Comparison between ingress controllers and methods to secure applications in k3s cluster based on authentication and automated certificates

Presenter → Dikshita Kalita
Supervisor → Prof. Dr Martin Leischner
Mentor → Richard Clauß

Agenda

01 Research Questions

02 Ingress controllers

- Introduction
- Types
- Comparison
- Preference

04 Demos

03 Security in Kubernetes

- Introduction
- Users
- Access Control
- Pipeline
- AuthN Strategies
- AuthZ Modules
- OIDC
- AuthN Protocol Comparison

01

Research Questions

1. Basic properties of underlying implementations
    a. What are the principal point of differences between Traefik, HaProxy and Nginx?

2. Security in kubernetes
    a. Analyse best modes to authenticate users accessing the cluster and its resources.
    b. Compare their key differences based on scenarios.

02

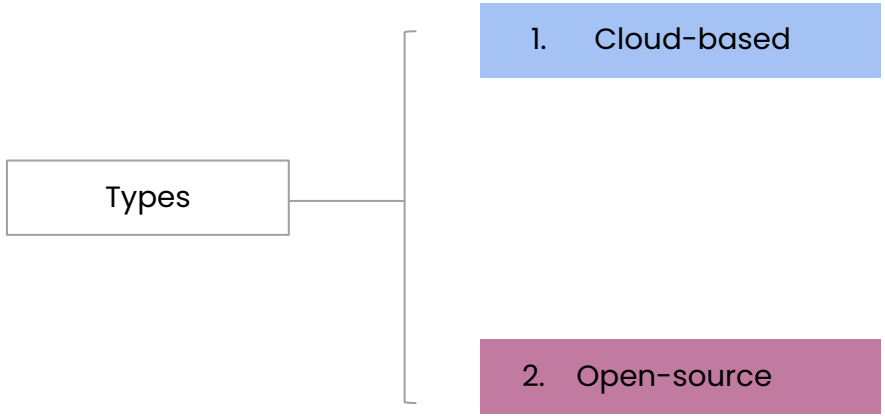
Ingress controllers

02 Ingress controllers | Introduction | 1

- Object that allows access to the Kubernetes services from outside of the cluster.
- Controls how the external users should access the services running in a kubernetes cluster typically via HTTP/HTTPS through a single externally reachable IP address.
- Operates at the application layer of the network stack.

Continued ...

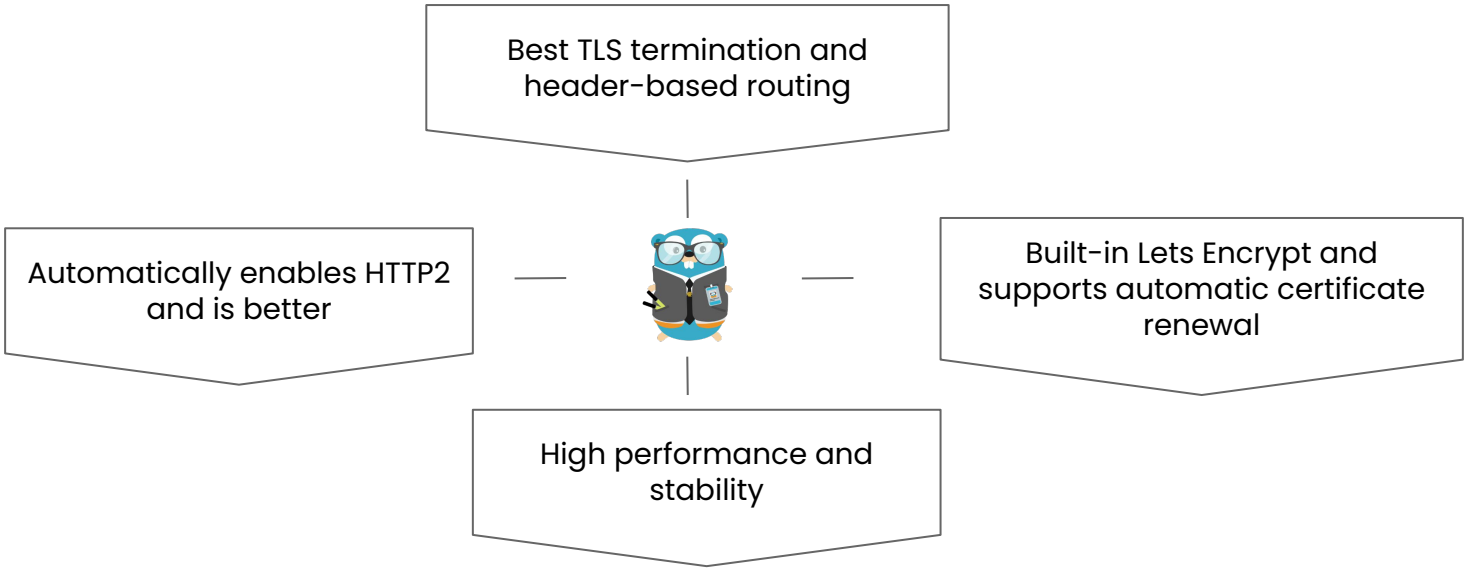
02 Ingress controllers | Types & Comparison



