# Controle de Acesso do Kubernetes

(Usando Active Directory)

# Quem sou?

IT Architect (Unicred do Brasil)

Technical Trainer (TargetTrust)

- OpenSource
- Cloud Native Solutions

# Agenda

- Jornada da Implementação
- Marco teórico
- Detalhar  o laboratório
- Demo

# Jornada da Implementação

- Avaliar o problema e justificar a solução
- Estudar os conceitos dos protocolos
- Criar um laboratório que nos permita:
  - Implementar a solução
  - Verificar os conceitos
  - Fazer a simulação de problemas
- Implementar em TST/HLG e finalmente em PRD

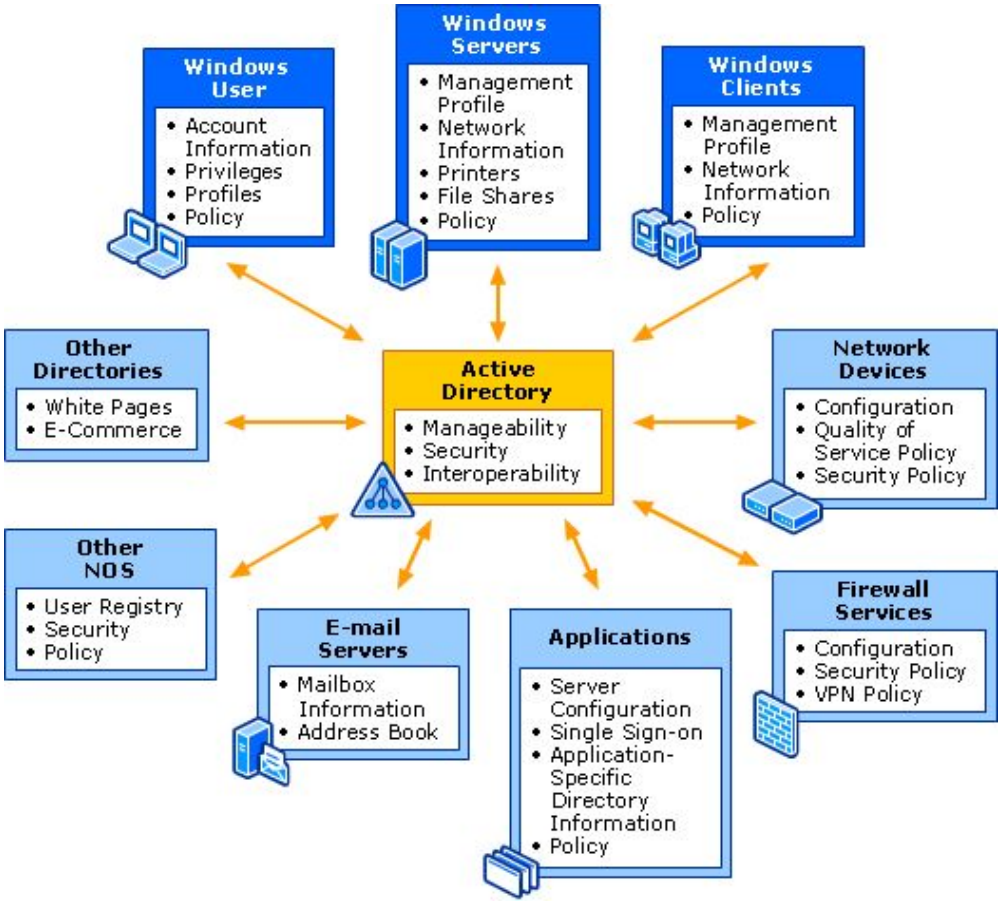
# Motivação

- Integrar o Active Directory ao Kubernetes para fornecer autenticação **(AuthN)** e autorização **(AuthZ)** centralizada
- Usar novos métodos para tratamento de identidade
  - OpenID Connect
  - OIDC
  - Oauth2
- Delegar os processos de assignação de credenciais para outras áreas

