AWS-DEVOPS

DATE	:	18.04.2023
DT/NT	:	NT
LESSON	:	DEVOPS
SUBJECT	:	PROMETHEUS-GRAFANA (Kubernetes Security)
BATCH	:	B 224

TECHPRO EDUCATION

techproeducation.com

+1 (585) 304 29 59

What is Prometheus?

Metrics-based monitoring & alerting stack

- Metrics collection and storage
- Querying, alerting, dashboarding
- For all levels of the stack!

Made for dynamic cloud/container environments

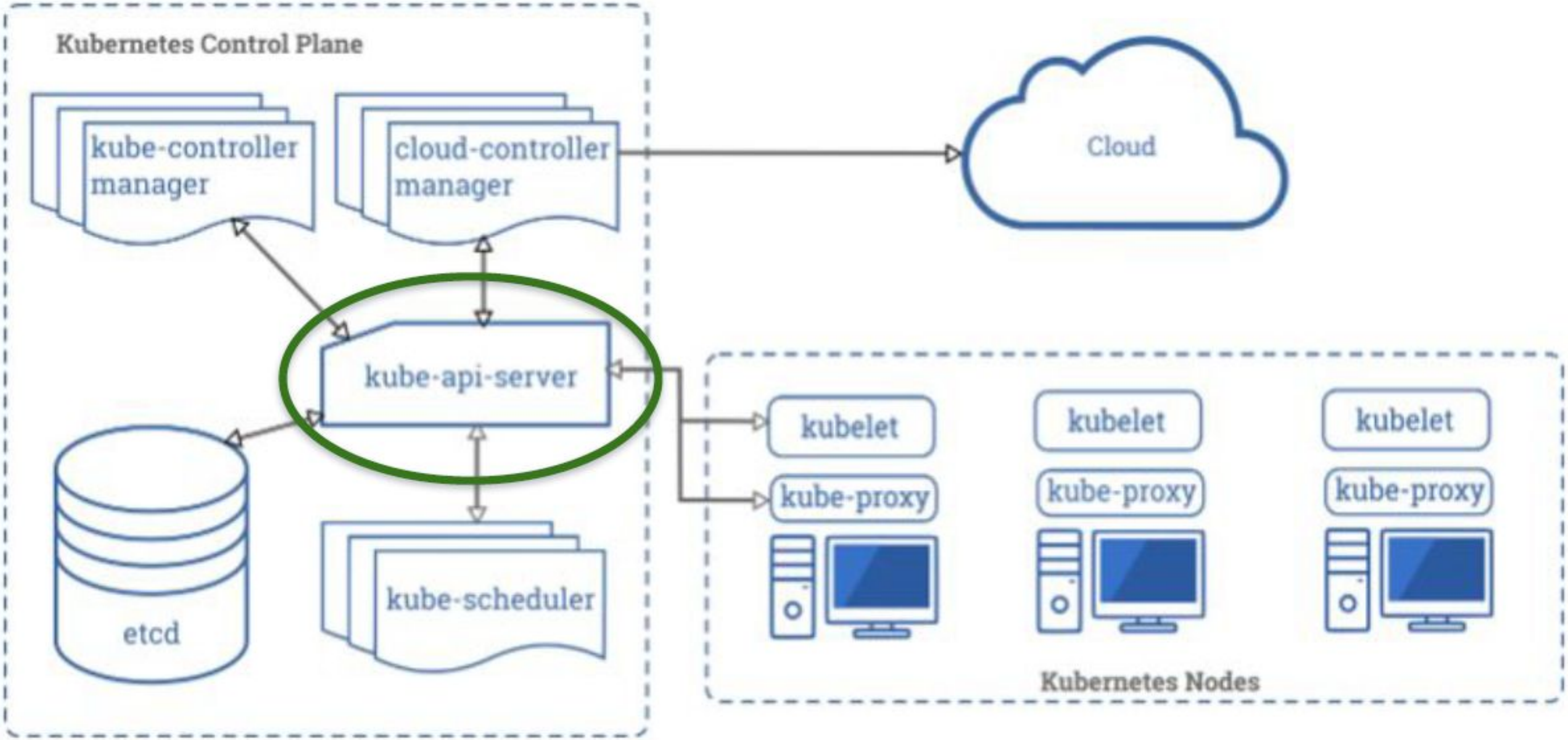
What is Grafana?

Grafana is an open-source analytics and interactive visualization web application.

It provides charts, graphs, and alerts for the web when connected to supported data sources.

