KubeCon | CloudNativeCon

North America 2023

KubeCon | CloudNativeCon

North America 2023

OIDC and Workload Identity in Kubernetes

Anish Ramasekar, Microsoft &
Ashutosh Kumar, Elastic

Intros

Anish Ramasekar

- Software Engineer at Microsoft working on security
- Maintainer of Secrets Store CSI Driver
- Member of Kubernetes SIG-Auth
- Seattle, WA

Intros

Ashutosh Kumar

- Software Engineer at Elastic
- Maintainer of Cluster API Provider Azure

Agenda

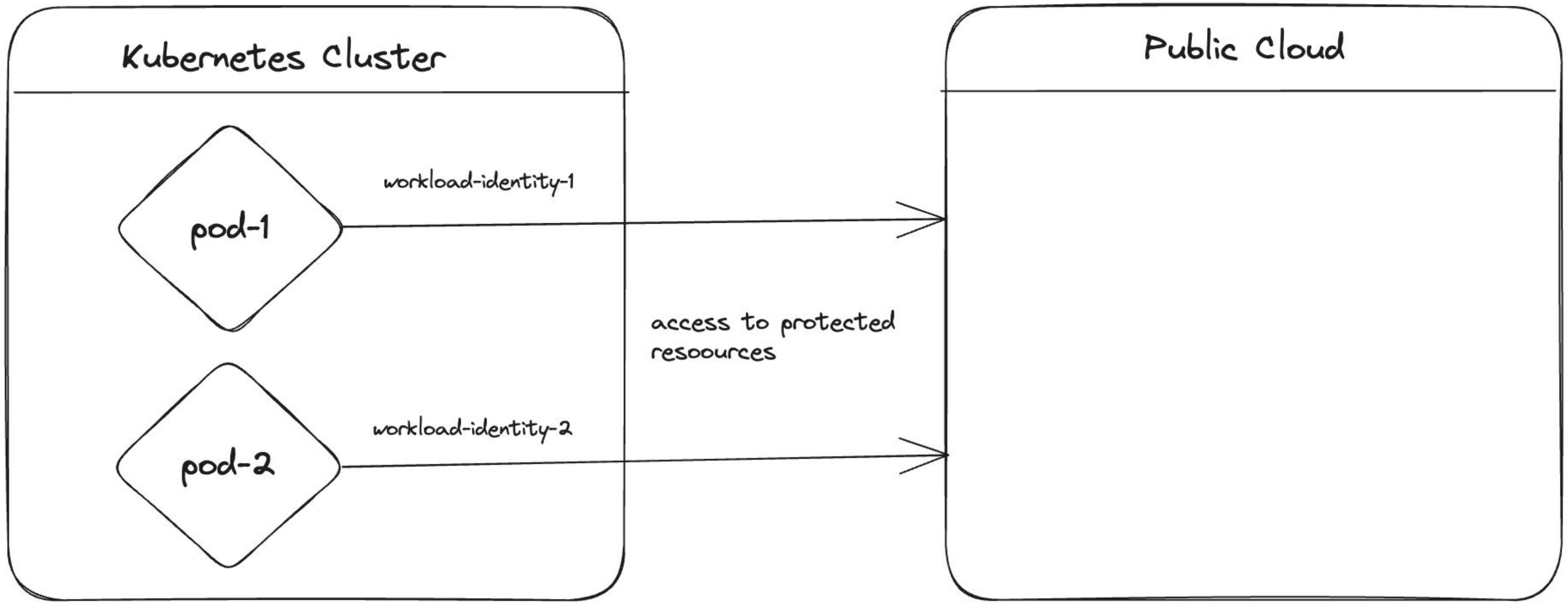
- Introduction: Workload Identity
- Authentication and Authorization
- Workload Identity: Functionality and Mechanics
- Demo
- Workload Identity usages in Kubernetes

Introduction: Workload Identity

- Workloads deployed in Kubernetes may require access to external protected resources e.g. on public cloud.

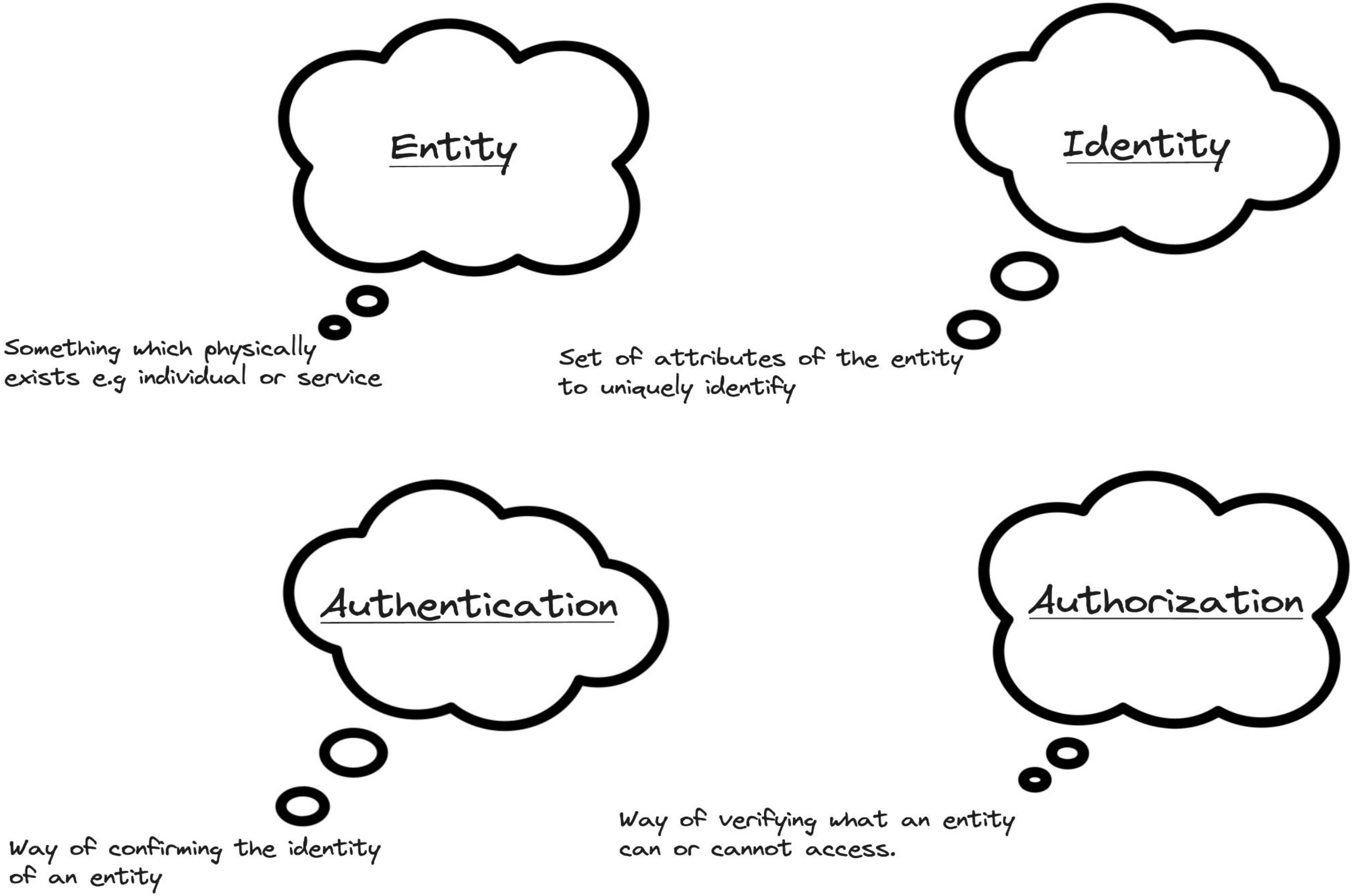
- Workload identity is a way to authenticate and authorize access to workloads(or pods in k8s context) of cloud or external resources in a much more secure way.

- Before we understand workload identity, we can do a refresher of some keywords and Oauth2.0 and OIDC which powers it.