# Marco Teórico

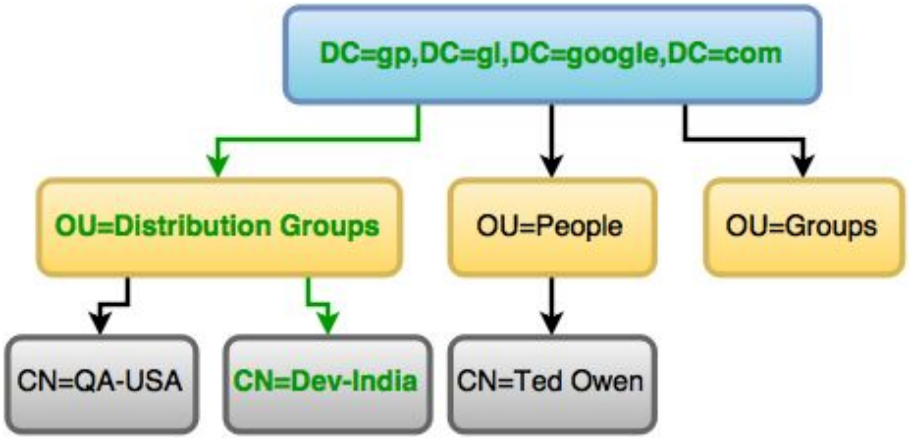
- Active Directory - Directory Services
- Oauth2 - OIDC - OpenID Connect
- Kubernetes AuthN - AuthZ

# Active Directory - Directory Services



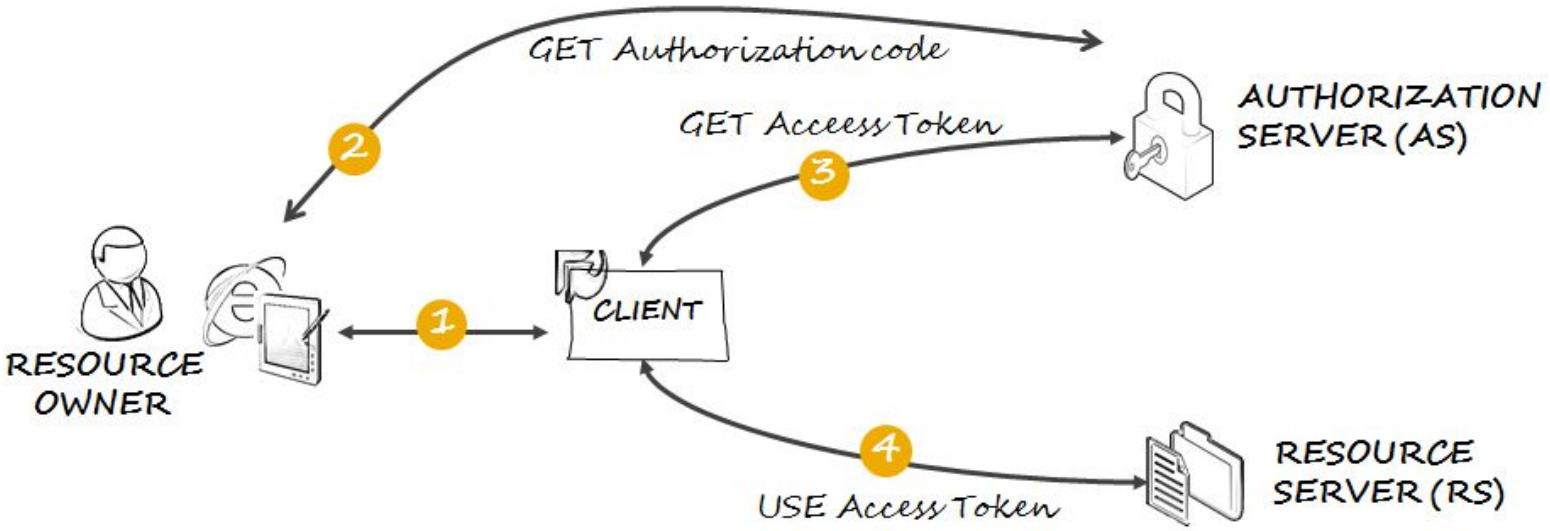
# Active Directory - X.500 Directory Specification