Kubernetes Security

Core Concepts



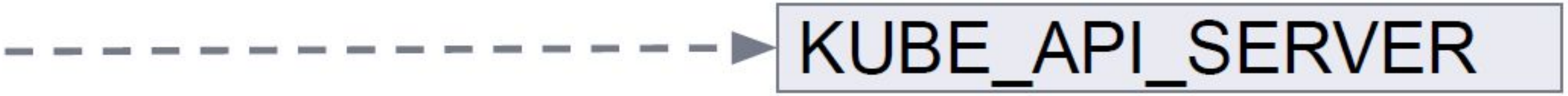
Core Concepts

kube-apiserver:

- Provides a forward facing REST interface into the Kubernetes control plane and datastore.
- All clients and other applications interact with kubernetes strictly through the API Server.
- Acts as the **gatekeeper** to the cluster by handling **authentication** and **authorization**, request validation, mutation, and admission control in addition to being the front-end to the backing datastore.

Core Concepts

Who can Access?



KUBE_API_SERVER

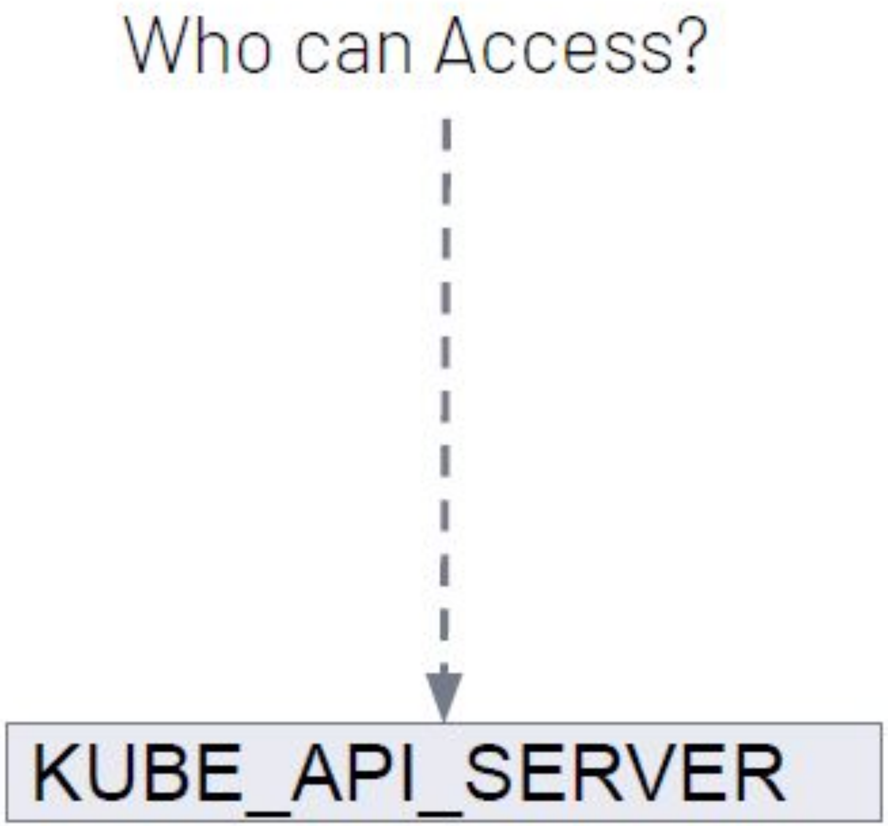
What can they do?