Authentication And Authorization

Let us understand some keywords…



Entity

Something which physically exists e.g individual or service

Identity

Set of attributes of the entity to uniquely identify

Authentication

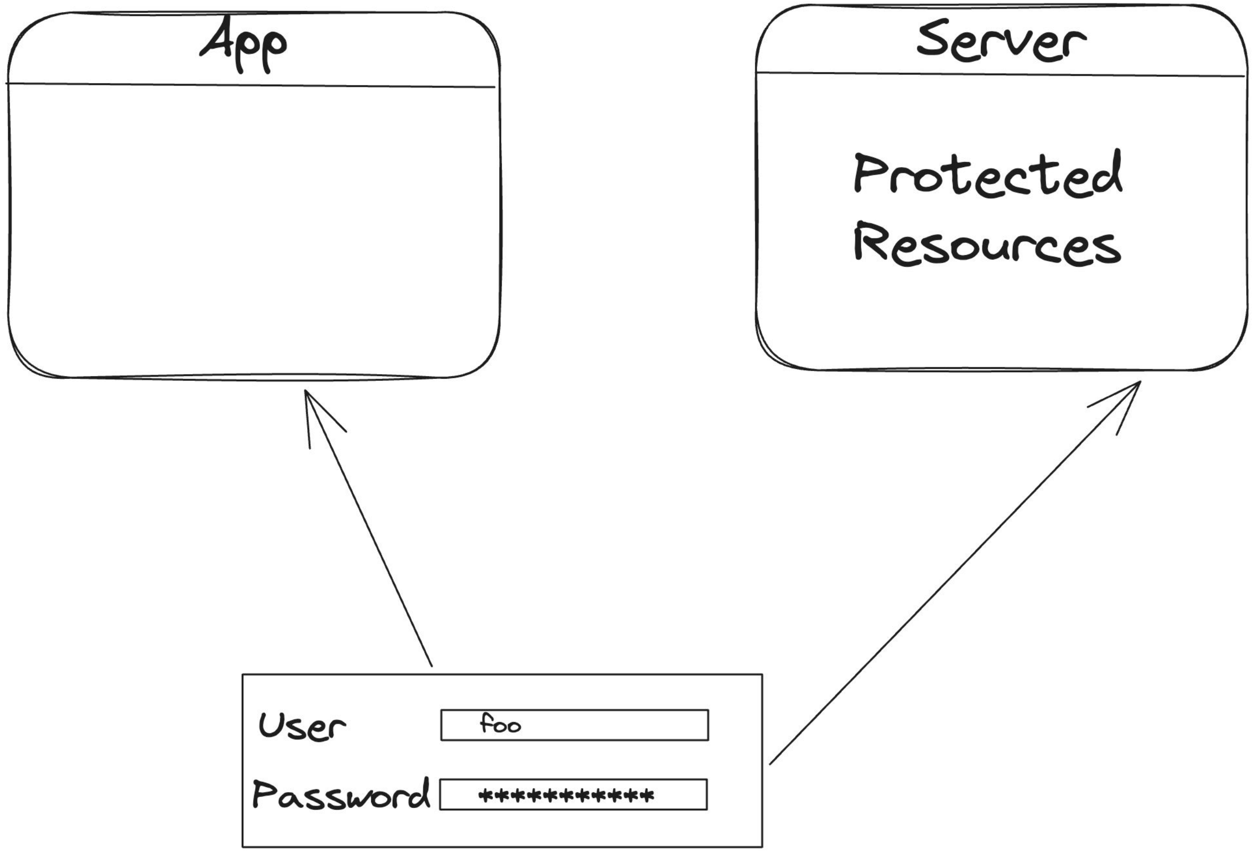
Way of confirming the identity of an entity

Authorization

Way of verifying what an entity can or cannot access.

Authentication And Authorization

- Quick look into traditional model…



Drawbacks

- Server need to support password authentication.
- Providing password to third party apps can be risky.
- No way to restricting access to only selected resources.
- No way to revoking access to an individual third party, need to revoke for all.

Authentication And Authorization

- Oauth2.0 addresses the limitation…

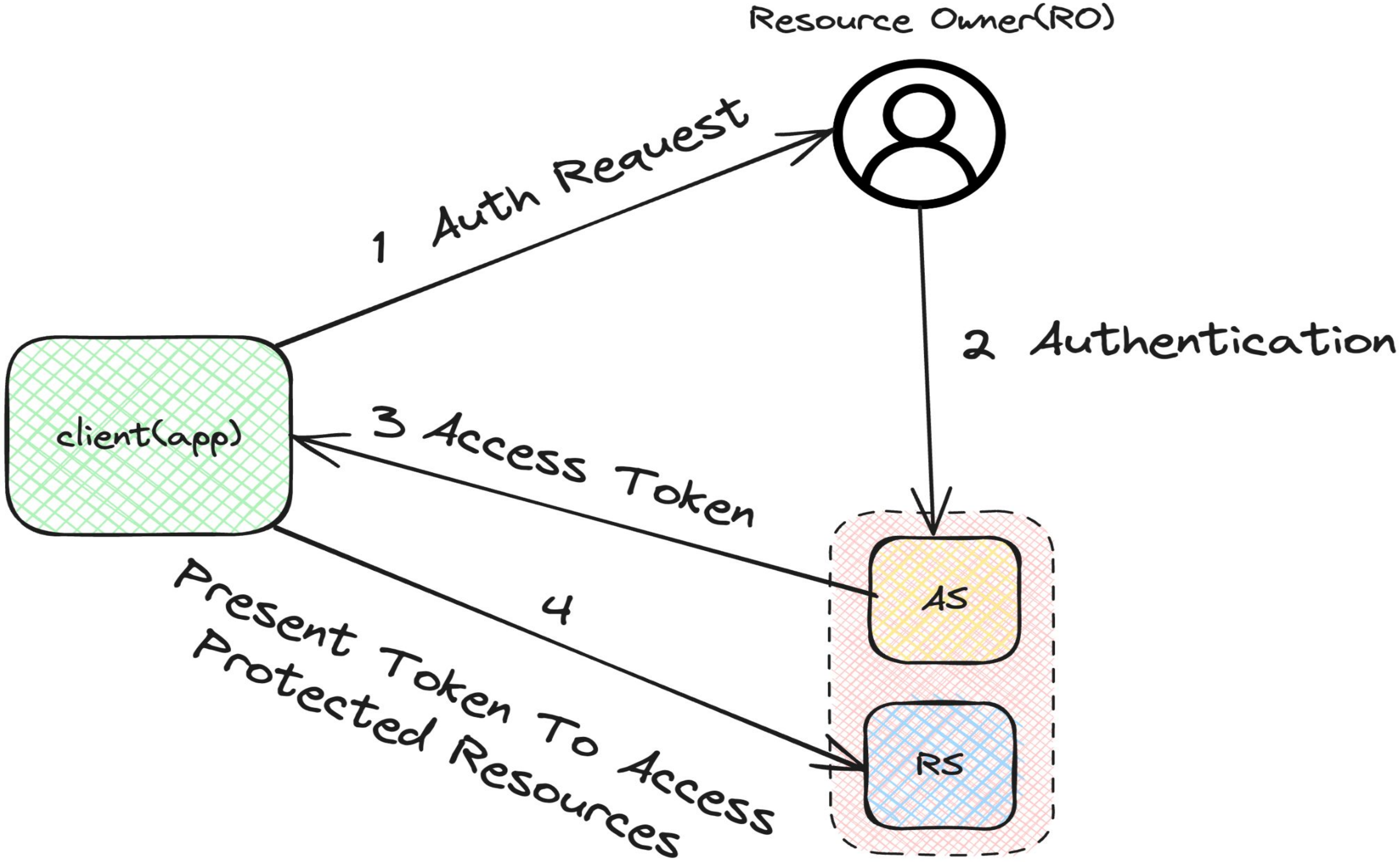
What is Oauth2.0

- A framework for delegated authority.
- Captured in RFC-6749
- No way to restricting access to only selected resources.
- Does not define authentication.

