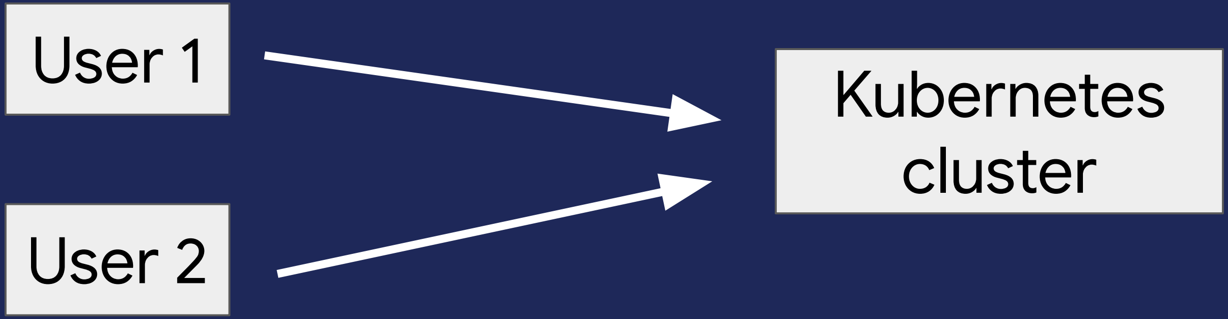
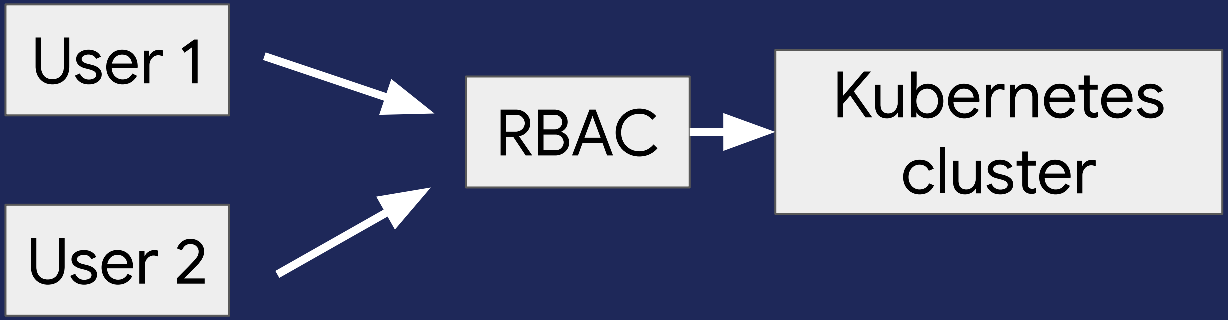
Intro to Kubernetes RBAC

Imre Nagi
GDE Cloud & CTO schoters.com

Background



RBAC (Role Based Access Control)



RBAC Overview

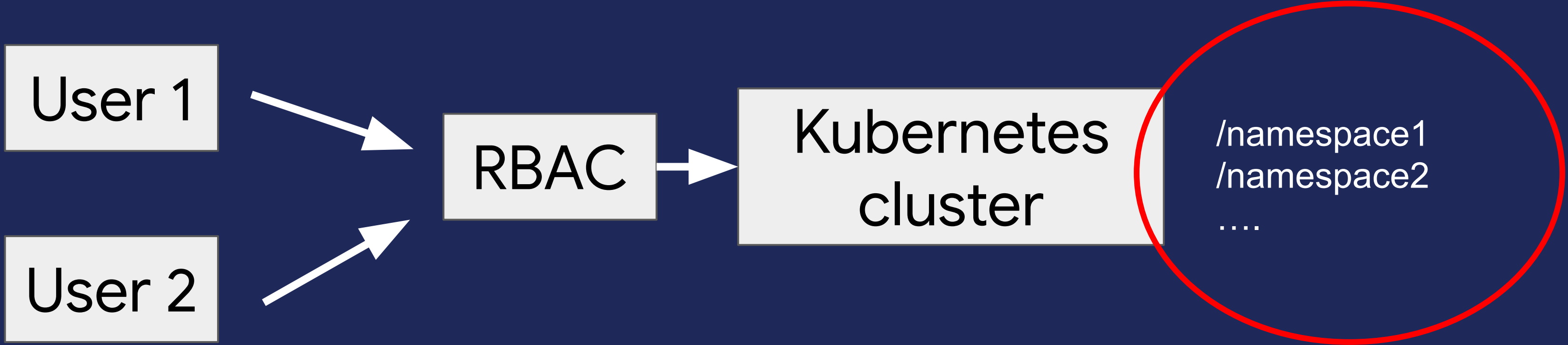
Can ____________ ____________ ____________ ?

