Securing Kubernetes Cluster with OIDC

Akanksha Kumari

Member of Technical Staff

Agenda

Challenges in Kubernetes

Introduction to OIDC

Key Concepts in OIDC

Authorization Code Flow

Configuring OIDC in Kubernetes

Benefits of OIDC in Kubernetes

OIDC Best Practices

Challenges in Kubernetes

- Static Password Files
- X.509 Certificates
- Service Account Tokens
- OpenID Connect (OIDC)

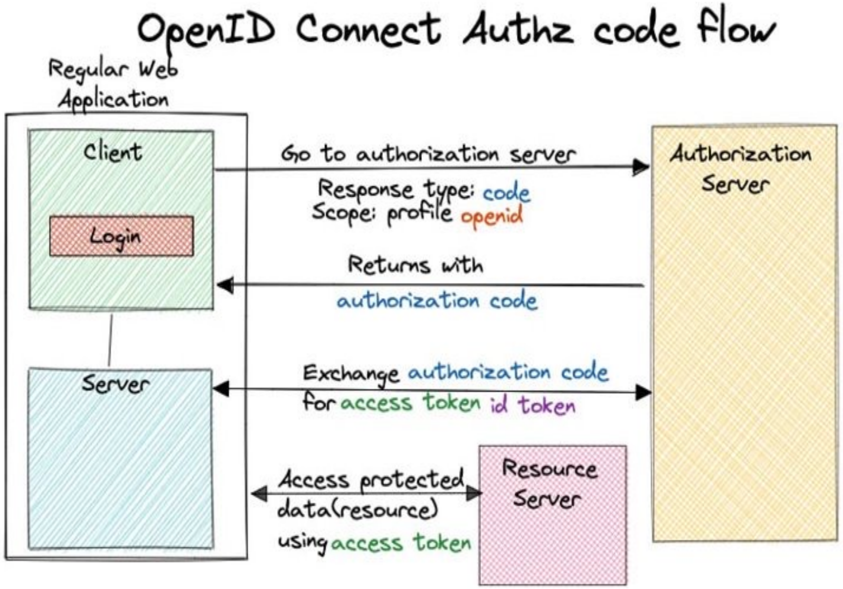
Introduction to OIDC

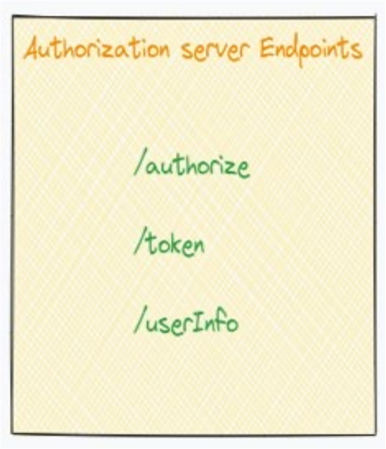
- OIDC stands for OpenID Connect.
- OIDC is an identity layer built on top of the OAuth 2.0 protocol.
- OIDC is commonly used for single sign-on (SSO) scenarios in web and mobile applications.
- Provides a standardized way for users to authenticate and authorize access to their resources.
- Popular identity providers that supports OIDC include Google, Microsoft Azure Active Directory, Okta, and Auth0

Key Concepts in OIDC

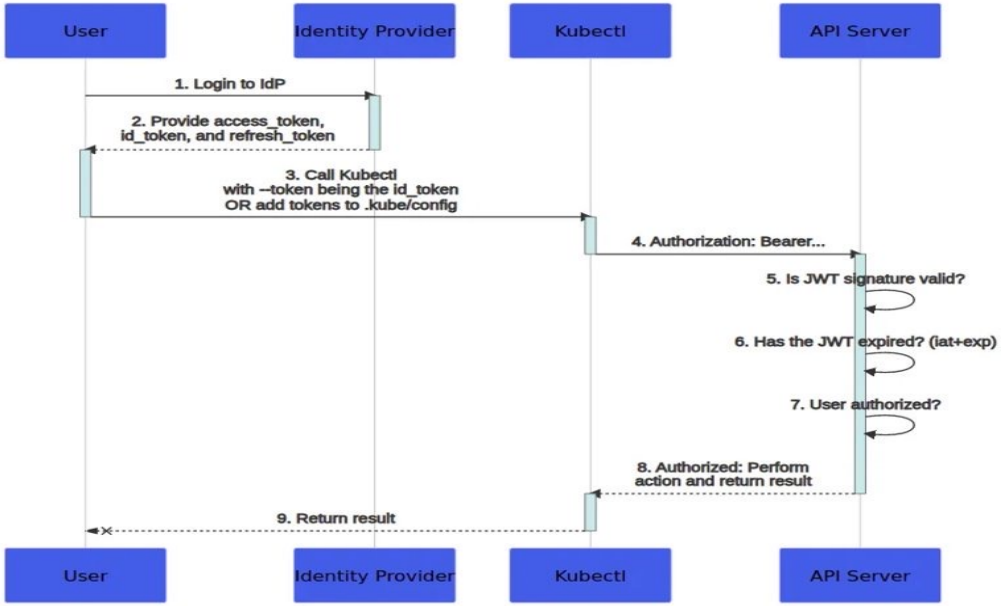
- Identity Providers IdPs (Okta, Microsoft Azure Directory)
- Client
- Tokens(ID Tokens, Access Tokens, Refresh Tokens)
- Scopes
- Flows:
  - Authorization Code Flow
  - Implicit Flow
  - Hybrid Flow

