

SINGLE SIGN-ON FOR KUBERNETES

A look at OIDC and Pusher's journey to SSO

@JoelASpeed



@Pusher

WHO AM I?

Cloud Infrastructure Engineer, Pusher

@JoelASpeed

Joel.Speed@Pusher.com

joelspeed.co.uk



"WE SHOULD START USING RBAC"

@JoelASpeed



@Pusher

OUR PLATFORM



@JoelASpeed

PUSHER

@Pusher

GETTING STARTED (OUR DARK PAST)

One x509 Certificate.

One Identity.

30 Engineers.

WHAT DID WE WANT?

Individual user accounts

Group management

Scalable

UX

AUTHENTICATION OPTIONS

- X.509 Client Certs
- Static Token File
- Bootstrap Tokens
- Static Password File
- Service Account Tokens
- OpenID Connect Tokens
- Webhook Token Authentication
- Authenticating Proxy
- Keystone Password

Source: <https://kubernetes.io/docs/reference/access-authn-authz/authentication/>

AUTHENTICATION OPTIONS

- X.509 Client Certs
- ~~Static Token File~~
- ~~Bootstrap Tokens~~
- Static Password File
- ~~Service Account Tokens~~
- OpenID Connect Tokens
- Webhook Token Authentication
- Authenticating Proxy
- Keystone Password

Source: <https://kubernetes.io/docs/reference/access-authn-authz/authentication/>

AUTHENTICATION OPTIONS

- X.509 Client Certs
- Static Token File
- Bootstrap Tokens
- Static Password File
- Service Account Tokens
- OpenID Connect Tokens
- Webhook Token Authentication
- Authenticating Proxy
- Keystone Password

Source: <https://kubernetes.io/docs/reference/access-authn-authz/authentication/>

