Certified Kubernetes Administrator

Kubernetes: Security

```
#include <iostream>
#include <cstdlib>
using namespace std;
int main()
{
    cout << "Hello, world!" << endl;
    return 0;
}
```

Learning Objectives

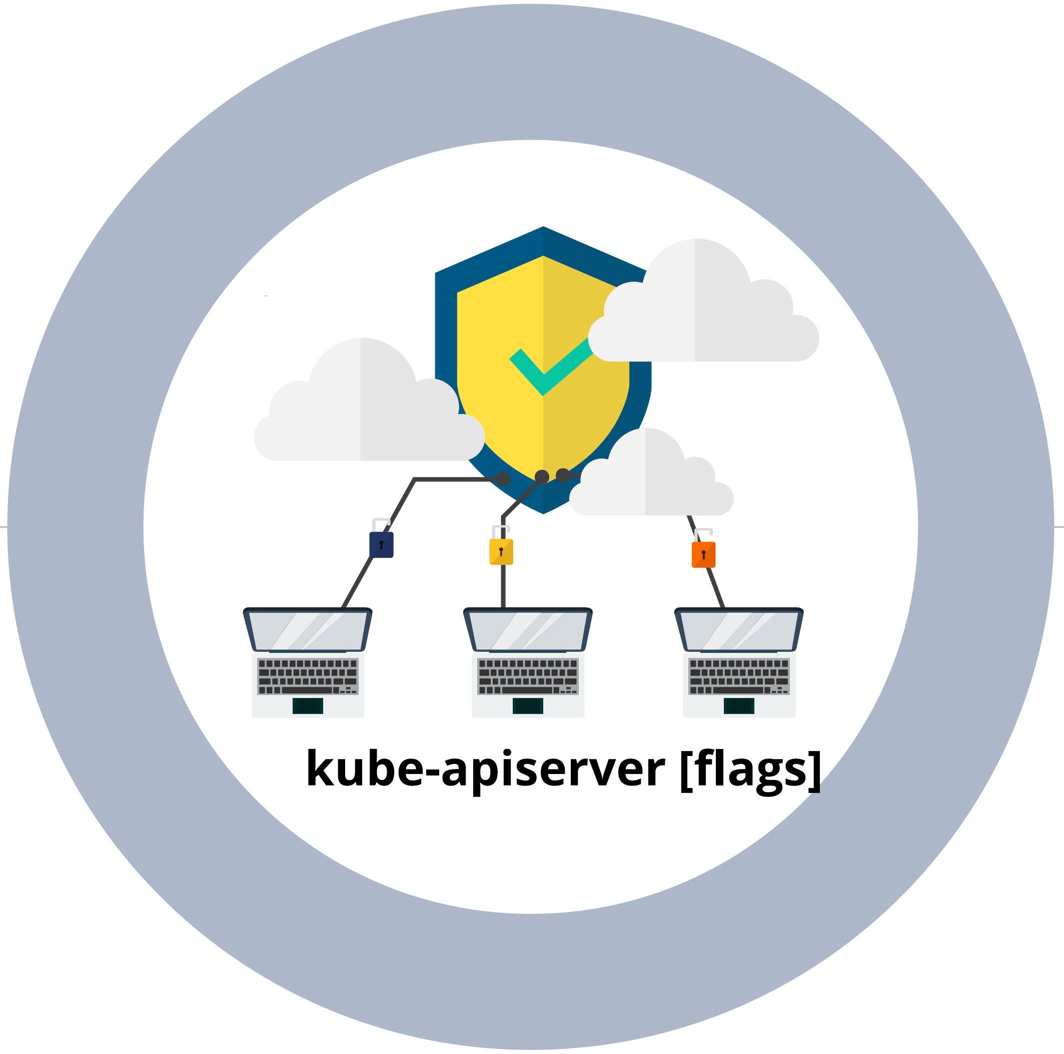
By the end of this lesson, you will be able to:

- Create an RBAC role and associate a user with that role
- Create a cluster role and associate a user with that role
- Work on secrets with private registry information
- Create a network policy
- Modify the pod settings to associate with a network policy

Kubernetes Security Primitives

Secure Kube-API Server

The Kubernetes API server validates and configures data for the API objects including pods, services, and replication controllers.



kube-apiserver [flags]

The API Server services the REST operations and provides the front end to the cluster’s shared state for all other elements to interact.

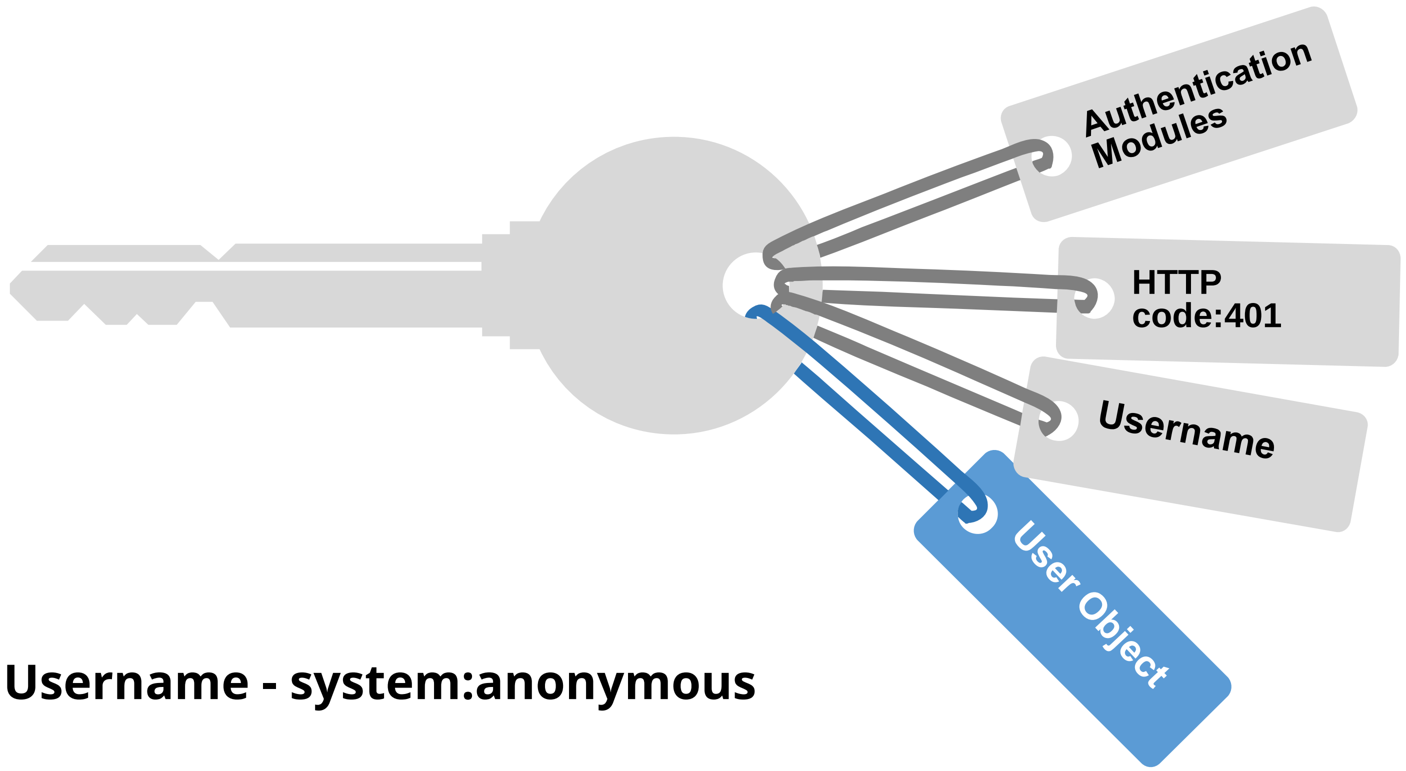
Secure Kube-API Server

Syntax: kube-apiserver [flags]