Oauth2.0 Roles

- Resource Owner (RO)
- Resource Server (RS)
- Client
- Authorization Server (AS)

Authentication And Authorization

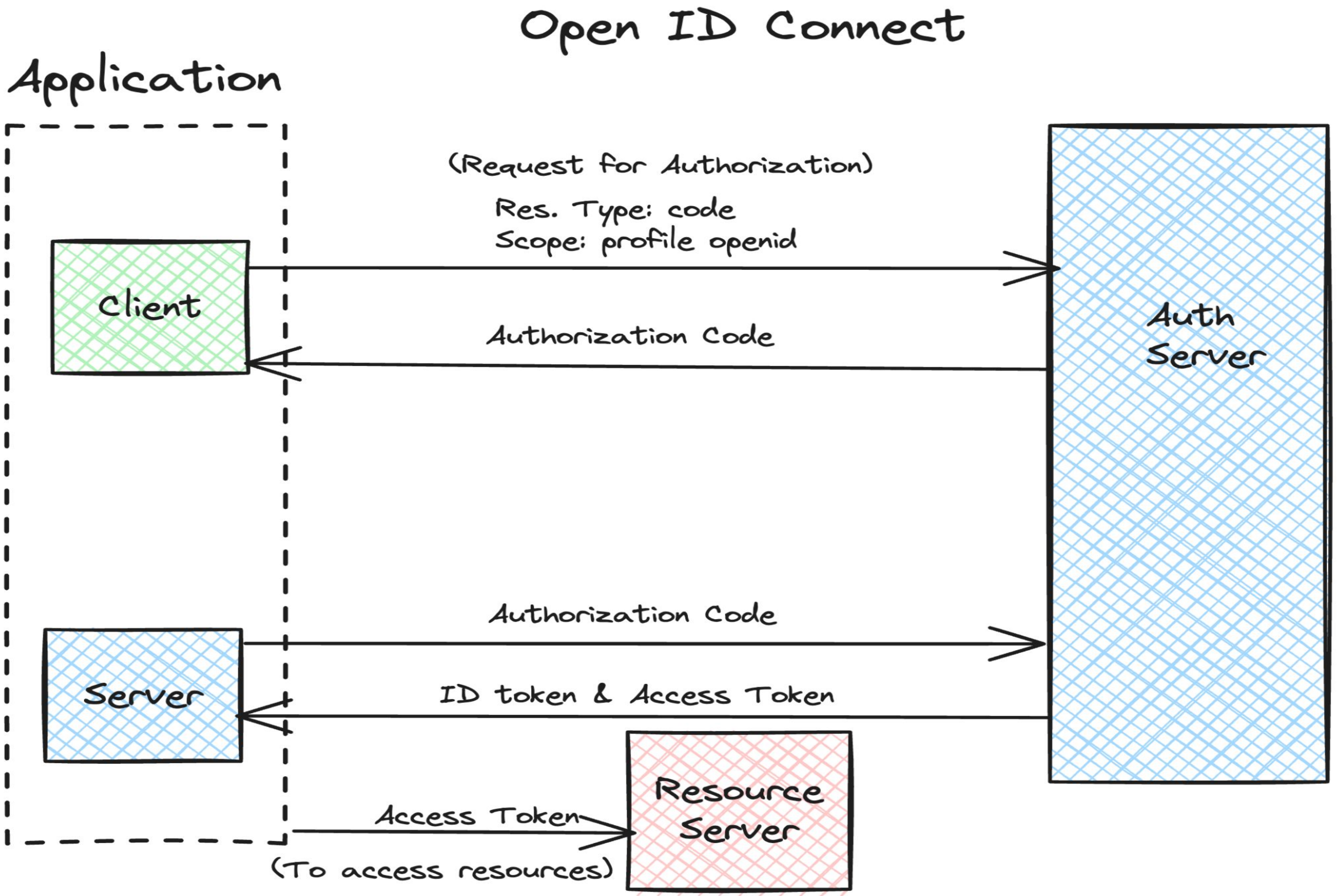


Authentication And Authorization

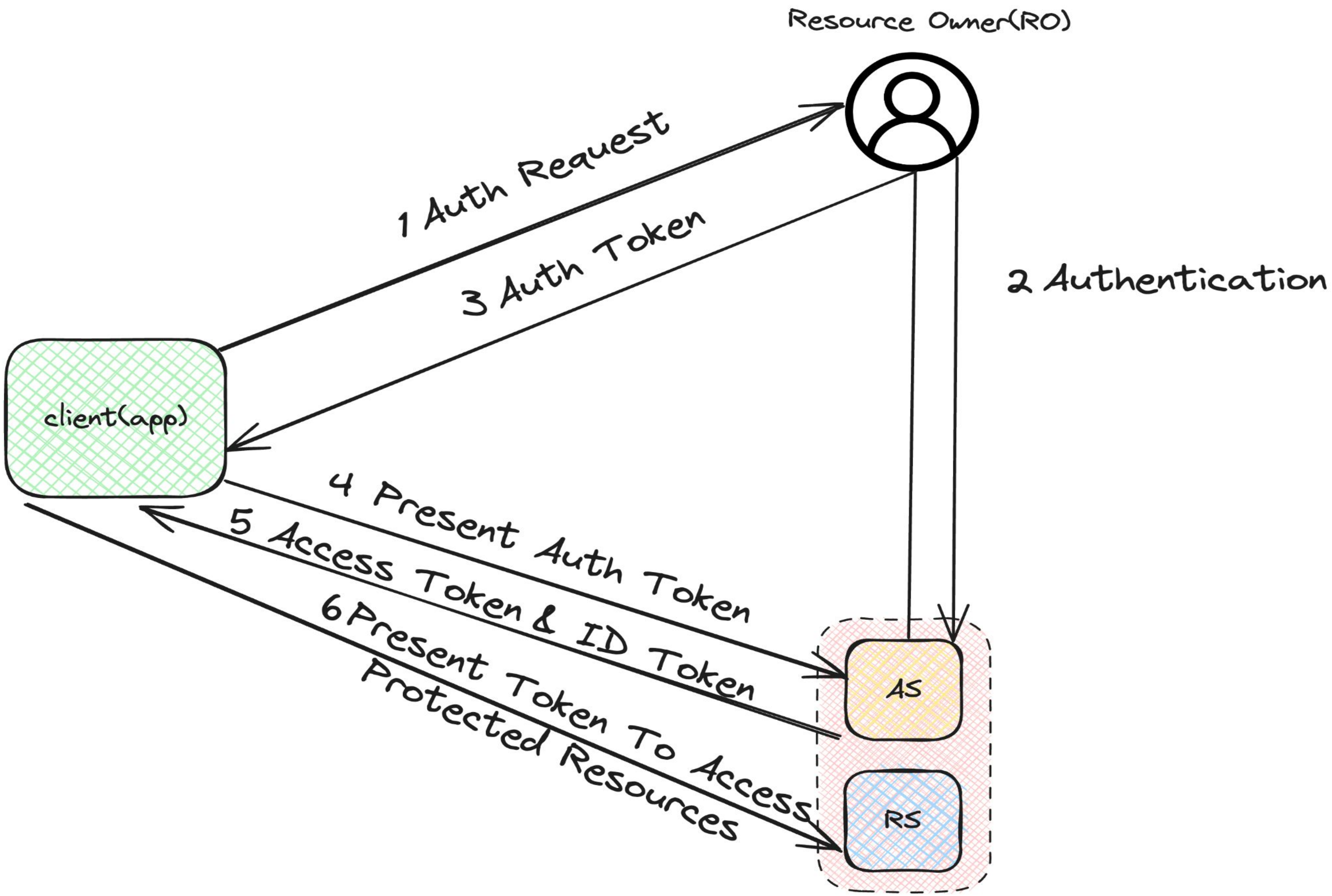
Open ID connect is a standard added on top of Oauth2.0

- Oauth2.0 does not define authentication mechanism.
- OIDC provides a method for user authentication.
- OIDC introduced ‘ID tokens’ in form of JWT.
- The JWT contains set of claims used to identify user e.g. email, name etc

Authentication And Authorization



Authentication And Authorization



Workload Identity

A **workload** is an application running on Kubernetes.

