


KubeCon | CloudNativeCon

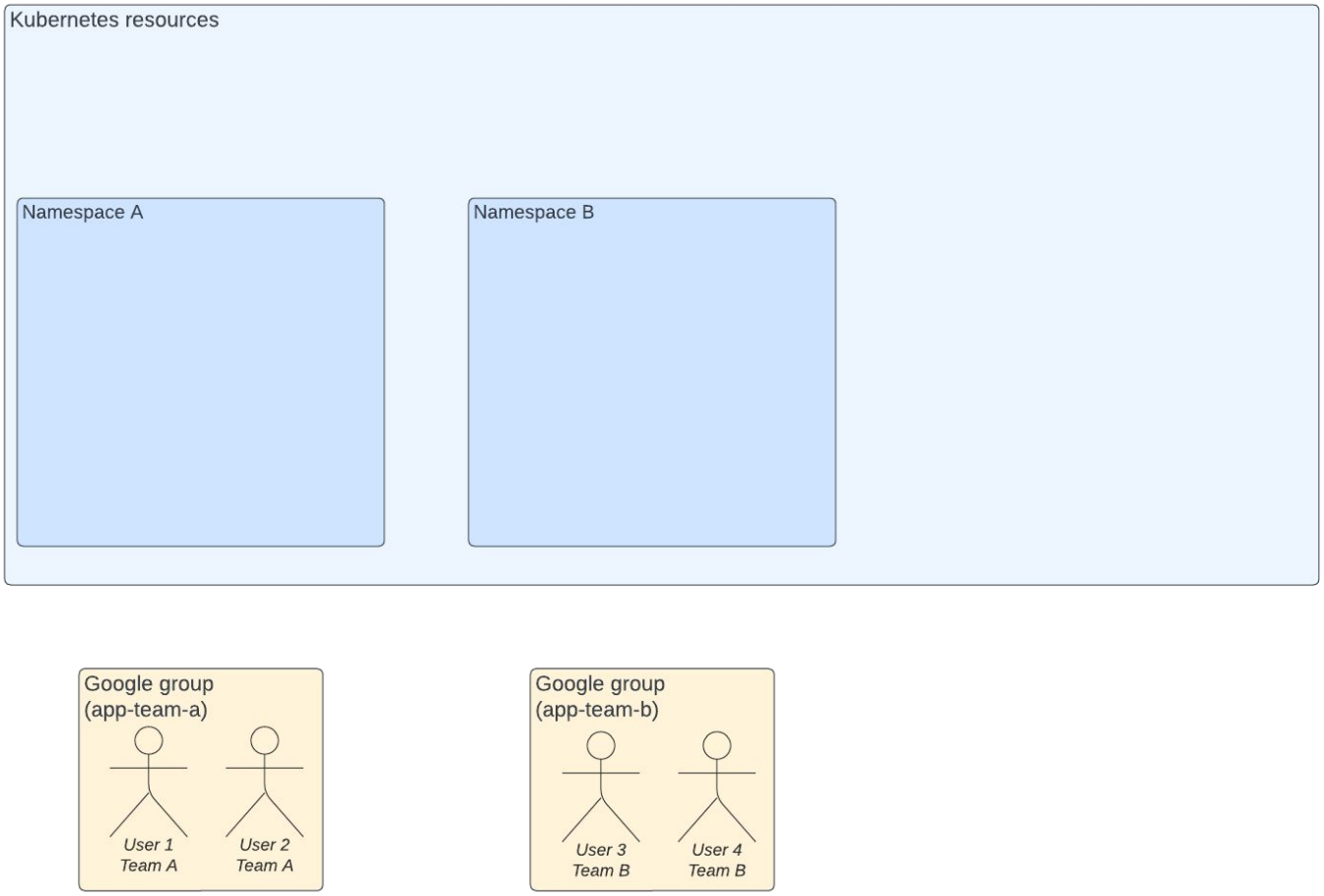
North America 2023

K8s Auth{n,z} at Robinhood

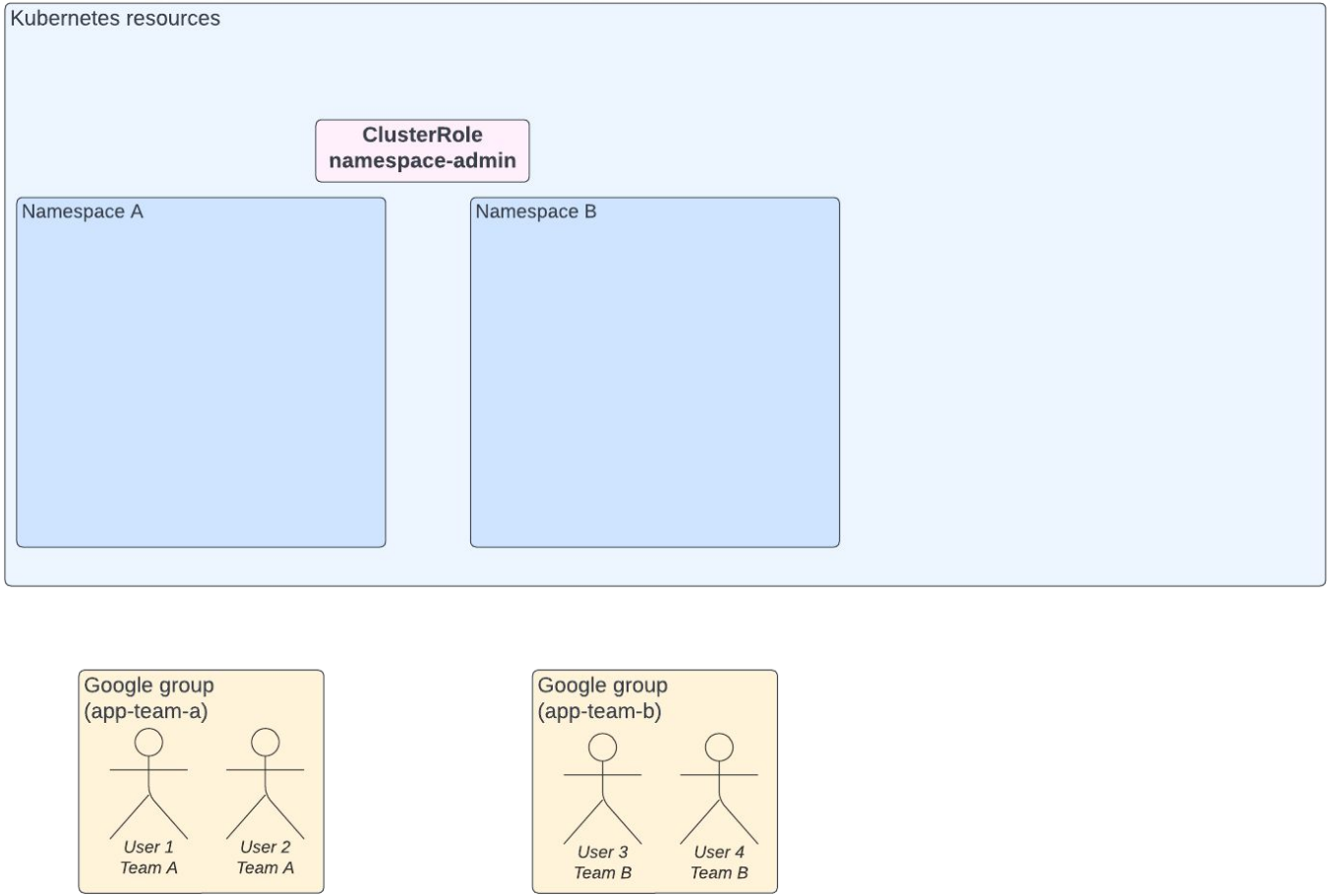
Learnings from reductions, migrations and designing automation

Karen Tu, Sujith Katakam
Robinhood Markets, Inc.