CN = Common Name
OU = Organizational Unit
DC = Domain Component

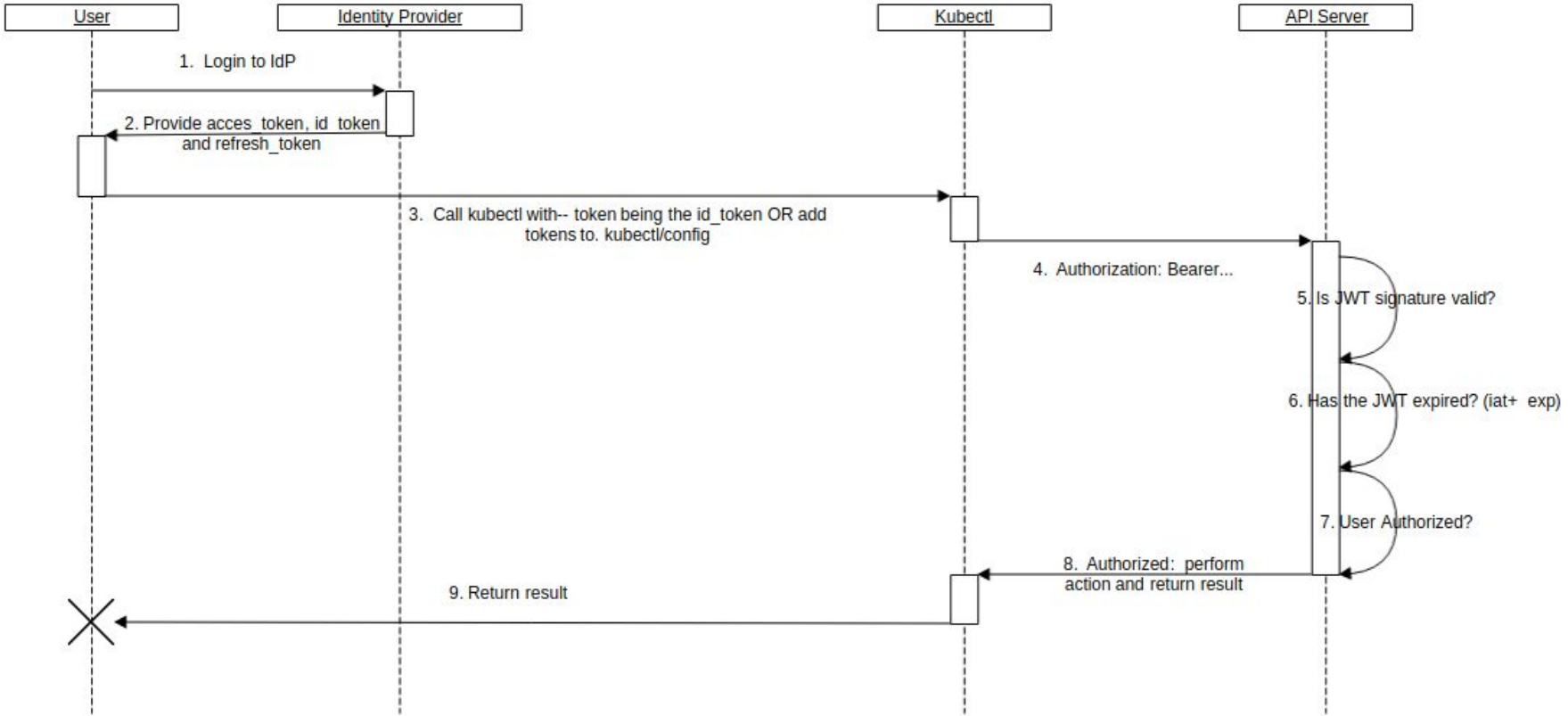



"CN=Dev-India,OU=Distribution Groups,DC=gp,DC=gl,DC=google,DC=com"

# Oauth2 - Authorization Code Flow



# Kubernetes - OpenID Connect Tokens



# Que precisa o Kubernetes?

- `--oidc-issuer-url=URL` identifies the URL of a provider, which allows the API server to discover public signing keys.
- `--oidc-client-id=ID` is the client ID for verifying signatures of the JSON web tokens.
- `--oidc-username-claim=email` specifies what parameter to use as a username.
- `--oidc-groups-claim=groups` specifies the parameter to get the groups.