02 Ingress controllers | Types & Comparison

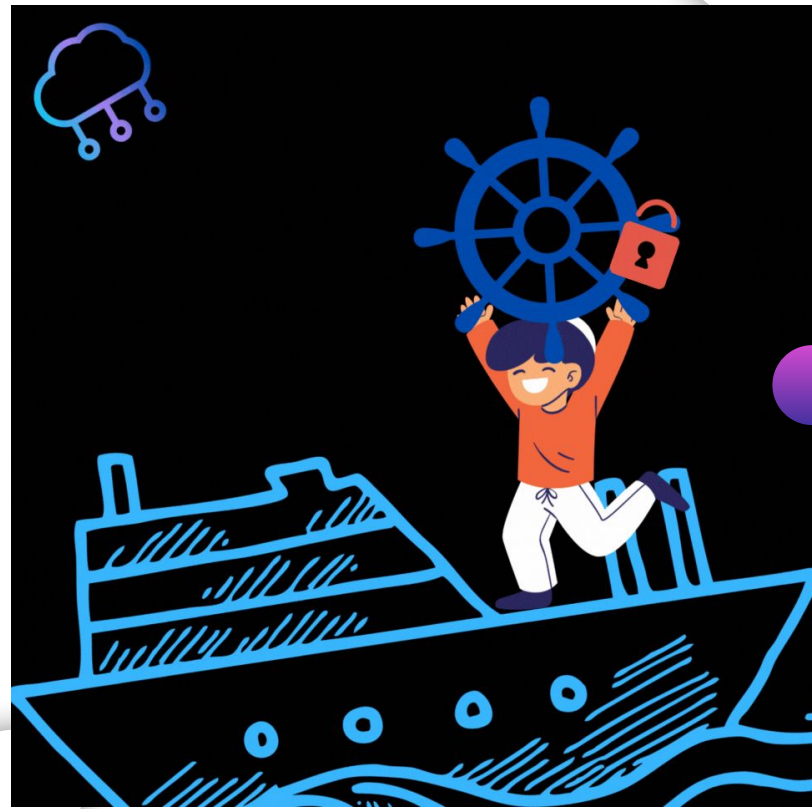
	Nginx	Ha-proxy	Traefik
Build on	nginx/nginx plus	haproxy	traefik
Protocols supported	• HTTP(s) • HTTP2 • TCP/UDP	• HTTP(s) • HTTP2 • TCP+TLS	• HTTP(s) • HTTP2 • TCP+TLS
Traffic routing logic	• Host • Path • Header • Method • Query params	• Host • Path	• Host • Path • Headers • Query • Methods • Path prefix