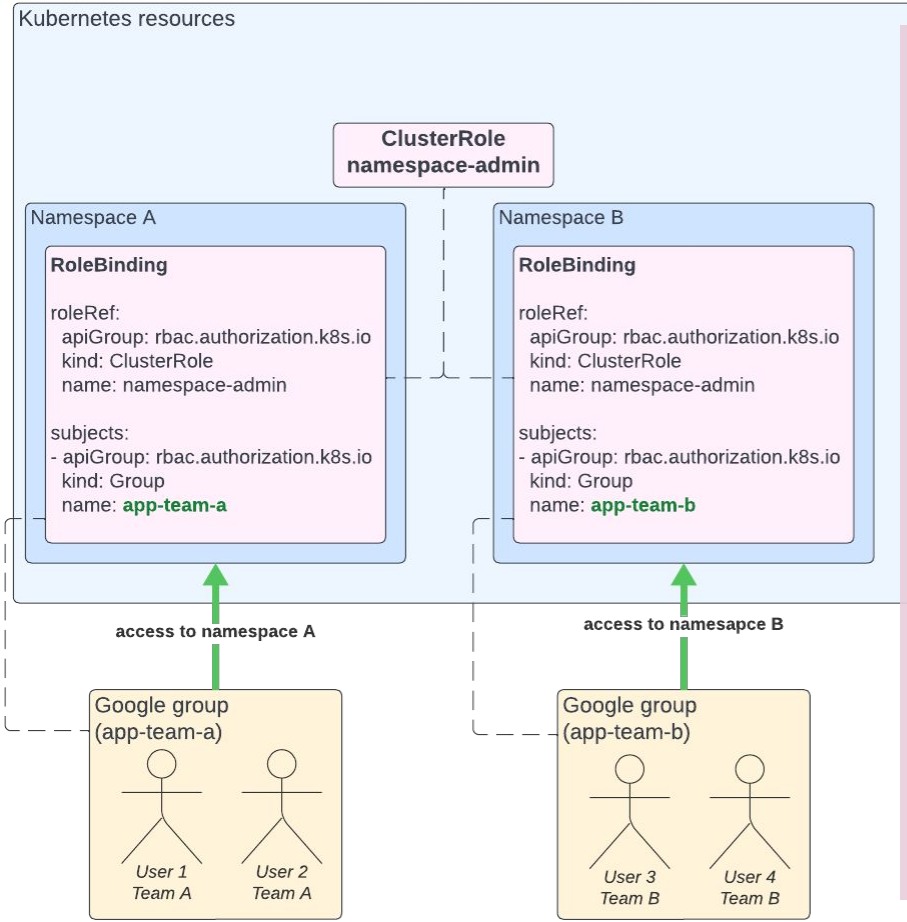
Initial design of k8s access control



Initial design of k8s access control

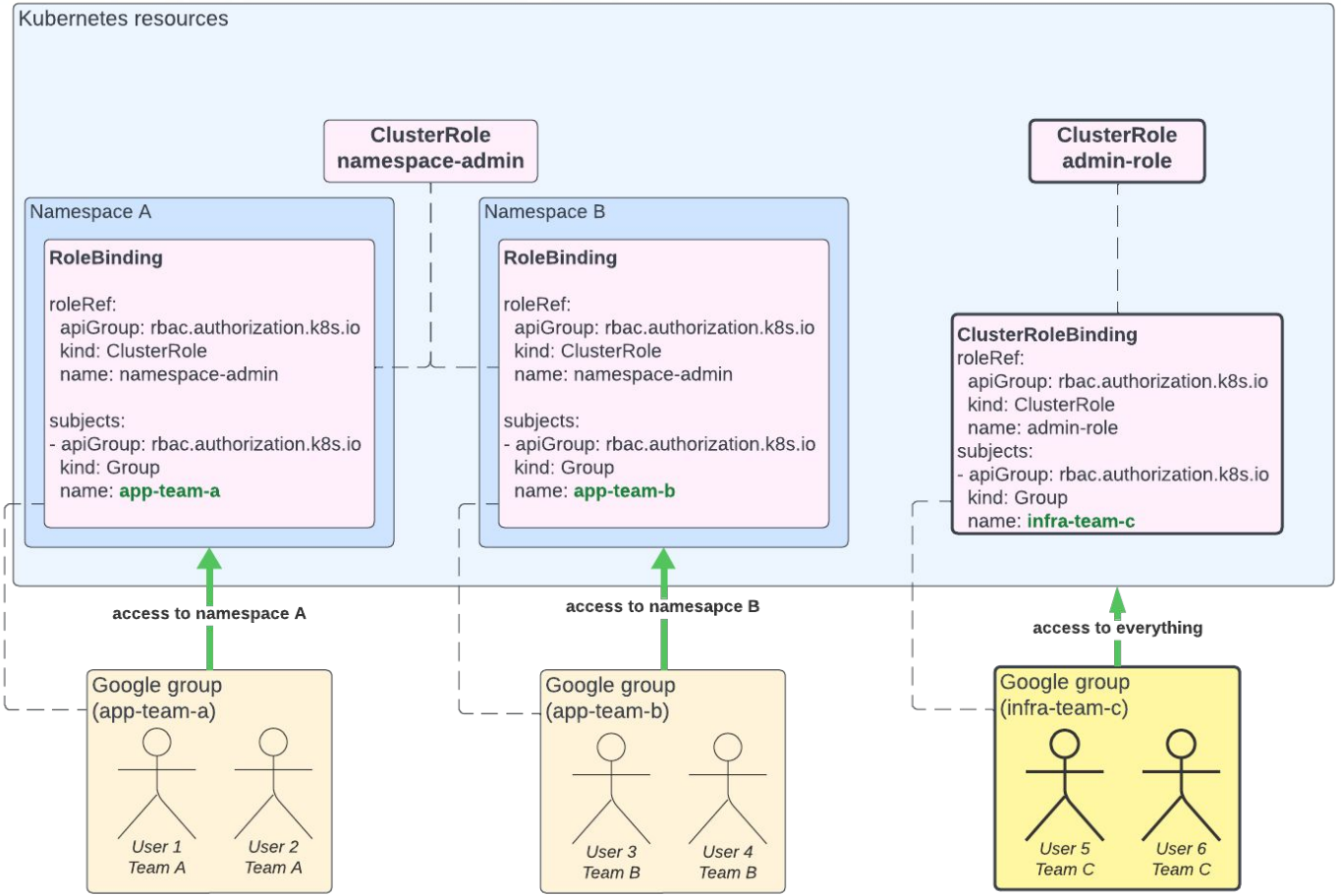


Initial design of k8s access control



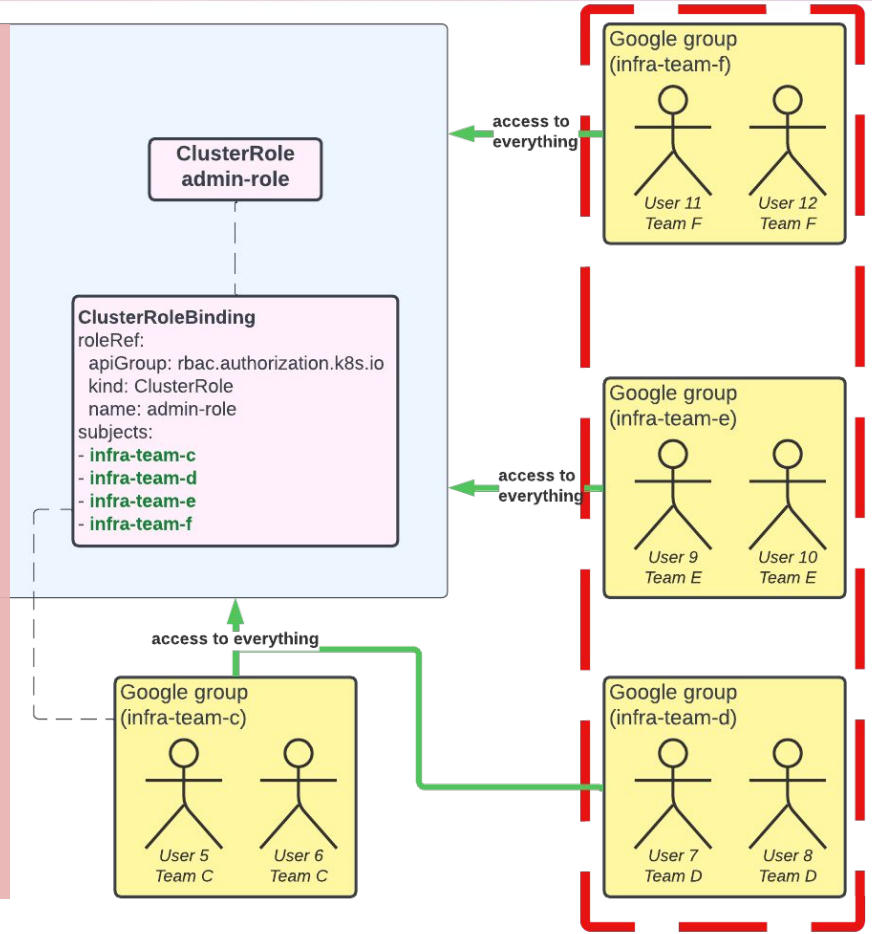
What about infrastructure teams?