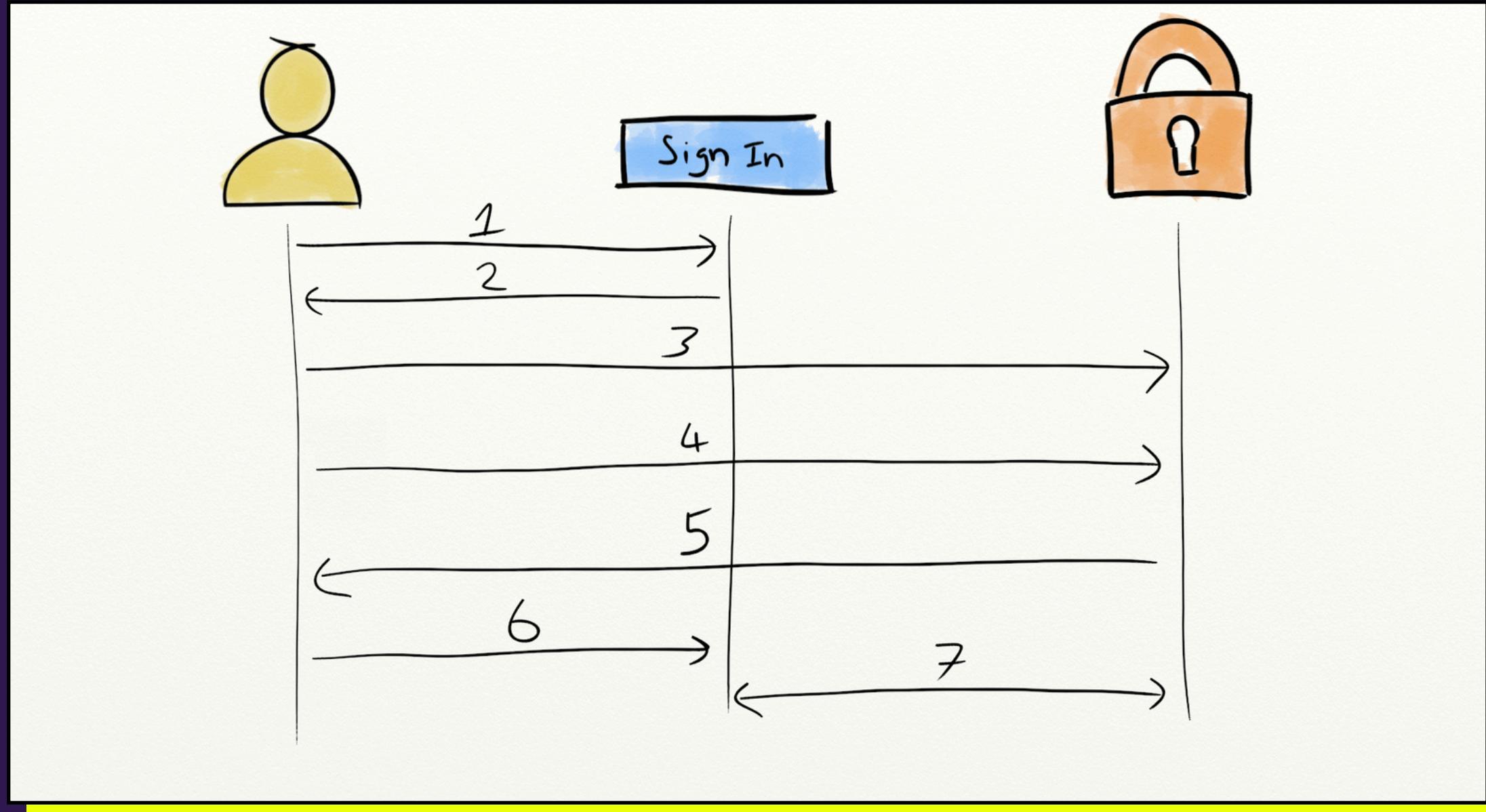
X.509 CLIENT CERTS

- Fixed lifetime. Cannot easily be revoked.
- Certificates must be signed by trusted CA.
- Self service is hard. Must verify CSR before signing certificate. How to manage users and groups?
- No Kubernetes Dashboard support
- Renewal is hard

OPEN ID CONNECT (OIDC)

- Fixed lifetime. Cannot easily be revoked (without control of the Identity Provider)
- Only a handful of providers (Google, Salesforce, Azure AD)
- Single Sign-On: Can re-use existing user accounts and groups
- Kubernetes Dashboard supports OIDC tokens
- Automatic refresh

AUTHENTICATION FLOW



1. Click Sign In
- 2/3. Redirect to Identity Provider
4. Enter username and password
- 5/6. Redirect back to the origin with authentication code
7. Origin server exchanges code for ID token

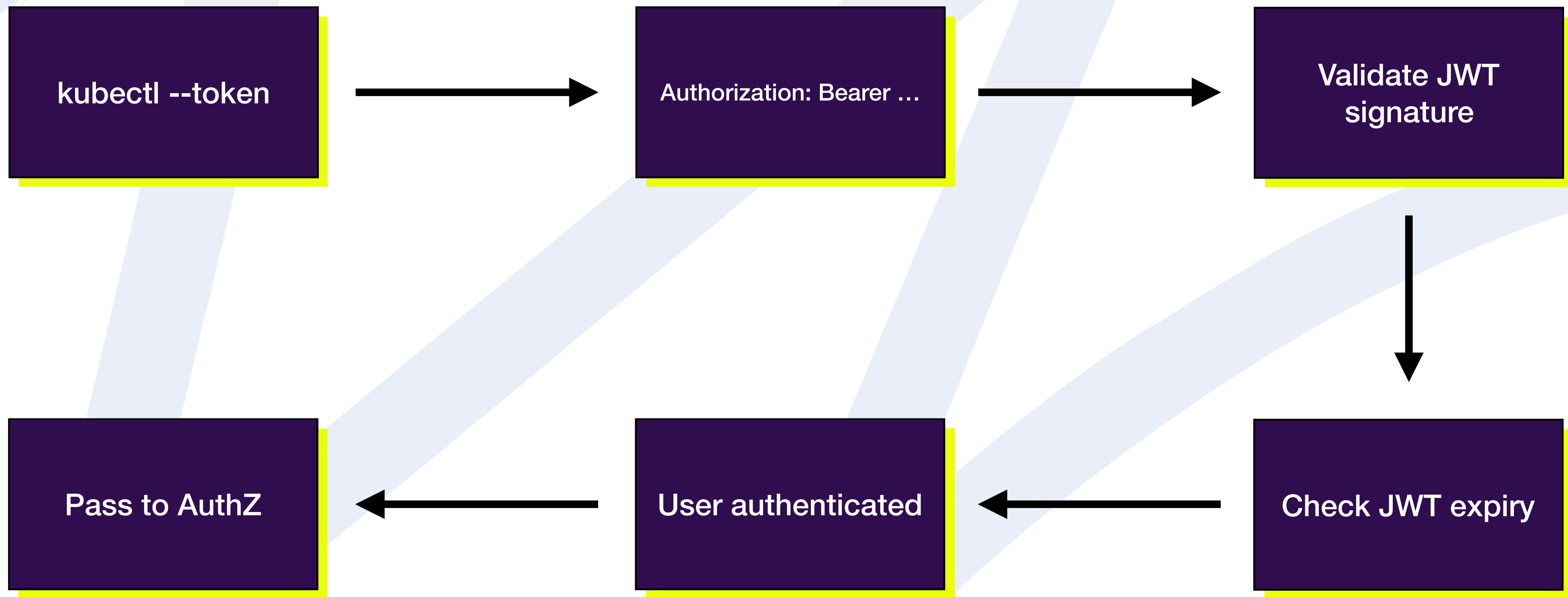
ID TOKENS (JWT)

<metadata>.<payload>.<signature>

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG91IiwiYWrtaw4iOnRydWV9.TJVA95OrM7E2cBab30RMHrHDcEfjoxYZgeFONFh7HgQ

```
"iss": "https://auth.example.com/dex",
"sub": "ChUXMDk0MzA2...",
"aud": "kubernetes",
"exp": 1519123284,
"iat": 1519036884,
"at_hash": "X2G33w55vEm39VwyOMMjzg",
"email": "joel.speed@pusher.com",
"email_verified": true,
"groups": [
    "group1@pusher.com",
    "group2@pusher.com"
],
"name": "Joel Speed"
```

USING ID TOKENS



OPEN ID CONNECT (OIDC)

- Fixed lifetime. Cannot easily be revoked (without control of the Identity Provider)
- Only a handful of providers (Google, Salesforce, Azure AD)
- Single Sign-On: Can re-use existing user accounts and groups
- Kubernetes Dashboard supports OIDC tokens
- Automatic refresh

INTRODUCING DEX



Dex is an identity service that uses OpenID Connect to drive authentication for other apps.