Example:

```
--oidc-issuer-url=https://keycloak.enciso.website/auth/realms/k8s \
--oidc-client-id=oidckube \
--oidc-username-claim=email \
--oidc-groups-prefix=oidc: \
--oidc-groups-claim=groups \
```

# Kubernetes - AuthN AuthZ

- **Authentication (AuthN)** determines the identity of a user, a server, or a client.

  Certificates
  Tokens
  Static Password
  **OpenID Connect**

- **Authorization (AuthZ)** determines if a user, a server, or a client has permission to execute specific tasks.

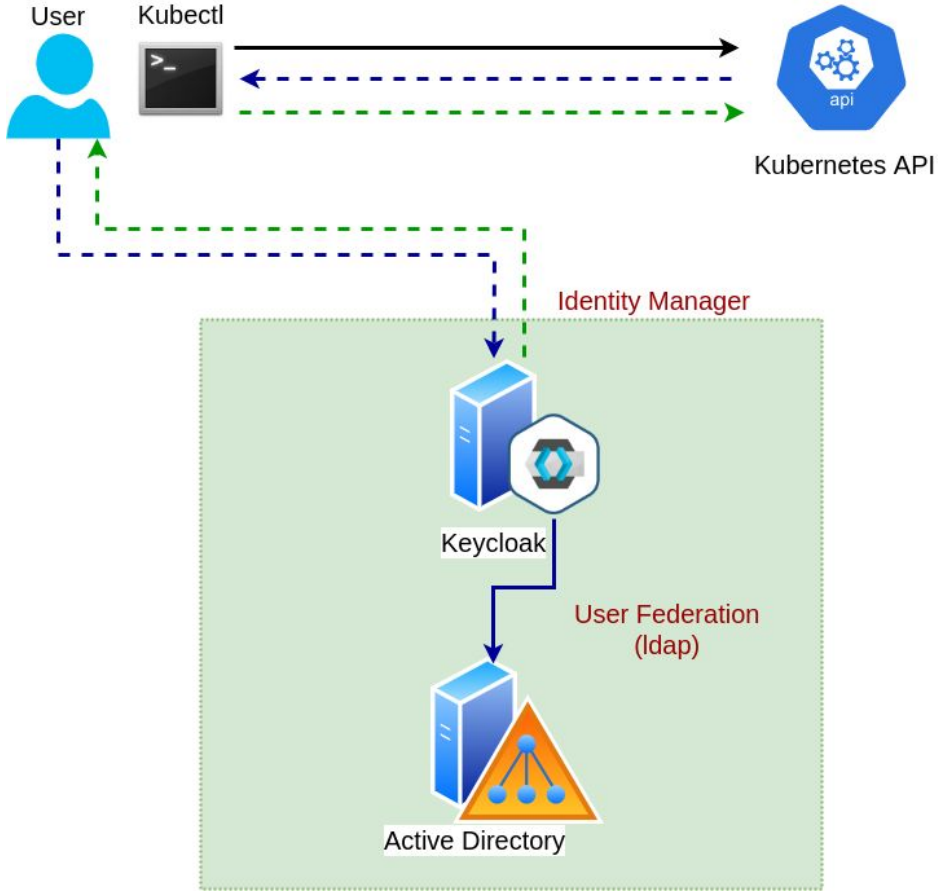
  ABAC
  Node
  Webhook
  **RBAC**

# Laboratório

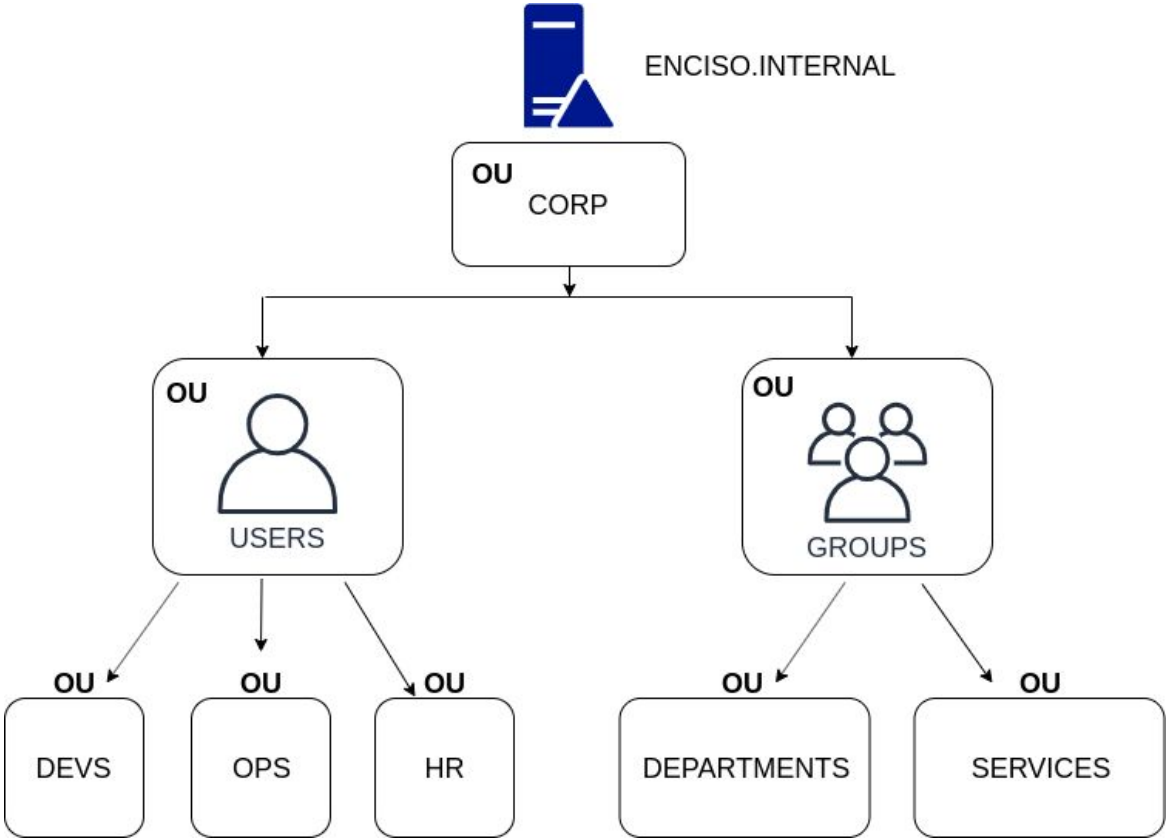
# Laboratório - Requerimentos

**SERVIÇOS**

- Kubernetes Cluster
- Identity Manager (Keycloak)
- Active Directory