02 Ingress controllers | Preference



03

Security in Kubernetes



03 Security in Kubernetes | Introduction | 1

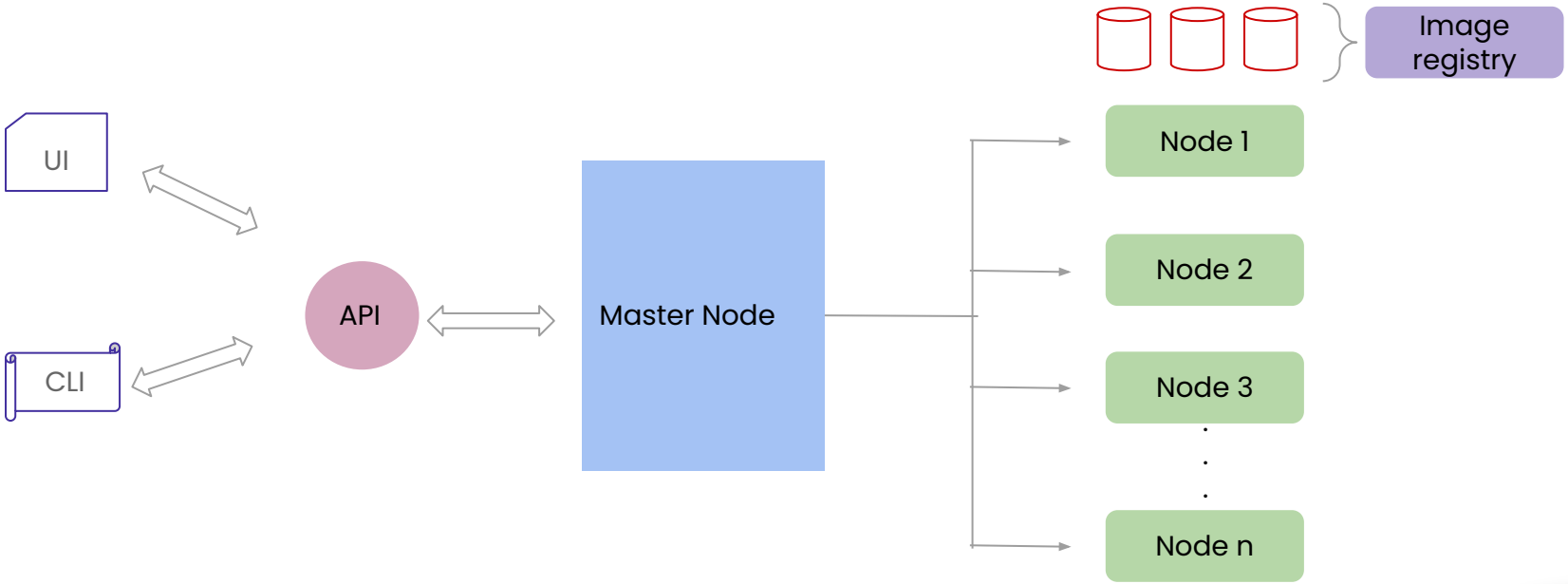
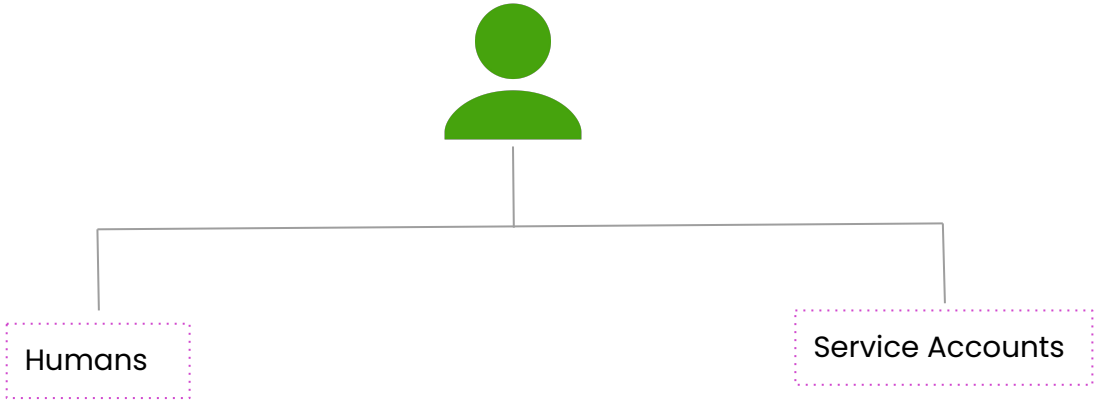
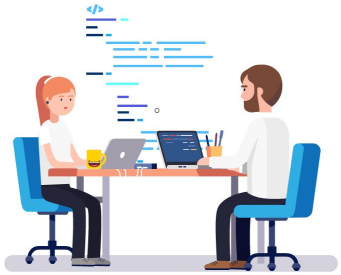