- Has a unique name within a namespace
- References an image that contains the actual code

Use cases for Workload Identity

- Authenticating with the Kubernetes API Server
- Authenticating communication between Kubernetes workloads
- Access to external protected resources

Workload Identity Options

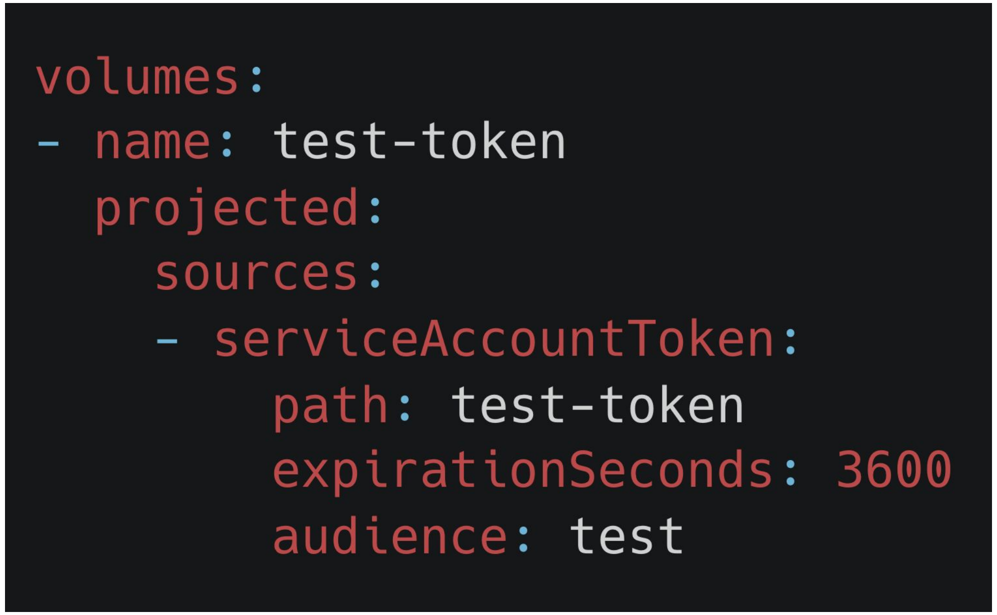
- Kubernetes Certificates API
- Service Mesh
- SPIFFE/SPIRE
- Kubernetes Service Account Token

Kubernetes Service Account Token

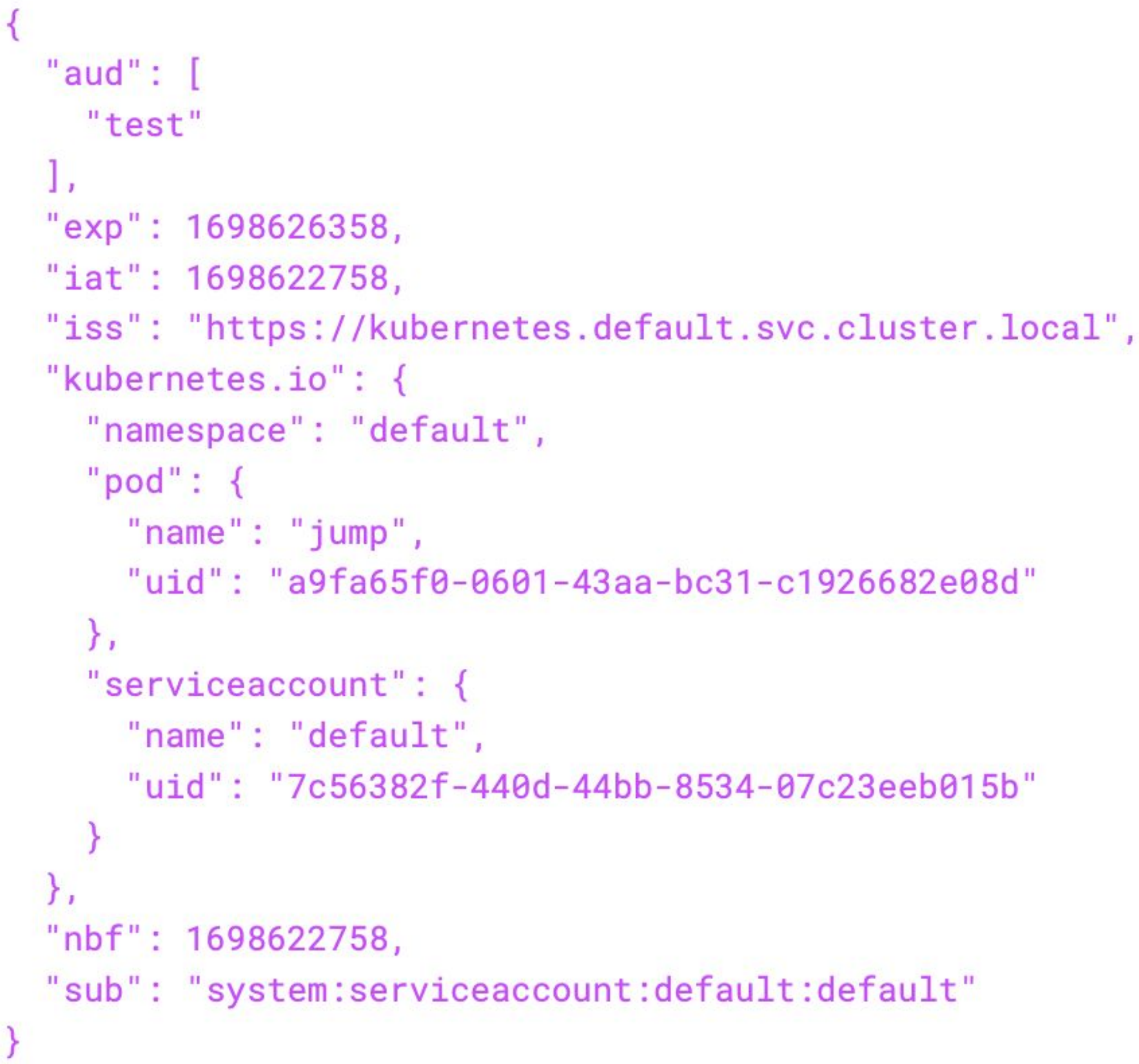
- Default Service Account Token
  - Automatically generated
  - **Not that secure**

- Projected Service Account Token

Projected Service Account Token

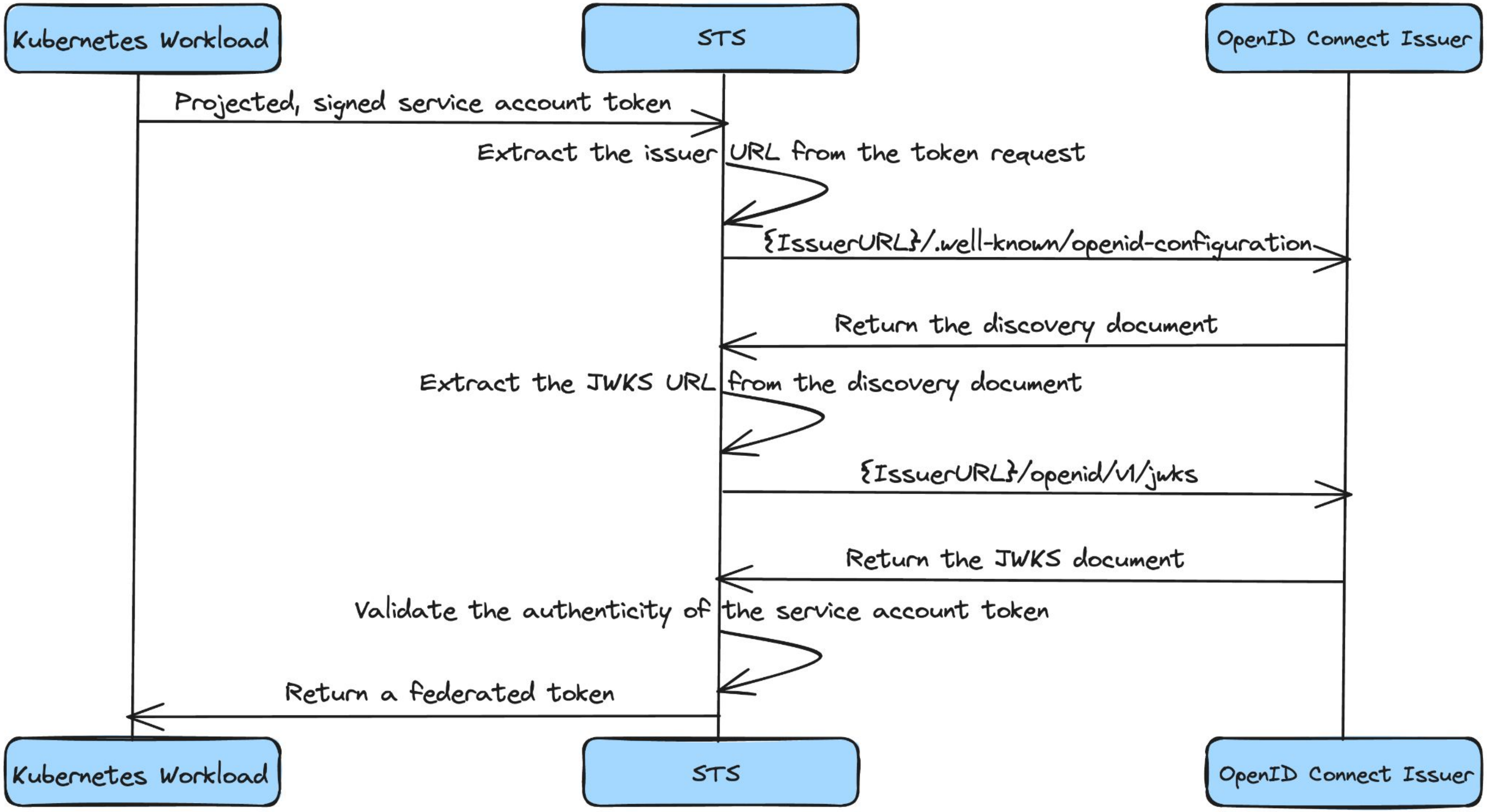


```yaml
volumes:
- name: test-token
  projected:
    sources:
    - serviceAccountToken:
        path: test-token
        expirationSeconds: 3600
        audience: test
```



```json
{
  "aud": [
    "test"
  ],
  "exp": 1698626358,
  "iat": 1698622758,
  "iss": "https://kubernetes.default.svc.cluster.local",
  "kubernetes.io": {
    "namespace": "default",
    "pod": {
      "name": "jump",
      "uid": "a9fa65f0-0601-43aa-bc31-c1926682e08d"
    },
    "serviceaccount": {
      "name": "default",
      "uid": "7c56382f-440d-44bb-8534-07c23eeb015b"
    }
  },
  "nbf": 1698622758,
  "sub": "system:serviceaccount:default:default"
}
```

