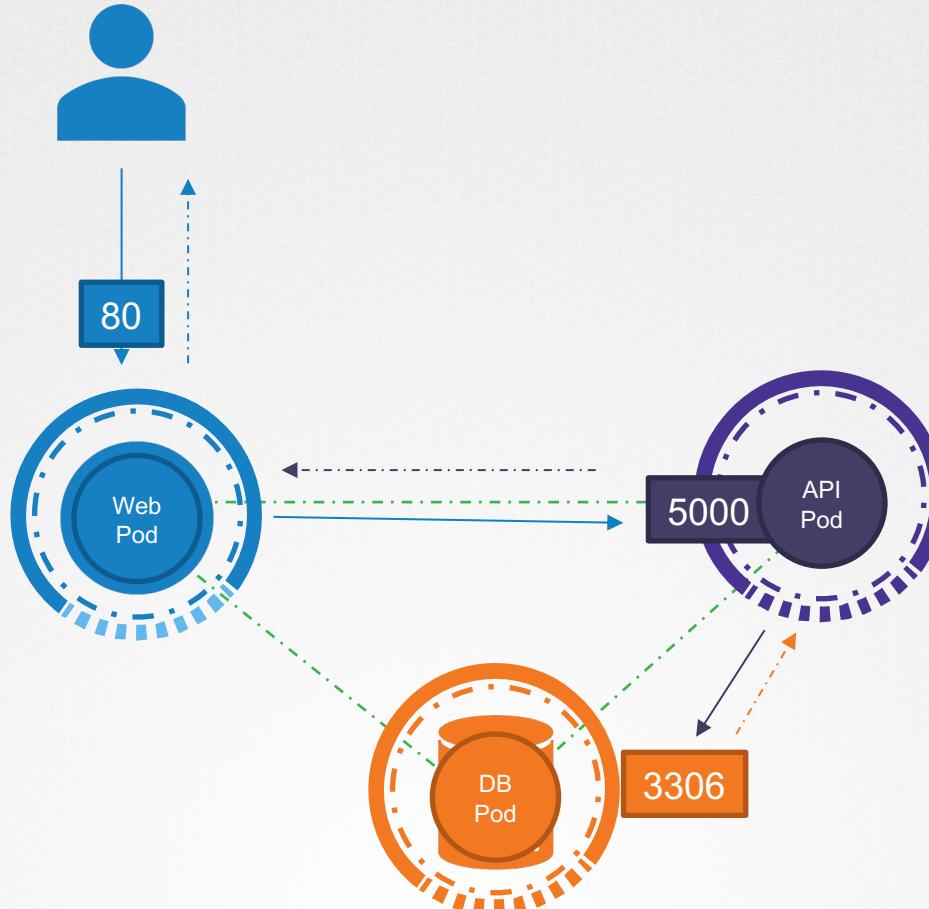


# Network Security

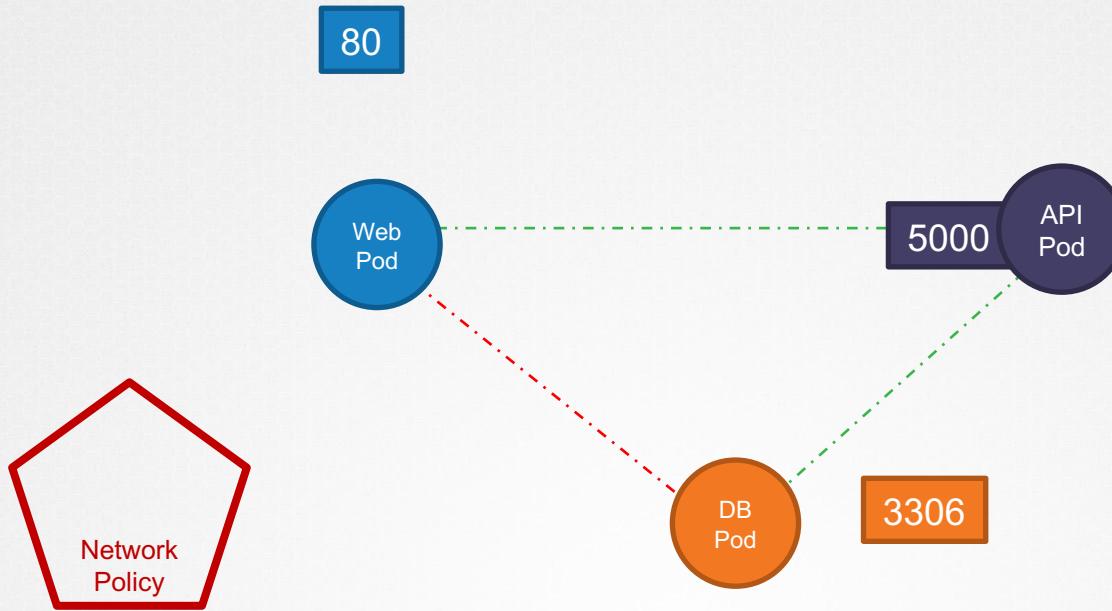


**“All Allow”**

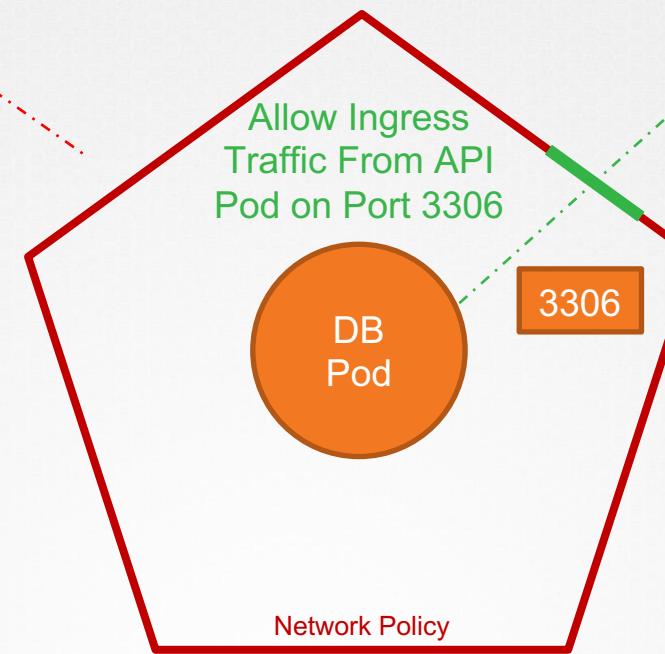
# Traffic



# Network Policy

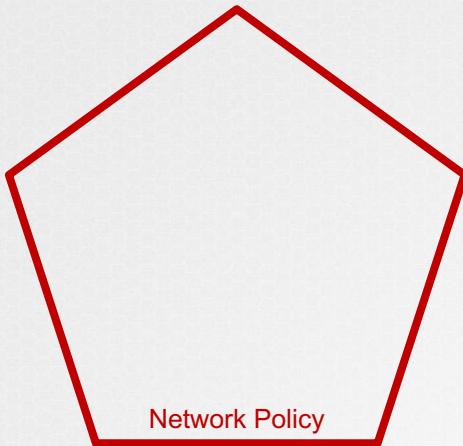


# Network Policy



# Network Policy - Selectors

Allow Ingress  
Traffic From API  
Pod on Port 3306



```
podSelector:  
  matchLabels:  
    role: db
```

```
labels:  
  role: db
```

# Network Policy - Rules

```
policyTypes:  
- Ingress  
ingress:  
- from:  
- podSelector:  
  matchLabels:  
    name: api-pod  
ports:  
- protocol: TCP  
  port: 3306
```

Allow

Ingress

Traffic

From  
API Pod

on

Port 3306

# Network Policy

```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: db-policy
spec:
```

```
▶ kubectl create -f policy-definition.yaml
matchLabels:
  role: db
```

```
policyTypes:
- Ingress
ingress:
- from:
  - podSelector:
    matchLabels:
      name: api-pod
ports:
- protocol: TCP
  port: 3306
```

# Note

## Solutions that Support Network Policies:

- Kube-router
- Calico
- Romana
- Weave-net

## Solutions that DO NOT Support Network Policies:

- Flannel



{KODE} {LOUD}

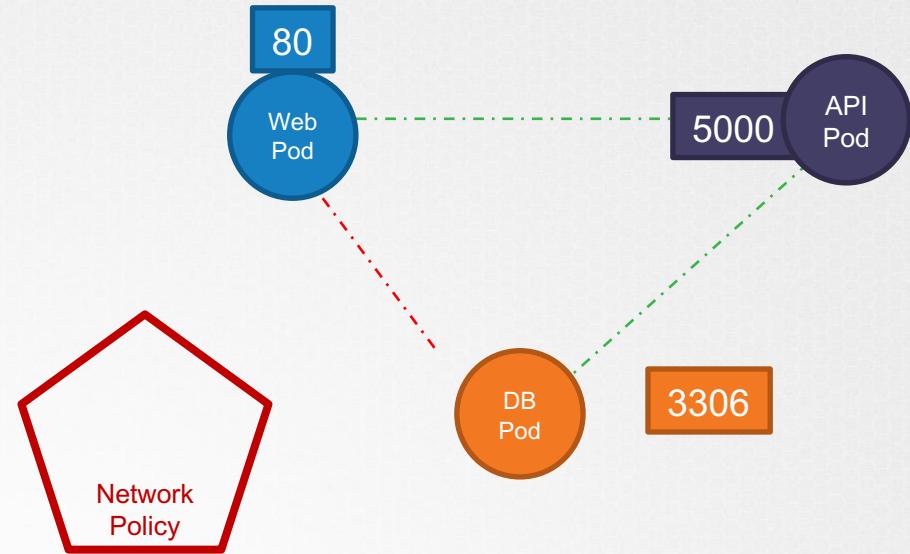
[www.kodekloud.com](http://www.kodekloud.com)