Initial design of k8s access control



Why did the initial design NOT scale?

Should **all** infrastructure teams still get full admin access, even though we have >4x the number of infrastructure engineers?




KubeCon | CloudNativeCon

North America 2023

Evolving our security posture

Evolving our security posture

	Original Access Policy	**New Access Policy**
Format	None	Formal document
Admin access definition	Full wildcard * access cluster-wide	1. First order - cluster-wide 2. Second order - indirect access
Who gets admin access?	All infrastructure teams	Fine grained guidelines
What can non-admins access?	Their own namespaces	Their own namespaces

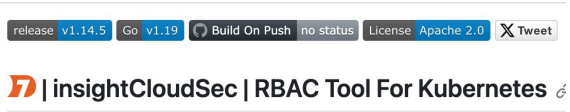

KubeCon | CloudNativeCon

North America 2023

Enforcing Fine-Grained Admin Access

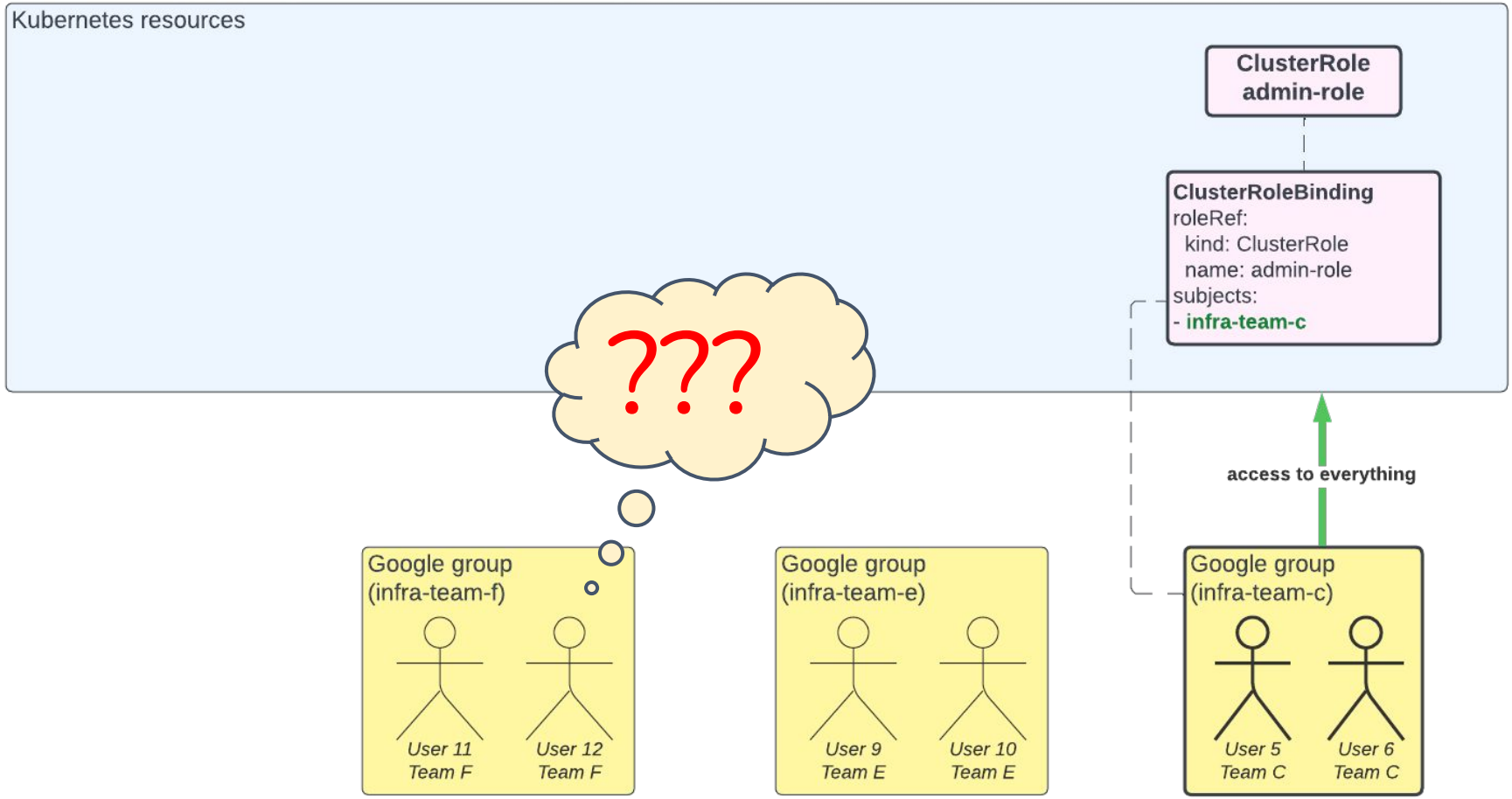
Surveying the existing state

- KubiScan
  - Good starting point for identifying risky permissions
  - Limitations
    - Noisy - included the namespace-admin RoleBinding that we create by default in all namespaces
    - Feature gaps - does not handle missing Roles well
- rbac-tool
  - Only for looking up specific subjects