Authorization Code Flow





Kubernetes OIDC Flow



User

Identity Provider

Kubectl

API Server

1. Login to IdP

2. Provide access_token, id_token, and refresh_token

3. Call Kubectl with --token being the id_token OR add tokens to .kube/config

4. Authorization: Bearer...

5. Is JWT signature valid?

6. Has the JWT expired? (iat+exp)

7. User authorized?

8. Authorized: Perform action and return result

9. Return result

User

Identity Provider

Kubectl

API Server

Configuring Kubernetes with OIDC

Configure the OIDC Provider

Register Client within the provider's administration console. Obtain clientID and client secret.

Update the Kubernetes API server configuration.

--oidc-issuer-url

--oidc-client-id

--oidc-username-claim(Optional)

Configure Kubectl (use OIDC Authenticator or –token Flag)

Authorization

User Mapping

The API server maps the user's identity from the ID Token to a Kubernetes user account.

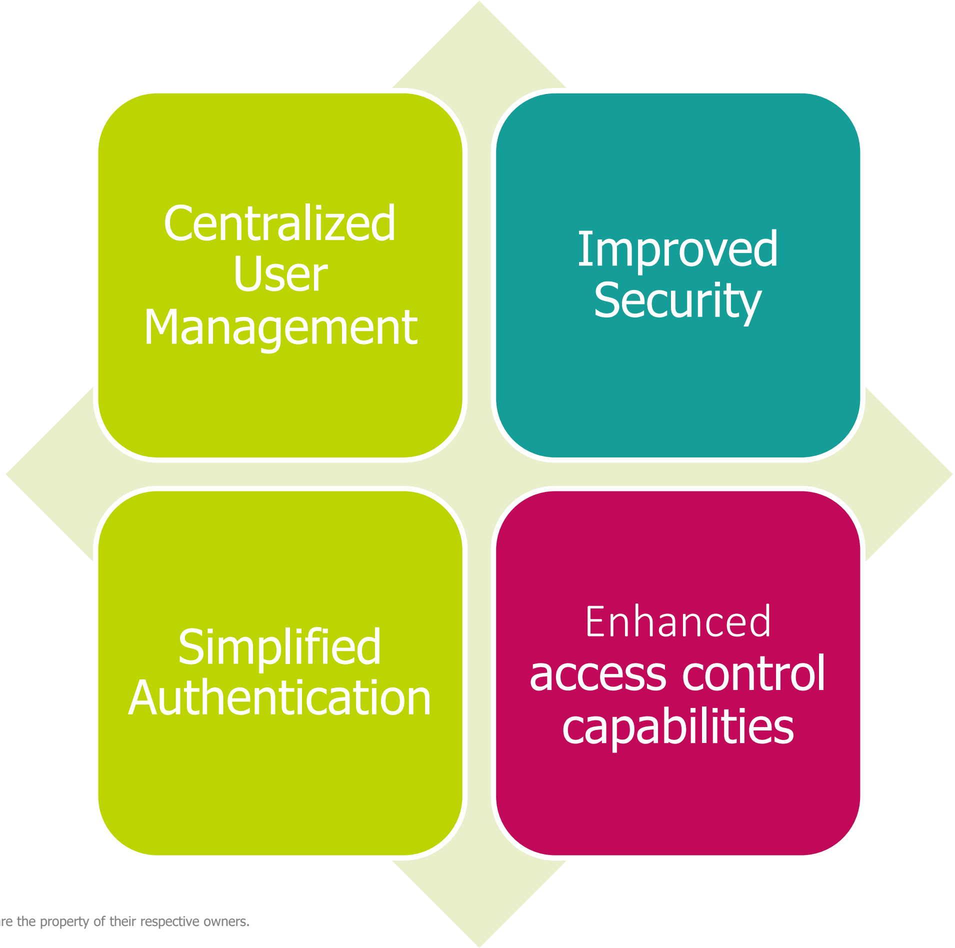
RBAC

Once the user is mapped, Kubernetes applies Role-Based Access Control (RBAC) to determine the user's permissions and access to Kubernetes resources.

Access Control Evaluation

When a user attempts to perform an action on a Kubernetes resource, API server evaluates the RBAC authorization rules.

Benefits Of Using OIDC



Centralized User Management

Improved Security

Simplified Authentication

Enhanced access control capabilities

Best Practices For OIDC Implementation

- Enable RBAC (Role-Based Access Control)
- Regularly Update Identity Providers and Kubernetes
- Implement Multi-Factor Authentication (MFA)
- Monitor and Audit OIDC Integration
- Use Strong Encryption and Secure Communication
- Securely Store and Rotate Client Secrets
- Regularly Review and Test Configuration

Questions?

KASTEN
by Veeam

Thank You

KASTEN
by Veeam