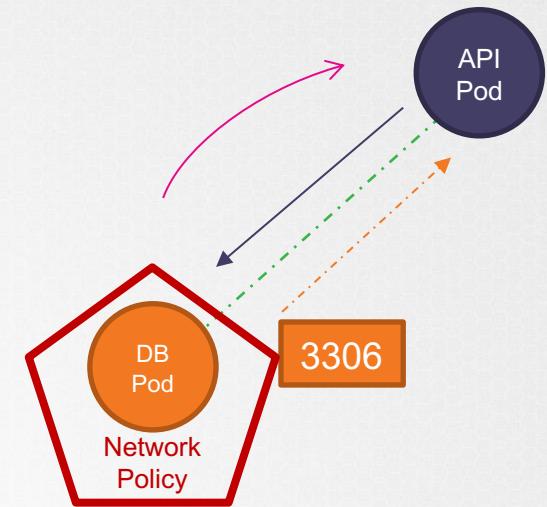
# Network Policies



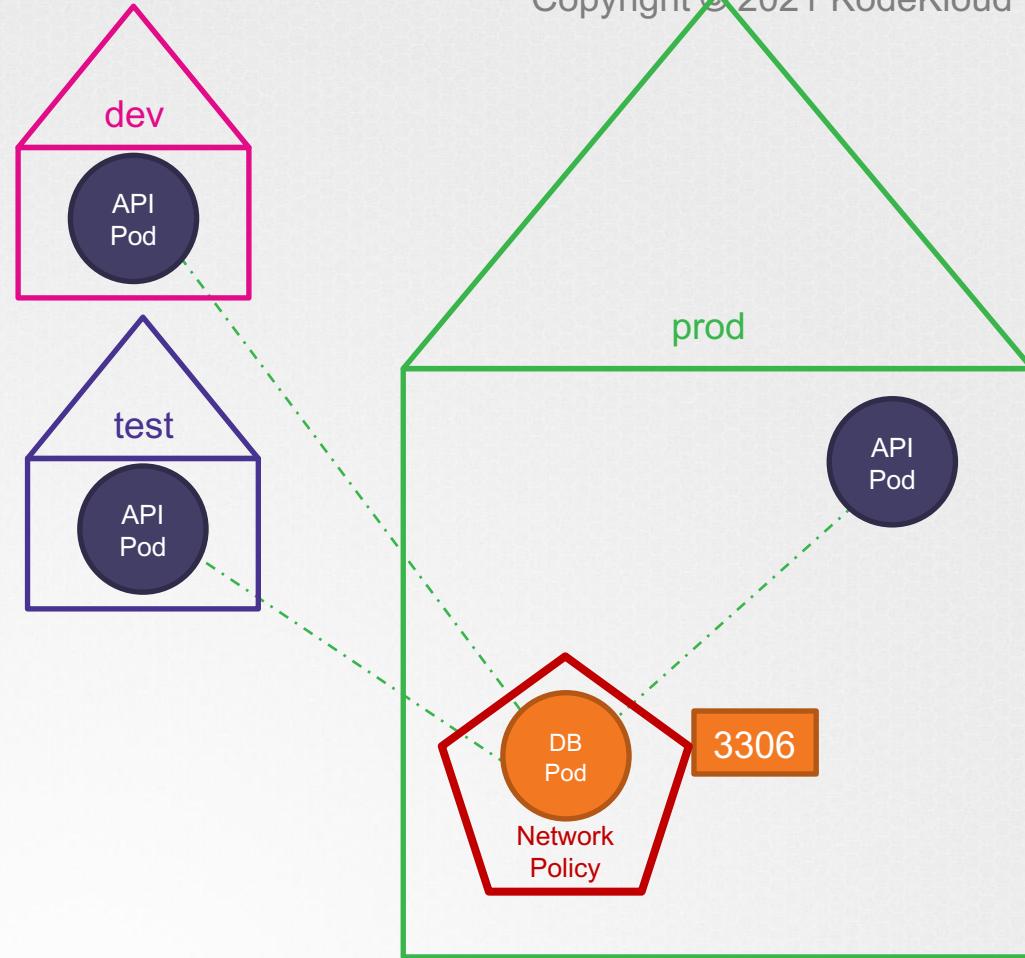
```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: db-policy
spec:
  podSelector:
    matchLabels:
      role: db
```



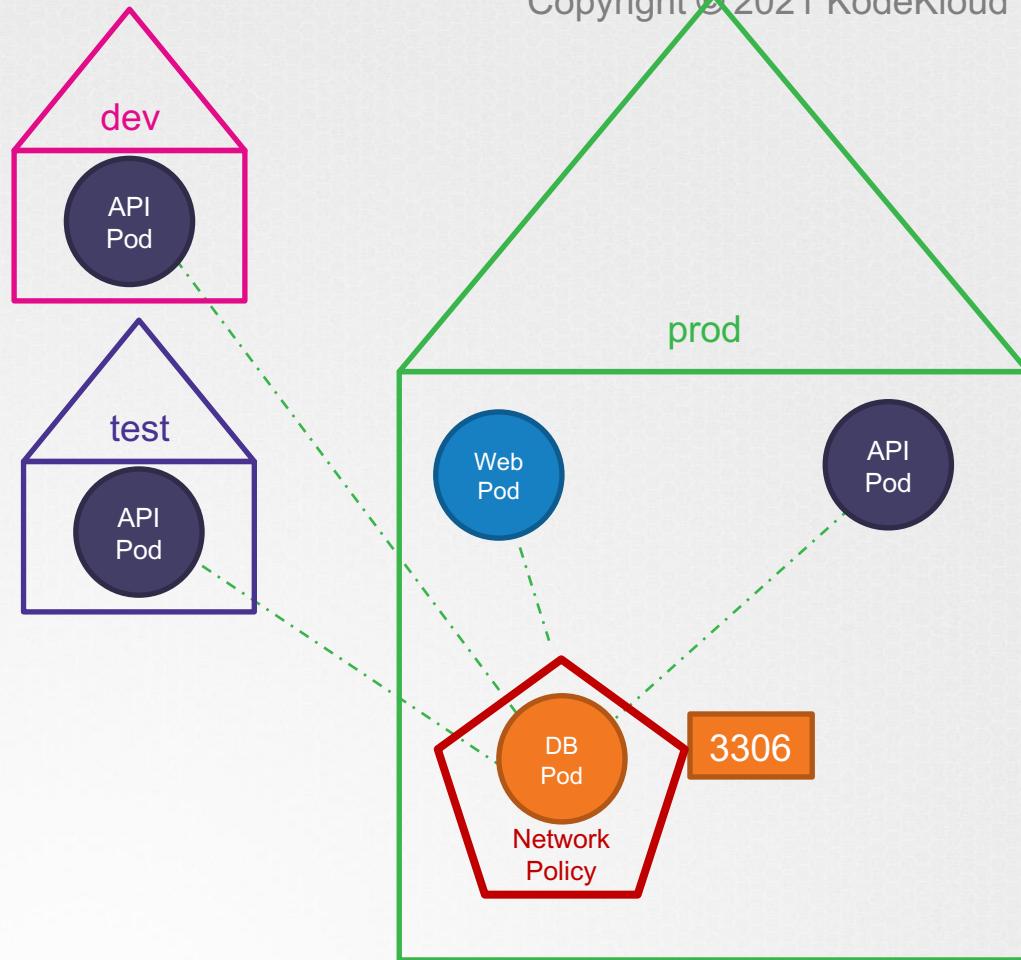
```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: db-policy
spec:
  podSelector:
    matchLabels:
      role: db
  policyTypes:
  - Ingress
```



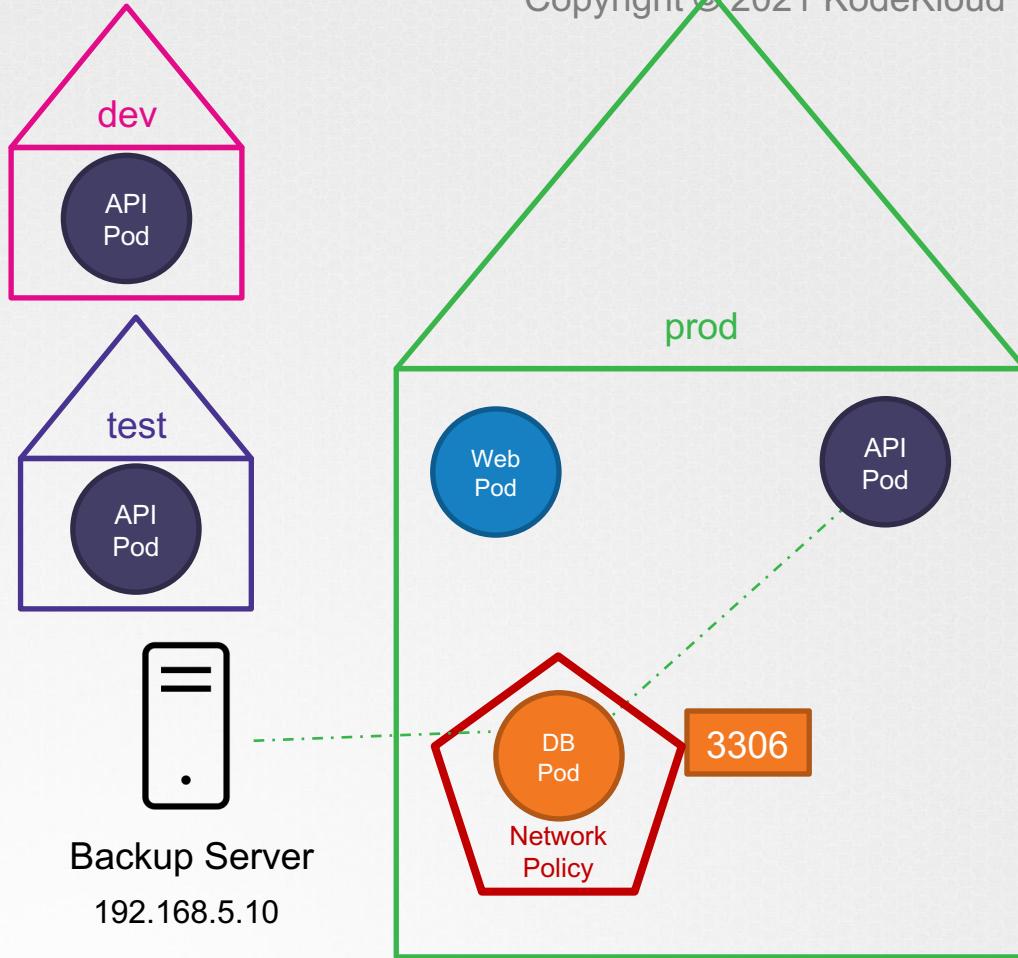
```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: db-policy
spec:
  podSelector:
    matchLabels:
      role: db
  policyTypes:
  - Ingress
  ingress:
  - from:
    - podSelector:
        matchLabels:
          name: api-pod
  ports:
  - protocol: TCP
    port: 3306
```