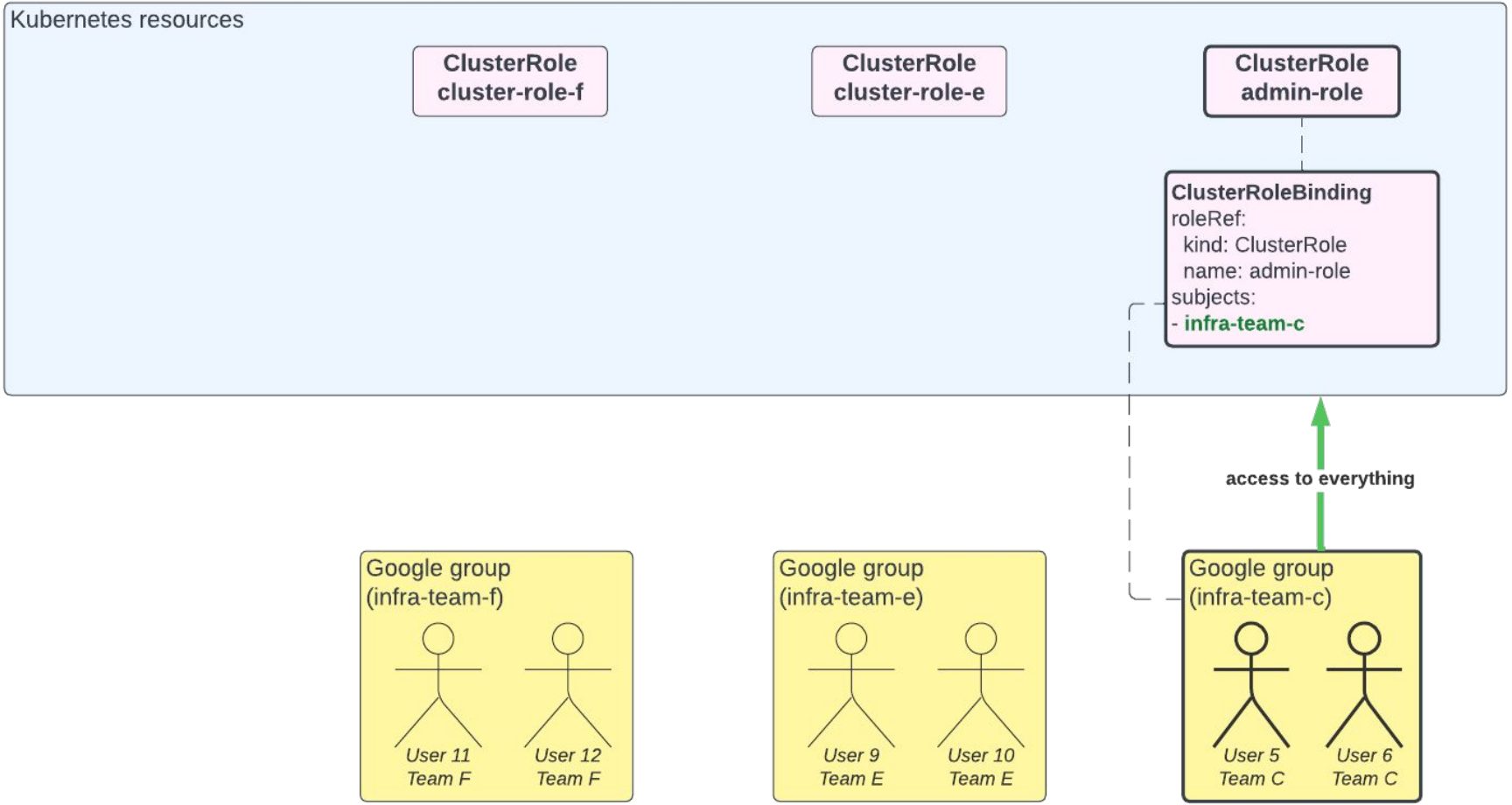


→ We wrote our own!