Options	Description
--admission-control-config-file string	File with admission control configuration
--allow-privileged	If true, allow privileged containers [default=false]
--audit-dynamic-configuration	It enables dynamic audit configuration. This feature also requires the DynamicAuditing feature flag
--audit-log-batch-buffer-size int Default: 10000	The size of the buffer to store events before batching and writing; it is only used in a batch mode
--audit-log-batch-max-size int Default: 1	It is the maximum size of a batch that is only used in a batch mode

Authentication

User and Password-Based Authentication

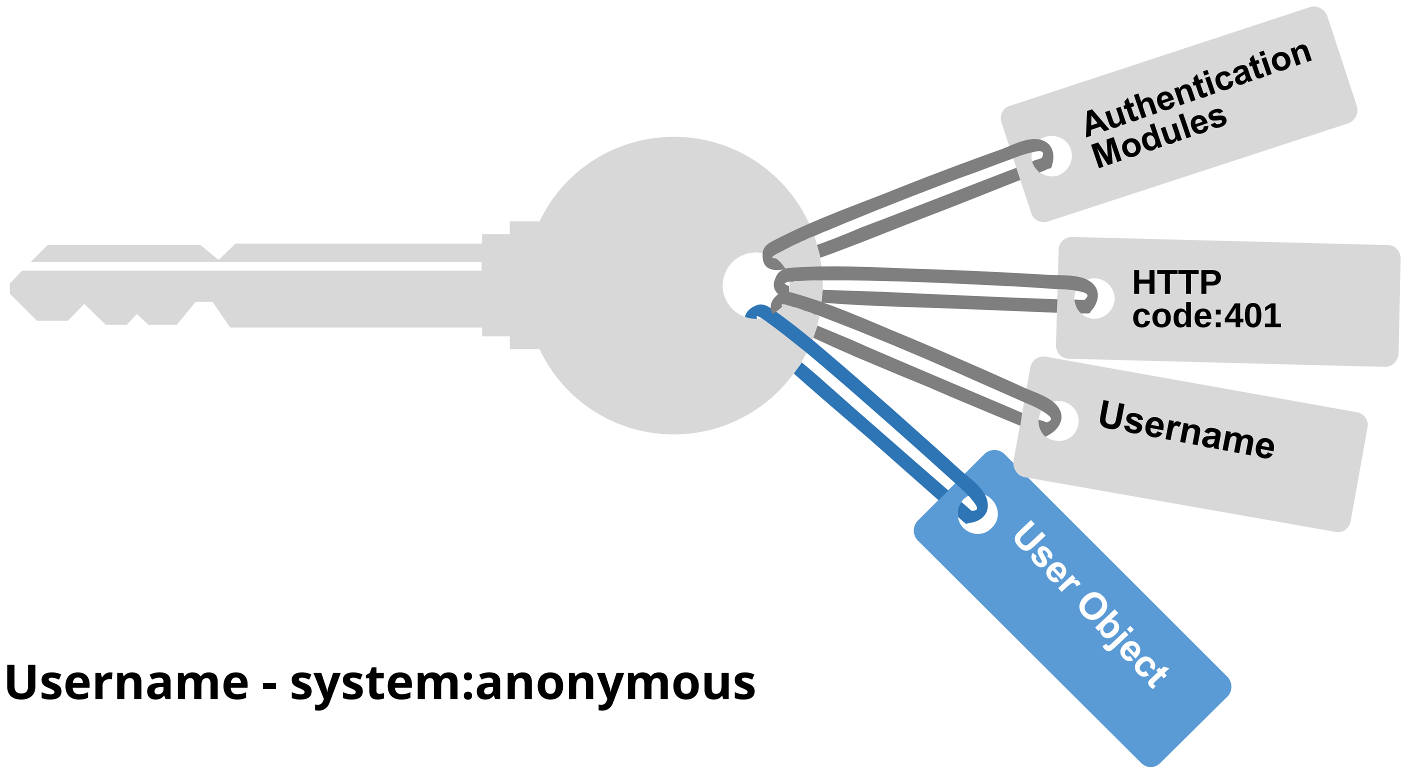


Username - system:anonymous

Authentication modules include client certificates, password, plain tokens, Bootstrap tokens, and JWT tokens (used for service accounts). Each of the authentication modules mentioned is tried in sequence until one of them succeeds.

On GCE, client certificates, password, plain tokens, and JWT tokens are all enabled. The request is rejected with HTTP status code 401 if it can not be authenticated.

User and Password-Based Authentication



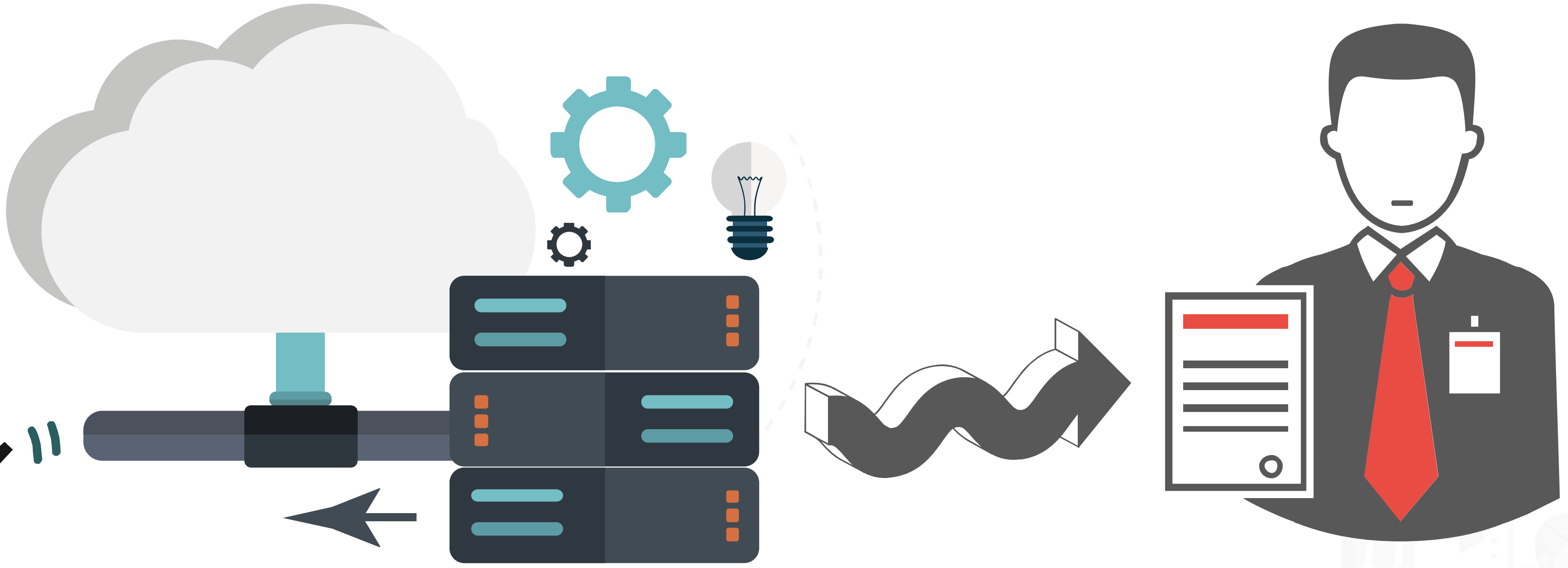
Username - system:anonymous

Otherwise, the user is authenticated as a specific username. This username is available to the subsequent steps for use in their decisions. Some authenticators also provide the group memberships of the user, while others will not.

Kubernetes neither has a user object nor does it store the usernames or any other user data in its object store. Instead, it uses usernames for access control decisions and in-request logging.