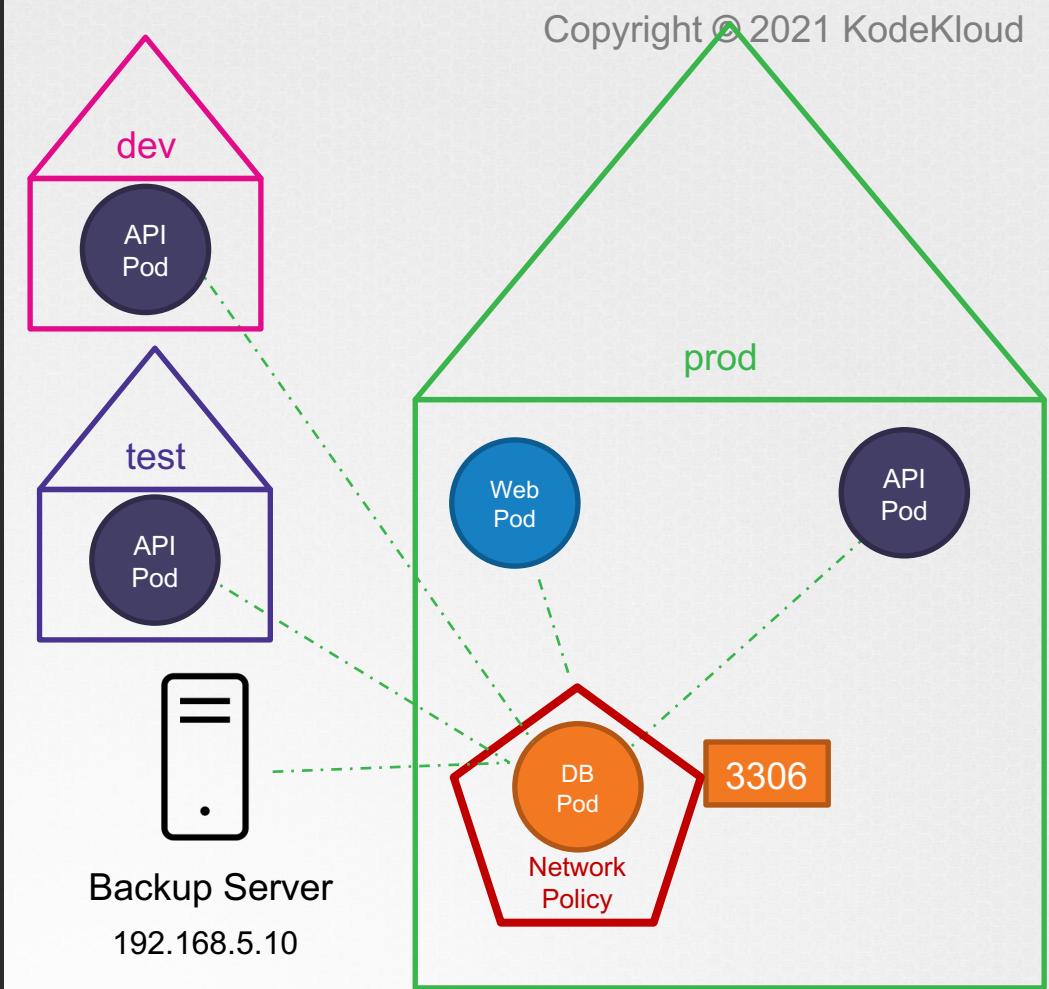
```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: db-policy
spec:
  podSelector:
    matchLabels:
      role: db
  policyTypes:
  - Ingress
  ingress:
  - from:
    - podSelector:
        matchLabels:
          name: api-pod
    namespaceSelector:
      matchLabels:
        name: prod
  ports:
  - protocol: TCP
    port: 3306
```



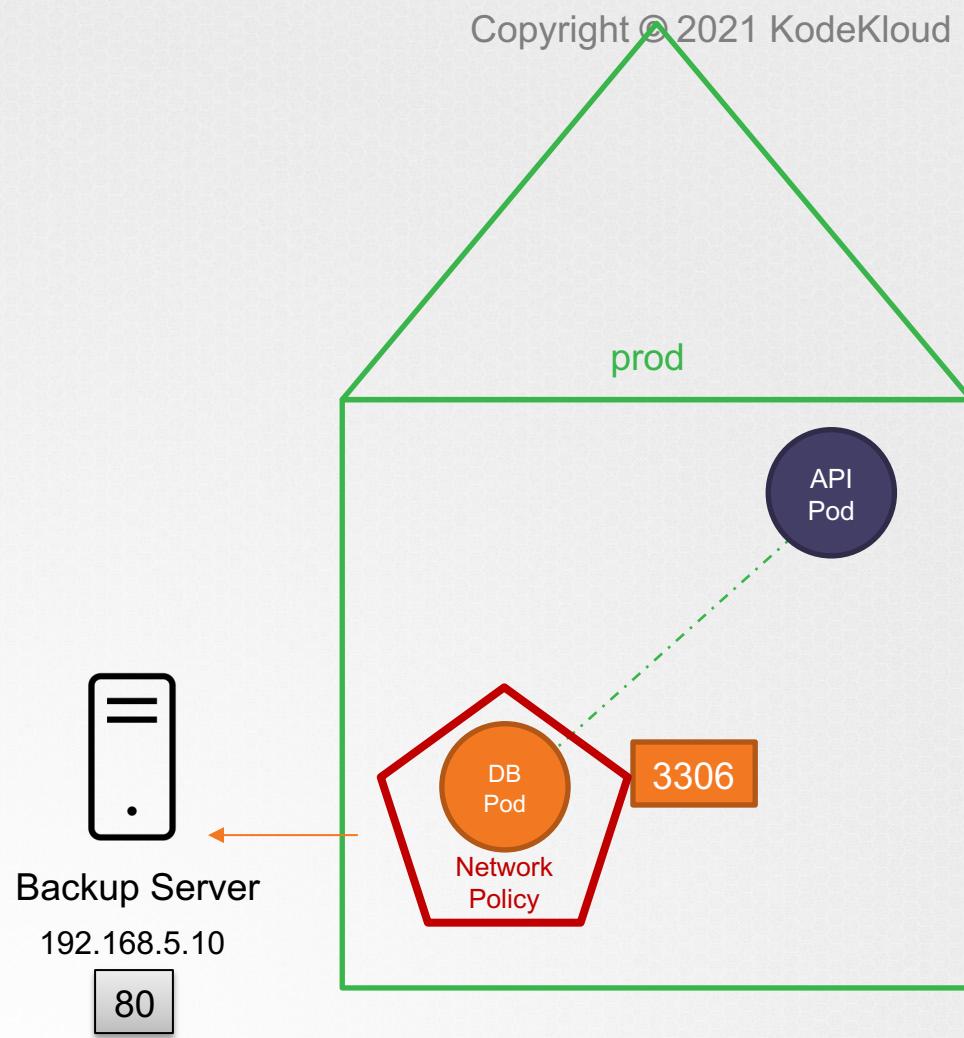
```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: db-policy
spec:
  podSelector:
    matchLabels:
      role: db
  policyTypes:
  - Ingress
  ingress:
  - from:
    - podSelector:
        matchLabels:
          name: api-pod
  namespaceSelector:
    matchLabels:
      name: prod
  ports:
  - protocol: TCP
    port: 3306
```



```
spec:  
  podSelector:  
    matchLabels:  
      role: db  
  
  policyTypes:  
  - Ingress  
  
  ingress:  
  - from:  
    - podSelector:  
        matchLabels:  
          name: api-pod  
    - namespaceSelector:  
        matchLabels:  
          name: prod  
    - ipBlock:  
        cidr: 192.168.5.10/32  
  
  ports:  
  - protocol: TCP  
    port: 3306
```



```
spec:  
  podSelector:  
    matchLabels:  
      role: db  
  
  policyTypes:  
    - Ingress  
    - Egress  
  
  ingress:  
    - from:  
      - podSelector:  
          matchLabels:  
            name: api-pod  
  
  ports:  
    - protocol: TCP  
      port: 3306  
  
  egress:  
    - to:  
      - ipBlock:  
          cidr: 192.168.5.10/32  
  
  ports:  
    - protocol: TCP  
      port: 80
```



# INGRESS





{KODE} {LOUD}

[www.kodekloud.com](http://www.kodekloud.com)

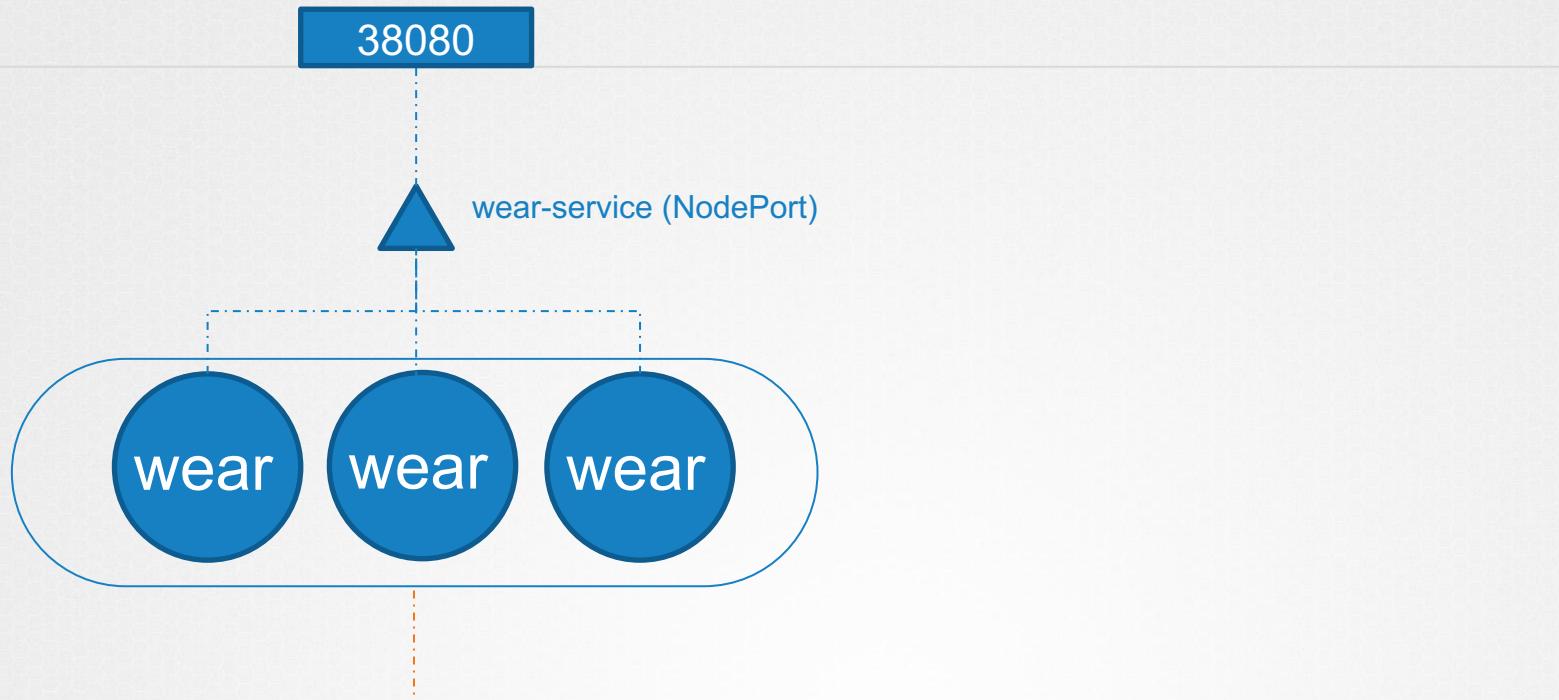
[www.my-online-store.com](http://www.my-online-store.com)