Fig: Architecture in bigger picture

03 Security in Kubernetes | Users

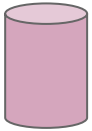


03 Security in Kubernetes | User 1- Humans



For human users

Kubernetes does not have



User, profile database / lookup table to store usernames, passwords to store usernames, passwords

i.e.

NO API calls to create user

Instead

RELIES on a **variety of techniques** to delegate that

03 Security in Kubernetes | User 2- Pods

Service Account / SA is

1. Internal representation of a set of credentials that are typically stored as Secrets.

2. Pods uses this SA to authenticate when they talk to internal API endpoint

3. By default, SA have no access permissions but can be configured using RBAC to give them access permission and role-based control so they can query and manipulate

03 Security in Kubernetes | ServiceAccount

Very useful if we have an automatized application set-up in the cluster to deploy the pods after its build up

1.
```
root@masterdev:/home/masterdev# k3s kubectl create serviceaccount sa -n default
serviceaccount/sa created
```

2. Create a Role:

   k3s kubectl apply -f default-role.yaml

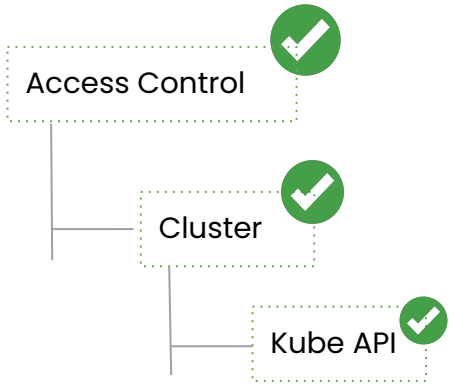
3. Create a RoleBinding:

   k3s kubectl apply -f role-binding.yaml

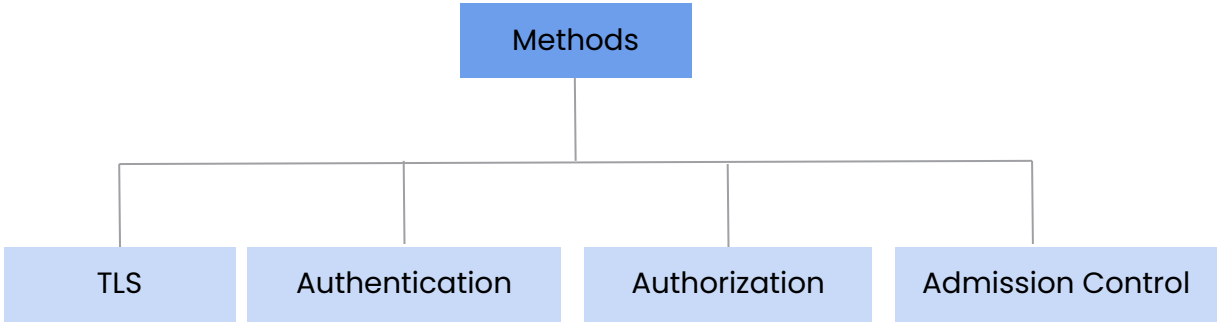
03 Security in Kubernetes | Access Control | 1

Kubernetes is **not** a security platform

- Native tooling to handle most security-related tasks
- Detecting vulnerabilities within applications
- Monitoring for attacks / breaches