Subject verb object

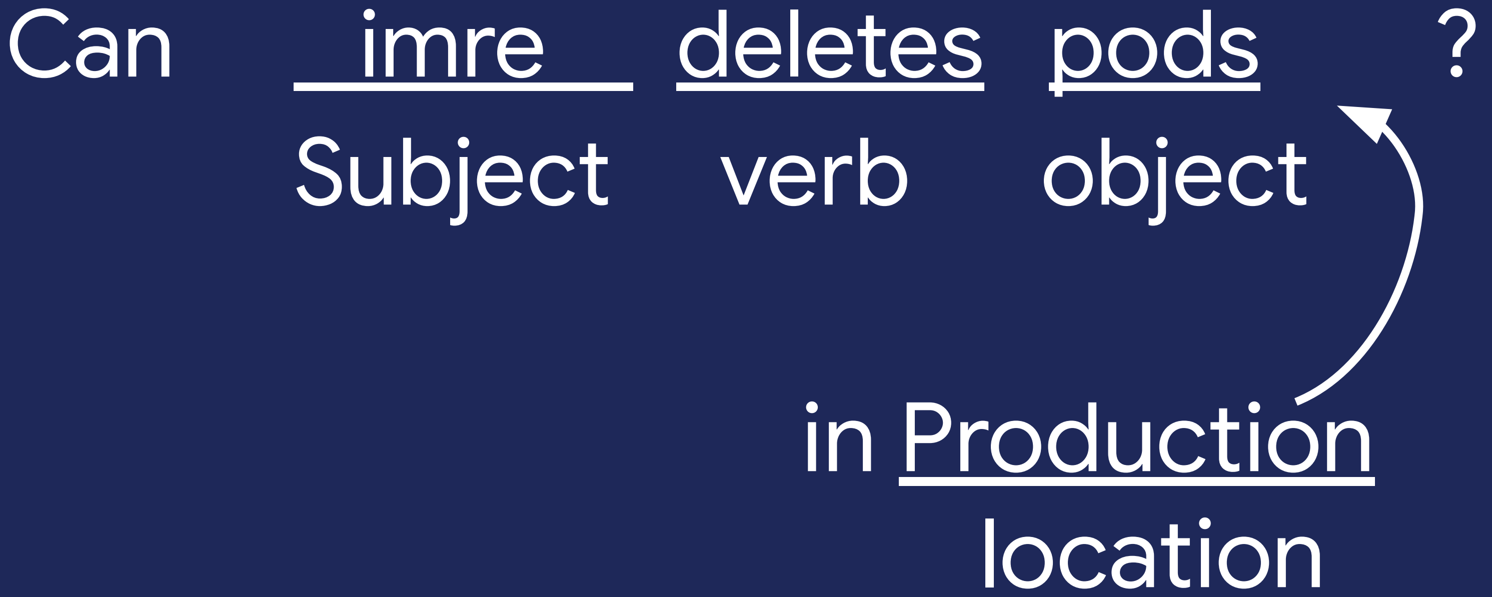
RBAC Overview

Can	imre	deletes	pods	?
	Subject	verb	object	

RBAC (Role Based Access Control)