`http://my-online-store.com:38080`

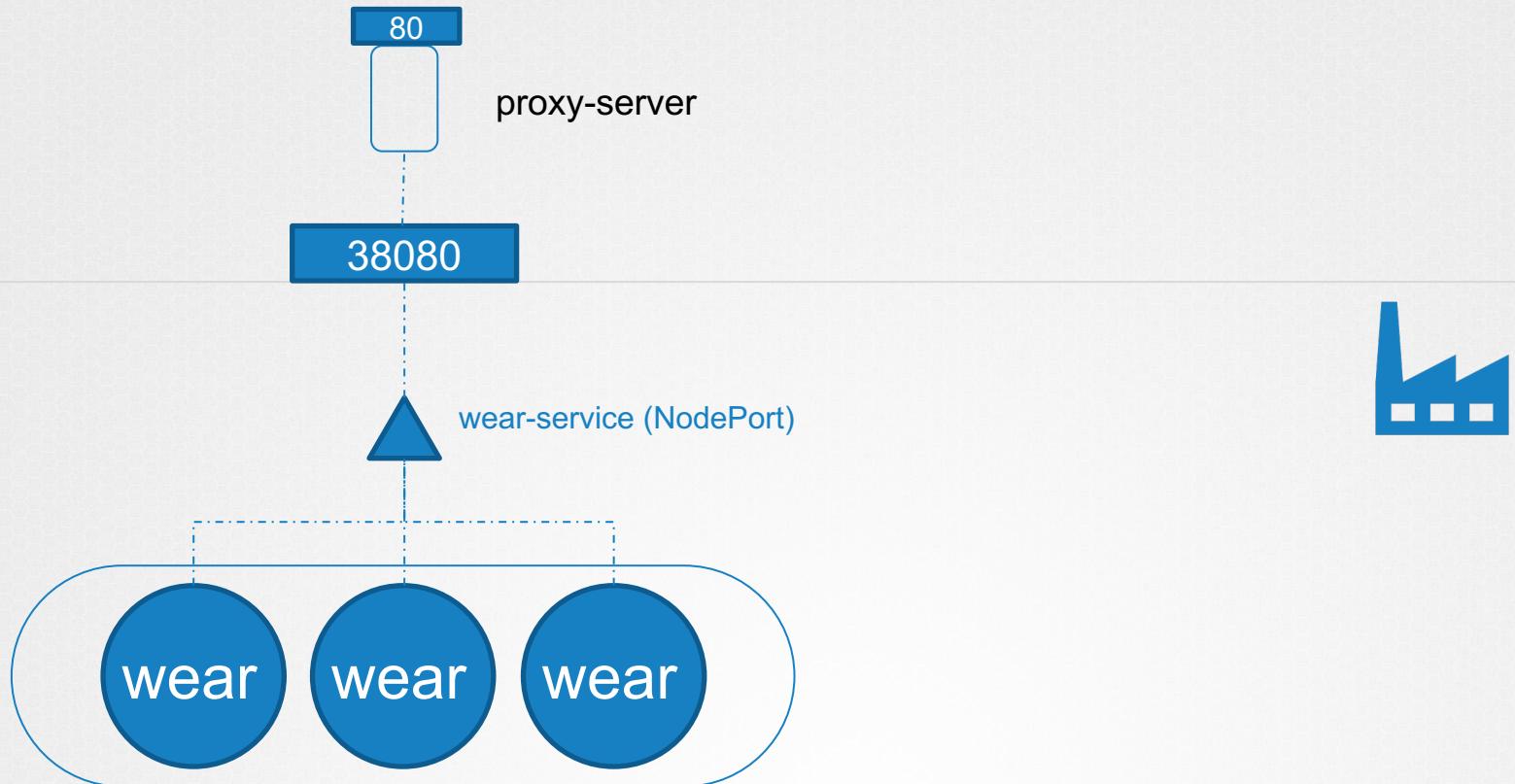


www.my-online-store.com



proxy-server

http://my-online-store.com

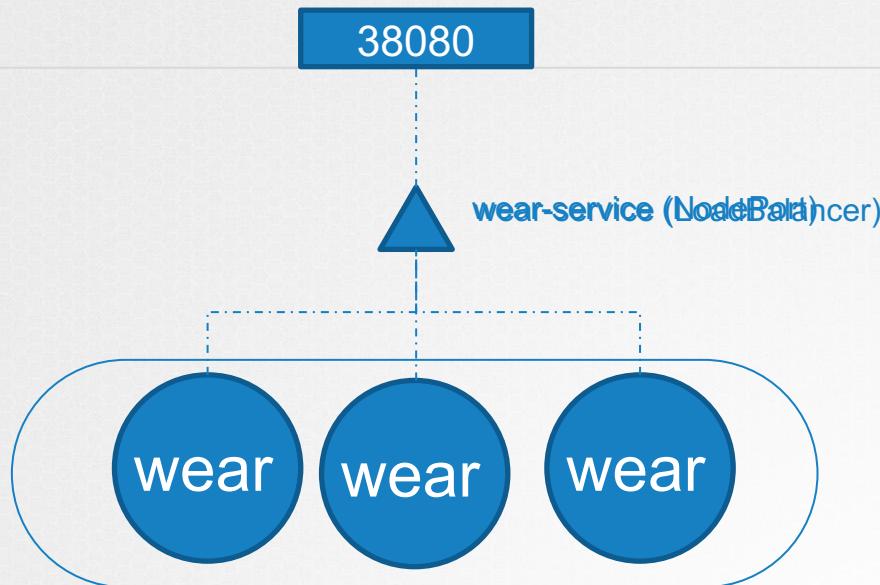


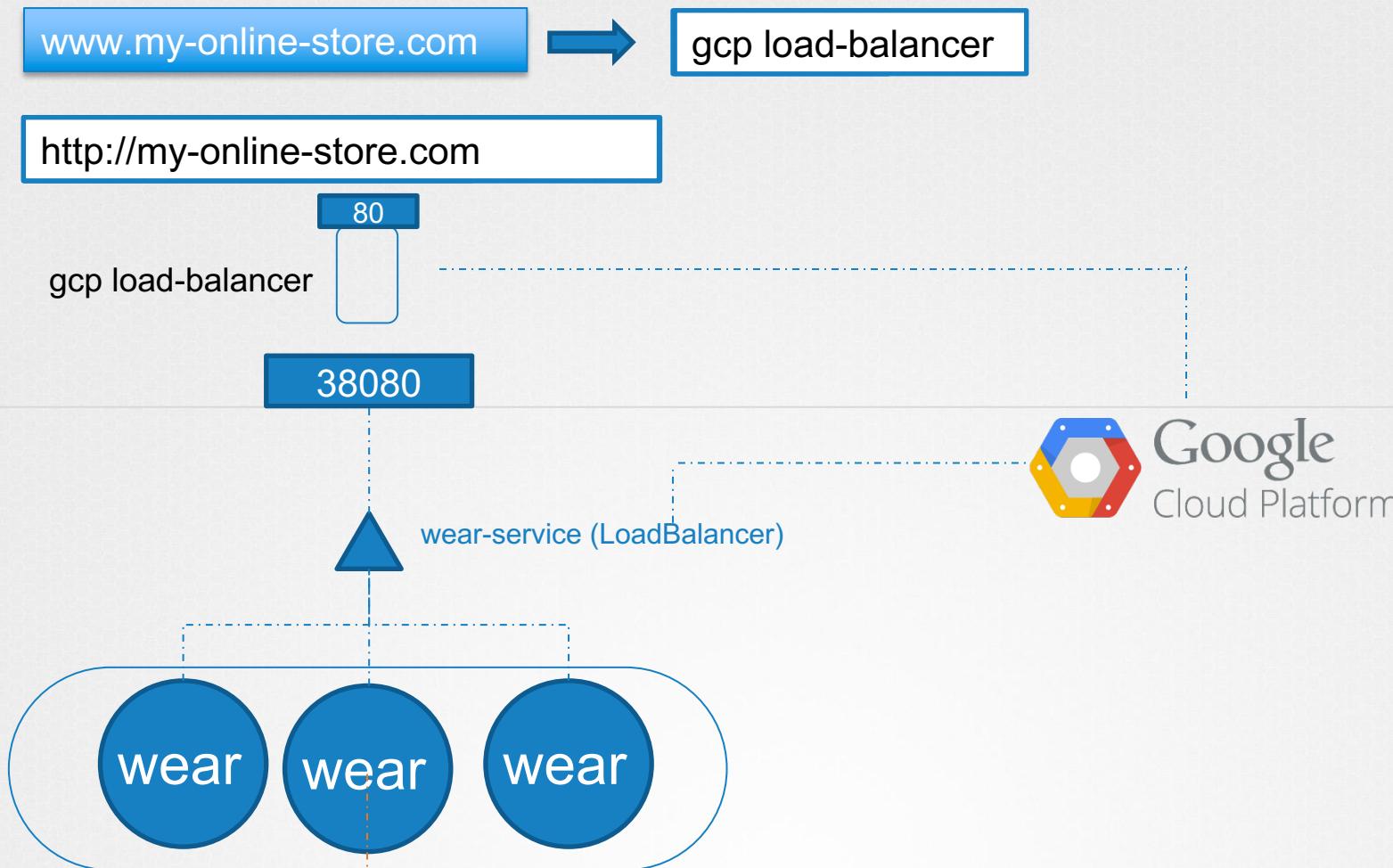
www.my-online-store.com



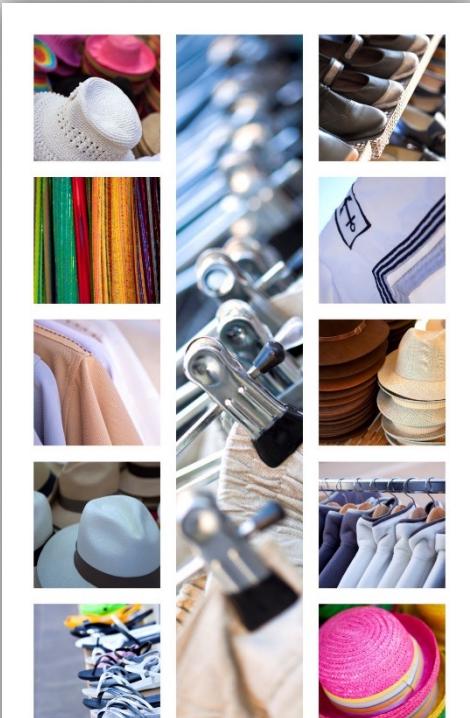