03 Security in Kubernetes | Access Control | 2



03 Security in Kubernetes | Security Pipeline

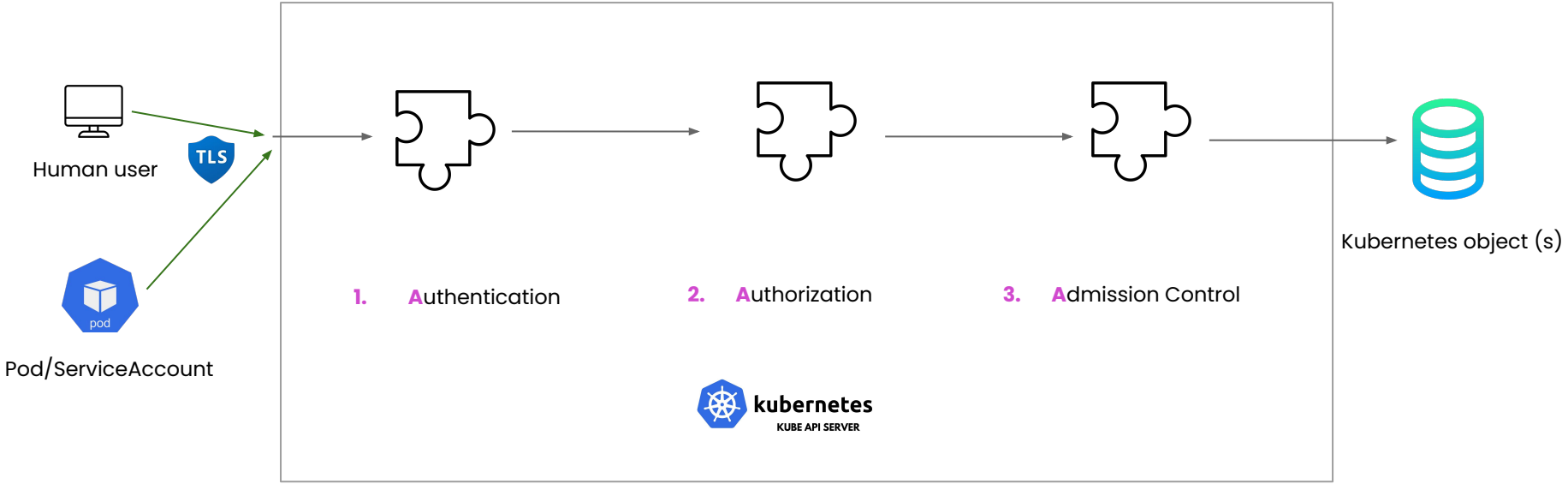


Fig: Kubernetes security pipeline, Source: Click here

Continued ...

03 Security in Kubernetes | AuthN Strategies

Basic Authentication

- Pass a CSV with the following:

```
<password>,<username>,<UID>,"<group1, group2>"
<password>,<username>,<UID>,"<group2, group4>"
```

- Not scalable

X509 Client Certificates

```
root@masterdev:/var/lib/rancher/k3s/server/tls# cat client-ca.crt
-----BEGIN CERTIFICATE-----
MIIBdzCCAR2gAwIBAgIBADAKBggqhkjOPQQDAjAjMSEwHwYDVQQDDBhrM3MtY2xp
ZW50LWNhQDE2Njk2NDU3MjEwHhcNMjIxMTI4MTQyODQxWhcNMzIxMTI1MTQyODQx
WjAjMSEwHwYDVQQDDBhrM3MtY2xpZW50LWNhQDE2Njk2NDU3MjEwWTATBgcqhkjO
PQIBBggqhkjOPQMBBwNCAASBll28qpxgEkZlrKH/ZWcu9hxv8PSVlrC1TK1I/CsK
g8GaDkoUl7r7M3dmM87Mh5h/9wRkIxd0JvgT7EN7zCCdo0IwQDAOBgNVHQ8BAf8E
BAMCAqQwDwYDVR0TAQH/BAUwAwEB/zAdBgNVHQ4EFgQUmfJYGHagEFd7W9P5bX8/
A5l9F2gwCgYIKoZIzj0EAwIDSAAwRQIgDzZi/E/MlPAwD+DQzU+f7ryqu+Tv15kZ
5/fKrL7nfLYCIQD1cJgFyyJ3F4lEEYQyt7cNEg5M8V6mBizNAZu1NsQXeA==
-----END CERTIFICATE-----
```

- Insecure
- Long-lived and can't be revoked effectively
- Makes it hard to use groups with RBAC

Tokens

- OpenID Connect

03 Security in Kubernetes | AuthZ Modules

1. AlwaysAllow
2. AlwaysDeny

Not for production **but for** specified use-case

3. Node — Using this node or kubelet would talk to API server
4. Attribute-Based Access Control(ABAC)
5. Role-based Access Control (RBAC)
6. Webhook — We can connect multiple webhooks for authorization

1. It evaluates attributes or characteristics in order to determine the access

2. The administrator defines a set of user authorization policies into a file with one JSON per line format.

Any modification in the file requires API server to be restarted.