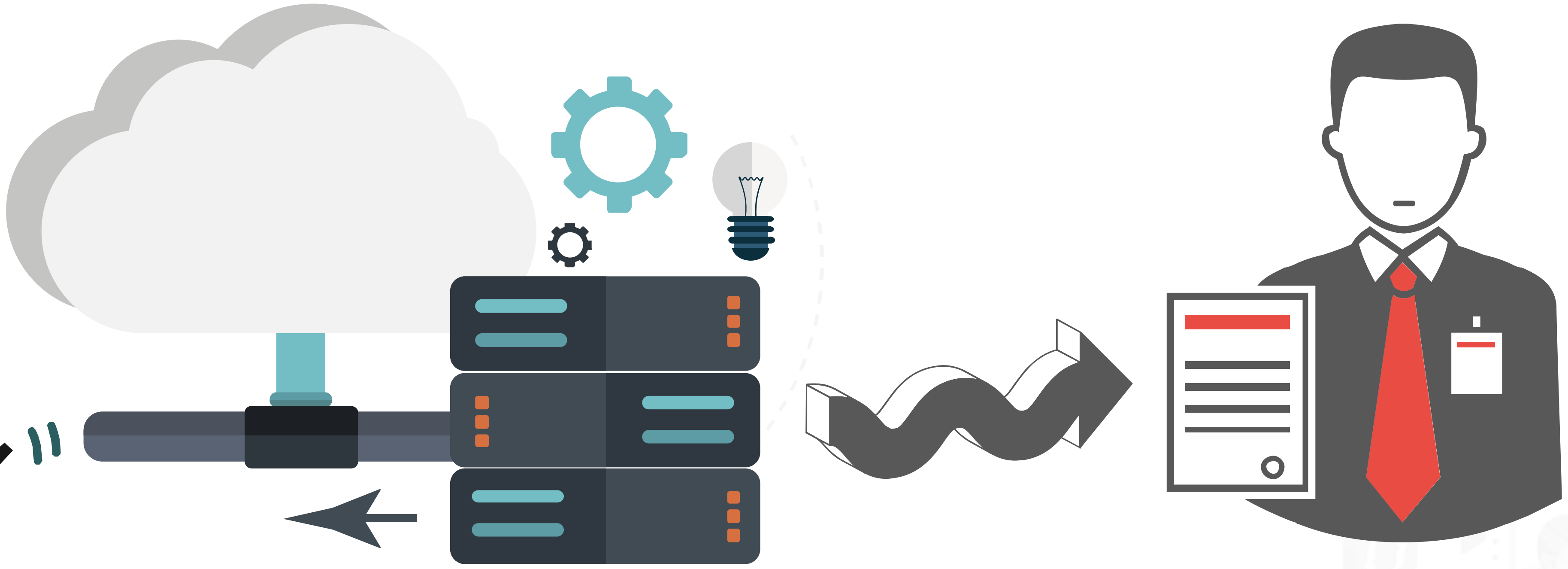
TLS

Securing the Kubernetes Cluster



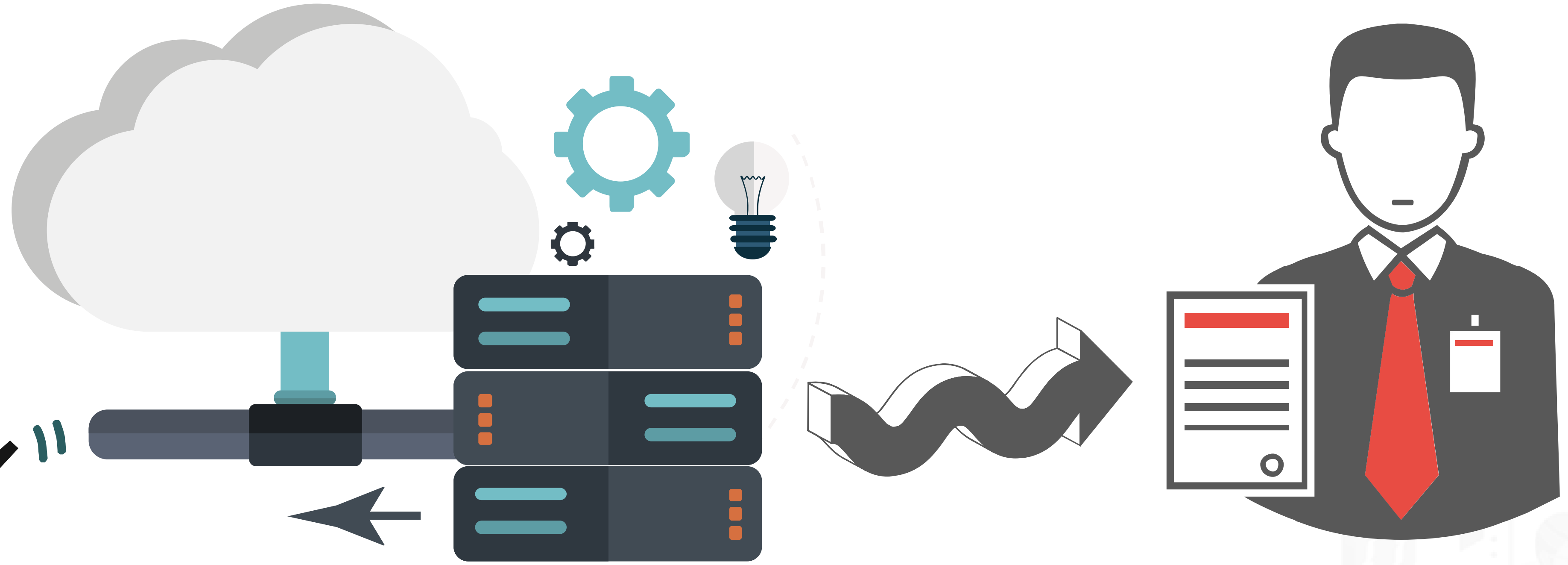
Kubernetes recommends making communication between node components such as kubelet, kube-proxy, and master components, private and secure.

Securing the Kubernetes Cluster



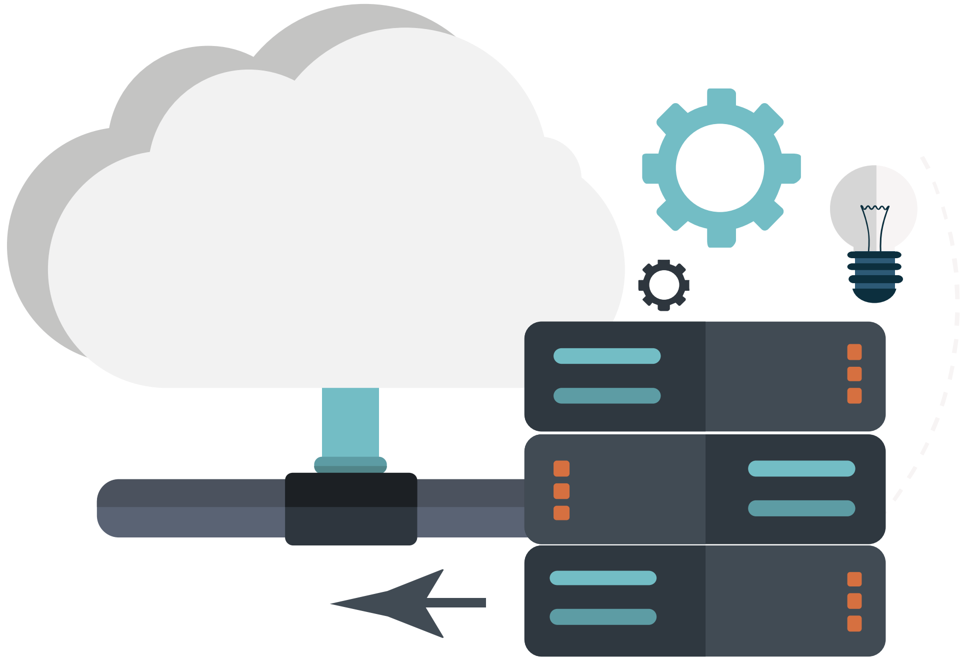
TLS certificates need to be distributed for the node components and are to be signed with the master Certificate Authority (CA). This is to achieve private and secure connection between node components and master components.