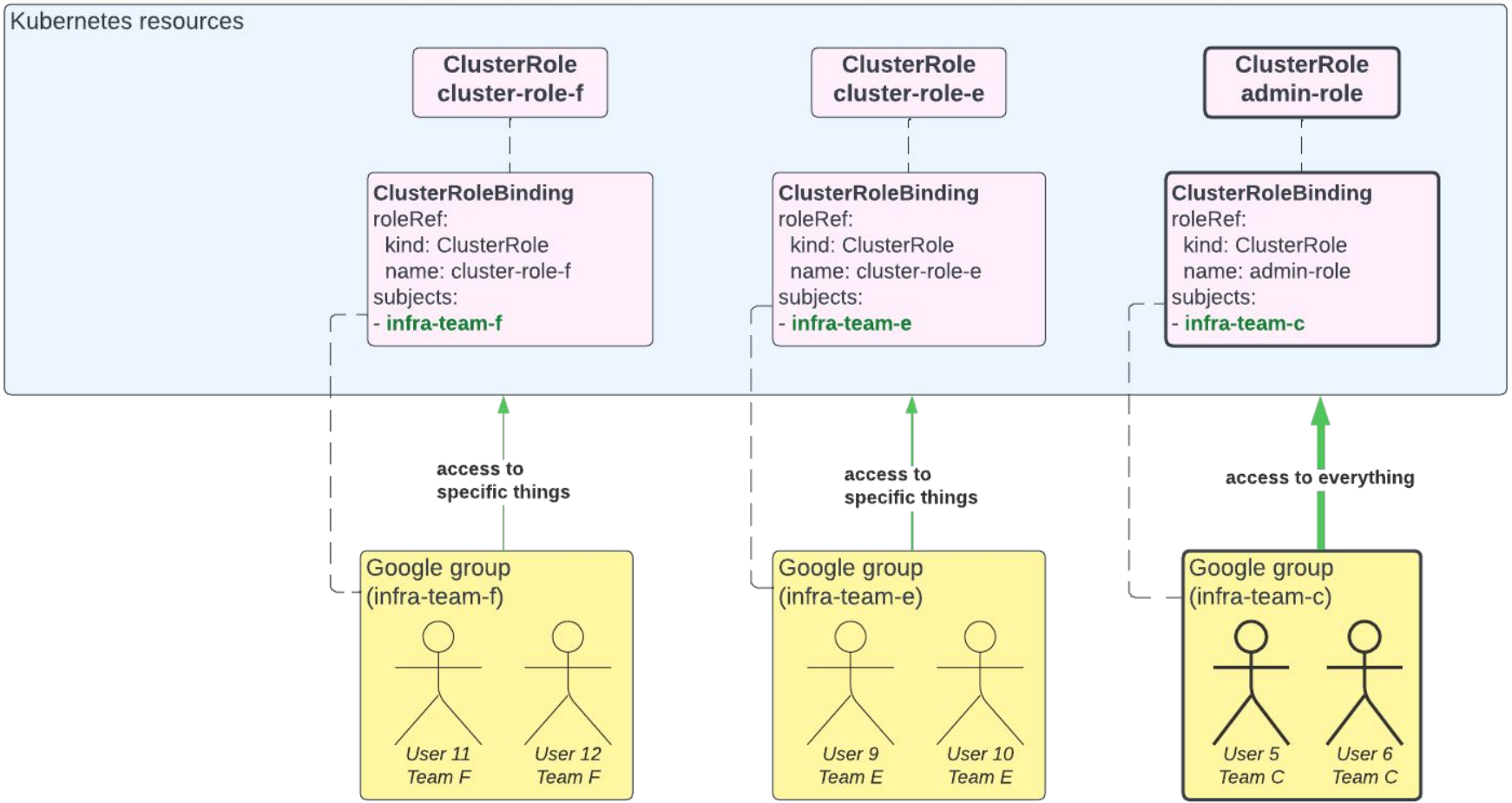
Reducing admin access



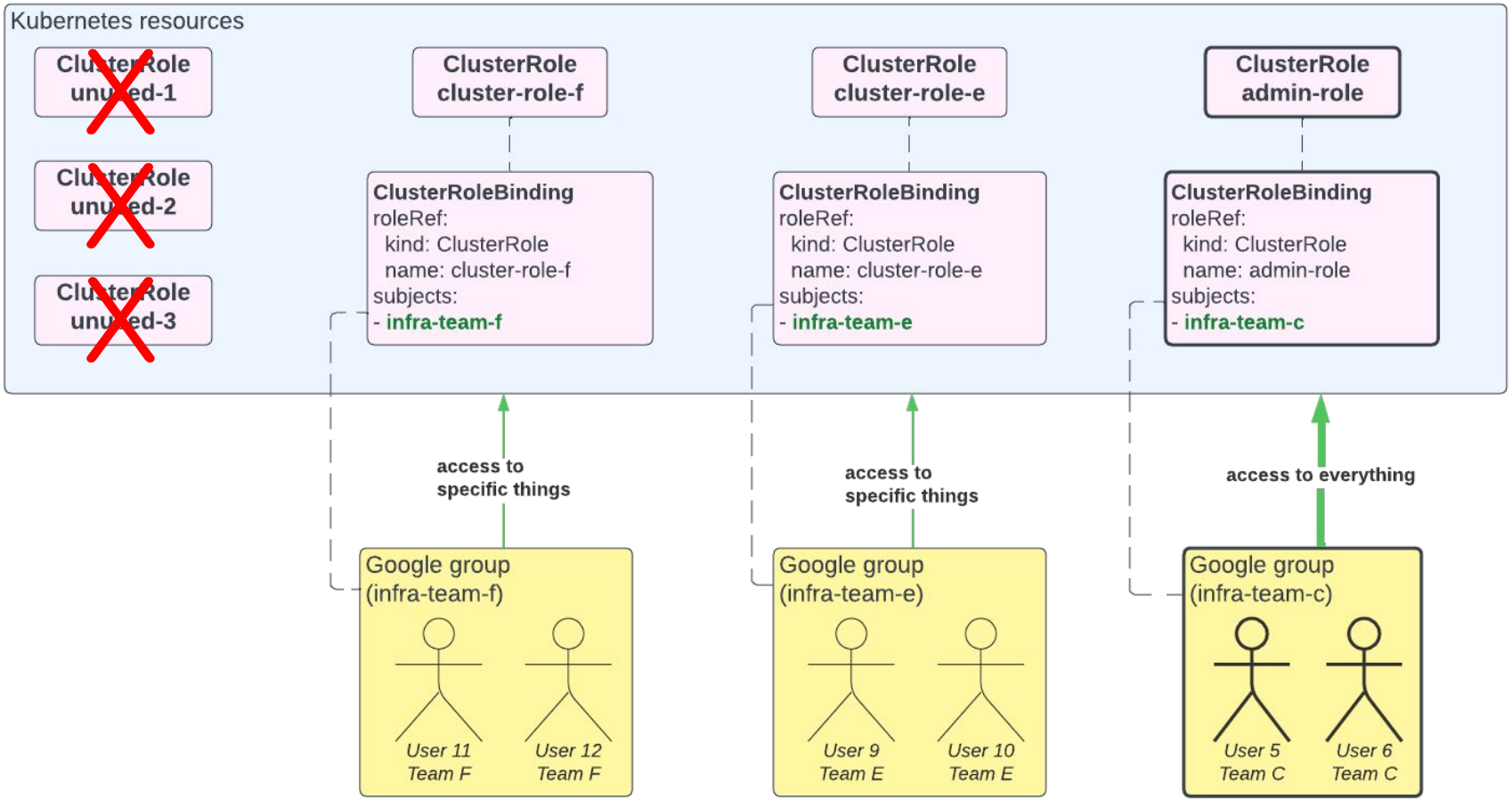
Reducing admin access



Reducing admin access



Reducing admin access

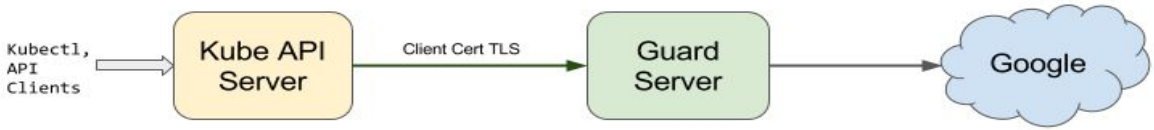



KubeCon | CloudNativeCon

North America 2023

Designing a better access control mechanism

Guard, a third-party webhook authenticator

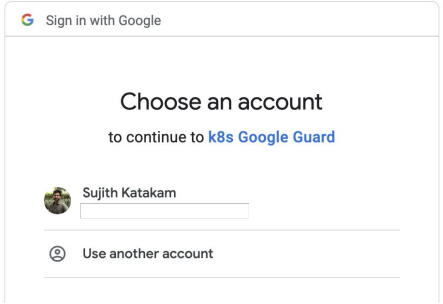



```
{
  "apiVersion": "authentication.k8s.io/v1beta1",
  "kind": "TokenReview",
  "status": {
    "authenticated": true,
    "user": {
      "username": "john@mycompany.com",
      "uid": "<google-id>",
      "groups": [
        "groups-1@mycompany.com",
        "groups-2@mycompany.com"
      ]
    }
  }
}
```

- Verify id-token received using `status.user.username`
- Secrets from GCP service accounts allowed to read Google Admin Directory
- Inject Google groups in TokenReview

How client-side Guard works with kubectl

- Local binary invoked once to write configuration to kubeconfig after user consents
  - `guard get token -o google`
- Once token expires, kubectl fetches a new one from the issuer using the refresh token
  - User doesn't need to login / consent again *(unless the id-token is removed from the kubeconfig)*




```
users:
- name: sample.user@robinhood.com
  user:
    auth-provider:
      config:
        client-id: 484925643224-2gpu9rh2d288mffd8nrqgt8pofrtebkr.apps.googleusercontent.com
        client-secret: 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
        id-token: REDACTED
        idp-issuer-url: https://accounts.google.com
        refresh-token: REDACTED
      name: oidc
```

What's the OAuth client? How is it configured?

guard / auth / providers / google / **google.go** ↑ Top

Code | Blame | 115 lines (94 loc) · 2.69 KB | Raw

```
13
14  const (
15          OrgType = "google"
16
17          googleIssuerUrl = "https://accounts.google.com"
18          // https://developers.google.com/identity/protocols/OAuth2InstalledApp
19          GoogleOauth2ClientID     = "37154062056-220683ek37naab43v23vc5qg01k1j14g.apps.googleusercontent.com"
20          GoogleOauth2ClientSecret = "pB9ITCuMPLj-bkObrTqKbt57"
21  )
```

- Hardcoding client ID and client secret is *okay* with "OAuth 2.0 for Mobile & Desktop Apps"
- Token always gets redirected to the configured redirect URI (controlled by us), which is user's `localhost:8000`
  - *oauth2 redirect URL*

An incident once upon a time

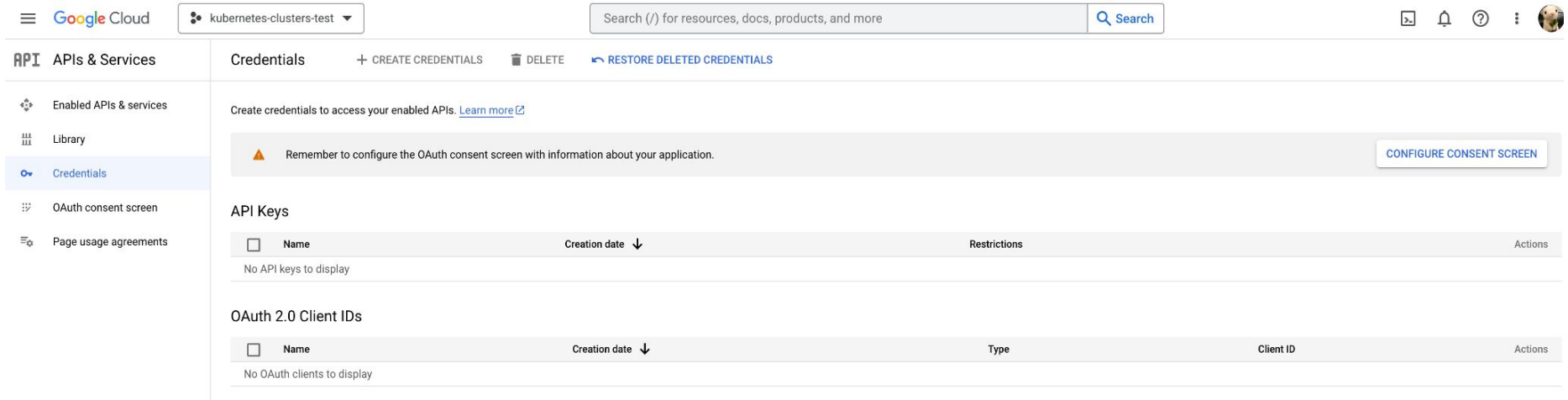

```
Unable to connect to the server: failed to refresh token: oauth2: cannot fetch
token: 401 Unauthorized

Response: {
  "error": "deleted_client",
  "error_description": "The OAuth client was deleted."
}
```

- Symptoms: No one was able to generate new `id-token` and `refresh-token`
- Logs don't show any errors, only logs like -
  Received token review request for google/robinhood.com
- Impact: Authentication & Authorization down for all clusters; no one can do anything (besides admins with break glass static key)

Where is the OAuth client?