LDAP, GitHub, SAML 2.0, GitLab, Open ID Connect, LinkedIn, Microsoft, AuthProxy

Image credits: Kubernetes, CoreOS, Google

@JoelASpeed

 PUSHER

@Pusher

WHY DEX IN THE MIDDLE?

@JoelASpeed



@Pusher

CONTROL OF TOKEN LIFETIME

@JoelASpeed



18

@Pusher

REVOKE TOKENS

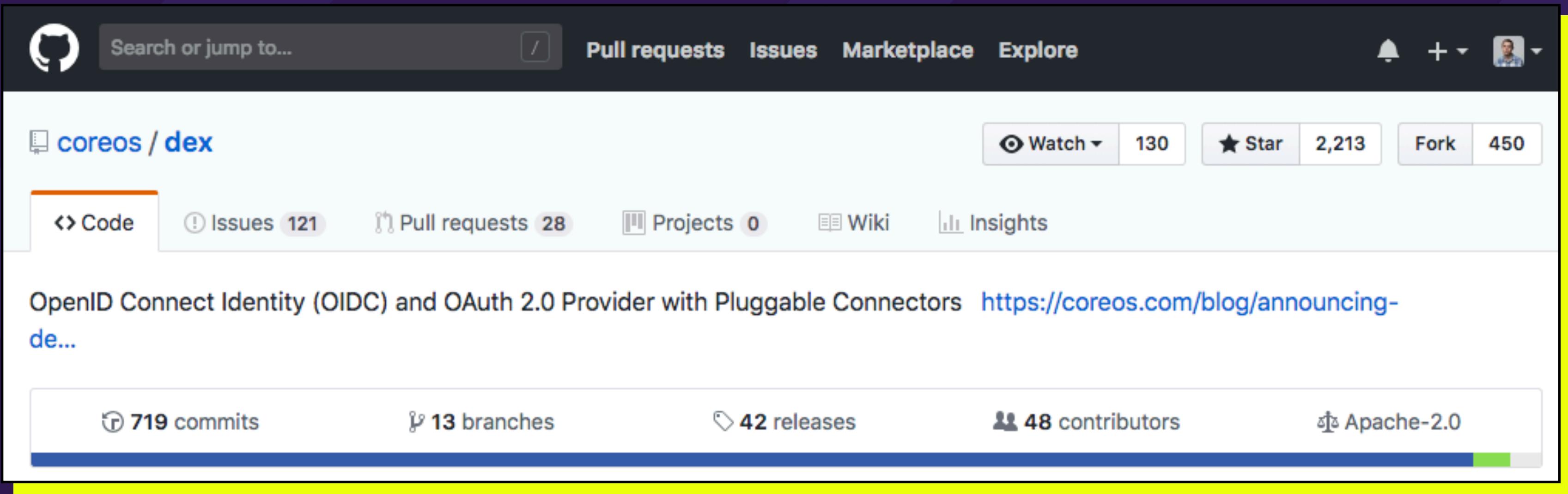
/DEX/.WELL-KNOWN/OPENID-CONFIGURATION

```
{  
  "issuer": "https://auth.domain.com/dex",  
  "authorization_endpoint": "https://auth.domain.com/dex/  
auth",  
  "token_endpoint": "https://auth.domain.com/dex/token",  
  "jwks_uri": "https://auth.domain.com/dex/keys",  
  "response_types_supported": [  
    "code"  
,  
    "subject_types_supported": [  
      "public"  
,  
      "id_token_signing_alg_values_supported": [  
        "RS256"  
,  
        "scopes_supported": [  
          "openid",  
          "email",  
          "groups",  
          "profile",  
          "offline_access"  
,  
          . . .  
        ]  
      ]  
    ]  
  }  
}
```

ADD NEW CLIENTS

```
staticClients:  
- id: kubernetes  
  redirectURIs:  
  - 'http://127.0.0.1:5555/callback'  
  name: 'Kubernetes API'  
  secret: <INSERT_CLIENT_SECRET_HERE>
```