Workload Identity Federation



Kubernetes Workload

STS

OpenID Connect Issuer

Projected, signed service account token

Extract the issuer URL from the token request

{IssuerURL}/.well-known/openid-configuration

Return the discovery document

Extract the JWKS URL from the discovery document

{IssuerURL}/openid/v1/jwks

Return the JWKS document

Validate the authenticity of the service account token

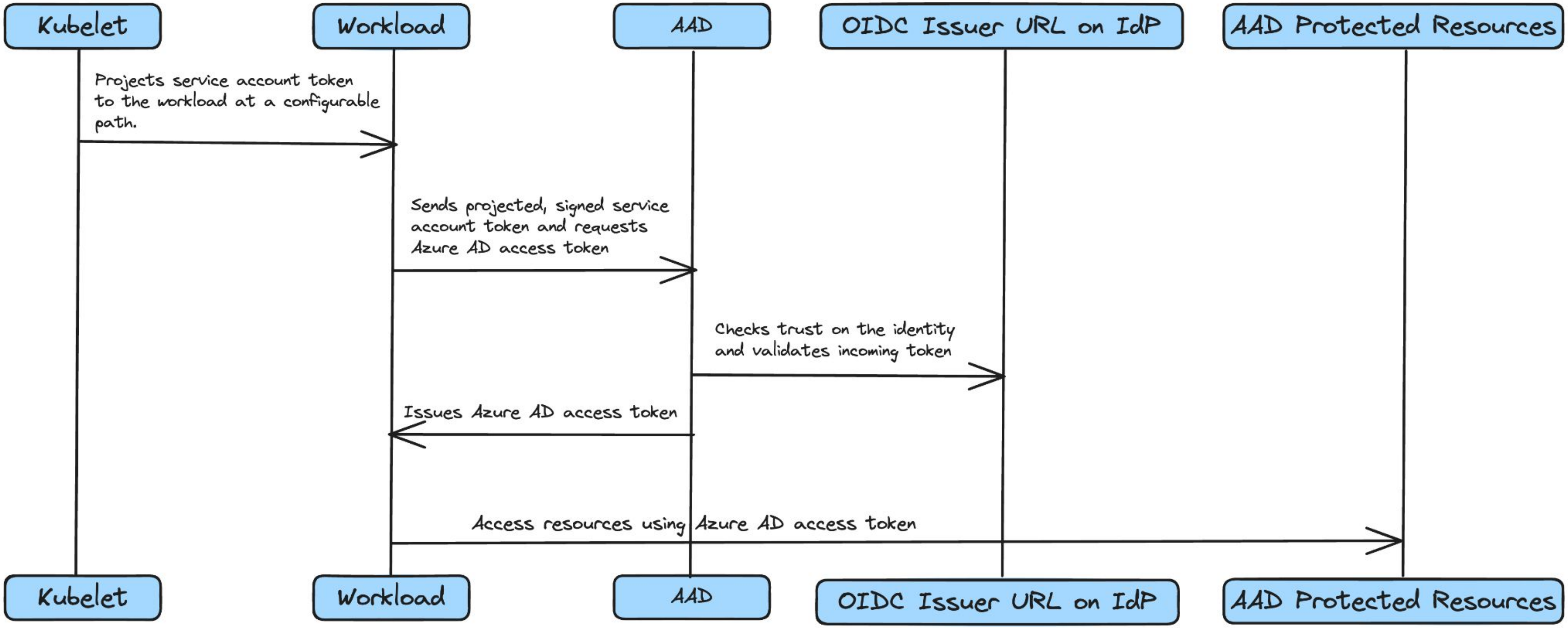
Return a federated token

Kubernetes Workload

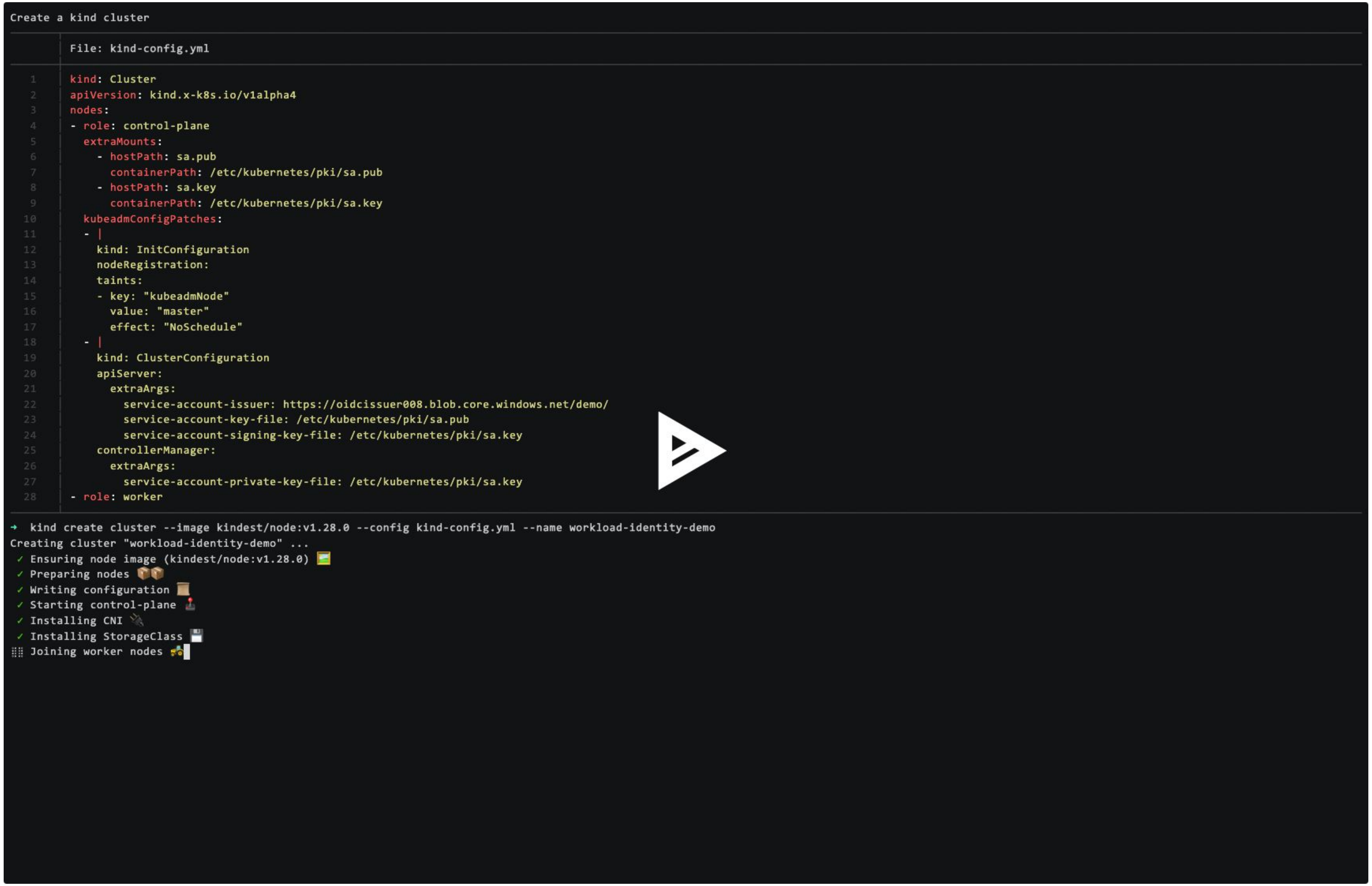
STS

OpenID Connect Issuer

Workload Identity Federation


Kubelet
Workload
AAD
OIDC Issuer URL on IdP
AAD Protected Resources
Projects service account token to the workload at a configurable path.
Sends projected, signed service account token and requests Azure AD access token
Checks trust on the identity and validates incoming token
Issues Azure AD access token
Access resources using Azure AD access token
Kubelet
Workload
AAD
OIDC Issuer URL on IdP
AAD Protected Resources

Demo



Create a kind cluster

File: kind-config.yml

```yaml
kind: Cluster
apiVersion: kind.x-k8s.io/v1alpha4
nodes:
- role: control-plane
  extraMounts:
    - hostPath: sa.pub
      containerPath: /etc/kubernetes/pki/sa.pub
    - hostPath: sa.key
      containerPath: /etc/kubernetes/pki/sa.key
  kubeadmConfigPatches:
  - |
    kind: InitConfiguration