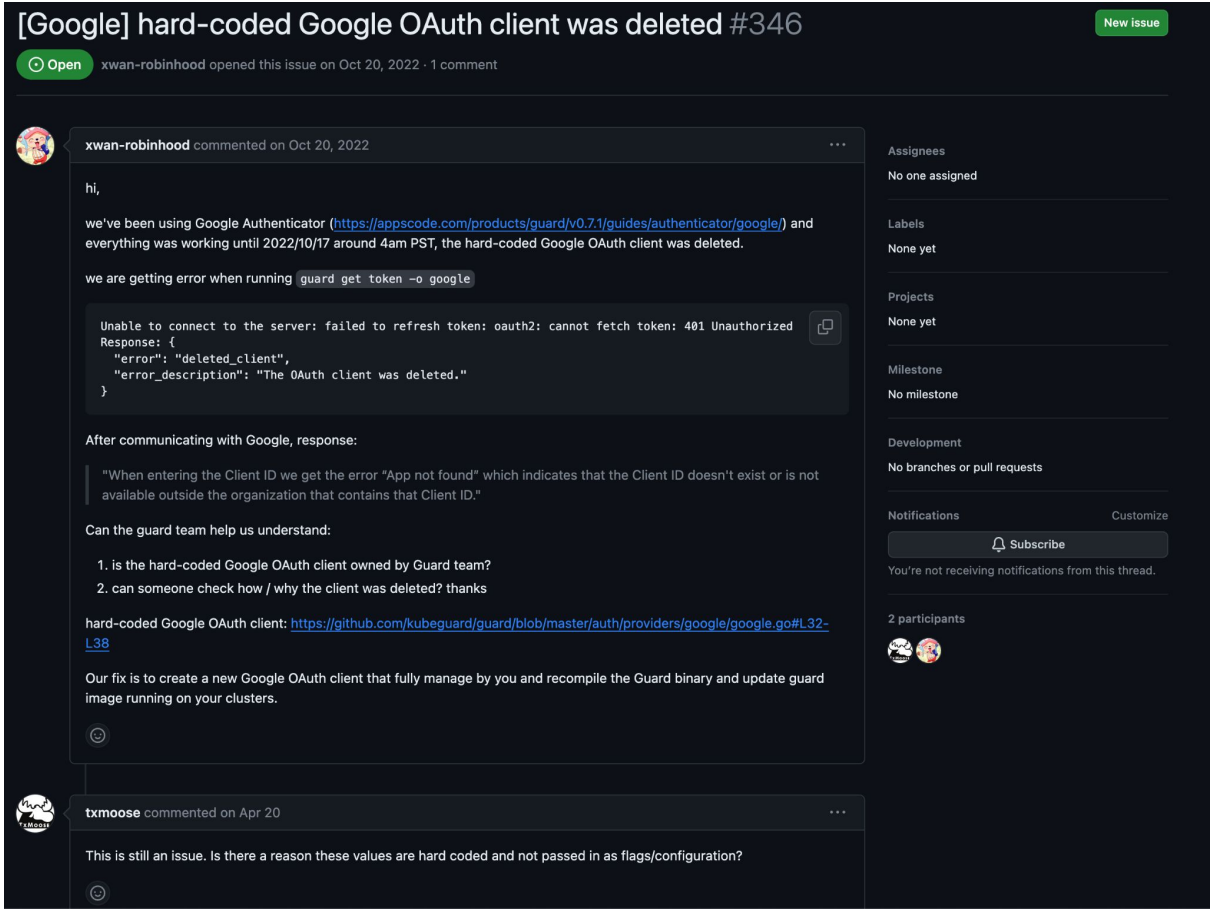
- We tried looking in Robinhood's GCP project, engaging GCP support
  - *There's no OAuth client configured at all in our GCP project!*
- Nothing in our Guard cluster add-on configurations



Did we find the OAuth client?

We still don't know what happened to the OAuth client!

Source: Open GitHub issue



[Google] hard-coded Google OAuth client was deleted #346

New issue

Open xwan-robinhood opened this issue on Oct 20, 2022 · 1 comment

xwan-robinhood commented on Oct 20, 2022

hi,

we've been using Google Authenticator (https://appscode.com/products/guard/v0.7.1/guides/authenticator/google/) and everything was working until 2022/10/17 around 4am PST, the hard-coded Google OAuth client was deleted.

we are getting error when running `guard get token -o google`

```
Unable to connect to the server: failed to refresh token: oauth2: cannot fetch token: 401 Unauthorized
Response: {
  "error": "deleted_client",
  "error_description": "The OAuth client was deleted."
}
```

After communicating with Google, response:

> "When entering the Client ID we get the error "App not found" which indicates that the Client ID doesn't exist or is not available outside the organization that contains that Client ID."

Can the guard team help us understand:

1. is the hard-coded Google OAuth client owned by Guard team?
2. can someone check how / why the client was deleted? thanks

hard-coded Google OAuth client: https://github.com/kubeguard/guard/blob/master/auth/providers/google/google.go#L32-L38

Our fix is to create a new Google OAuth client that fully manage by you and recompile the Guard binary and update guard image running on your clusters.

txmoose commented on Apr 20

This is still an issue. Is there a reason these values are hard coded and not passed in as flags/configuration?

Assignees
No one assigned

Labels
None yet

Projects
None yet

Milestone
No milestone

Development
No branches or pull requests

Notifications Customize
Subscribe
You're not receiving notifications from this thread.

2 participants

Additional problems with Guard

- Obviously, the OAuth client situation was not ideal
- Security
  - Long lived refresh-tokens
  - Tokens shared among all environments
  - No MFA to retrieve tokens
- Operations
  - Drift from central identity management
  - GCP only used for Guard
- Reliability
  - Guard's logs aren't helpful
  - Additional dependency on the control plane

Additional problems with Guard

- Obviously, the OAuth client situation was not ideal
- Security
  - Long lived refresh-tokens
  - Tokens shared among all environments
  - No MFA to retrieve tokens
- Operations
  - Drift from central identity management
  - GCP only used for Guard
- Reliability
  - Guard's logs aren't helpful
  - Additional dependency on the control plane

Defining requirements of Auth{n,z}

Security

- Groups mechanisms used for authorization have tightly controlled procedure of managing membership, dedicated to K8s access only
- Credentials issued should be short-lived, retrieved with multi-factor authentication (MFA)

Reliability

- The identity provider (IdP) should be able to handle our load
- The IdP and associated authentication mechanism should export informative error logs and reliable metrics

Token lifetime

- Balance between security and productivity

Potential providers for Auth{n,z}

- We want to use OIDC
  - "*Since all of the data needed to validate who you are is in the id_token, Kubernetes doesn't need to "phone home" to the identity provider*"

- Okta vs AWS as IdP

Q: My application supports OpenID Connect (OIDC) only. Can I use it with IAM Identity Center?

No. IAM Identity Center supports only SAML 2.0–based applications.

Potential providers for Auth{n,z}

- We want to use OIDC
  - *"Since all of the data needed to validate who you are is in the id_token, Kubernetes doesn't need to "phone home" to the identity provider"*

✅

- Okta vs AWS as IdP

❌

Q: My application supports OpenID Connect (OIDC) only. Can I use it with IAM Identity Center?

No. IAM Identity Center supports only SAML 2.0–based applications.