RBAC Overview



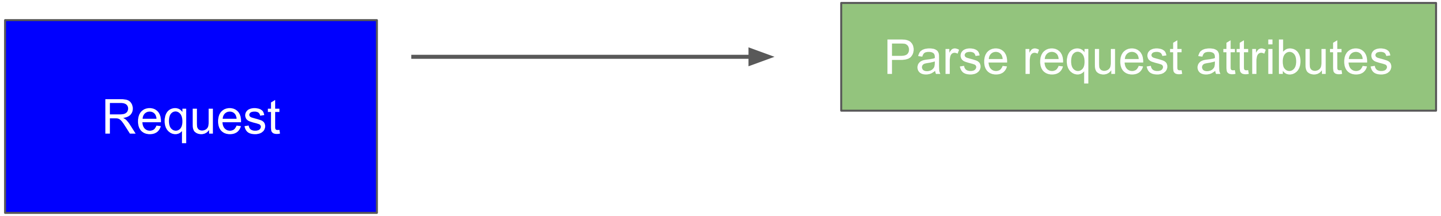
Can imre deletes pods ?

Subject verb object

in Production

location

Request Handling

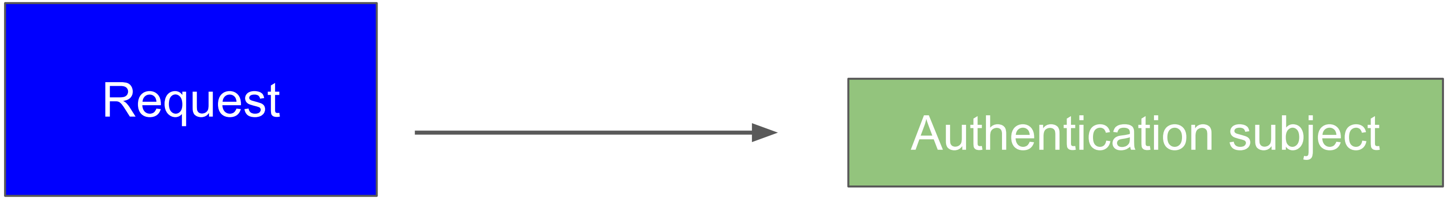


GET /user/john/orders
Authorization: Bearer abasdAZKLJDA....
Content-Type: application/json
Accept: application/json

{“id”: 123, “code”: “mantap jiwa”.......

Verb	get
API group	user
User	john
resource	orders

Request Handling



Request

Authentication subject

GET /user/john/orders

Authorization: Bearer abasdAZKLJDA....

Content-Type: application/json

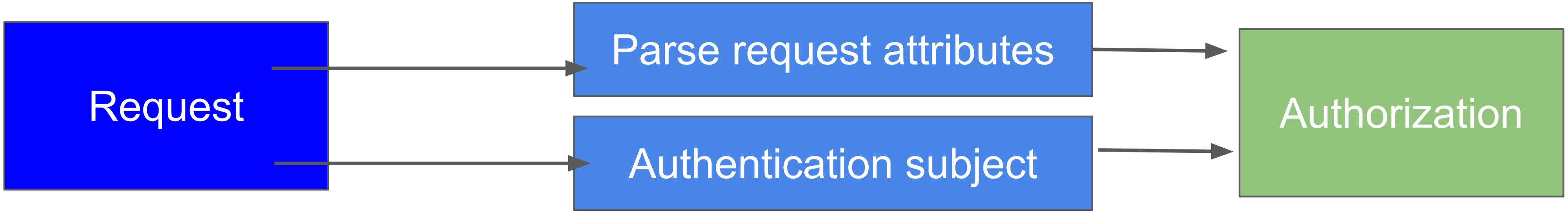
Accept: application/json

{“id”: 123, “code”: “mantap jiwa”.......

Verb	get
API group	user
User	john
resource	orders

user_id	123124
username	john

Request Handling



Can john in group normal_user

get

order for john?