    nodeRegistration:
    taints:
    - key: "kubeadmNode"
      value: "master"
      effect: "NoSchedule"
  - |
    kind: ClusterConfiguration
    apiServer:
      extraArgs:
        service-account-issuer: https://oidcissuer008.blob.core.windows.net/demo/
        service-account-key-file: /etc/kubernetes/pki/sa.pub
        service-account-signing-key-file: /etc/kubernetes/pki/sa.key
    controllerManager:
      extraArgs:
        service-account-private-key-file: /etc/kubernetes/pki/sa.key
- role: worker
```

```
➜  kind create cluster --image kindest/node:v1.28.0 --config kind-config.yml --name workload-identity-demo
Creating cluster "workload-identity-demo" ...
 ✓ Ensuring node image (kindest/node:v1.28.0) 🖼
 ✓ Preparing nodes 📦📦
 ✓ Writing configuration 📜
 ✓ Starting control-plane 🕹️
 ✓ Installing CNI 🔌
 ✓ Installing StorageClass 💾
⣿ Joining worker nodes 🚜
```

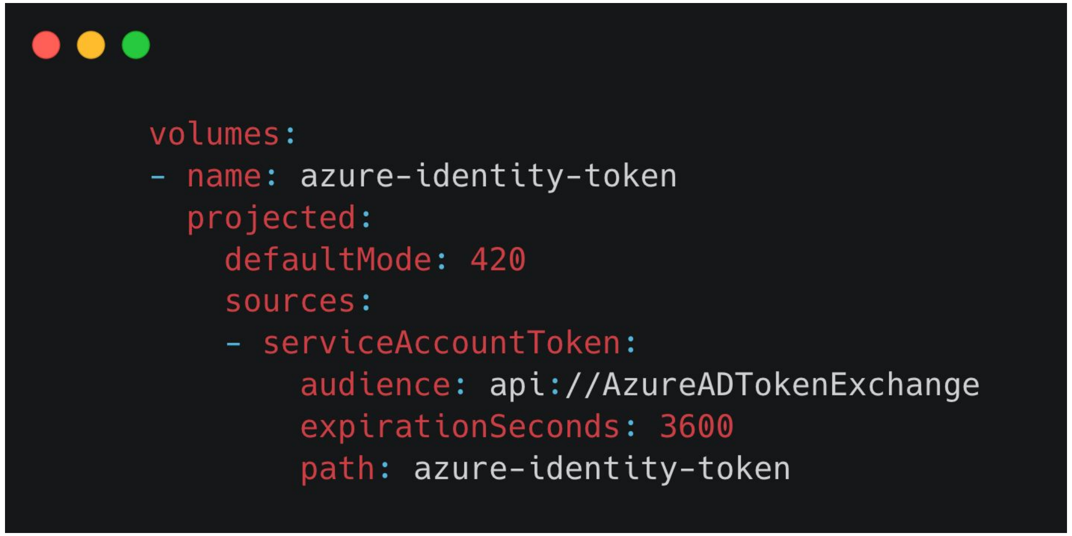
Other usages in Kubernetes

- Cluster API Provider Azure
- Token requests support in Kubernetes CSI
  - Workload Identity in Secrets Store CSI Driver

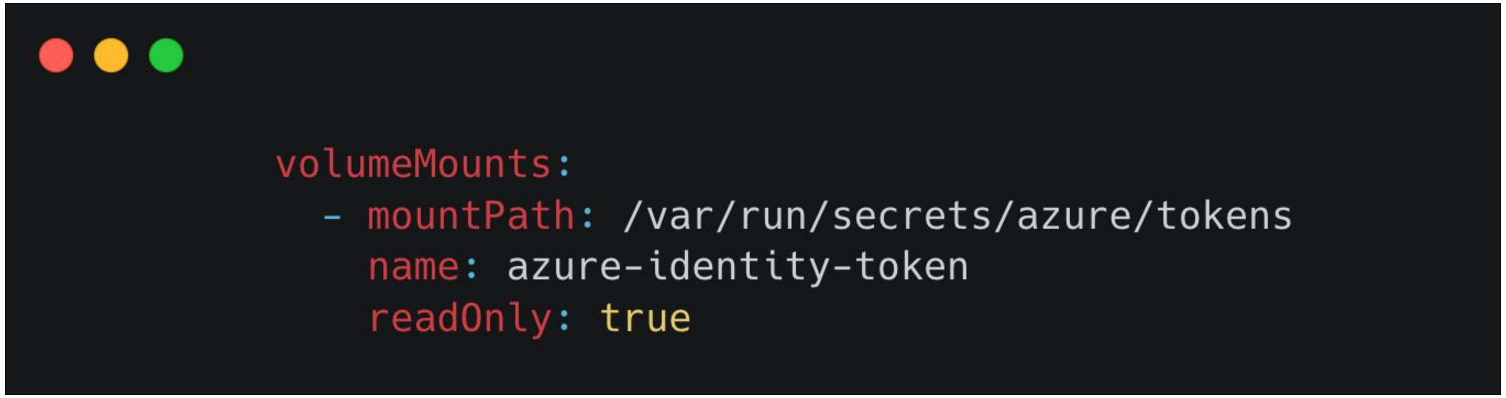
Workload Identity in CAPZ

- Cluster API Provider Azure is a SIG Cluster Lifecycle sub project which helps provision k8s clusters on Azure in a declarative way.
- Using service principal to access Azure resources is roughly using user/passwords.