<node-ip>

http://my-online-store.com:38080



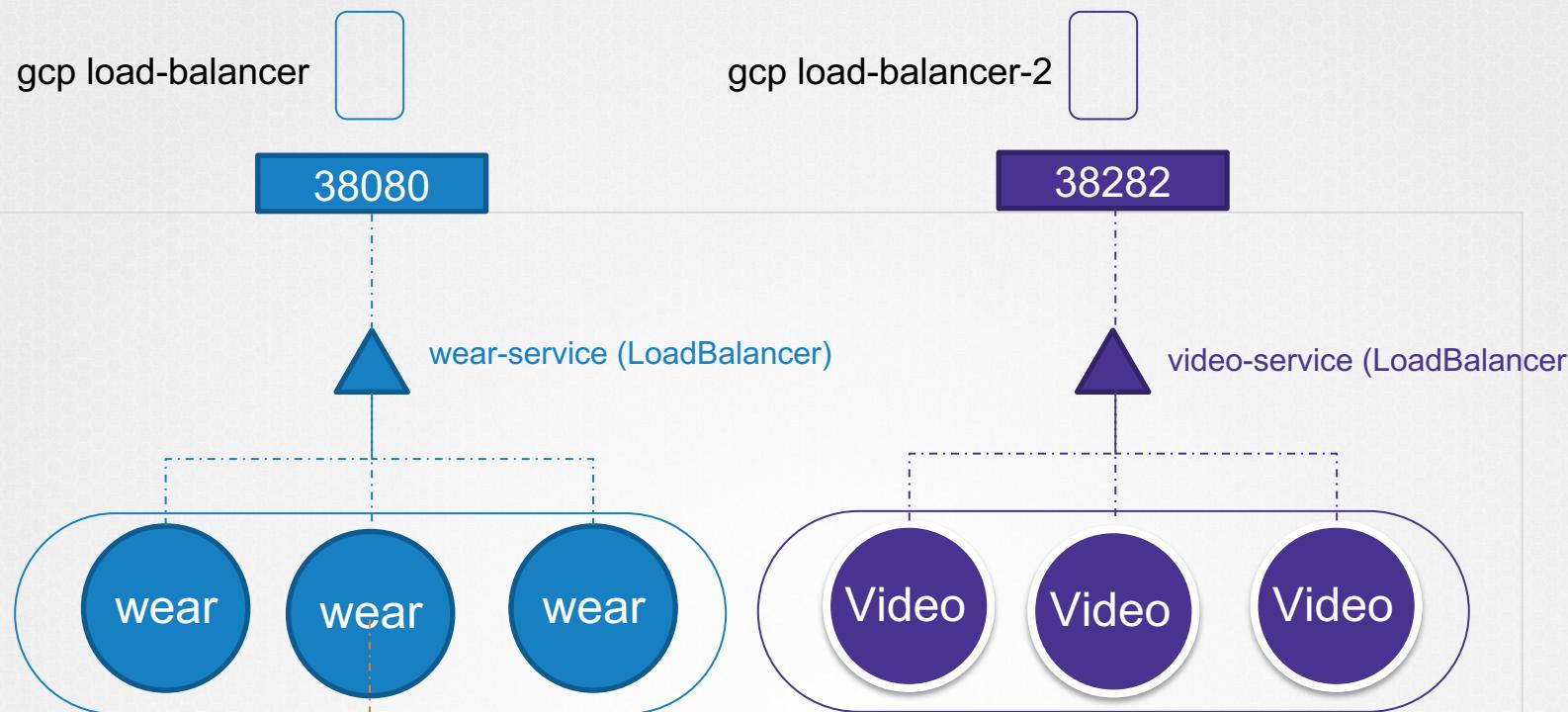


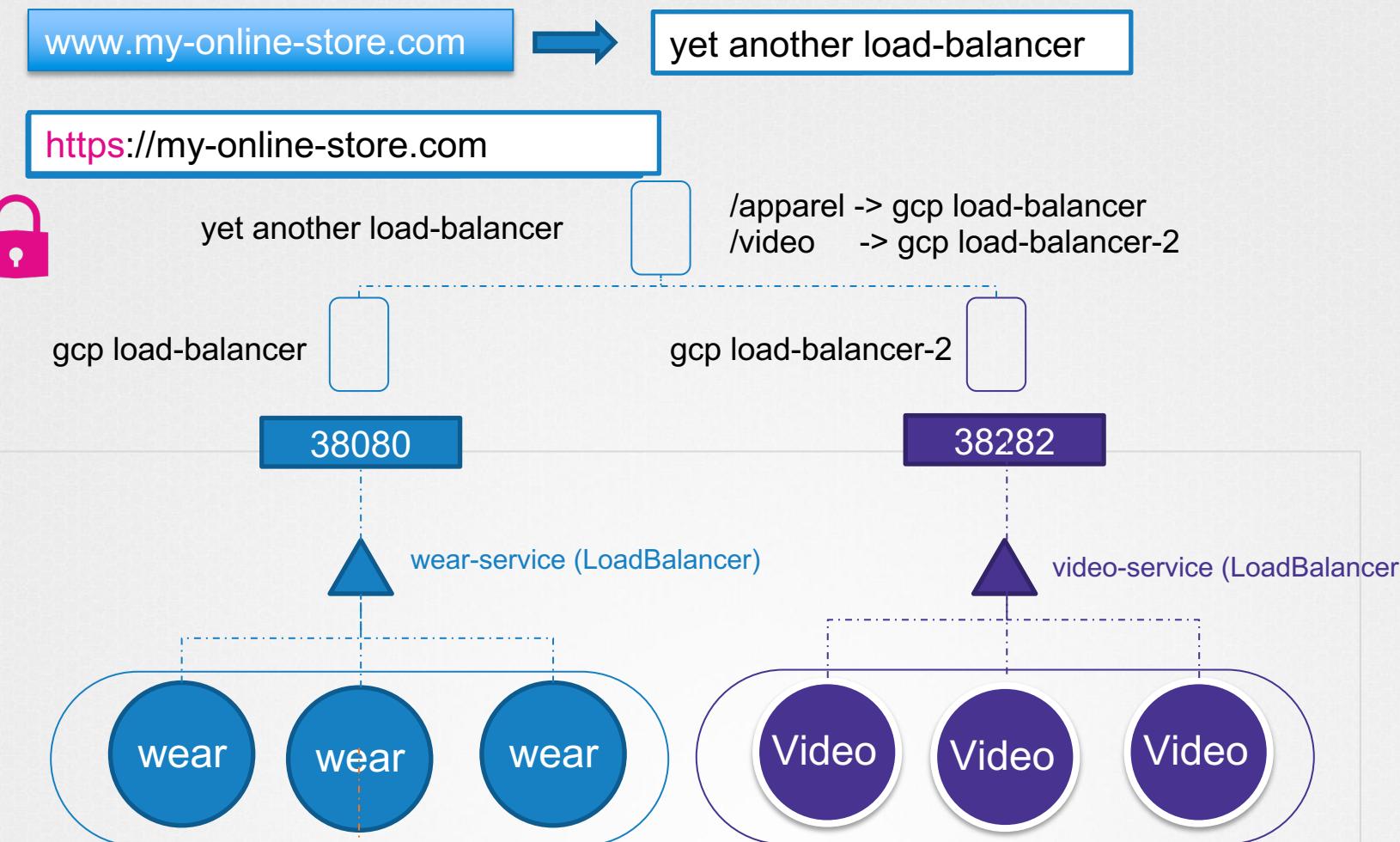
[www.my-online-store.com/wear](http://www.my-online-store.com/wear)



[www.my-online-store.com/watch](http://www.my-online-store.com/watch)