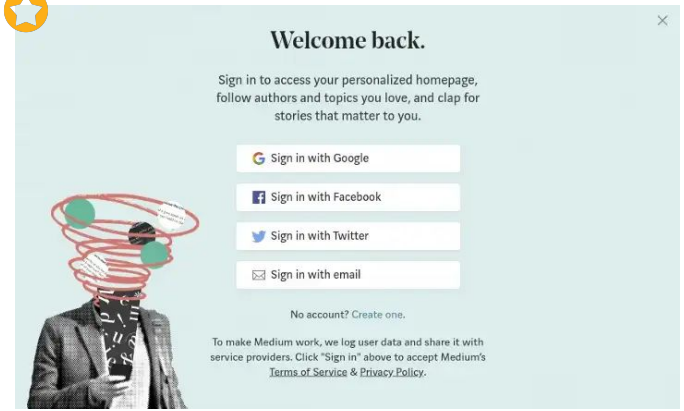
1. Here we define **Roles** and with these roles a user can access kubernetes objects

03 Security in Kubernetes | Auth | OIDC

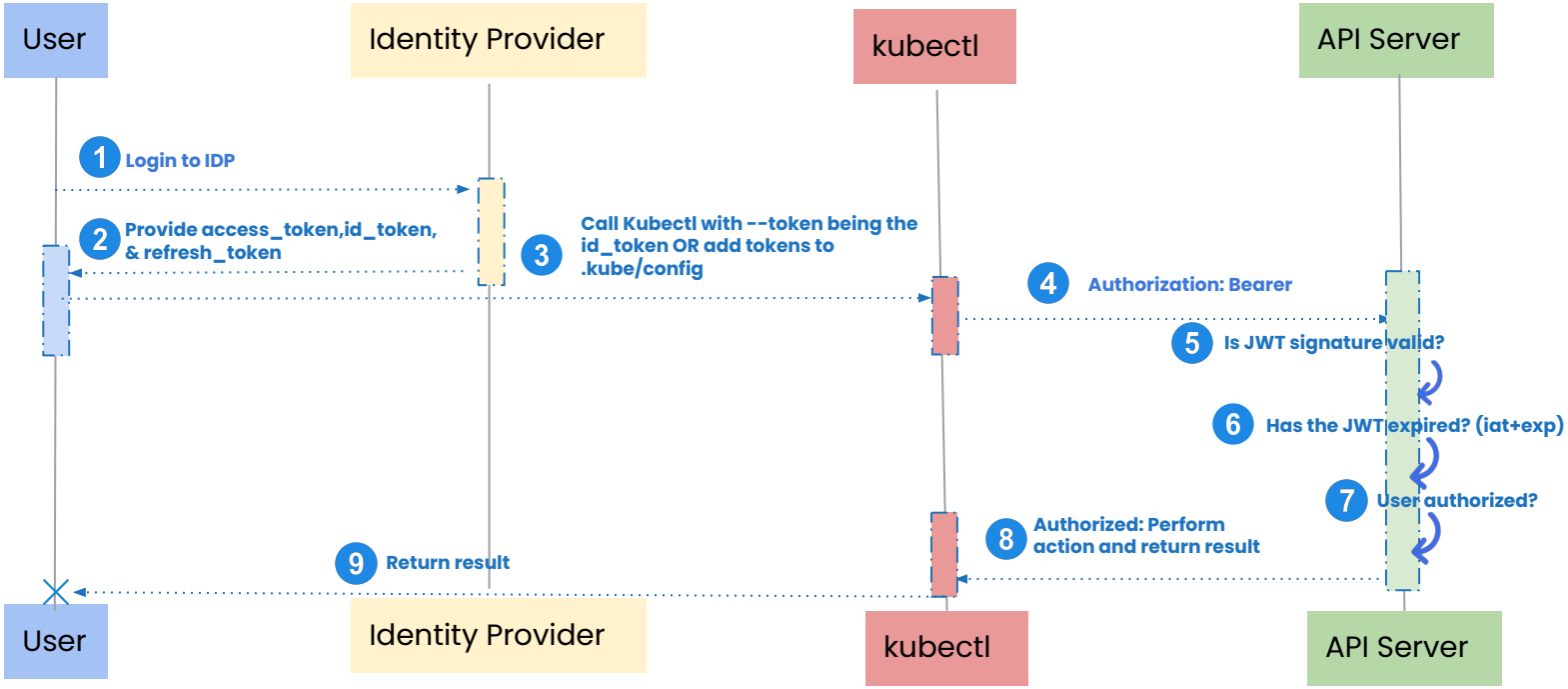
1. Widely accepted as a solution to authentication of users on web services
2. Easy to use because it sits on top of existing technology OAuth2.0
3. Typical use-case: Identity Federation