Workload Identity in CAPZ



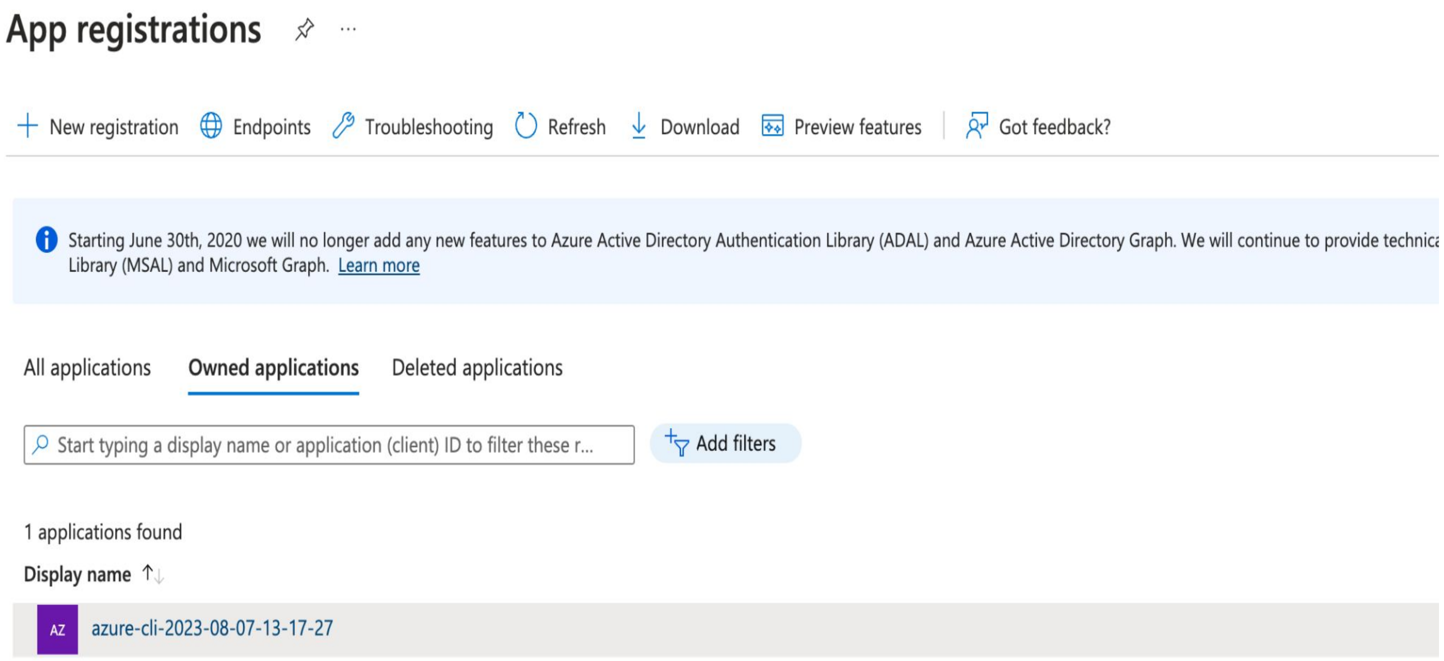
```yaml
volumes:
- name: azure-identity-token
  projected:
    defaultMode: 420
    sources:
    - serviceAccountToken:
        audience: api://AzureADTokenExchange
        expirationSeconds: 3600
        path: azure-identity-token
```



```yaml
volumeMounts:
  - mountPath: /var/run/secrets/azure/tokens
    name: azure-identity-token
    readOnly: true
```

Workload Identity in CAPZ

Create an app on azure.



App registrations

Starting June 30th, 2020 we will no longer add any new features to Azure Active Directory Authentication Library (ADAL) and Azure Active Directory Graph. We will continue to provide technica
Library (MSAL) and Microsoft Graph. Learn more

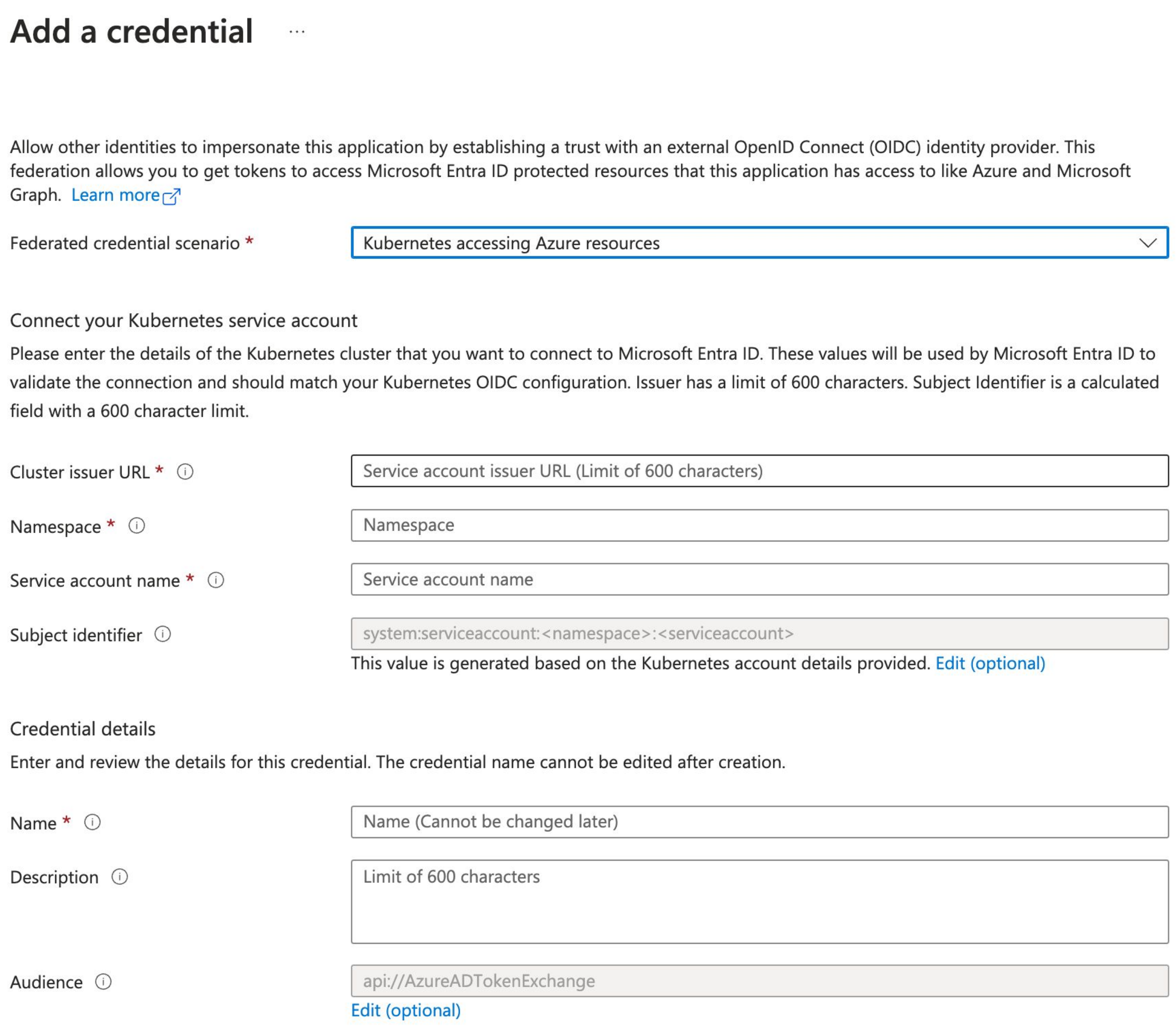
All applications
Owned applications
Deleted applications
Start typing a display name or application (client) ID to filter these r...
Add filters

1 applications found

Display name ↑↓
AZ azure-cli-2023-08-07-13-17-27

Workload Identity in CAPZ

Create federated credential



Add a credential

Allow other identities to impersonate this application by establishing a trust with an external OpenID Connect (OIDC) identity provider. This federation allows you to get tokens to access Microsoft Entra ID protected resources that this application has access to like Azure and Microsoft Graph. Learn more

Federated credential scenario * Kubernetes accessing Azure resources

Connect your Kubernetes service account

Please enter the details of the Kubernetes cluster that you want to connect to Microsoft Entra ID. These values will be used by Microsoft Entra ID to validate the connection and should match your Kubernetes OIDC configuration. Issuer has a limit of 600 characters. Subject Identifier is a calculated field with a 600 character limit.