www.my-online-store.com



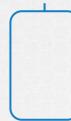
yet another load-balancer

<https://my-online-store.com>



yet another load-balancer

gcp load-balancer



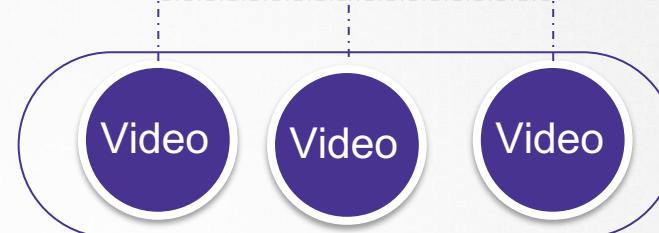
38080

/apparel -> gcp load-balancer  
/video -> gcp load-balancer-2

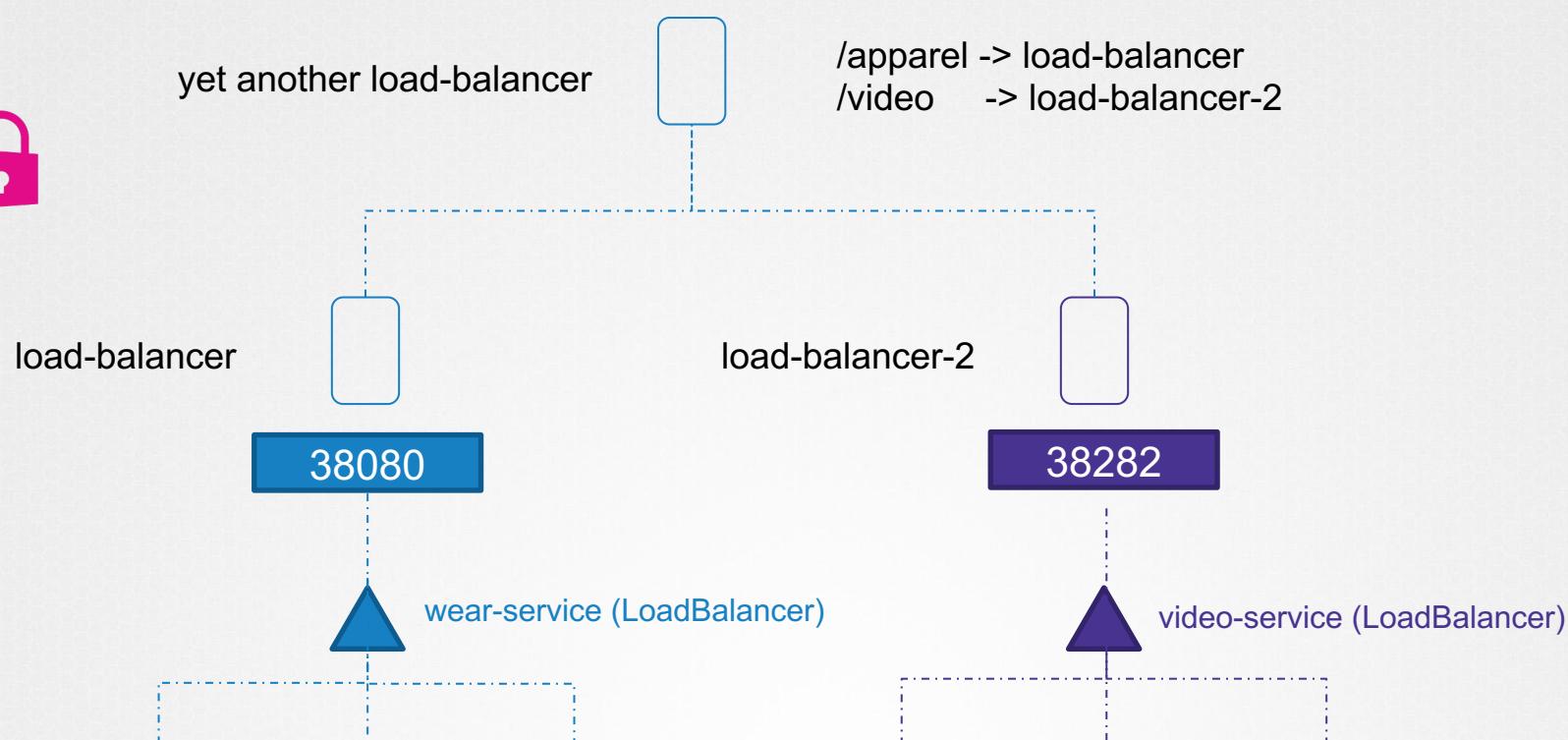
gcp load-balancer-2



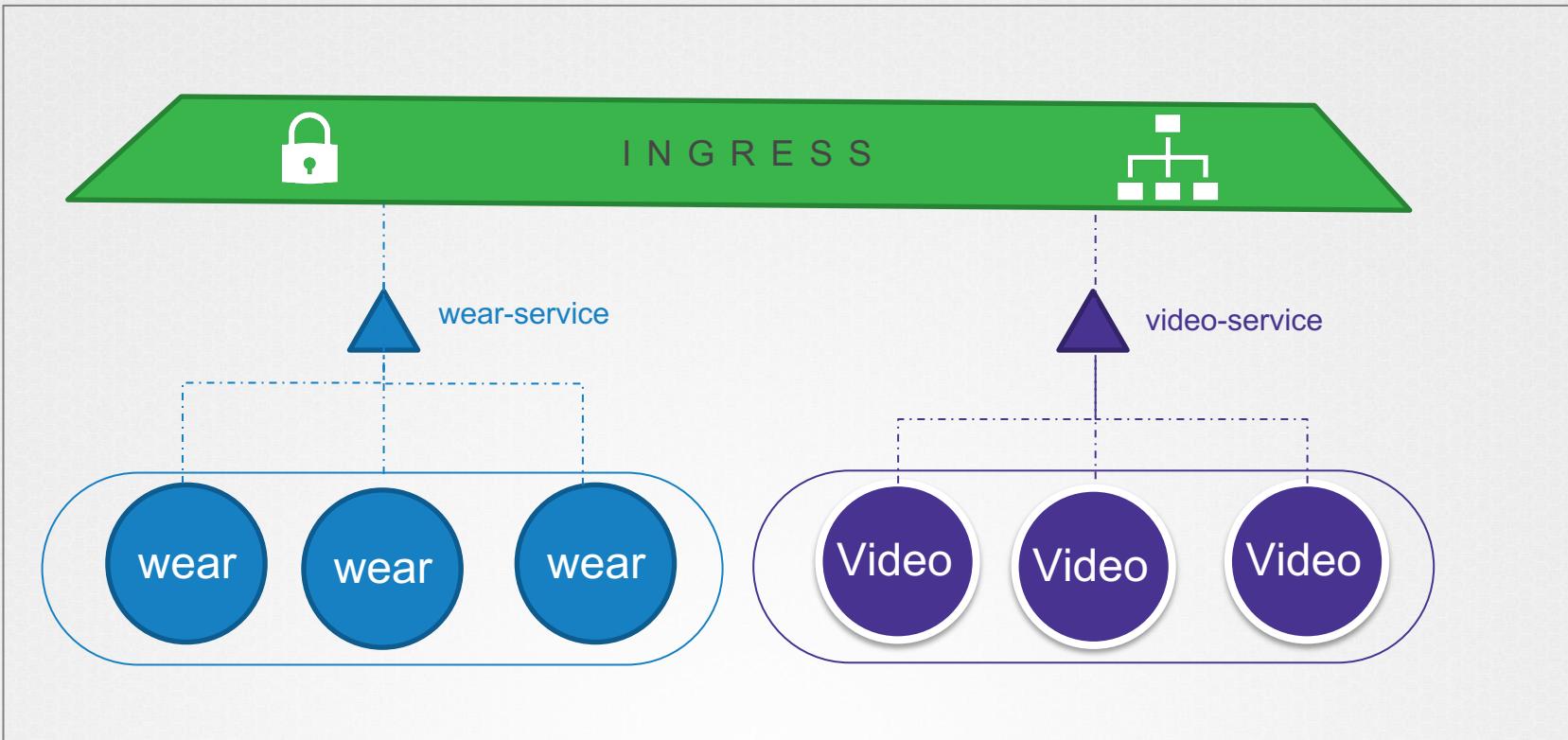
38282



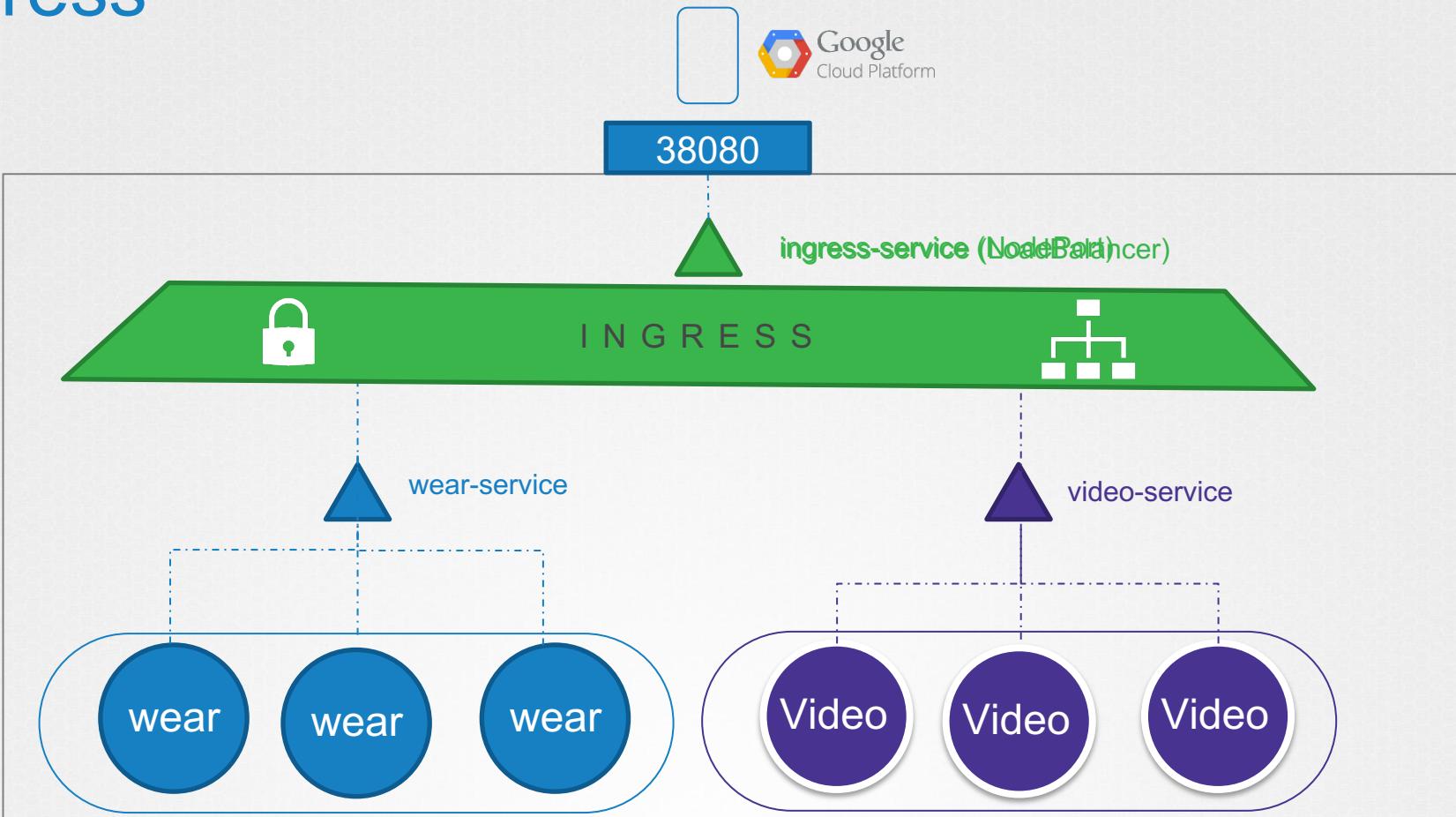
# Ingress



# Ingress



# Ingress



# Ingress

1. Deploy

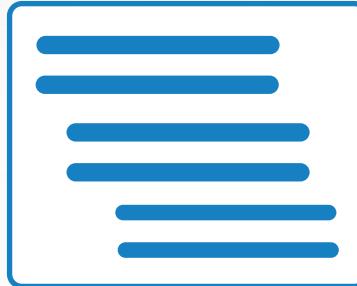


INGRESS CONTROLLER



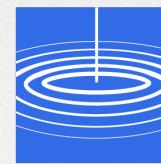
2. Configure

INGRESS RESOURCES



# INGRESS CONTROLLER

GCP HTTP(S)  
Load Balancer (GCE)



Contour



HAProxy



Istio

# INGRESS CONTROLLER



ConfigMap  
nginx-configuration

```
kind: ConfigMap
apiVersion: v1
metadata:
  name: nginx-configuration
```

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: nginx-ingress-controller
spec:
  replicas: 1
  selector:
    matchLabels:
      name: nginx-ingress
  template:
    metadata:
      labels:
        name: nginx-ingress
    spec:
      containers:
        - name: nginx-ingress-controller
          image: quay.io/kubernetes-ingress-
                controller/nginx-ingress-controller:0.21.0
      args:
        - /nginx-ingress-controller
        - --configmap=$(POD_NAMESPACE)/nginx-configuration
```

# INGRESS CONTROLLER



**ConfigMap**  
nginx-configuration

```
kind: ConfigMap
apiVersion: v1
metadata:
  name: nginx-configuration
```

```
name: nginx-ingress-controller
spec:
  replicas: 1
  selector:
    matchLabels:
      name: nginx-ingress
  template:
    metadata:
      labels:
        name: nginx-ingress
  spec:
    containers:
      - name: nginx-ingress-controller
        image: quay.io/kubernetes-ingress-
              controller/nginx-ingress-controller:0.21.0
    args:
      - /nginx-ingress-controller
      - --configmap=$(POD_NAMESPACE)/nginx-configuration
    env:
      - name: POD_NAME
        valueFrom:
          fieldRef:
            fieldPath: metadata.name
      - name: POD_NAMESPACE
        valueFrom:
          fieldRef:
            fieldPath: metadata.namespace
```

# INGRESS CONTROLLER



ConfigMap  
nginx-configuration