Securing the Kubernetes Cluster



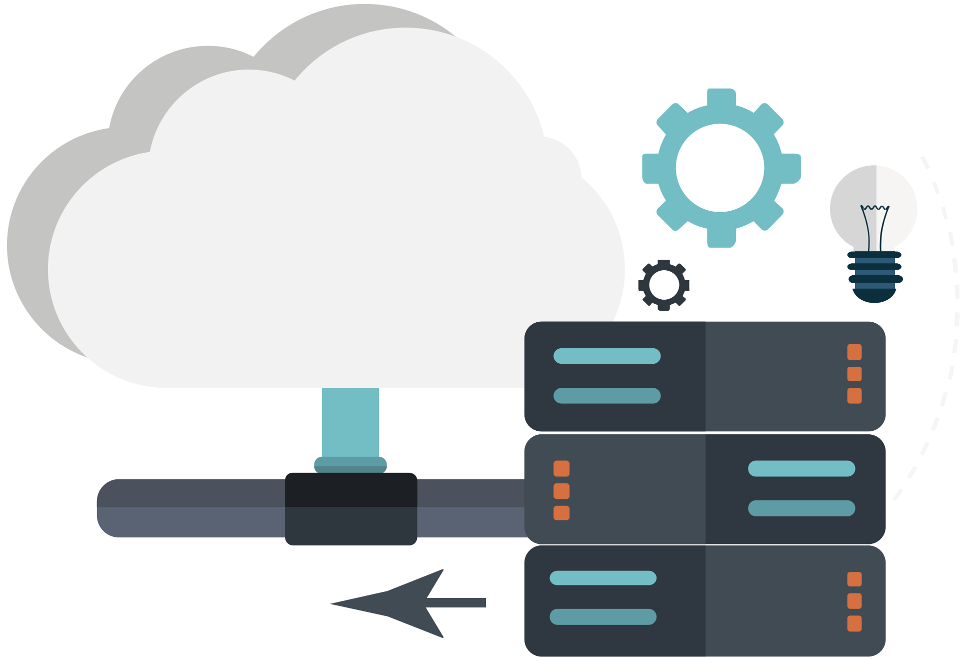
TLS bootstrapping can be described as a mechanism for the automated retrieval of certificates from the API to permit newly created node(s) to communicate with the master(s) securely.

Server Certificates



A Certificate Authority (CA) key and a certificate, without bootstrapping, are used to sign the kubelet certificate. Also, it is the user’s responsibility to distribute them to master nodes.

Server Certificates



All Kubernetes components assume the key and certificate to be PEM-encoded using certificates like kubelet, kube-apiserver, and kube-controller-manager.

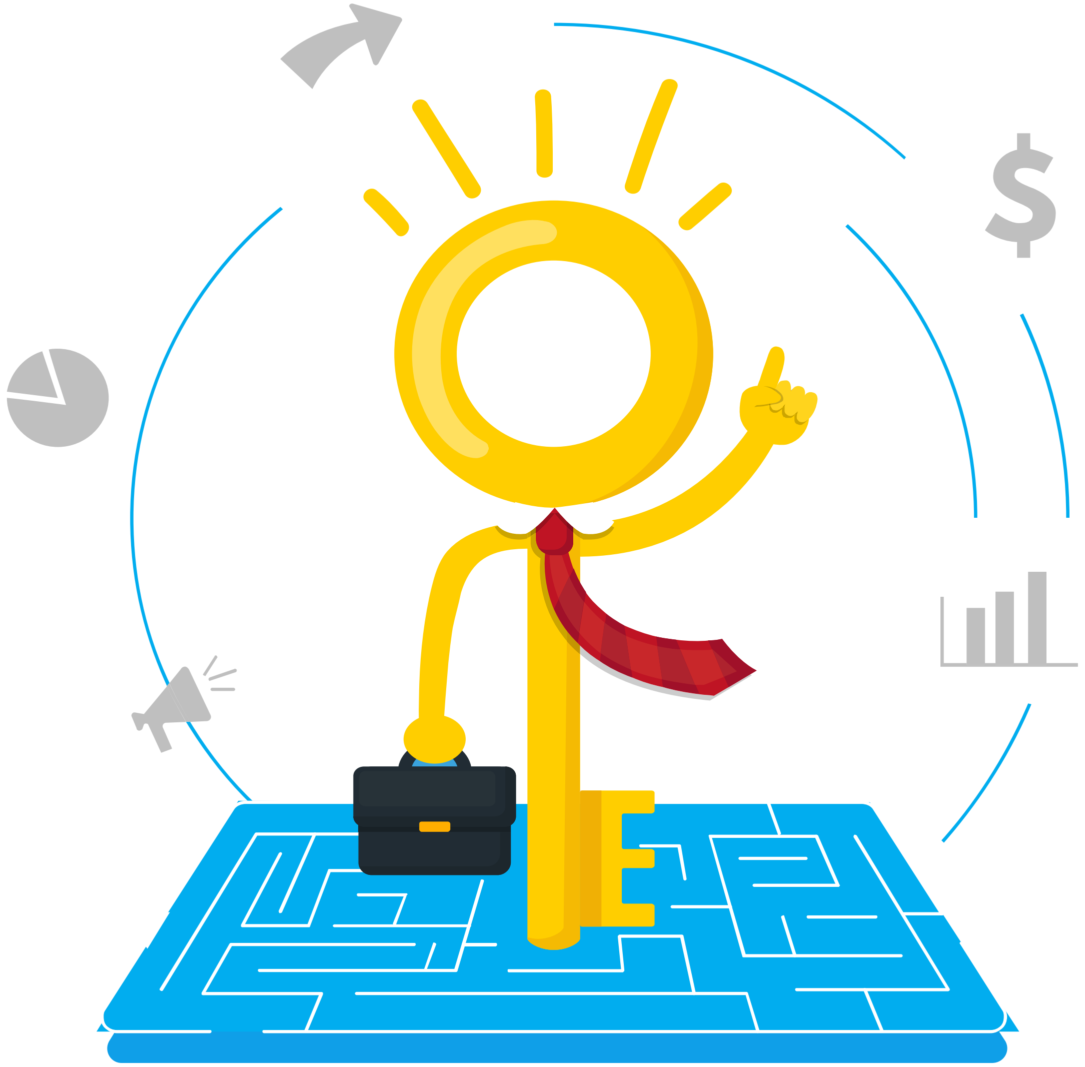
Client Certificates

A notable component is kube-proxy that controls the plane and runs on each node. It can also include other components like monitoring or networking.

Other components may have to communicate directly with the kube-apiserver.