Cluster issuer URL * ⓘ	Service account issuer URL (Limit of 600 characters)
Namespace * ⓘ	Namespace
Service account name * ⓘ	Service account name
Subject identifier ⓘ	system:serviceaccount:<namespace>:<serviceaccount> This value is generated based on the Kubernetes account details provided. Edit (optional)

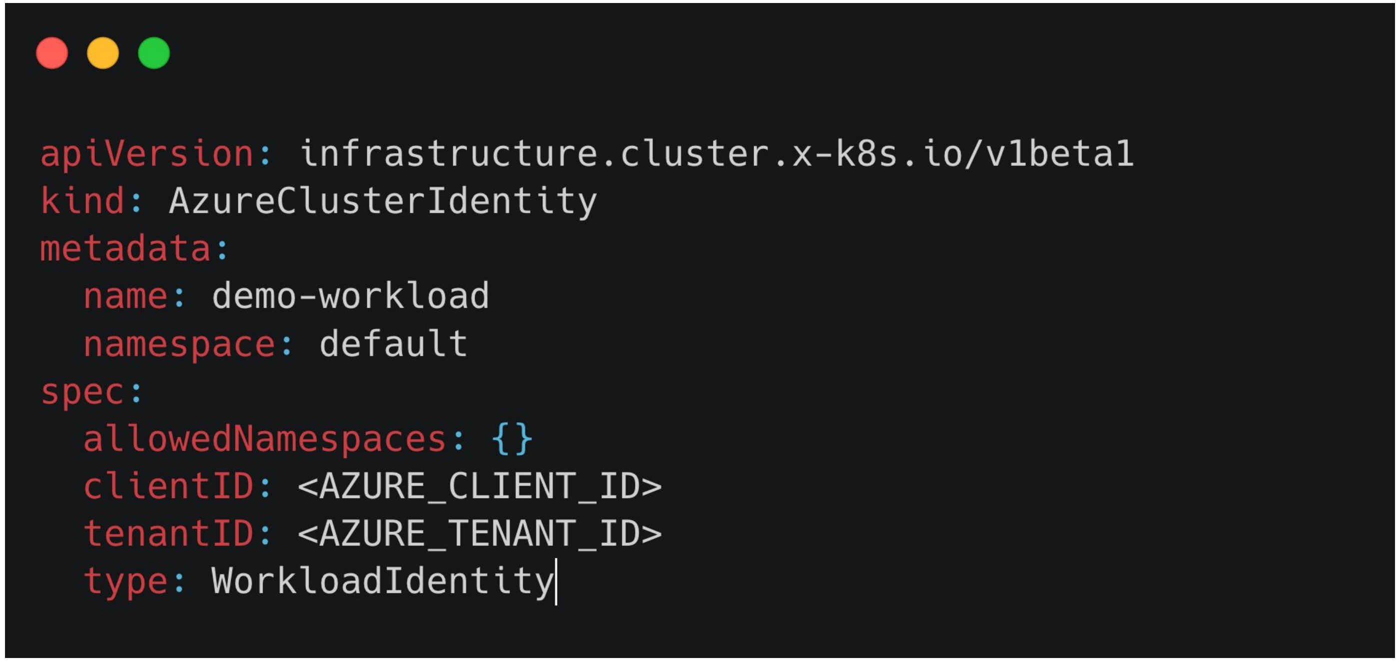
Credential details

Enter and review the details for this credential. The credential name cannot be edited after creation.

Name * ⓘ	Name (Cannot be changed later)
Description ⓘ	Limit of 600 characters
Audience ⓘ	api://AzureADTokenExchange Edit (optional)

Workload Identity in CAPZ

Supply the app client ID to CAPZ

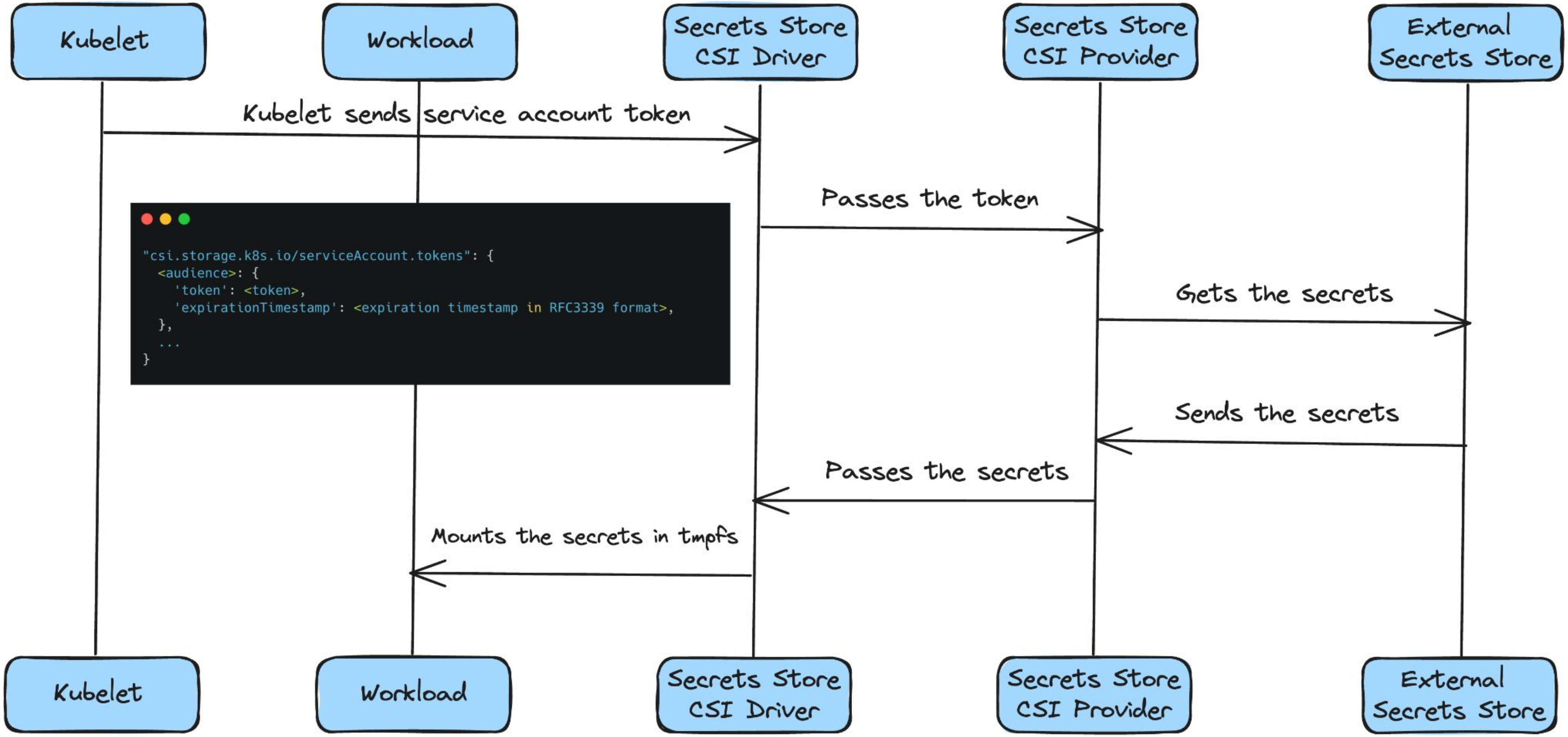


```
apiVersion: infrastructure.cluster.x-k8s.io/v1beta1
kind: AzureClusterIdentity
metadata:
  name: demo-workload
  namespace: default
spec:
  allowedNamespaces: {}
  clientID: <AZURE_CLIENT_ID>
  tenantID: <AZURE_TENANT_ID>
  type: WorkloadIdentity
```

Secrets Store CSI Driver

- SIG Auth subproject
- Container Storage Interface (CSI) Storage Driver
- Loads secret into tmpfs mount

Secrets Store CSI Driver



```
"csi.storage.k8s.io/serviceAccount.tokens": {
  <audience>: {
    'token': <token>,
    'expirationTimestamp': <expiration timestamp in RFC3339 format>,
  },
  ...
}
```

Workload Identity

Resources

- Service Account token volume projection
- Service Account issuer discovery
- Workload Identity in GKE
- Workload Identity In CAPZ
  - CAPZ Workload Identity
  - Azure Workload Identity
- Secrets Store CSI Driver

Q & A

**Please scan the QR Code above
to leave feedback on this session**