PLUGGABLE



coreos / dex

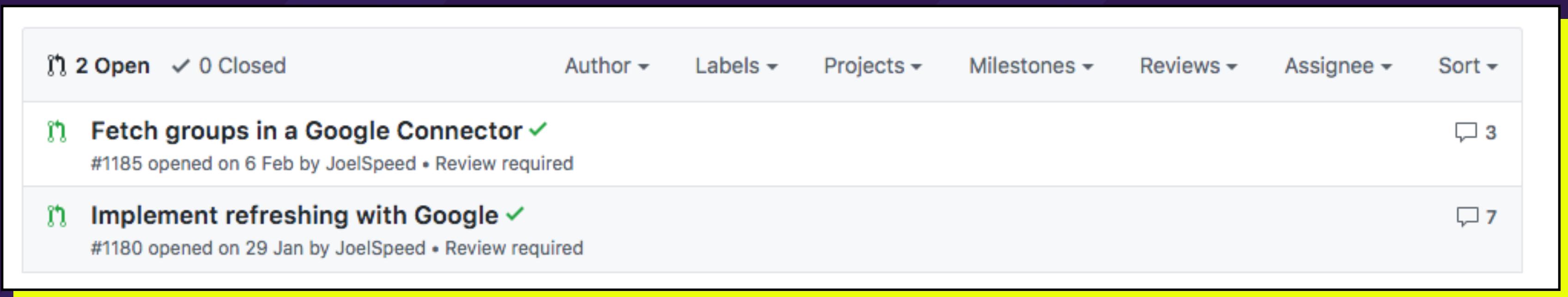
Pull requests Issues Marketplace Explore

Watch 130 Star 2,213 Fork 450

Code Issues 121 Pull requests 28 Projects 0 Wiki Insights

OpenID Connect Identity (OIDC) and OAuth 2.0 Provider with Pluggable Connectors <https://coreos.com/blog/announcing-dex...>

719 commits 13 branches 42 releases 48 contributors Apache-2.0



2 Open 0 Closed

Author Labels Projects Milestones Reviews Assignee Sort

Fetch groups in a Google Connector ✓ #1185 opened on 6 Feb by JoelSpeed • Review required

Implement refreshing with Google ✓ #1180 opened on 29 Jan by JoelSpeed • Review required

@JoelASpeed



21

@Pusher

HOW DO I USE THIS?

@JoelASpeed



22

@Pusher

CONNECT K8S TO DEX

```
# The URL where Dex was available  
--oidc-issuer-url=https://auth.example.com/dex  
  
# The client ID configured in dex.  
--oidc-client-id=kubernetes  
  
# CA cert to verify Dex's serving cert  
--oidc-ca-file=/etc/kubernetes/ssl/dex-ca.pem  
  
# The claim field to identify users  
--oidc-username-claim=email  
  
# The claim field to identify user's group membership  
--oidc-groups-claim=groups
```

CONFIGURE KUBECTL

```
users:  
- name: my.email@my.domain.com  
  user:  
    auth-provider:  
      config:  
        client-id: kubernetes  
        client-secret: <INSERT_CLIENT_SECRET_HERE>  
        id-token: <GO_FETCH_YOURSELF_AN_ID_TOKEN>  
        idp-issuer-url: https://auth.domain.com/dex  
        refresh-token: <YOU'LL_PROBABLY_WANT_A_REFRESH_TOKEN_TOO>  
    name: oidc
```

COREOS/DEX/CMD/EXAMPLE-APP

Token:

```
eyJhbGciOiJSUzI1NiIsImtpZCI6IjZiZjU1YmM0YzIzMmAzZWUwYjI1ZDViNTAxYjIxMzUzMWE0NGVjNTIifQ.eyJpc3MiOiJodHRwczovL2F1dGgucHVzaGVycGxhdGZvcn0uaW8vZGV4Iiwic3ViIjoiQ2hVeE1EazBNekEyTWpRd05UY3dORGmzTURFNE1Ua1NCbWR2YjKc1pRIiwiYXVkJjoia3ViZXJuZXRLcyIsImV4cCI6MTUyNzg0MjE5MiwiaWF0IjoxNTI3ODM4NTkyLCJhdF9oYXNoIjoiVEg0dzNwWnF1TmhDZ0pNQX1FTlg5dyIsImVtYWlsIjoiam9lbC5zcGVLZEBwdXNoZXIuY29tIiwiZWlhaWxfdmVyaWZpZWQiOnRydWUsImdyb3VwcyI6WyJhbGVydHNAcHVzaGVyLmNvbSIsImVsZW1lbRzQHB1c2hlc15jb20iLCJlbmdpbmVlcmluZ0BwdXNoZXIuY29tI10sIm5hbWUiOiJKb2VsIFNwZWVkIn0.na6xEleWw2qN9zOf_syTWMs85B-rvo6piAclBj6Z-
```

Claims:

```
{
  "iss": "https://auth.exampledomain.com/dex",
  "sub": "ChUxMDk0MzA2MjQwNTcwNDc3MDE4MTkSBmdvb2dsZQ",
  "aud": "kubernetes",
  "exp": 1527842192,
  "iat": 1527838592,
  "at_hash": "TH4w3pZquNhCgJMAyENX9w",
  "email": "joel.speed@pusher.com",
  "email_verified": true,
  "groups": [
    "group @pusher.com",
    "another @pusher.com",
    "andanother @pusher.com"
  ],
  "name": "Joel Speed"
}
```