Core Concepts

Who can Access? → **Authentication**

What can they do? → **Authorization**

Authentication



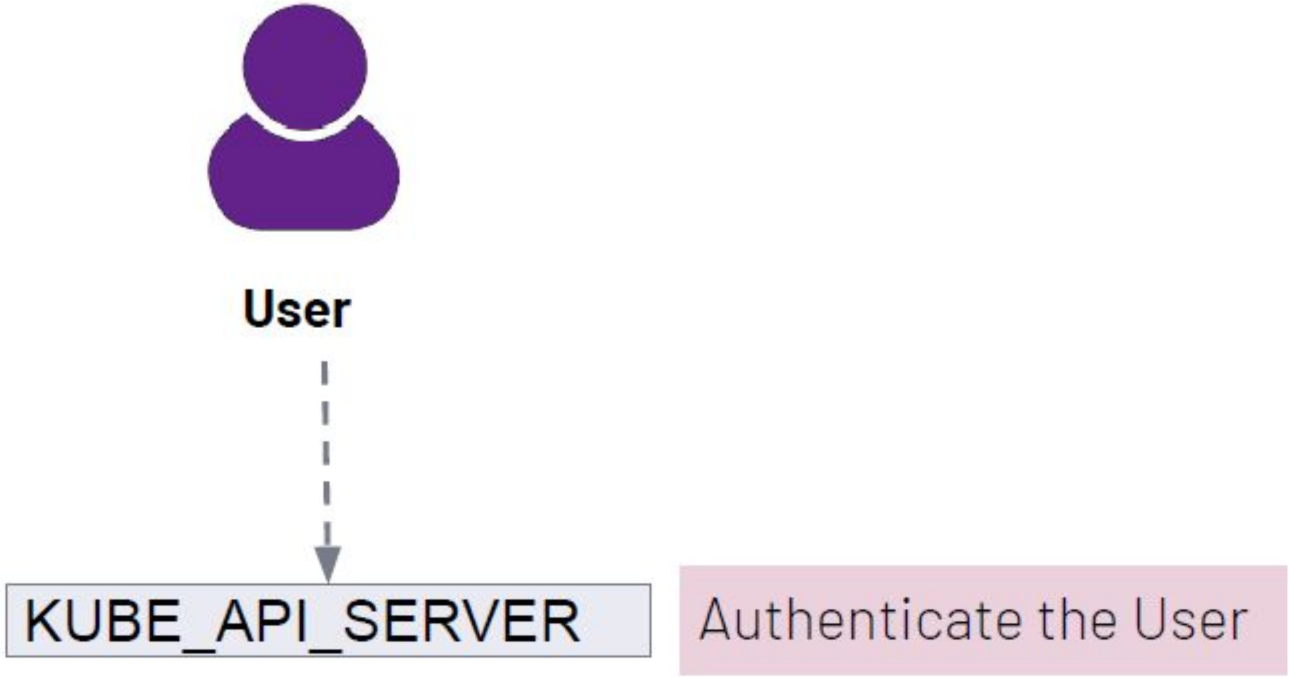
Who can Access?

KUBE_API_SERVER

Authentication

- ✔ **User accounts** are for humans. **Service accounts** are for **processes**, which run in pods.
- ✔ **User accounts** are intended to be **global**. Names must be unique across all namespaces of a cluster.
- ✔ Service accounts are namespaced.

Authentication



Authenticate the User

Authentication Strategies

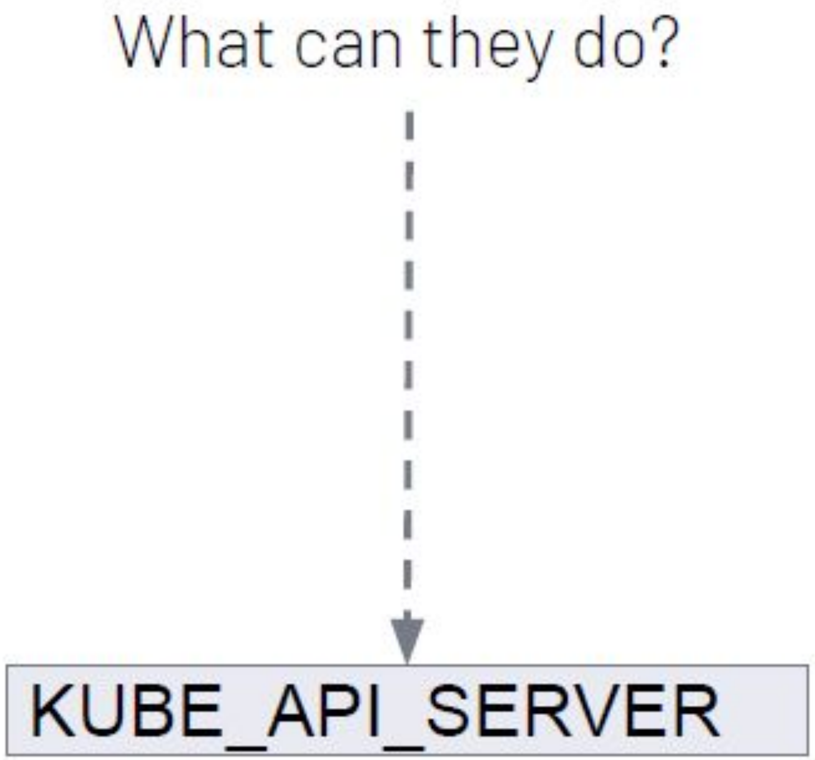
client certificates

Static Token File

Identity Services

IDENTITY
ACCESS CONTROL
AUTHENTICATION
AUTHORIZATION

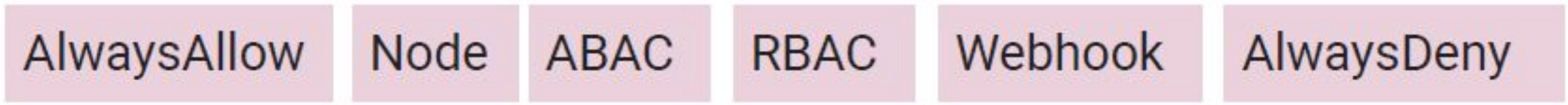
Authorization



What can they do?