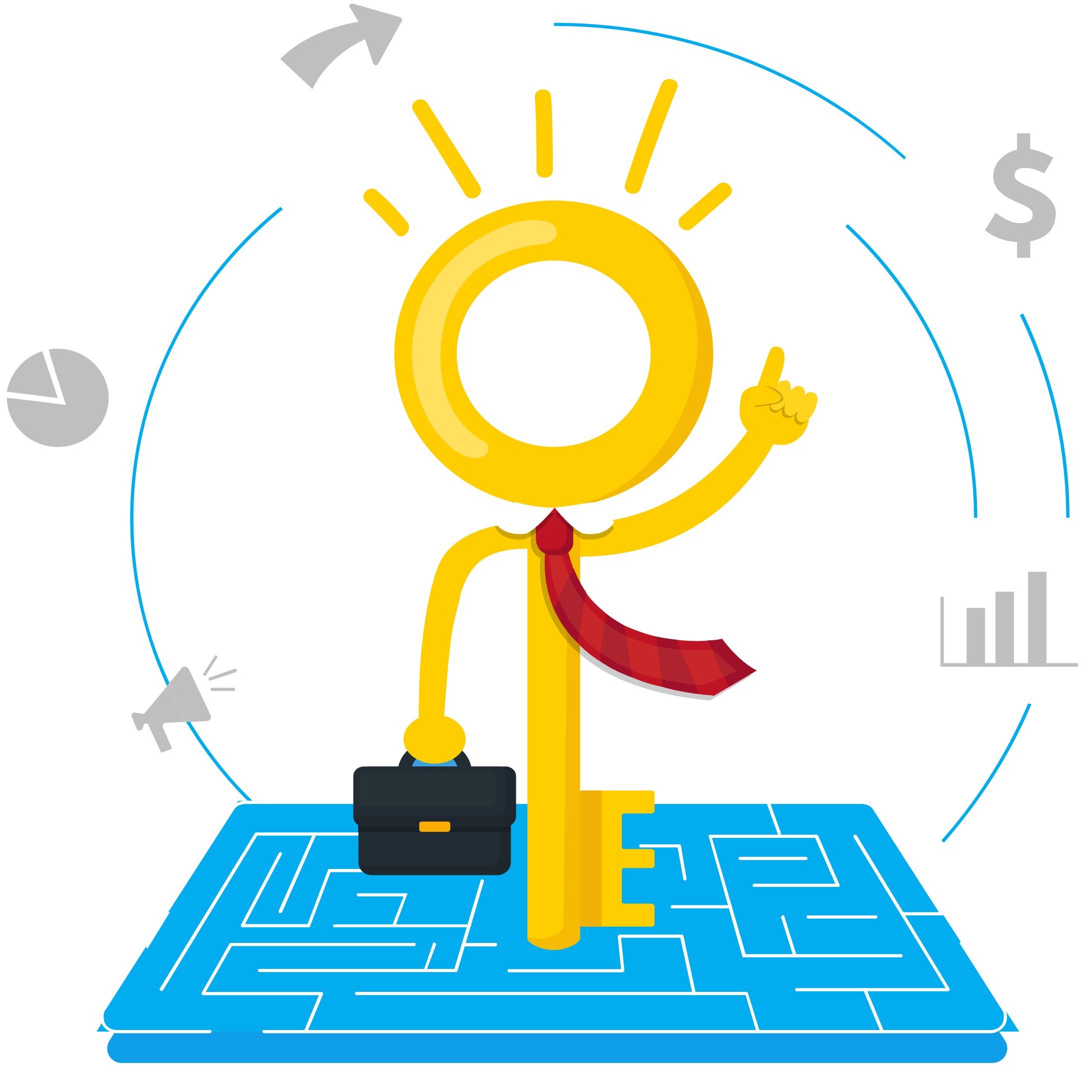
Certificates API

Certificate Authentication Server



Bootstrap tokens are defined with a unique type of secret that resides in the kube-system namespace. The Bootstrap authenticator of the API Server is used to read it.

Certificate Authentication Server



The expired tokens are removed using the TokenCleaner controller in the Controller Manager.

The tokens are also used to create a signature for a specific ConfigMap used in a ***discovery*** process using the BootstrapSigner controller.

Certificates API Process

- Use the following flag on the API server to enable the Bootstrap Token authenticator:

  --enable-bootstrap-token-auth

- To authenticate requests against the API server, bootstrapping tokens can be used as bearer token credentials when enabled

  Authorization: Bearer 07401b.f395accd246ae52d

Certificates API Process

- The username system : bootstrap: are members of the group system : bootstrappers are authenticated using the active tokens

- Expired tokens can be deleted automatically by enabling the tokencleaner controller on the controller manager

```
--controllers=*,tokencleaner
```

FULL STACK

Kubeconfig

Kubeconfig File Requirements

Imagine you have defined several clusters, your users, and components authentications in a number of ways.

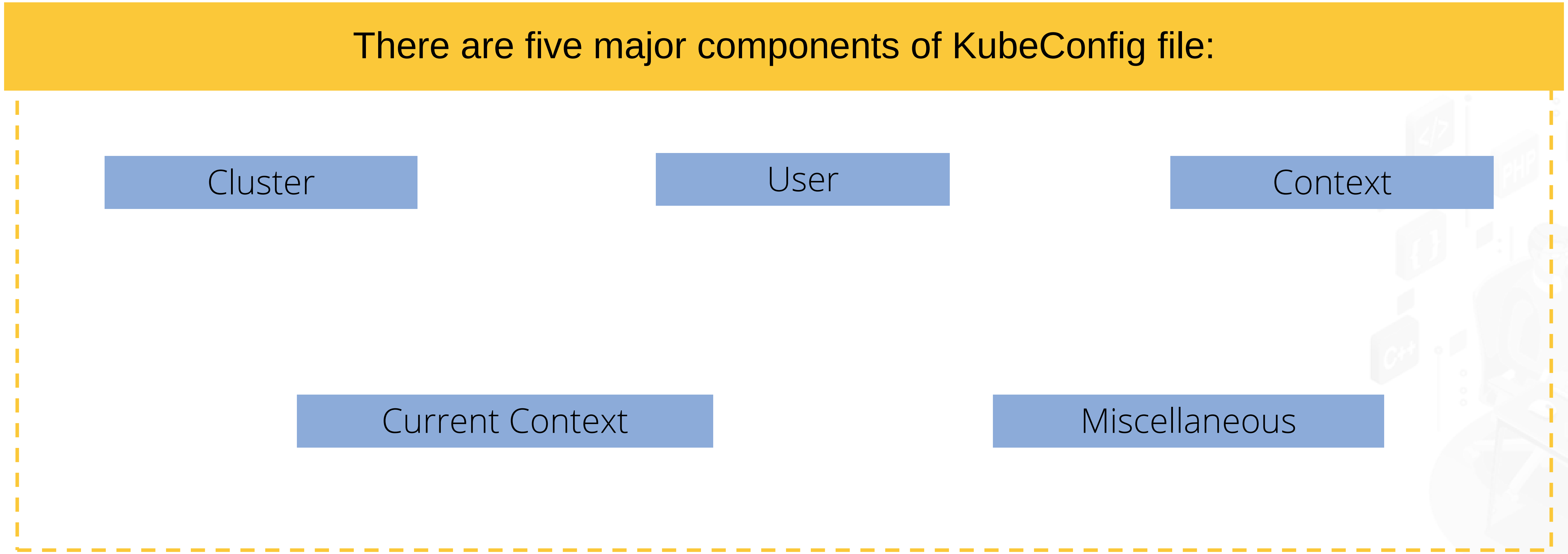
For example:

- A running kubelet might authenticate using certificates
- A user might authenticate using tokens
- Administrators may have several certificate sets that they use to provide individual users

Kubeconfig File Requirements

Using kubeconfig files, you can organize your clusters, users, and namespaces. You can additionally outline contexts to quickly and simply switch between clusters and namespaces.

Major Components of KubeConfig File



There are five major components of KubeConfig file:

Cluster

User

Context

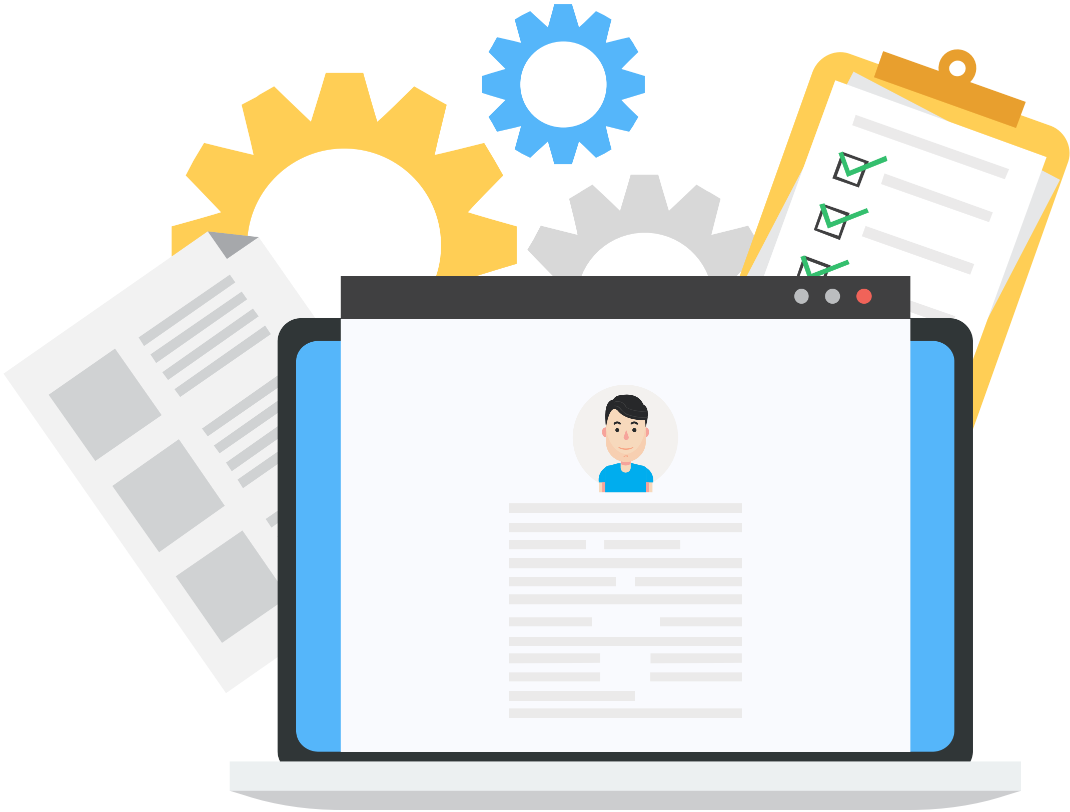
Current Context

Miscellaneous

Components of Kubeconfig File

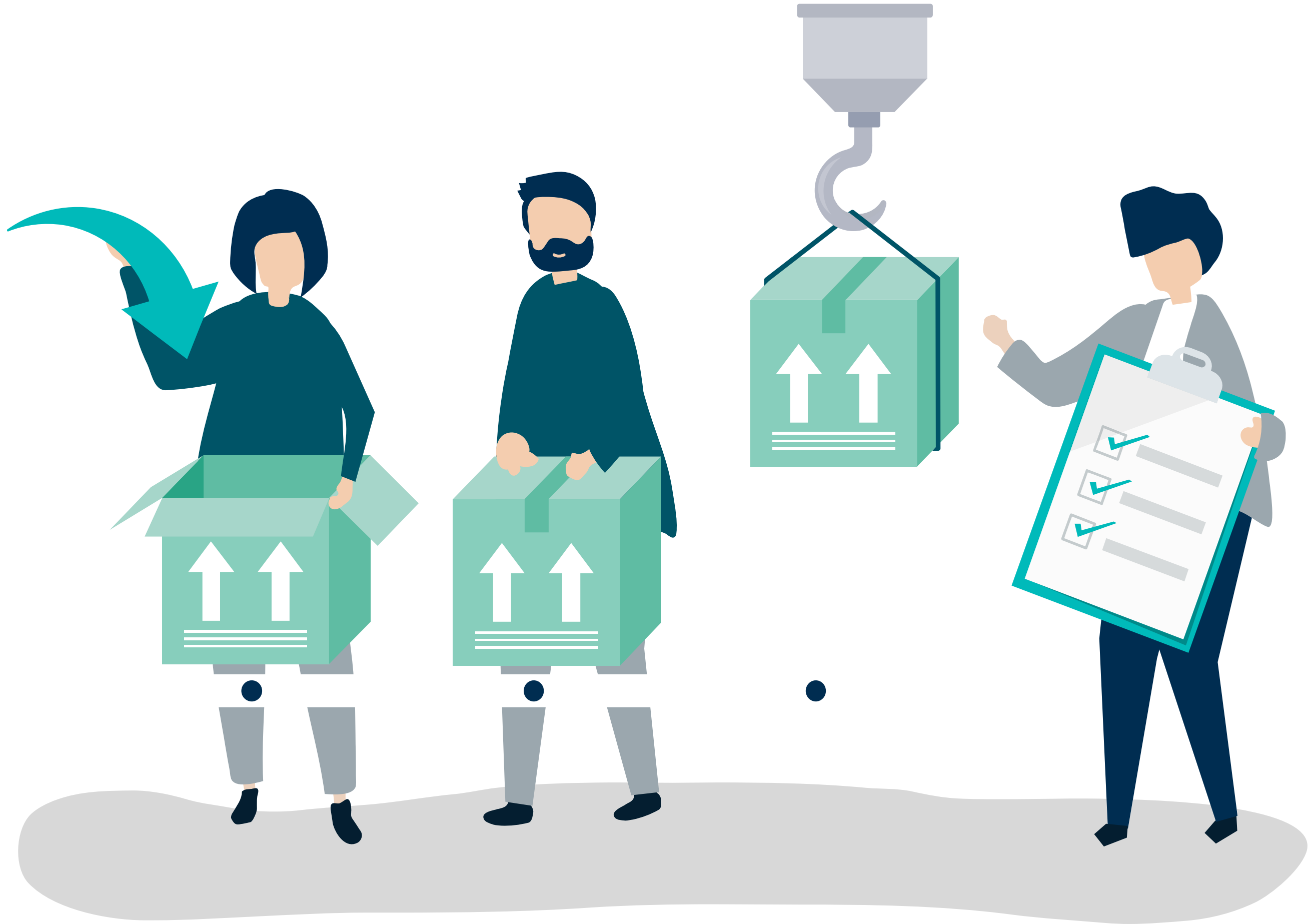
- The Kubeconfig environment variable has a list of kubeconfig files.
- For Linux and Mac, the list is colon-delimited.
- For Windows, the list is semicolon-delimited. The Kubeconfig environment variable is not required.

Components of Kubeconfig File



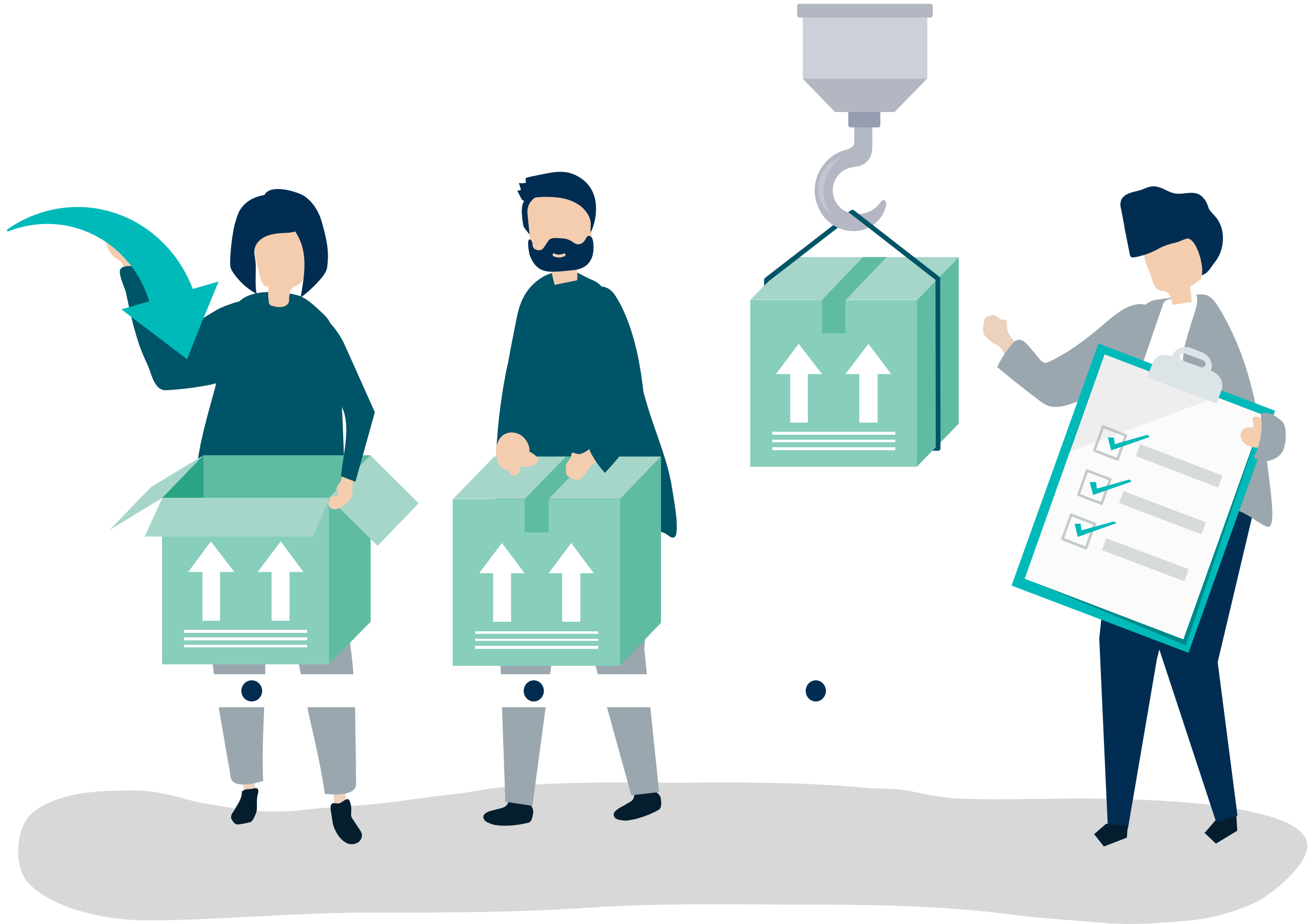
- kubectl uses the default kubeconfig file $HOME/.kube/config if the Kubeconfig environment variable does not exist.
- It then uses an effective configuration which is the result of merging files listed in the Kubeconfig environment variable.