Simple identity layer for **Authentication**

Standard protocol for **Authorization**



03 Security in Kubernetes | Auth | OIDC



03 Security in Kubernetes | OIDC | IdP

- A system entity that creates, maintains, and manages identity information for users and also provides authentication services to relying applications.

Does Kubernetes provide an OpenID Connect Identity Provider ?

Solution ?

03 Security in Kubernetes | OIDC | IdP

We can

1. Use an existing public OpenID Connect IdP

- Google
- Microsoft
- Amazon
- Okta
- PayPal
- SalesForce
- PhantAuth
- Github

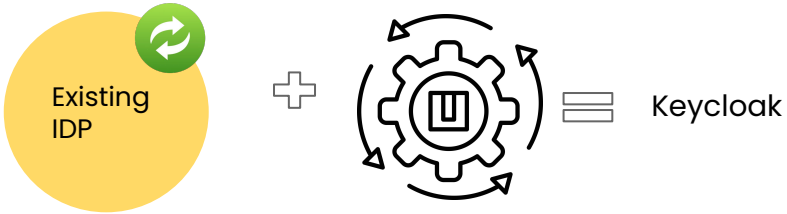
2. Run our own Identity Provider

- Keycloak
- Dex
- CloudFoundry User Account and Authentication
- Tremolo Security's OpenUnison