KUBE_API_SERVER

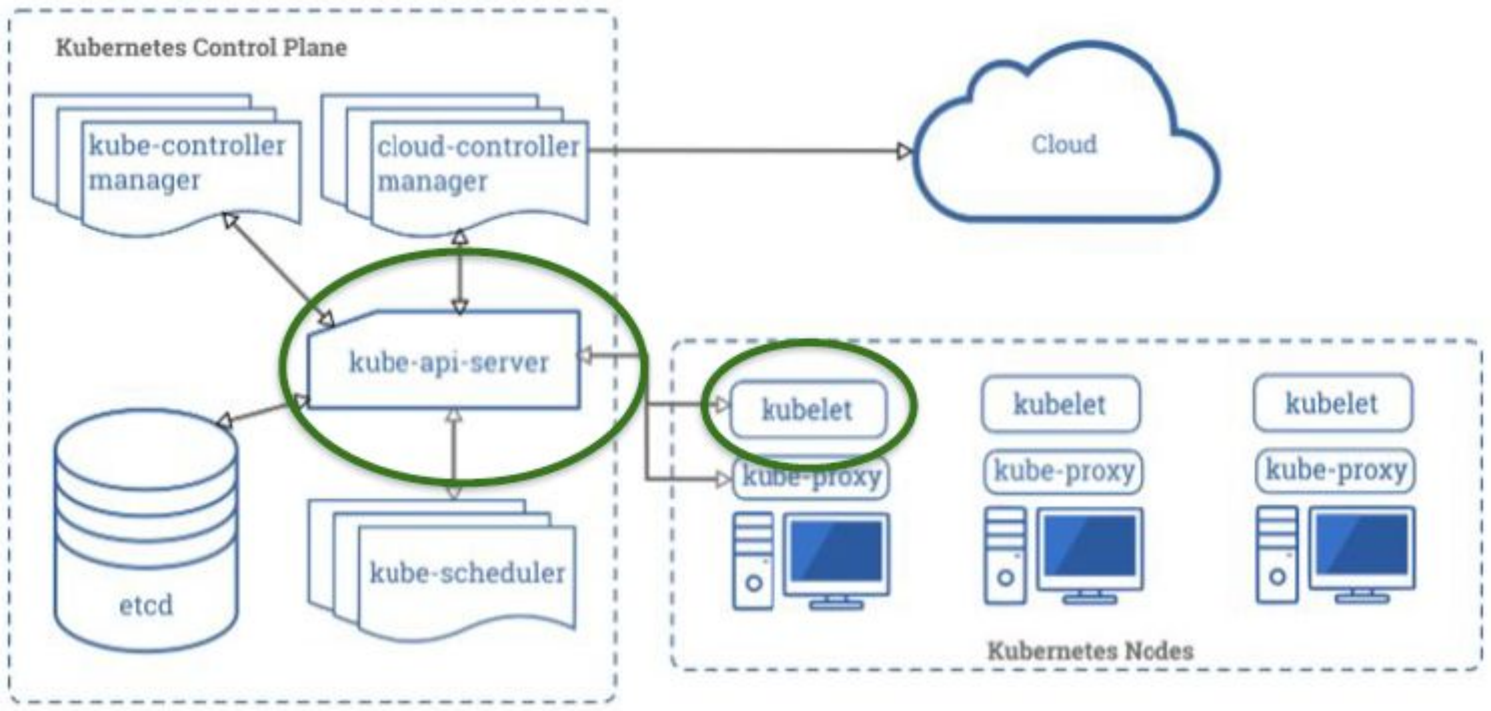
Authorization Modes



AlwaysAllow Node ABAC RBAC Webhook AlwaysDeny

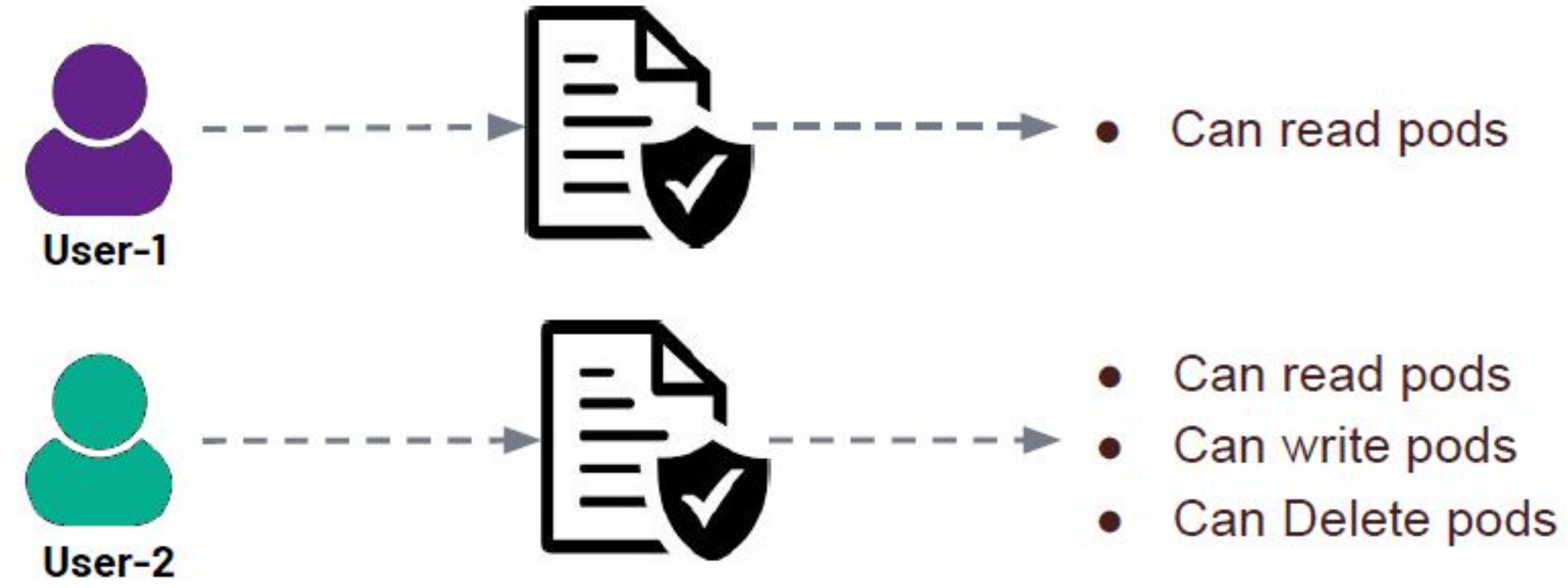
Node

Node authorization is a special-purpose authorization mode that specifically authorizes API requests made by kubelets.