Refresh Token:

```
ChlwCWIjenFjY2hwd21hd3
```

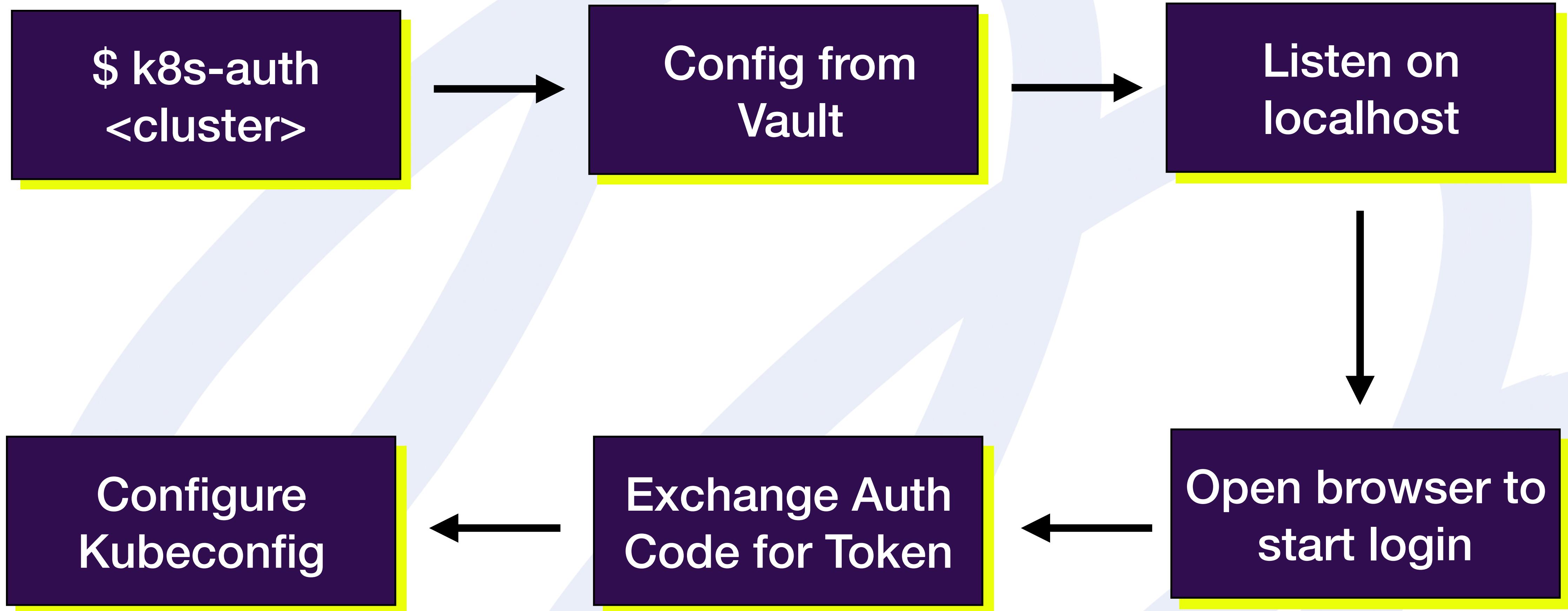
[Redeem refresh token](#)