Certificates in Kubeconfig File



To extract the certificate data from the kubeconfig file, use a kubeconfig file named config in the .kube subdirectory of your home directory.

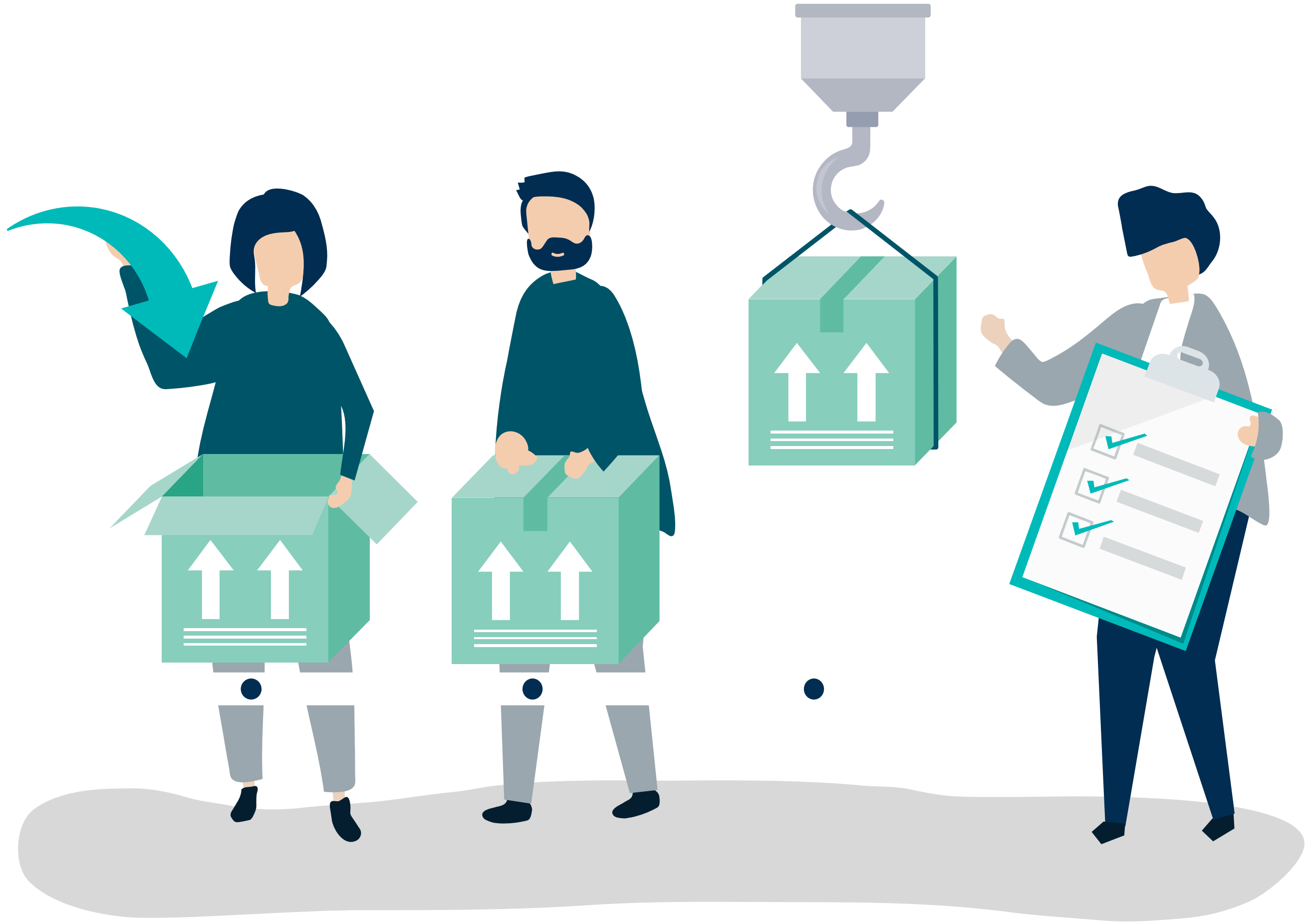
Certificates in Kubeconfig File



A simple grep statement can be used to isolate this information:

```
grep 'client-certificate-data' $HOME/.kube/config
```

Certificates in Kubeconfig File



Combine that with awk to isolate only the certificate data:
grep 'client-certificate-data' $HOME/.kube/config | awk '{print $2}'

Persistent Key-Value Store

Persistent Key-Value Store in etcd

Kubernetes uses etcd for persistent storage of all of its REST API objects. Kubernetes objects are stored under the /registry key in etcd.

Here are a few best practices for persistent storage key value pair in etcd:

- etcd should have only kube-apiserver read/write access
- etcd should run as a cluster or its data directory must be located on double storage
- Write a value on your master VM to test the working of etcd

API Groups

API Group

The main aim of an API group is to extend the Kubernetes API specified in a REST path and the apiVersion field of a serialized object.

Here are the two widely used API groups:

- Core API Group
- Named API Group

API Group

Below are the version and path components of API groups:

- The core group is also called legacy group. It uses **apiVersion: V1** and is at REST path **/api/v1**.
- The name group uses **apiVersion: $GROUP_NAME/$VERSION** and is at REST path **/apis/$GROUP_NAME/$VERSION**.

Role-Based Access Control

Role Binding

Used to grant permission to a set of users defined in a role

Consists of the subject, i.e., a user, group, or account and the reference to which the role will be granted

Used to grant permissions within a namespace and not cluster-wide; ClusterRoleBinding is used for granting permissions cluster-wide

RBAC Role Creation

Problem Statement: You are given a project to demonstrate the workflow of creating an RBAC role.

ASSISTED PRACTICE

Associate the User with the Role

Problem Statement: You are given a project to demonstrate the association of a user with the role.

ASSISTED PRACTICE

FULL STACK

Cluster Roles and Role Bindings

Cluster Role

AlikeRole and ClusterRole can also be used to grant permission. They can grant permissions for:

- Non-resource endpoints
- Clustered scooped resources like nodes
- Namespace resources

It is also used to grant read access to secrets across all namespaces or a particular namespace (if necessary).

ClusterRoleBinding

Used to grant permission at the cluster level and in all namespaces

Used to bind a role to user, groups, and service accounts. These are also called subjects

The two default ClusterRoleBinding are:

- system:authenticated group
- system:unauthenticated group

ClusterRoleBinding

Let us see an example of ClusterRoleBinding that allows users to read secrets in any namespace:

Example:

```
apiVersion: rbac.authorization.k8s.io/v1
# This cluster role binding allows anyone in the group
“SampleGroup” to read secrets in any namespace.
kind: ClusterRoleBinding
metadata:
  name: read-secrets-global
subjects:
- kind: Group
  name: SampleGroup
apiGroup: rbac.authorization.k8s.io
roleRef:
  kind: ClusterRole
  name: secret-reader
  apiGroup: rbac.authorization.k8s.io
```

Cluster Role Creation

Problem Statement: You are given a project to demonstrate the creation of a cluster role.

Associate the User with Cluster Role

Problem Statement: You are given a project to demonstrate the association of a user with cluster role.