```
kind: ConfigMap
apiVersion: v1
metadata:
  name: nginx-configuration
```

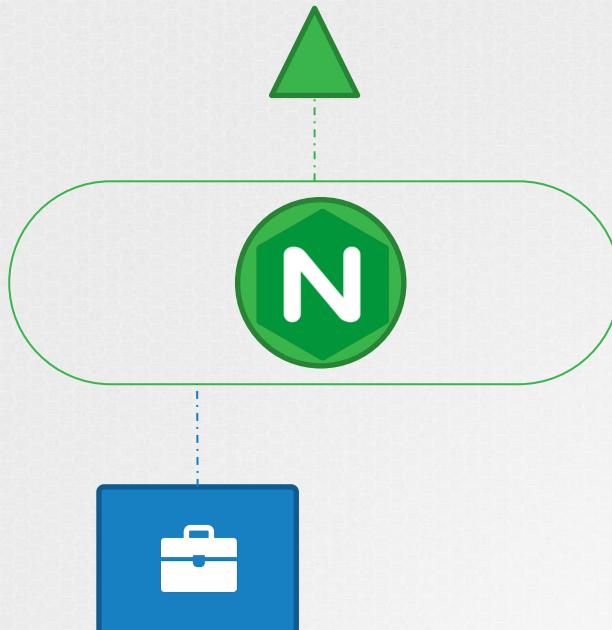
```
- name: nginx-ingress-controller
  image: quay.io/kubernetes-ingress-
        controller/nginx-ingress-controller:0.21.0

  args:
    - /nginx-ingress-controller
    - --configmap=$(POD_NAMESPACE)/nginx-configuration

  env:
    - name: POD_NAME
      valueFrom:
        fieldRef:
          fieldPath: metadata.name
    - name: POD_NAMESPACE
      valueFrom:
        fieldRef:
          fieldPath: metadata.namespace

  ports:
    - name: http
      containerPort: 80
    - name: https
      containerPort: 443
```

# INGRESS CONTROLLER



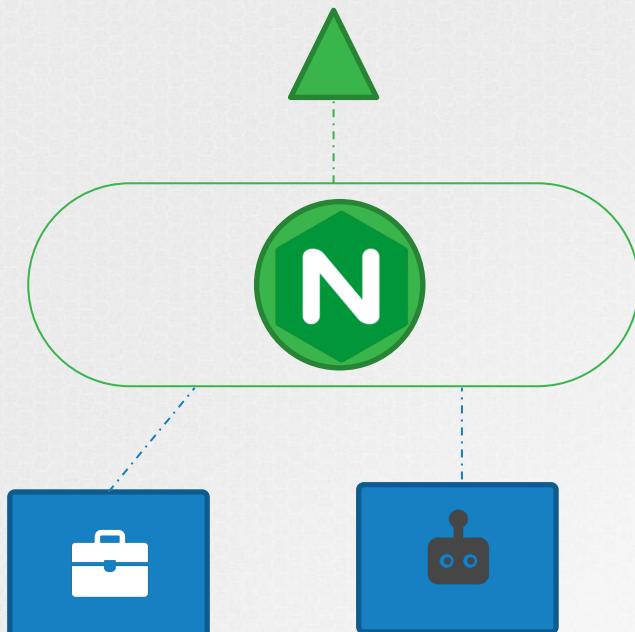
ConfigMap  
nginx-configuration

```
kind: ConfigMap
apiVersion: v1
metadata:
  name: nginx-configuration
```

```
ports:
  - name: http
    containerPort: 80
  - name: https
    containerPort: 443
```

```
apiVersion: v1
kind: Service
metadata:
  name: nginx-ingress
spec:
  type: NodePort
  ports:
    - port: 80
      targetPort: 80
      protocol: TCP
      name: http
    - port: 443
      targetPort: 443
      protocol: TCP
      name: https
  selector:
    name: nginx-ingress
```

# INGRESS CONTROLLER



```

protocol: TCP
name: http
- port: 443
  targetPort: 443
  protocol: TCP
  name: https
  selector:
    name: nginx-ingress
  
```

```

apiVersion: v1
kind: ServiceAccount
metadata:
  name: nginx-ingress-serviceaccount
  
```

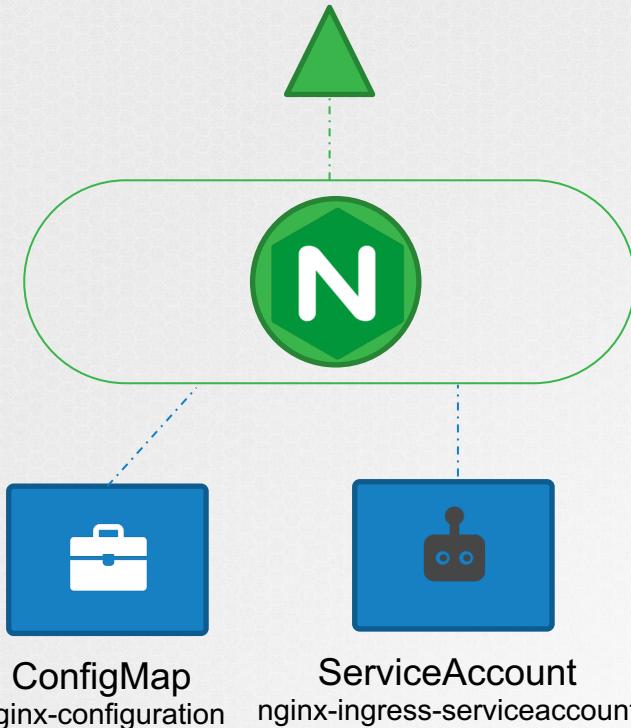
Roles

ClusterRole  
sRoleBindin  
gs

```

kind: ConfigMap
apiVersion: v1
metadata:
  name: nginx-configuration
  
```

# INGRESS CONTROLLER



**Deployment**

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: nginx-ingress-controller
spec:
  replicas: 1
  selector:
    matchLabels:
      name: nginx-ingress
  template:
    metadata:
      labels:
        name: nginx-ingress
    spec:
      containers:
        - name: nginx-ingress-controller
          image: quay.io/kubernetes-ingress-controller/nginx-ingress-
          args:
            - /nginx-ingress-controller
            - --configmap=$(POD_NAMESPACE)/nginx-configuration
      env:
        - name: POD_NAME
          valueFrom:
            fieldRef:
              fieldPath: metadata.name
        - name: POD_NAMESPACE
          valueFrom:
            fieldRef:
              fieldPath: metadata.namespace
      ports:
        - name: http
          containerPort: 80
        - name: https
          containerPort: 443
```

**Service**

```
apiVersion: v1
kind: Service
metadata:
  name: nginx-ingress
spec:
  type: NodePort
  ports:
    - port: 80
      targetPort: 80
      protocol: TCP
      name: http
    - port: 443
      targetPort: 443
      protocol: TCP
      name: https
  selector:
    name: nginx-ingress
```

**ConfigMap**

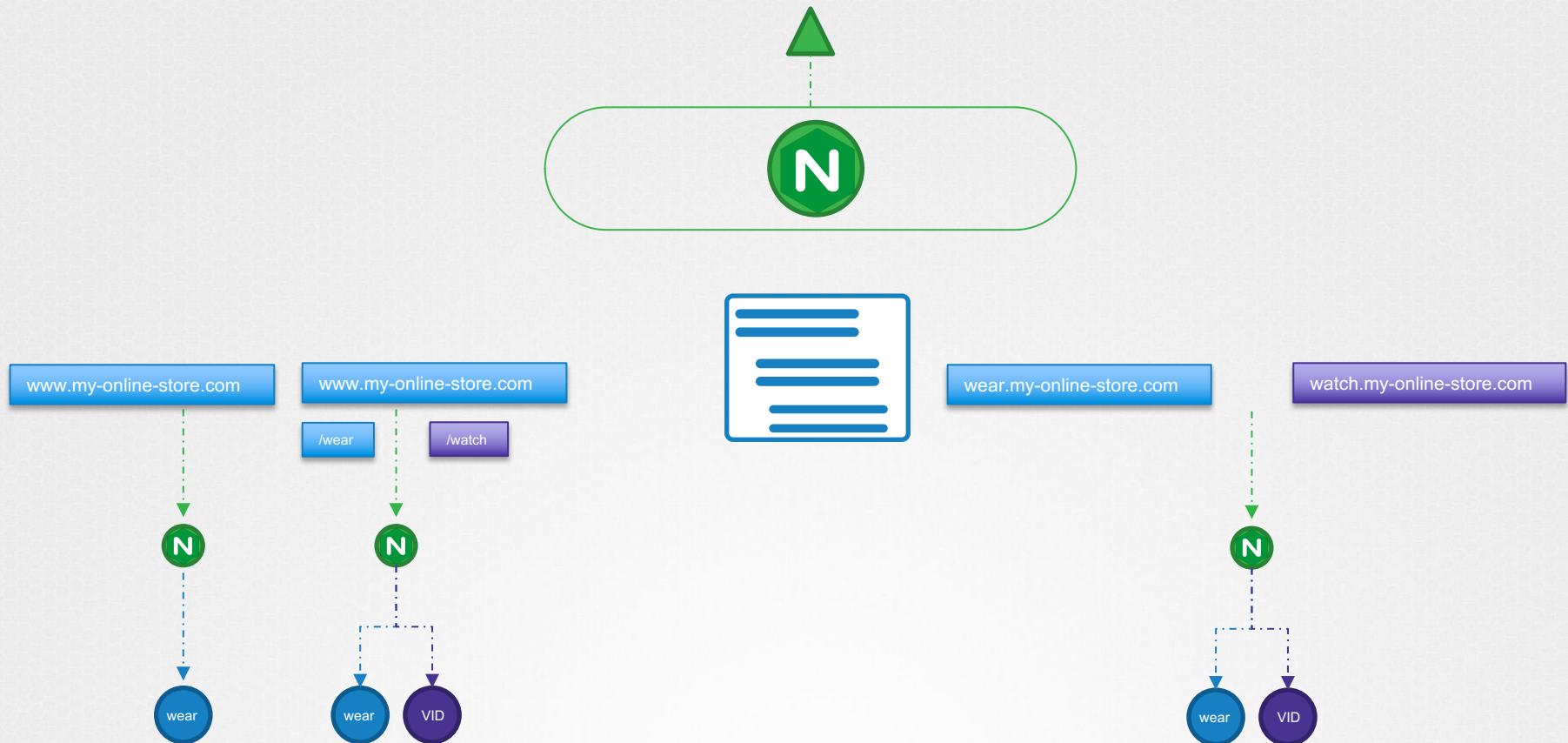
```
kind: ConfigMap
apiVersion: v1
metadata:
  name: nginx-configuration
```