@JoelASpeed



@Pusher

K8S-AUTH

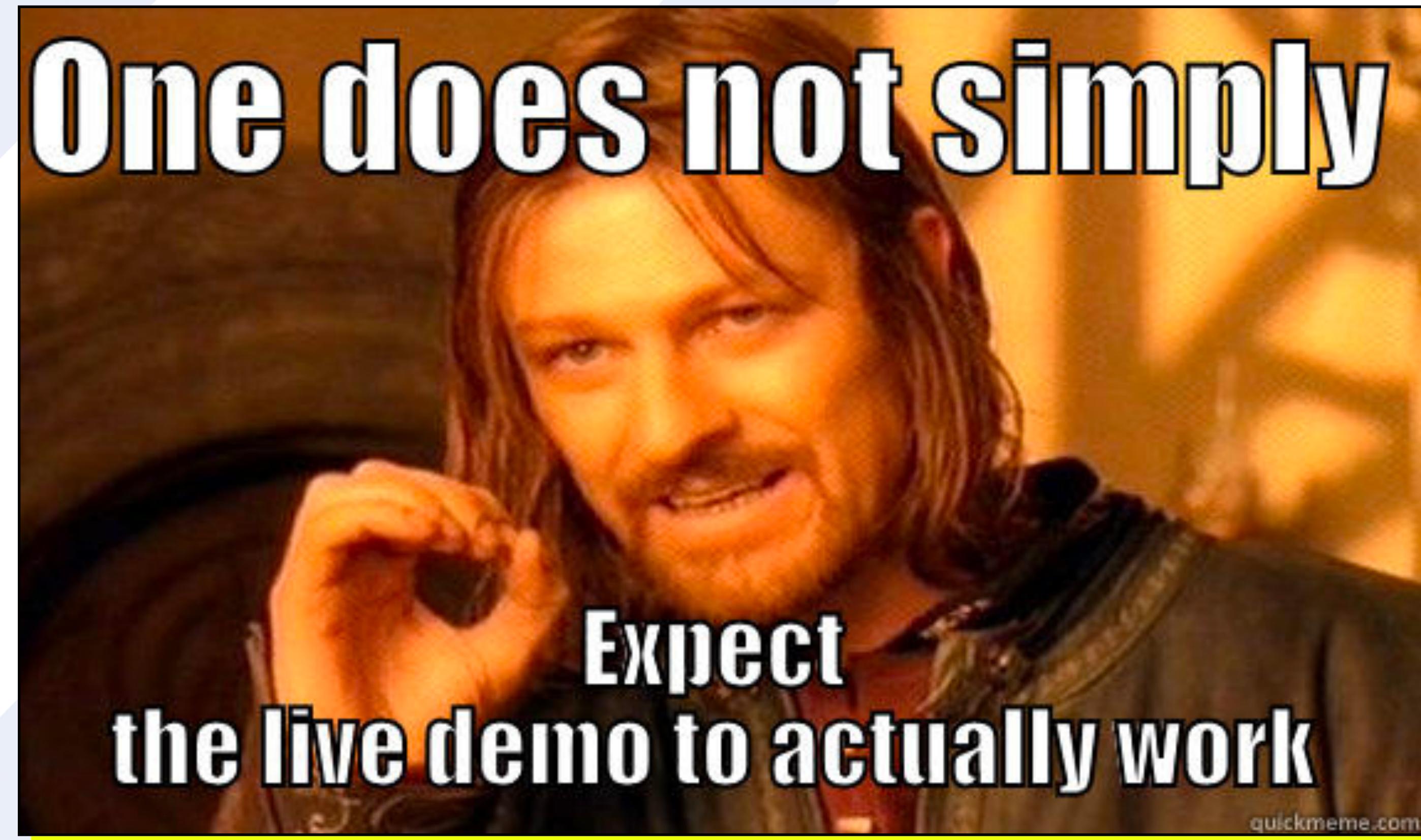


@JoelASpeed

PUSHER

@Pusher

DEMO



@JoelASpeed

 **PUSHER**

@Pusher

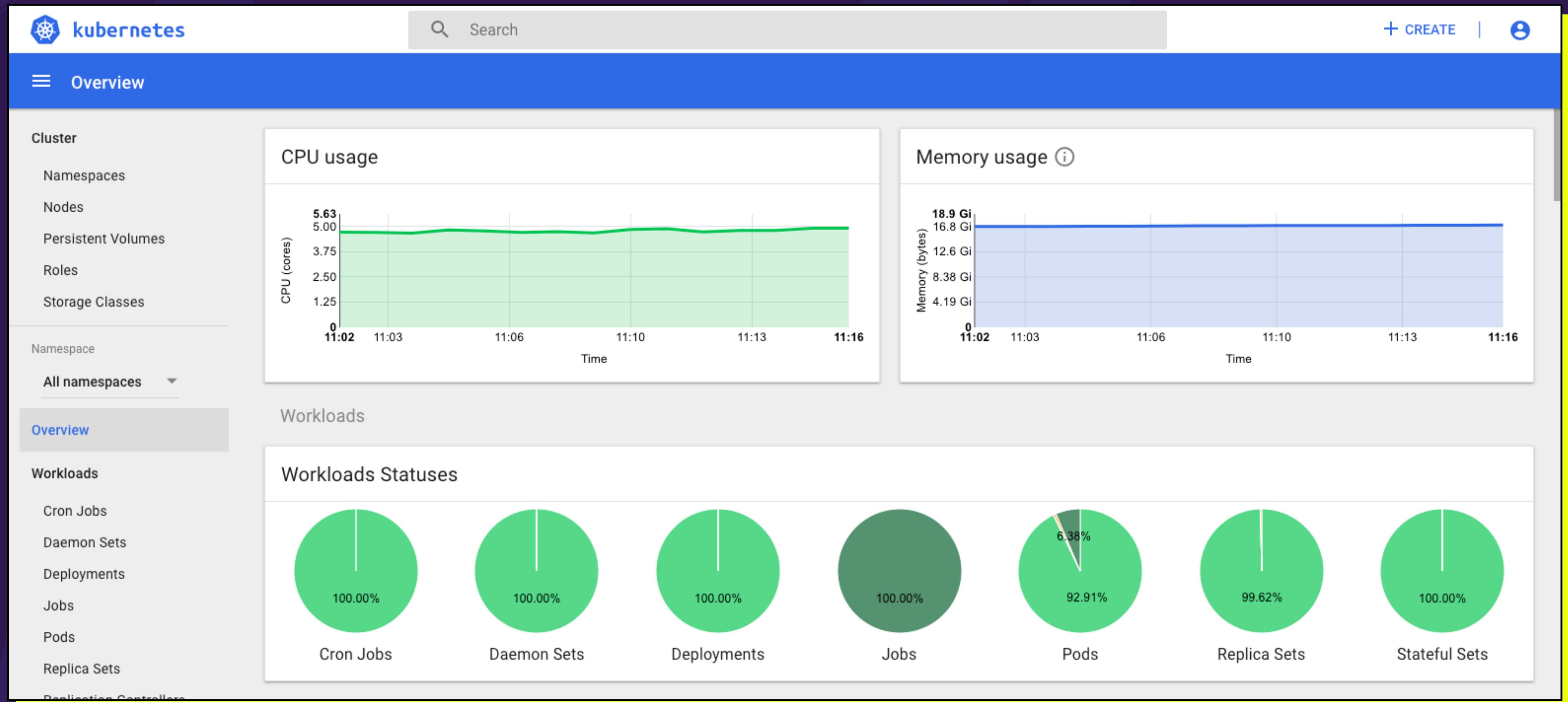
GITHUB.COM/PUSHER/K8S-AUTH-EXAMPLE

@JoelASpeed



@Pusher

KUBERNETES DASHBOARD



@JoelASpeed

 PUSHER

@Pusher

LOGIN

Kubernetes Dashboard