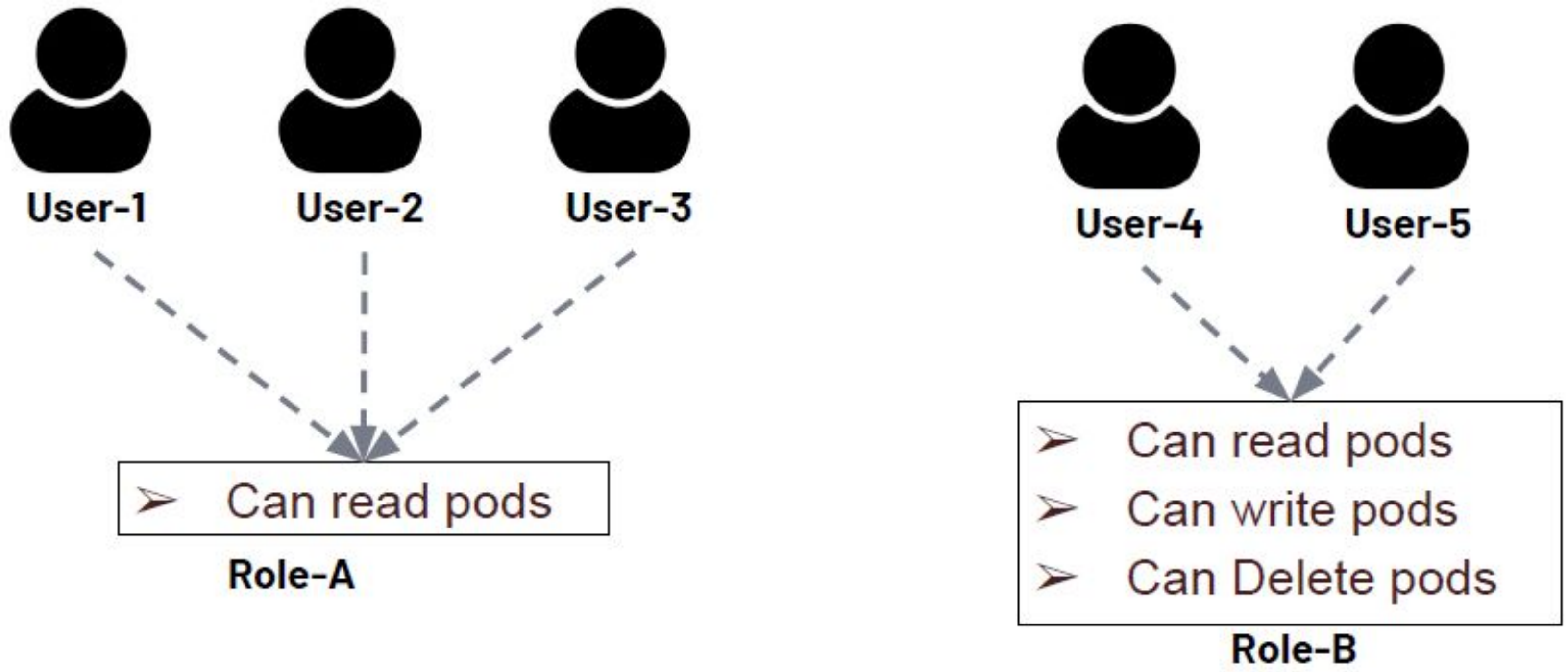
ABAC

Attribute-based access control (ABAC) defines an access control paradigm whereby access rights are granted to users through the use of policies which combine attributes together.



RBAC

Role-based access control (RBAC) is a method of regulating access to computer or network resources based on the roles of individual users within your organization.



Webhook

A **WebHook** is an HTTP callback: an HTTP POST that occurs when something happens; a simple event-notification via HTTP POST. A web application implementing WebHooks will POST a message to a URL when certain things happen.

When specified, mode Webhook causes Kubernetes to query an outside REST service when determining user privileges.

Role and ClusterRole

RBAC Role or ClusterRole contains rules that represent a set of permissions.

- A **Role** always sets permissions within a **particular namespace**; when you create a Role, you have to specify the namespace it belongs in.
- **ClusterRole**, by contrast, is a **non-namespaced** resource.

RoleBinding and ClusterRoleBinding

- A **role binding** grants the permissions defined in a role to a user or set of users.
- A **RoleBinding** grants permissions within a specific namespace whereas a **ClusterRoleBinding** grants that access cluster-wide.

API Groups

- Kubernetes API is grouped into multiple such groups based on their purpose. Such as one for apis, one for healthz, metrics and logs etc.
- The version API is for viewing the version of the cluster.
- metrics and healthz api are used to monitor the health of the cluster.