Args in api-server container

Webhook authenticator

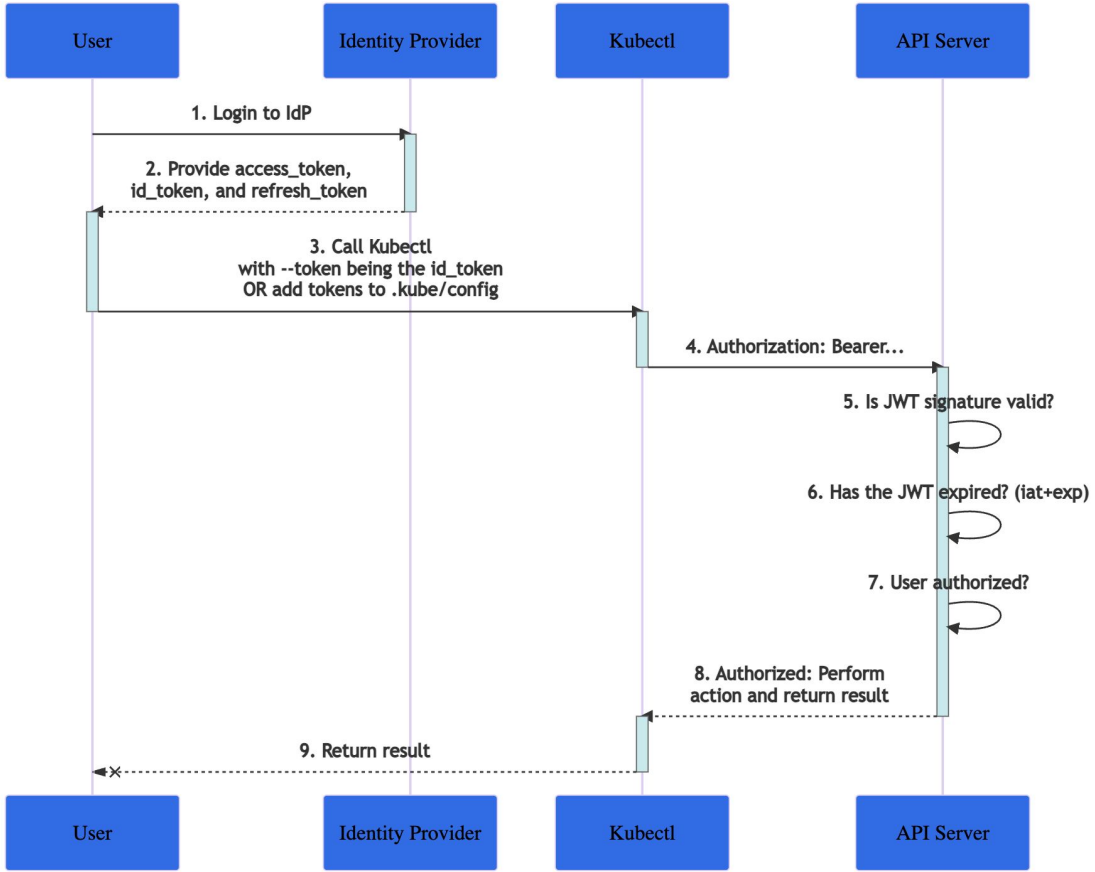
```
- --authentication-token-webhook-cache-ttl=5m
- --authentication-token-webhook-config-file=/srv/kubernetes/webhook-guard-config
- --authentication-token-webhook-version=v1
```

OIDC token authenticator

```
- --oidc-issuer-url=ISSUER_URL
- --oidc-client-id=PER_ENV_ID
- --oidc-username-claim=email
- --oidc-groups-claim=groups
```

Note: apiserver can support more than one authenticator, which we took advantage of during the migration!

Authentication: api-server natively validates id-token



What our id-token looks like

Per-environment OAuth Client ID

Group-based access control

- But… how can users get the id-token?

Client application to retrieve token

README.md

kubelogin

go passing | go report A+

This is a kubectl plugin for Kubernetes OpenID Connect (OIDC) authentication, also known as `kubectl oidc-login`.

```yaml
- name: oidc-user
  user:
    exec:
      apiVersion: client.authentication.k8s.io/v1beta1
      args:
      - oidc-login
      - get-token
      - --oidc-issuer-url=ISSUER_URL
      - --oidc-client-id=PER_ENV_ID
      - --oidc-extra-scope=email
      - --oidc-extra-scope=groups
      - --oidc-extra-scope=offline_access
      - --listen-address=127.0.0.1:18000
      command: kubectl
      env: null
      interactiveMode: IfAvailable
      provideClusterInfo: false
```

client-go credential plugins

FEATURE STATE: `Kubernetes v1.22 [stable]`

`k8s.io/client-go` and tools using it such as `kubectl` and `kubelet` are able to execute an external command to receive user credentials.

This feature is intended for client side integrations with authentication protocols not natively supported by `k8s.io/client-go` (LDAP, Kerberos, OAuth2, SAML, etc.). The plugin implements the protocol specific logic, then returns opaque credentials to use. Almost all credential plugin use cases require a server side component with support for the webhook token authenticator to interpret the credential format produced by the client plugin.


KubeCon | CloudNativeCon

North America 2023

Challenges with Onboarding to new system

Authorization: Group-based access

Biggest questions to answer were -

- What would each group represent?
  - RoleBinding? ClusterRoleBinding? Team?
    - *Get data and talk to users!*
- How to govern group membership?
    - *Third party? Build on our own solution?*

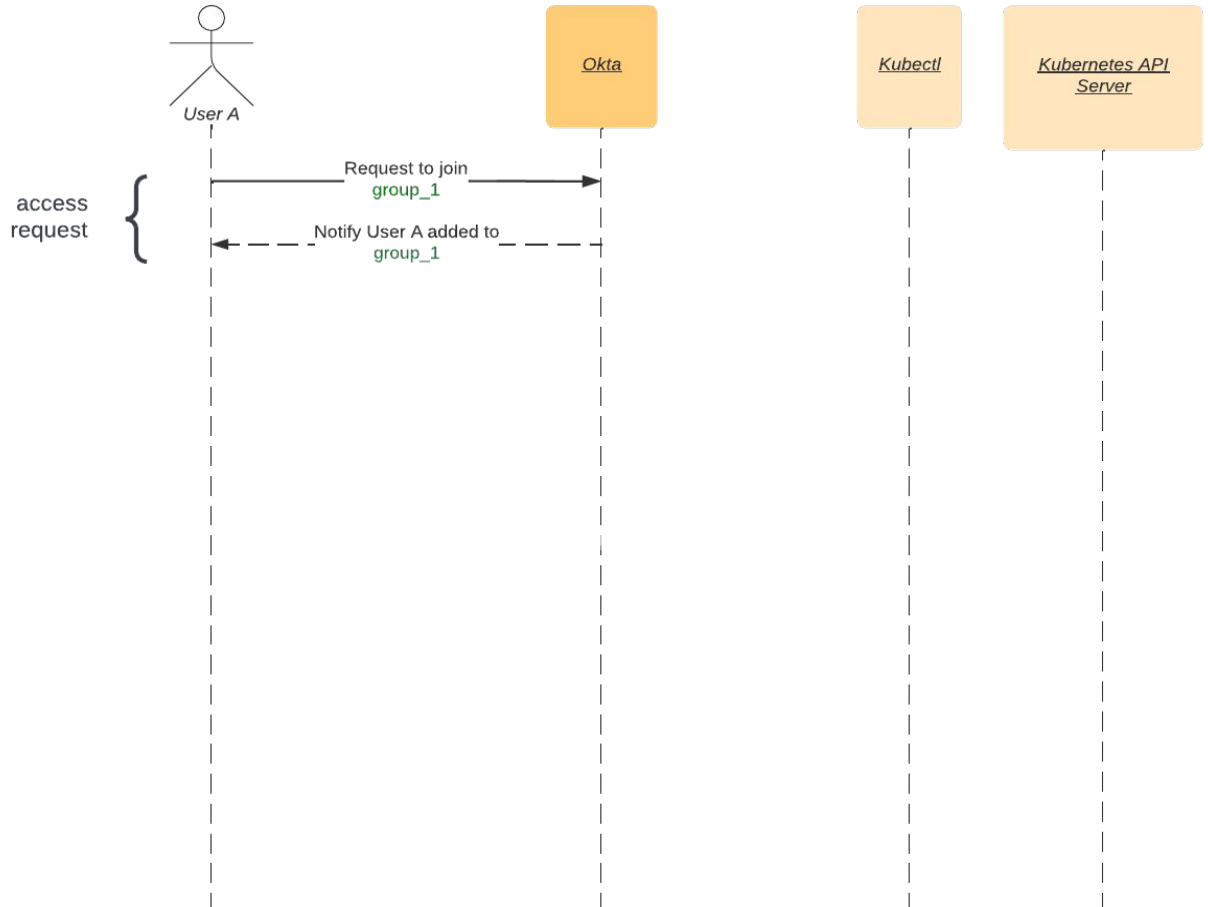
```
kind: ClusterRoleBinding
metadata:
  name: pod-writer
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: pod-writer
subjects:
- apiGroup: rbac.authorization.k8s.io
  kind: Group
  name: group_1
- apiGroup: rbac.authorization.k8s.io
  kind: Group
  name: group_2
```