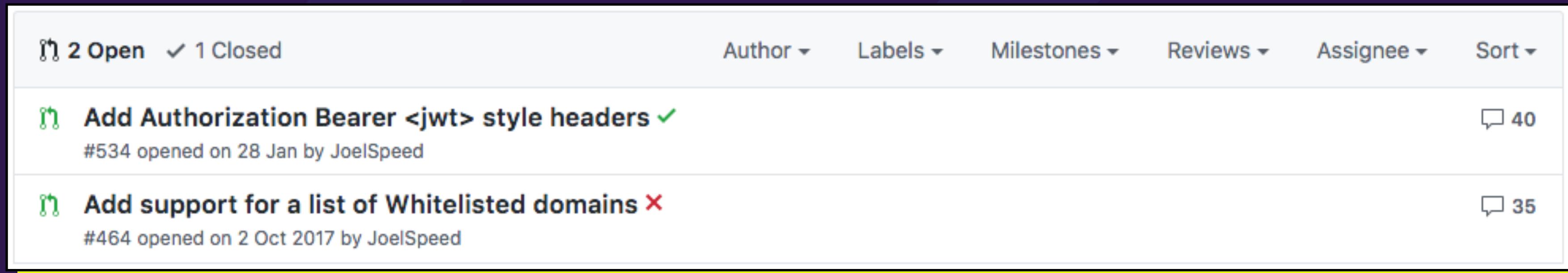
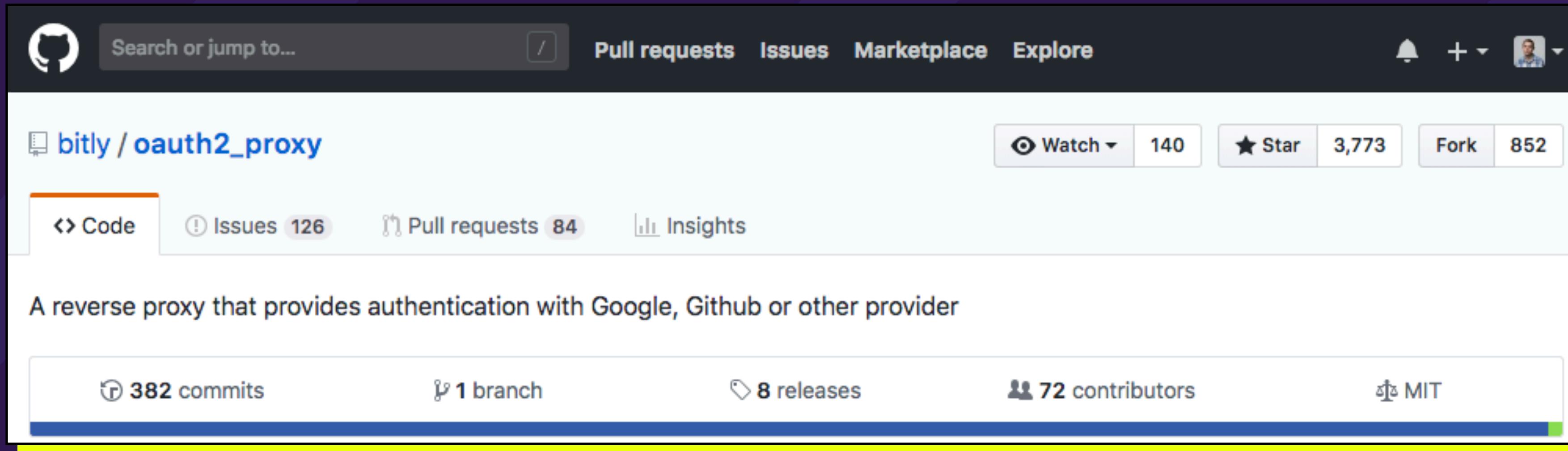
Kubeconfig
Please select the kubeconfig file that you have created to configure access to the cluster. To find out more about how to configure and use kubeconfig file, please refer to the [Configure Access to Multiple Clusters](#) section.