# Active Directory - OU Structure



# Active Directory - CN (Dados)

**OU = USERS**

**OU**
DEVS

**OU**
OPS

**OU**
HR

**OU = GROUPS**

**OU**
DEPARTMENTS

CN = DEVS

CN = OPS

CN = HR

**OU**
SERVICES

CN = kubernetes-admins

CN = kubernetes-users

CN = jira-users

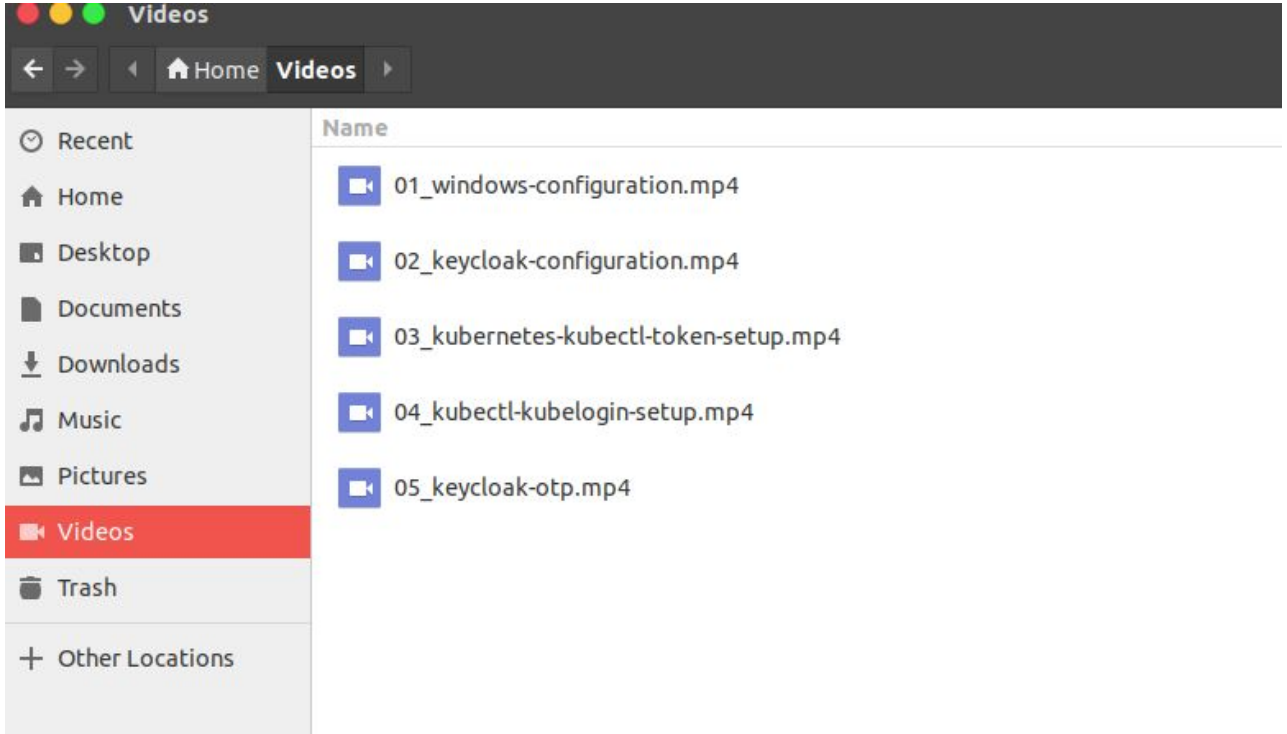
CN = gitlab-users

# Demo

Passos:

1. Criar os usuarios e grupos na estrutura do AD
2. Configurar o realm k8s no Keycloak
3. Verificar a geração de token no Keycloak
4. Configurar o kubernetes para o suporte OpenID
5. Configurar o kubectl com suporte OpenID
6. Configurar o Keycloak com OTP (One time password ou Dynamic Password)

# Videos


Videos
Home
Videos
Recent
Home
Desktop
Documents
Downloads
Music
Pictures
Videos
Trash
Other Locations
Name
01_windows-configuration.mp4
02_keycloak-configuration.mp4
03_kubernetes-kubectl-token-setup.mp4
04_kubectl-kubelogin-setup.mp4
05_keycloak-otp.mp4

# Video References

# Active Directory - Directory Services

# An Illustrated Guide to OAuth and OpenID Connect


An Illustrated Guide to OAuth and OpenID Connect

# References


Kubernetes day 2 - Operations authn authz with OIDC https://bit.ly/33x00w7

Active Directory - Structure https://www.youtube.com/watch?v=IFwek_OuYZ8&t=891s

Guide Oauth - OpenID Connect https://www.youtube.com/watch?v=t18YB3xDfXI

Kubucation - Kubernetes - Keycloak https://www.youtube.com/watch?v=gJ81eaGIN_I&t=202s

Kubelogin https://github.com/int128/kubelogin/

Oidckube - Template Keycloak https://github.com/mrbobbytables/oidckube