ASSISTED PRACTICE

Image Security

Registry Requirements

Docker images are pushed to registry before referring it in a pod. Private registries require keys to read those images.

The types of private registries are:

- AWS EC2 Container Registry (ECR)
- Google Container Registry
- Oracle Cloud Infrastructure Registry (OCIR)
- Azure Container Registry (ACR)
- IBM Cloud Container Registry

Registry Requirements

The requirements of AWS EC2 Container Registry (ECR) are:

- Use of kubelet version v1.2.0 or latest
- Use of kubelet version v1.3.0 or latest, if the nodes and the registry are in different regions
- Offer ECR in your region

Secrets with Private Registry Information Stored

Problem Statement: You are given a project to demonstrate the use of the secrets with private registry information stored.

ASSISTED PRACTICE

Network Policy

Ingress and Egress Traffic

Default network policy type in which the NetworkPolicy includes a list of ingress rules

Each ingress rule is used to allow traffic that matched the **from** and **port** section

In this, the NetworkPolicy includes a list of egress rules. Each egress rule is used to allow traffic that matched the **to** and **port** section

Ingress and Egress Traffic

Here are the four selectors that can be specified in ingress **from** and egress **to** section:

- podSelector
- namespaceSelctor
- namespace and podselector
- ipBlock

Creating Network Policy

Problem Statement: You are given a task to create a network policy.

ASSISTED PRACTICE

Modifying the Pod Settings to Associate with Network Policy

Problem Statement: You are given a project to modify the pod settings to associate with the network policy.

Key Takeaways

You are now able to:

- Create an RBAC role and associate a user with that role
- Create a cluster role and associate a user with that role
- Work on secrets with private registry information
- Create a network policy
- Modify the pod settings to associate with a network policy

Knowledge Check

Knowledge Check 1	____________ is used to grant permissions at the cluster level.

a. Role

b. RoleBinding

c. Cluster Role

d. ClusterBinding

Knowledge Check

1

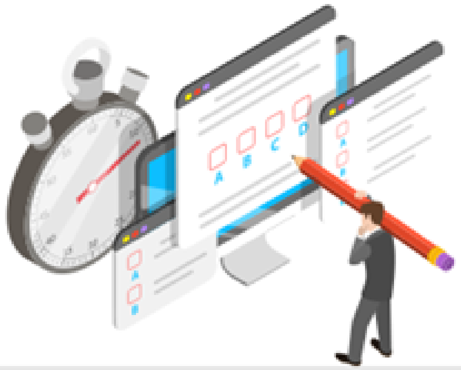
____________ is used to grant permissions at the cluster level.

a. Role

b. RoleBinding

c. Cluster Role

d. ClusterBinding



The correct answer is **d**

ClusterBinding is used to grant permissions at the cluster level.

Knowledge Check

2

Which of the following rules is used to allow traffic that matches the from and port section?

a. Ingress rule

b. Egress rule

c. podSelector

d. namespaceSelector

Knowledge Check

2

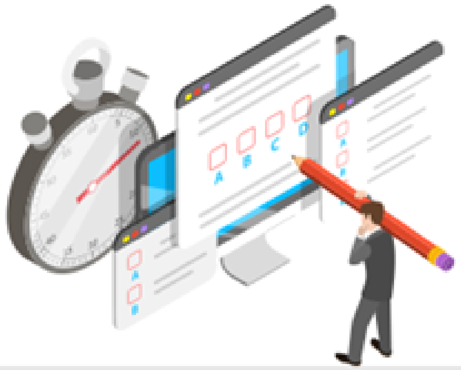
Which of the following rules is used to allow traffic that matches the from and port section?

a. Ingress rule

b. Egress rule

c. podSelector

d. namespaceSelector



The correct answer is **a**

Ingress rule is used to allow traffic that matches the from and port section.

Knowledge Check

3

Which of the following registries uses the kubelet version v1.3.0 or latest if the nodes and the registry are in different regions?

a. Google Container Registry

b. AWS EC2 Container Registry (ECR)

c. Oracle Cloud Infrastructure Registry (OCIR)

d. IBM Cloud Container Registry

Knowledge Check

3

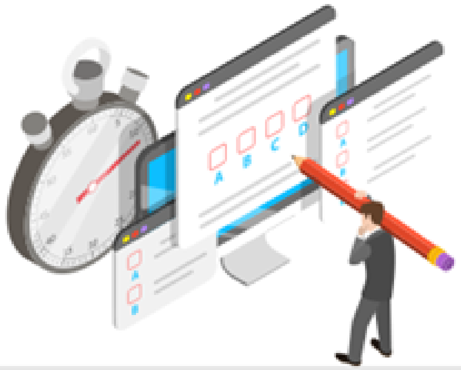
Which of the following registries uses the kubelet version v1.3.0 or latest if the nodes and the registry are in different regions?

a. Google Container Registry

b. AWS EC2 Container Registry (ECR)

c. Oracle Cloud Infrastructure Registry (OCIR)

d. IBM Cloud Container Registry



The correct answer is **b**

AWS EC2 Container Registry (ECR) uses the kubelet version v1.3.0 or latest if the nodes and the registry are in different regions.

Knowledge Check

4

Which of the following is a requirement of **AWS EC2 Container Registry (ECR)?**

a. Use of kubelet version v1.2.0 or latest

b. Use of kubelet version v1.3.0 or latest if the nodes and the registry are in different regions

c. Offering of ECR in your region

d. All of the above

Knowledge Check

4

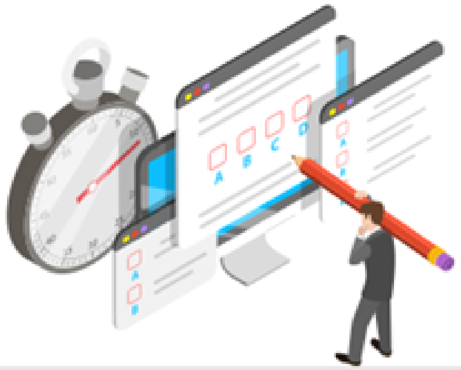
Which of the following is a requirement of AWS EC2 Container Registry (ECR)?

a. Use of kubelet version v1.2.0 or latest

b. Use of kubelet version v1.3.0 or latest if the nodes and the registry are in different regions

c. Offering of ECR in your region

d. All of the above



The correct answer is **d**

Use of kubelet version v1.2.0 or latest, use of kubelet version v1.3.0 or latest if the nodes and the registry are in different regions, and offering of ECR in your region are the requirements of AWS ECR.

Knowledge Check

5

Which of the following kube-apiserver flags specifies the maximum size of a batch and can be used only in a batch mode?

a. --admission-control-config-file string

b. --audit-dynamic-configuration

c. --audit-log-batch-buffer-size int Default: 10000

d. --audit-log-batch-max-size int Default: 1

Knowledge Check

5

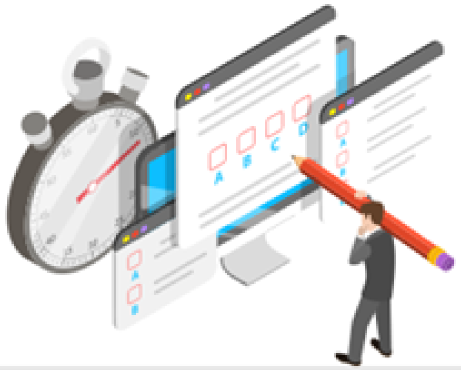
Which of the following kube-apiserver flags specifies the maximum size of a batch and can be used only in a batch mode?

a. --admission-control-config-file string

b. --audit-dynamic-configuration

c. --audit-log-batch-buffer-size int Default: 10000

d. --audit-log-batch-max-size int Default: 1



The correct answer is **d**

--audit-log-batch-max-size int Default: 1 specifies the maximum size of a batch and can be sued only in a batch mode.

Problem Statement: Corporate banks have a lot of sensitive information pertaining to millions of people. With online banking, one can get rid of many challenges of traditional banking. But, how can you ensure that data transfer and visibility are only limited to authorized users?

Objective: Authenticate the scenario using ID and password in online banking sector.

LESSON-END PROJECT