Token
Every Service Account has a Secret with valid Bearer Token that can be used to log in to Dashboard. To find out more about how to configure and use Bearer Tokens, please refer to the [Authentication](#) section.

Enter token

[SIGN IN](#) [SKIP](#)

BITLY OAUTH2 PROXY



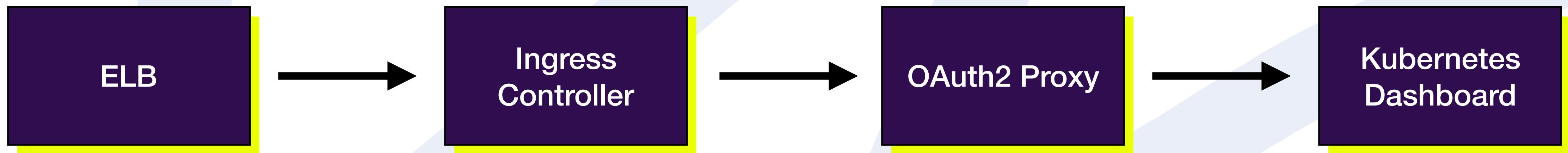
@JoelASpeed

PUSHER

@Pusher

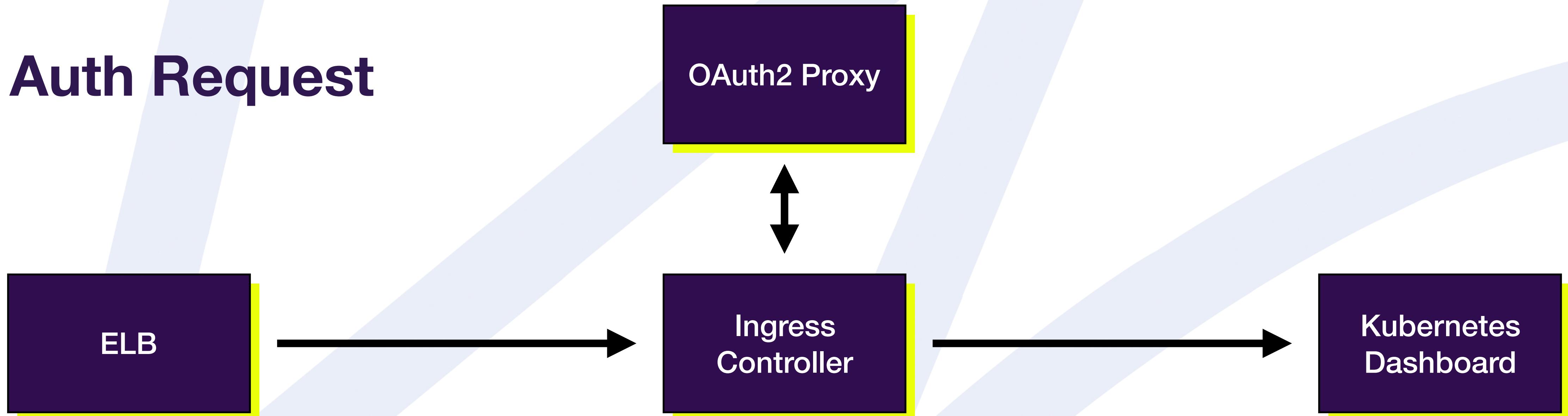
UPSTREAM VS AUTH REQUEST

Upstream



UPSTREAM VS AUTH REQUEST

Auth Request



NGINX CONFIG SNIPPET

```
# Configure Nginx Auth Request Module
ingress.kubernetes.io/auth-url: "https://auth.example.com/oauth2/auth"
ingress.kubernetes.io/auth-signin: "https://auth.example.com/oauth2/start?
                                         rd=https://$host$request_uri$is_args$args"

# Proxy Authentication header to Dashboard
# adds authorization header for kubernetes-dashboard
ingress.kubernetes.io/configuration-snippet: |
  auth_request_set $token $upstream_http_authorization;
  proxy_set_header Authorization $token;
```

DEMO



@JoelASpeed

 **PUSHER**

@Pusher

WHAT HAVE WE ACHIEVED?

Individual user accounts

Group management

Short lived tokens

Scalable

UX

WE'RE HIRING!

pusher.com/careers

@JoelASpeed



@Pusher

Dex

<https://github.com/coreos/dex>

PR #1180: Token Refresh for Google

PR #1185: Fetch Groups from Google

Pusher

@Pusher

pusher.com

<https://github.com/pusher/k8s-auth-example>

OAuth2 Proxy

https://github.com/bitly/oauth2_proxy

PR #464: Whitelist redirect domains

PR #534: Authorization headers

Me

@JoelASpeed

joelspeed.co.uk

Joel.Speed@pusher.com



PUSHER