KubeCon | CloudNativeCon

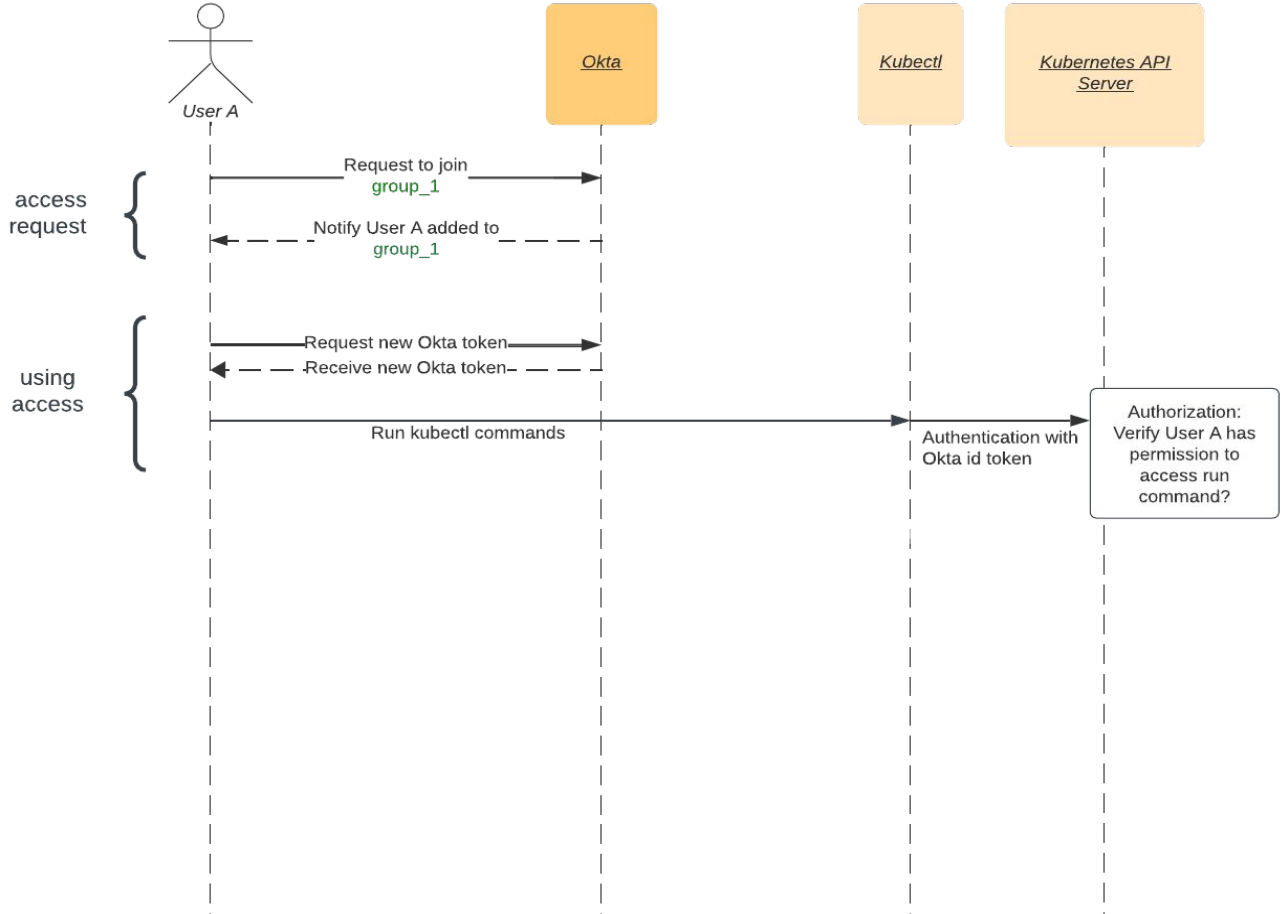
North America 2023

How can we remediate Persistent Privileged Access ?

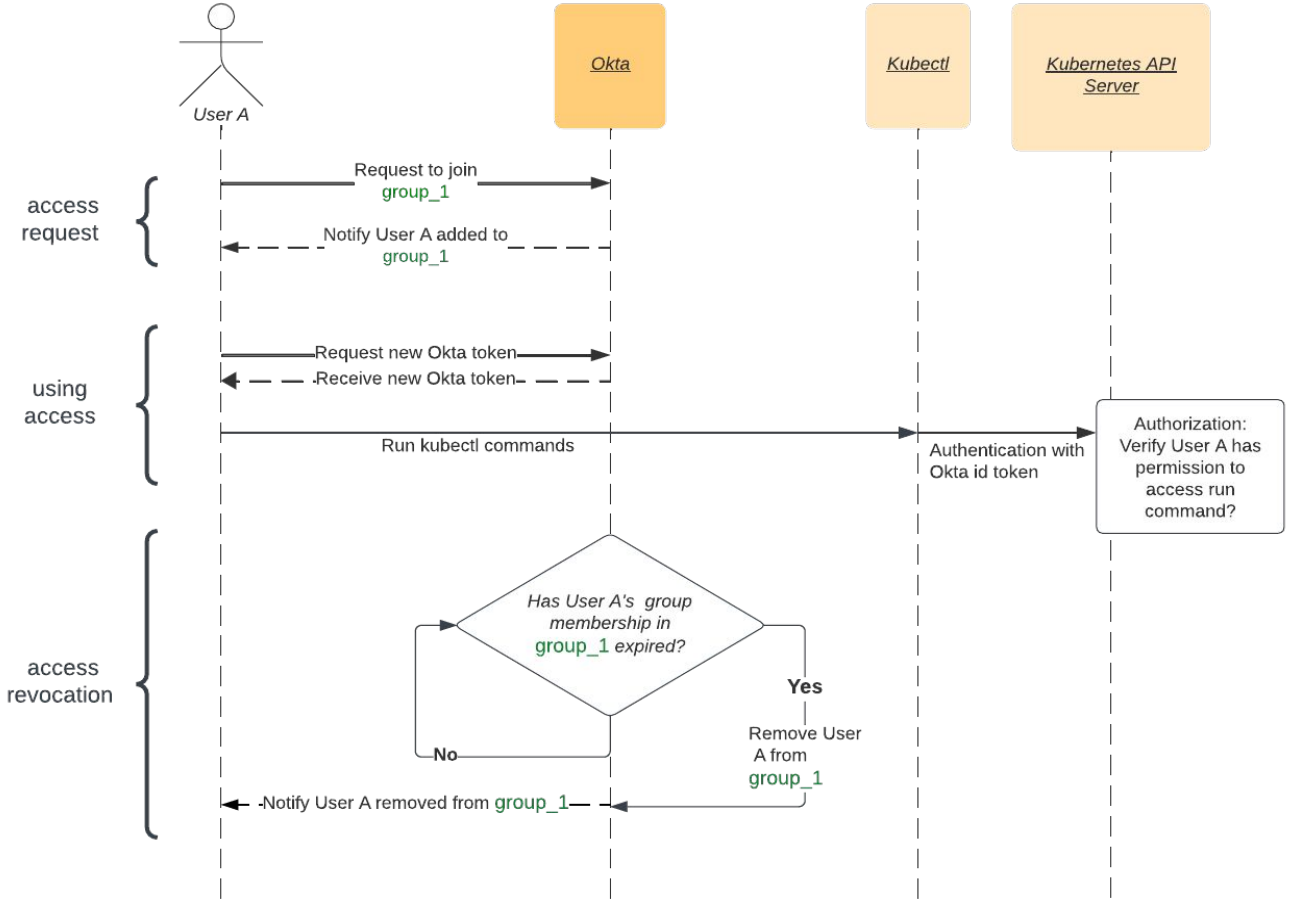
JIT (Just-In-Time) Privileged Access



JIT (Just-In-Time) Privileged Access



JIT (Just-In-Time) Privileged Access



Takeaways

- Talk to the users
  - But compare data to what users tell you
- Implement k8s rbac & infrastructure best practices
  - Continuous deployment!
- Proper governance on user groups
- Provide a way to request temporary access


KubeCon | CloudNativeCon

North America 2023

We're hiring!

careers.robinhood.com

Senior Software Engineer - Kubernetes (Container Orchestration)
Staff Software Engineer - Kubernetes (Container Orchestration)


KubeCon | CloudNativeCon

North America 2023

Questions ?

KubeCon | CloudNativeCon

North America 2023

Please scan the QR code above to leave feedback on this session