03 Security in Kubernetes | OIDC | IdP

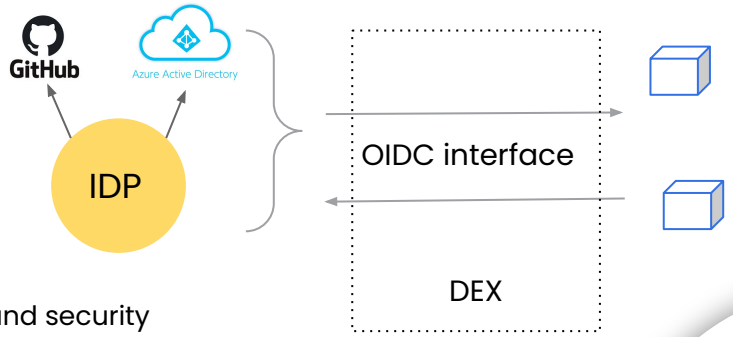
Keycloak

- Complex to operate since it needs standalone database
- **Scenario:**



Dex

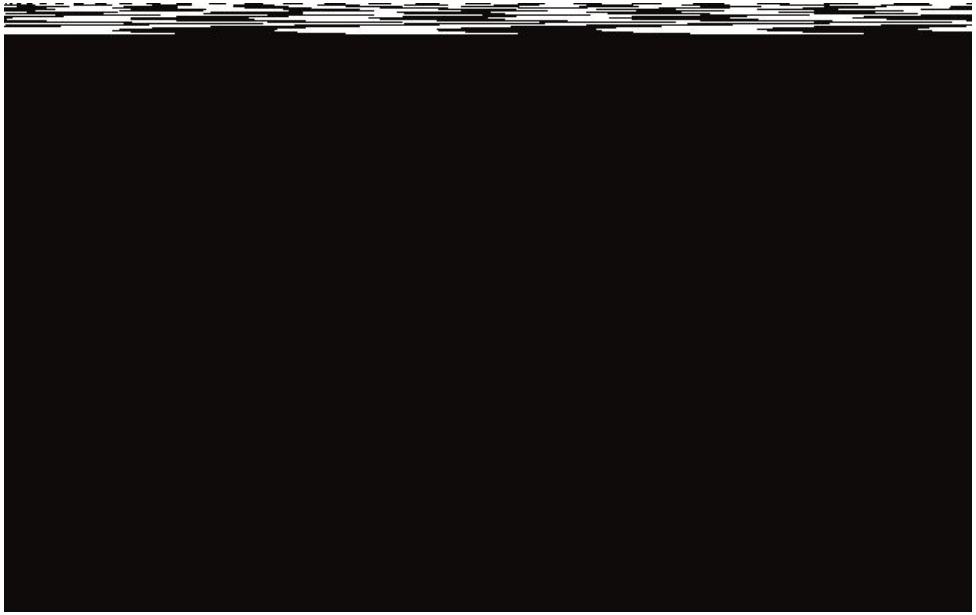
- Simple and easy to set up
- **Scenario:**



03 Security in Kubernetes | Auth | Comparison

	Kerberos	Lightweight Directory Access Protocol | LDAP	OAuth2.0	Security Assertion Markup Language | SAML	OIDC
Function	Aids in network authentication	Used for determining any individuals, organizations, and other devices during a network regardless of being on public or corporate internet. It is practiced as Directories-as-a-Service and is the grounds for Microsoft building Activity Directory.	Authorization framework that grants limited access to the user on its account through an HTTP service	XML-based authentication data format which provides authorization between identity provider(Google, AWS, Microsoft Active Directory or Azure) and service provider(Salesforce and other CRM solutions).	Authentication protocol that has API-centered architecture that uses JSON tokens (ID token) and is currently supported by many popular web services, including Google, Paypal, Microsoft and Amazon.
Pros	1. Supports many OS 2. Authentication key is shared much efficiently than public sharing	1. Automated protocol which makes it modernize easily 2. Supports existing technologies and allows multiple directories	1. Simple protocol and is easy to implement 2. Provides server-side authorization of code	1. Reduced the administrative costs for the end-users 2. Provides SSO for authenticating across service providers	1. Fast and easy implementation 2. Lightweight and more performance-friendly than SAML 3. Provides a frictionless user experience for mobile and single-page web applications.
Cons	1. Used only to authenticate clients and services used by them 2. Vulnerable to soft or weak passwords	1. Directory servers are required to be LDAP obedient for deployment 2. No 2FA 3. Not scalable	1. Vulnerable to manage different sets of code	1. Dependent on the identity provider	1. Still new and evolving and lacks some high-security features that are needed by certain sectors, such as the banking industry.

03 Security in Kubernetes | Demos



Resources

- https://kubernetes.io/docs/reference/access-authn-authz/authentication/
- https://www.tremolosecurity.com/post/kubernetes-dont-use-certificates-for-authentication
- https://cloudinfrastructureservices.co.uk/oauth2-vs-openid-whats-the-difference/
- https://sysdig.com/learn-cloud-native/kubernetes-security/kubernetes-rbac/
- https://www.geeksforgeeks.org/types-of-authentication-protocols/
- https://www.strongdm.com/blog/oidc-vs-saml#:~:text=OIDC%3A%20What's%20the%20Difference%3F,to%20obtain%20the%20security%20token.
- https://www.varonis.com/blog/what-is-oauth
- https://dinika-15.medium.com/identity-federation-a-brief-introduction-f2f823f8795a
- https://en.wikipedia.org/wiki/Identity_provider
- https://metal-k8s.readthedocs.io/en/latest/developer/architecture/authentication.html
- https://medium.com/@sct10876/keycloak-vs-dex-71f7fab29919

Note of gesture

Thank you for your attention
and patience