**Auth**

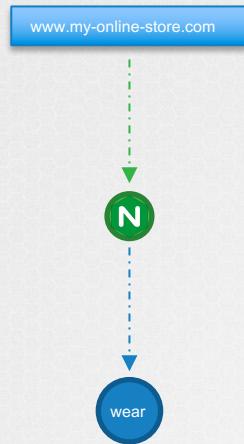
```
apiVersion: v1
kind: ServiceAccount
metadata:
  name: nginx-ingress-serviceaccount
```

Roles      ClusterRoles      RoleBindings

# INGRESS RESOURCE



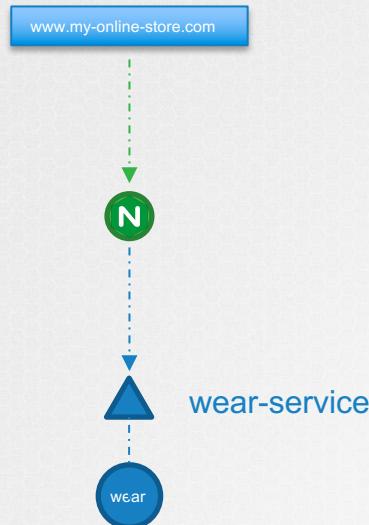
# INGRESS RESOURCE



Ingress-wear.yaml

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: ingress-wear
spec:
```

# INGRESS RESOURCE



Ingress-wear.yaml

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: ingress-wear
spec:
  backend:
    serviceName: wear-service
    servicePort: 80
```

```
▶ kubectl create -f Ingress-wear.yaml
```

```
ingress.extensions/ingress-wear created
```

```
▶ kubectl get ingress
```

NAME	HOSTS	ADDRESS	PORTS
ingress-wear	*	80	2s

# INGRESS RESOURCE - RULES

www.my-online-store.com

Rule 1

www.wear.my-online-store.com

Rule 2

www.watch.my-online-store.com

Rule 3

Everything Else

Rule 4

# INGRESS RESOURCE - RULES

DNS Name	Forward IP
<a href="#"><u>www.my-online-store.com</u></a>	10.123.23.12 (INGRESS SERVICE)
<a href="#"><u>www.wear.my-online-store.com</u></a>	10.123.23.12
<a href="#"><u>www.watch.my-online.store.com</u></a>	10.123.23.12
<a href="#"><u>www.my-wear-store.com</u></a>	10.123.23.12
<a href="#"><u>www.my-watch-store.com</u></a>	10.123.23.12

[www.my-online-store.com](#)[www.wear.my-online-store.com](#)[www.watch.my-online-store.com](#)[Everything Else](#)

Rule 1

Rule 2

Rule 3

Rule 4

# INGRESS RESOURCE - RULES

www.my-online-store.com

www.wear.my-online-store.com

www.watch.my-online-store.com

Everything Else

http://www.my-online-store.com/wear

http://www.my-online-store.com/watch

http://www.my-online-store.com/listen

Rule 1

Path /wear



Rule 2

Rule 3

Rule 4

Path /watch



Path /



# INGRESS RESOURCE - RULES

www.my-online-store.com

www.wear.my-online-store.com

www.watch.my-online-store.com

Everything Else

http://www.my-online-store.com/wear

http://www.wear.my-online-store.com/

http://www.my-online-store.com/watch

http://www.wear.my-online-store.com/returns

http://www.my-online-store.com/listen

http://www.wear.my-online-store.com/support

Rule 1

Path /wear



Path /watch



Path /



Rule 2

Path /



Path /returns



Path /support



Rule 3

Rule 4

# INGRESS RESOURCE - RULES

www.my-online-store.com

www.wear.my-online-store.com

www.watch.my-online-store.com

Everything Else

http://www.my-online-store.com/wear

http://www.wear.my-online-store.com/

http://www.watch.my-online-store.com/

http://www.my-online-store.com/watch

http://www.wear.my-online-store.com/returns

http://www.watch.my-online-store.com/movies

http://www.my-online-store.com/listen

http://www.wear.my-online-store.com/support

http://www.watch.my-online-store.com/tv

Rule 1

Path /wear



Path /watch



Path /



Rule 2

Path /



Path /returns



Path /support



Rule 3

Path /



Path /movies



Path /tv



Rule 4

# INGRESS RESOURCE - RULES

[www.my-online-store.com](http://www.my-online-store.com)[www.wear.my-online-store.com](http://www.wear.my-online-store.com)[www.watch.my-online-store.com](http://www.watch.my-online-store.com)

Everything Else

<http://www.my-online-store.com/wear><http://www.wear.my-online-store.com/><http://www.watch.my-online-store.com/><http://www.listen.my-online-store.com/><http://www.my-online-store.com/watch><http://www.wear.my-online-store.com/returns><http://www.watch.my-online-store.com/movies><http://www.eat.my-online-store.com/><http://www.my-online-store.com/listen><http://www.wear.my-online-store.com/support><http://www.watch.my-online-store.com/tv><http://www.drink.my-online-store.com/tv>

Rule 1

Path /wear



Path /watch



Path /



Rule 2

Path /



Path /returns



Path /support



Rule 3

Path /



Path /movies

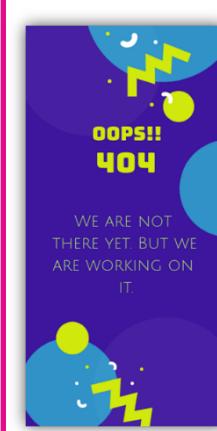


Path /tv

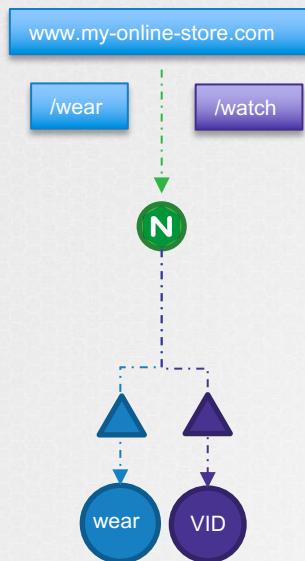


Rule 4

Path /



# INGRESS RESOURCE



## Ingress-wear.yaml

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: ingress-wear
spec:
  backend:
    serviceName: wear-service
    servicePort: 80
```

## Ingress-wear-watch.yaml

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: ingress-wear-watch
spec:
  rules:
  - http:
      paths:
      - path: /wear
      - path: /watch
    backend:
      serviceName: watch-service
      servicePort: 80
```

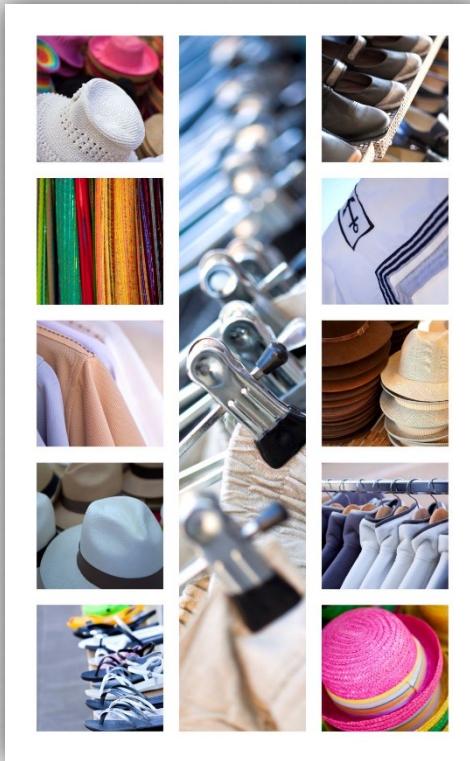
# INGRESS RESOURCE

```
▶ kubectl describe ingress ingress-wear-watch
```

```
Name:           ingress-wear-watch
Namespace:      default
Address:
Default backend: default-http-backend:80 (<none>)
Rules:
Host    Path    Backends
----  -----
*
    /wear    wear-service:80 (<none>)
    /watch   watch-service:80 (<none>)
Annotations:
Events:
Type    Reason  Age     From                      Message
----  -----
Normal  CREATE  14s    nginx-ingress-controller  Ingress  default/ingress-wear-watch
```

# INGRESS RESOURCE

[www.my-online-store.com/wear](http://www.my-online-store.com/wear)



[www.my-online-store.com/watch](http://www.my-online-store.com/watch)

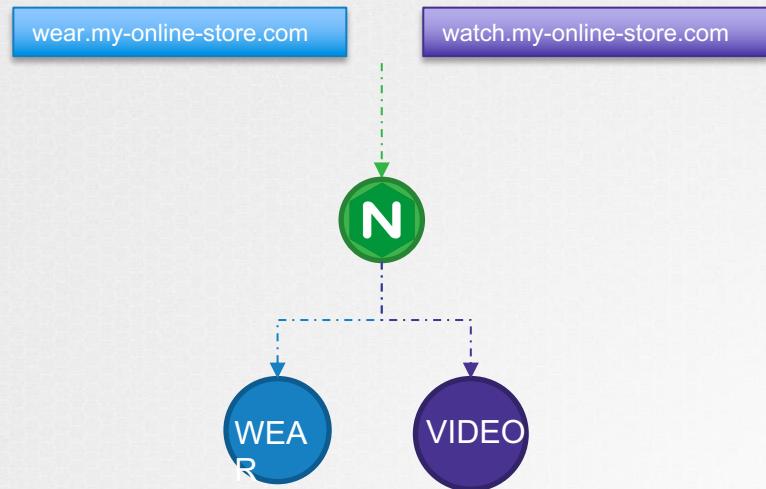


[www.my-online-store.com/eat](http://www.my-online-store.com/eat)

[www.my-online-store.com/listen](http://www.my-online-store.com/listen)



# INGRESS RESOURCE



## Ingress-wear-watch.yaml

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: ingress-wear-watch
spec:

  rules:
  - host: wear.my-online-store.com
    http:
      paths:
      - backend:
          serviceName: wear-service
          servicePort: 80
  - host: watch.my-online-store.com
    http:
      paths:
      - backend:
          serviceName: watch-service
          servicePort: 80
```

# INGRESS RESOURCE

Ingress-wear-watch.yaml

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: ingress-wear-watch
spec:
  rules:
  - http:
    paths:
    - path: /wear
      backend:
        serviceName: wear-service
        servicePort: 80
    - path: /watch
      backend:
        serviceName: watch-service
        servicePort: 80
```

Ingress-wear-watch.yaml

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: ingress-wear-watch
spec:
  rules:
  - host: wear.my-online-store.com
    http:
      paths:
      - backend:
          serviceName: wear-service
          servicePort: 80
  - host: watch.my-online-store.com
    http:
      paths:
      - backend:
          serviceName: watch-service
          servicePort: 80
```



{KODE} {LOUD}

[www.kodekloud.com](http://www.kodekloud.com)