  /api
  /apis
  /logs
  /healthz
  /metrics
  /version

API Groups

/api

/apis

/healthz

/metrics

/logs

/version

API Groups

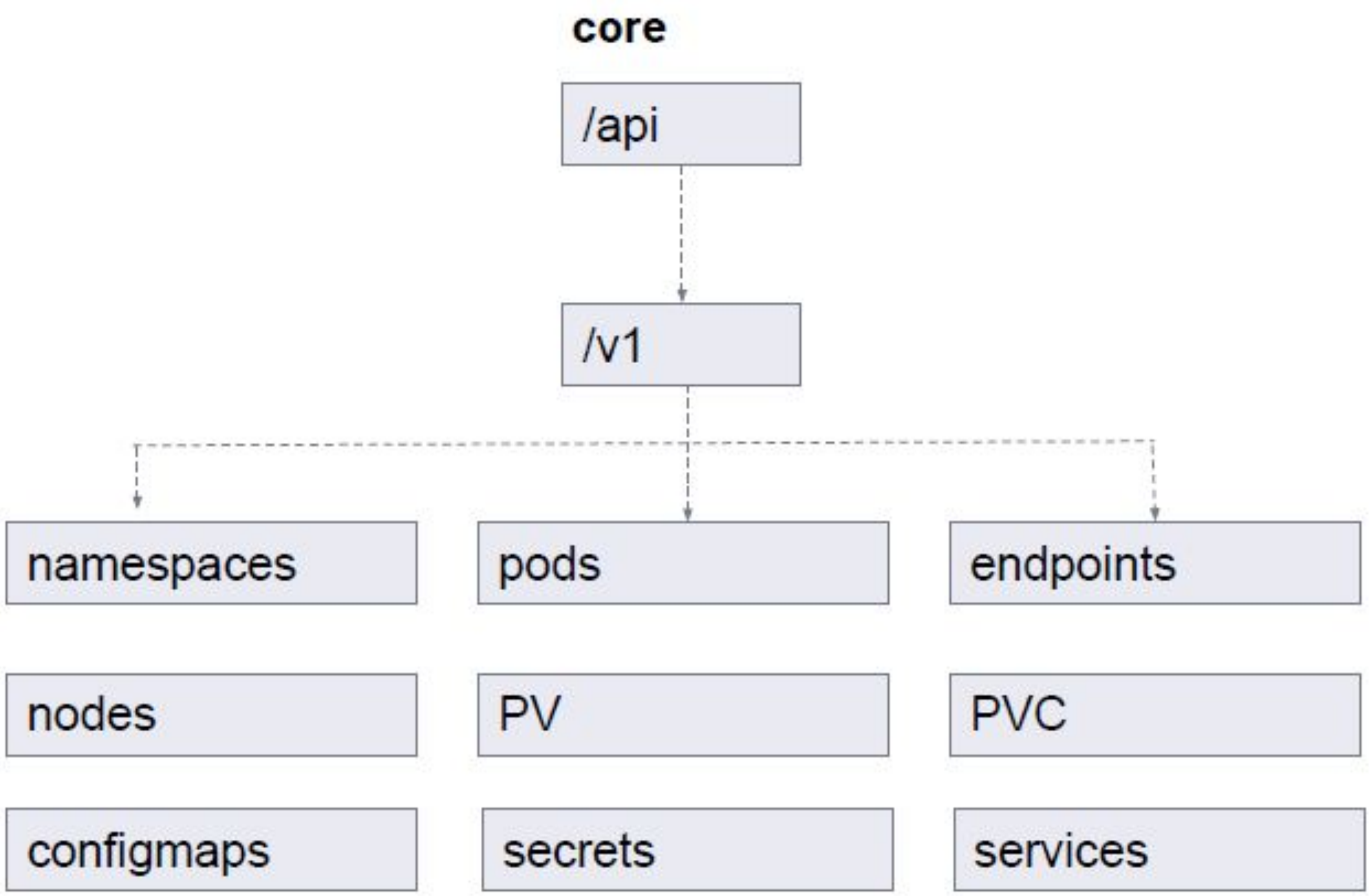
core

/api

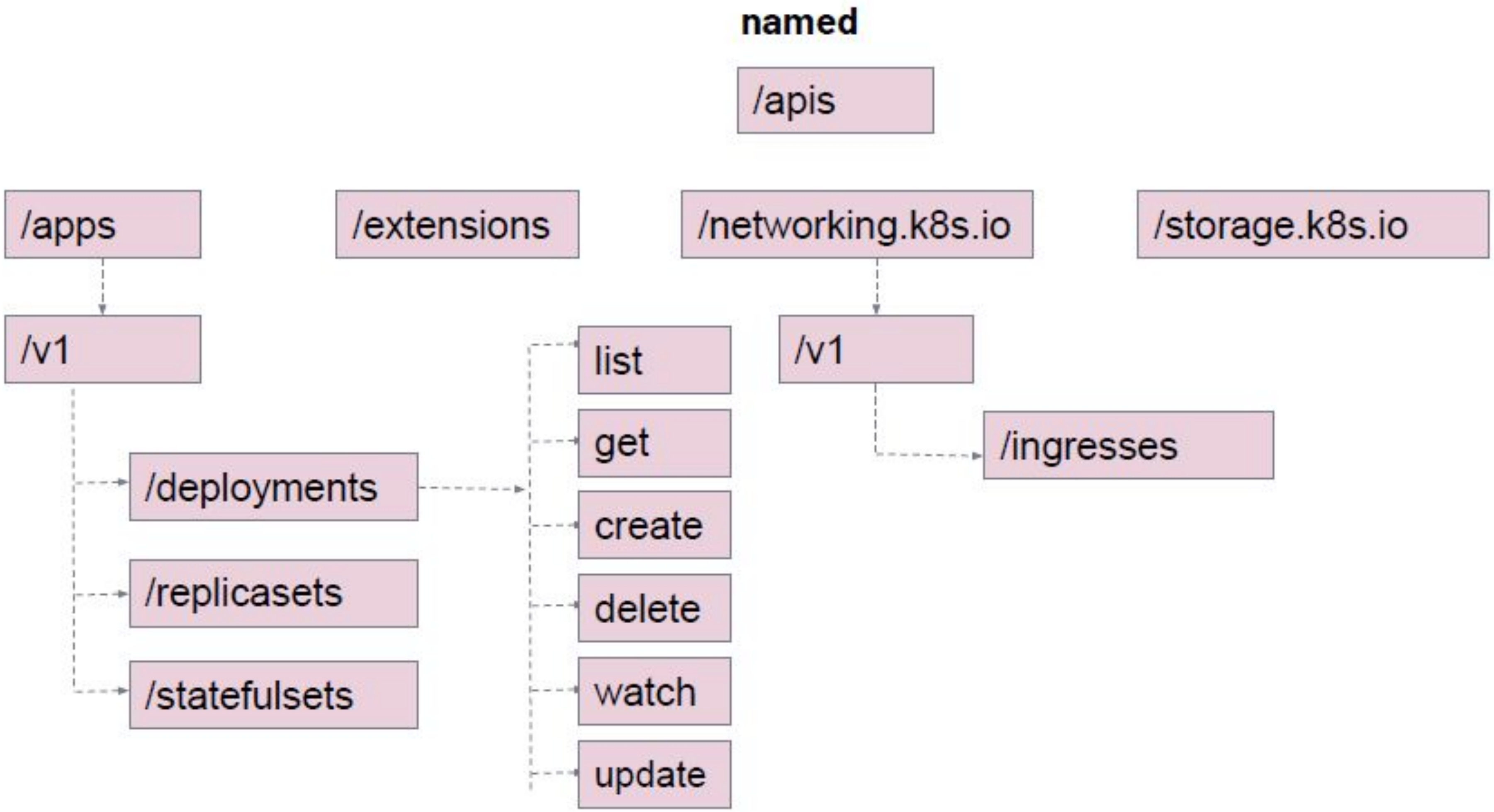
named

/apis

API Groups


core
/api
/v1
namespaces
pods
endpoints
nodes
PV
PVC
configmaps
secrets
services

API Groups


named

/apis

/apps

/extensions

/networking.k8s.io

/storage.k8s.io

/v1

/deployments

/replicasets

/statefulsets

list

get

create

delete

watch

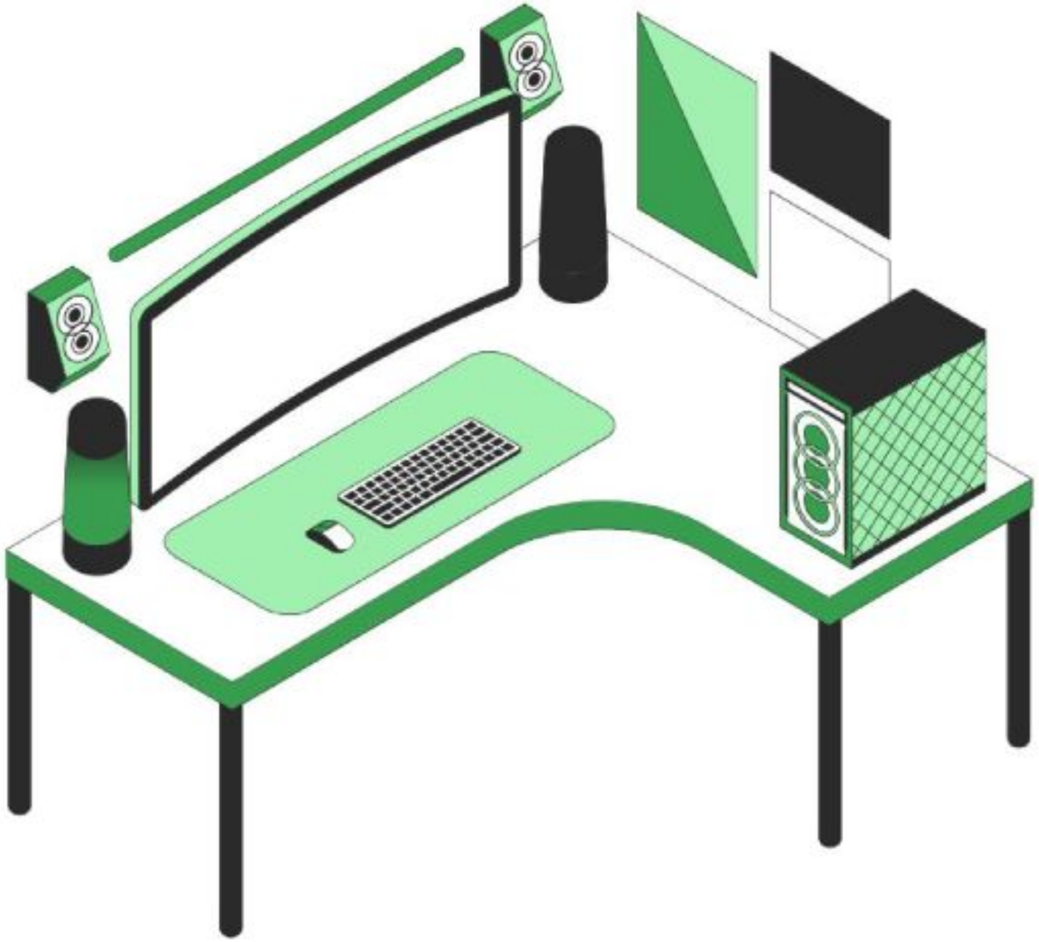
update

/v1

/ingresses

API Groups

- kubectl proxy --port=8080 &
- curl localhost:8080
- curl localhost:8080/version → kubectl version
- curl localhost:8080/api/v1/pods



Do you have any questions?

Send it to us! We hope you learned something new.