Role

A role contains rules that represent a set of permissions

Role: Rider
Permissions:
- create:order
- get:order

Role: Driver
Permissions:
- accept:order
- get:order

Role: Driver
Permissions:
- accept:order
- get:order
- assign:order

Role

```
apiVersion:
rbac.authorization.k8s.io/v1
kind: Role
metadata:a
  namespace: default
  name: pod-reader
rules:
- apiGroups: [""] # core API group
  resources: ["pods"]
  verbs: ["get", "watch", "list"]
```

ClusterRole

```
apiVersion:
rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: secret-reader
rules:
- apiGroups: [""]
  resources: ["secrets"]
  verbs: ["get", "watch", "list"]
```

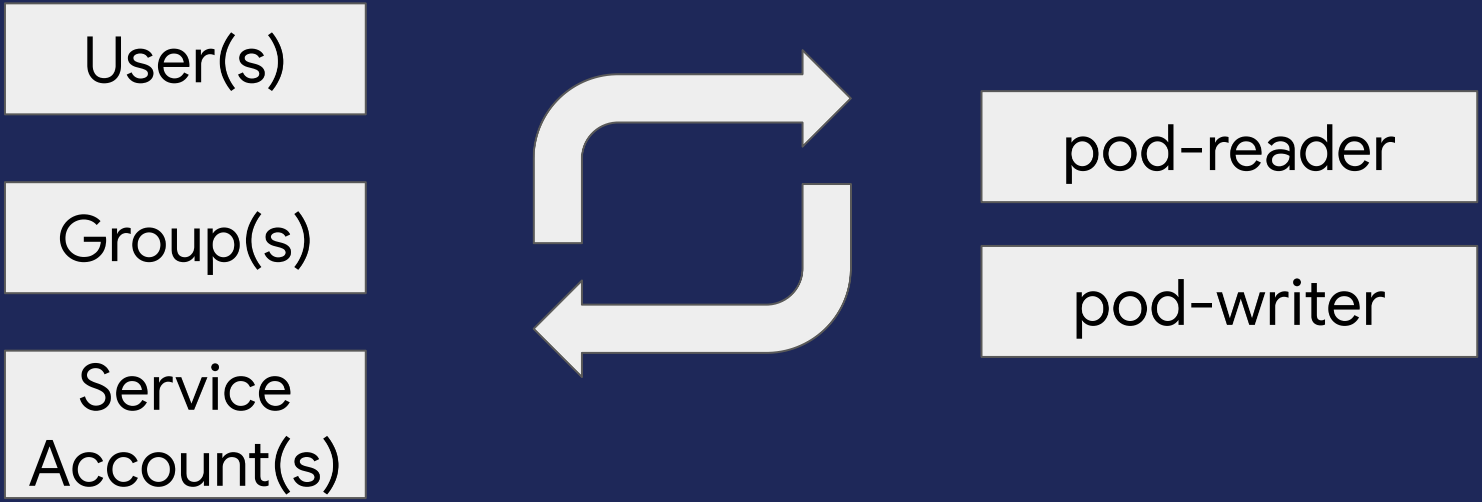
Role Binding

- A role binding grants the permissions defined in a role to a user or set of users.
- It holds a list of subjects (users, groups, or service accounts), and a reference to the role being granted

Subject

A role binding grants the permissions defined in a role to a user or set of users. It holds a list of subjects (users, groups, or service accounts), and a reference to the role being granted

Binding



User(s)

Group(s)

Service Account(s)

pod-reader

pod-writer

RoleBinding

```
apiVersion: rbac.authorization.k8s.io/v1
namespace.
kind: RoleBinding
metadata:
  name: read-pods
  namespace: default
subjects:
- kind: User
  name: jane@google.com # Name is case sensitive
  apiGroup: rbac.authorization.k8s.io
roleRef:
  kind: Role #this must be Role or ClusterRole
  name: pod-reader # must match Role or ClusterRole Name
  apiGroup: rbac.authorization.k8s.io
```

Extending RoleBinding with ClusterRole

```
apiVersion: rbac.authorization.k8s.io/v1
namespace.
kind: RoleBinding
metadata:
  name: read-pods
  namespace: development
subjects:
- kind: User
  name: dave@google.com # Name is case sensitive
  apiGroup: rbac.authorization.k8s.io
roleRef:
  kind: ClusterRole
  name: secret-reader
  apiGroup: rbac.authorization.k8s.io
```

ClusterRoleBinding

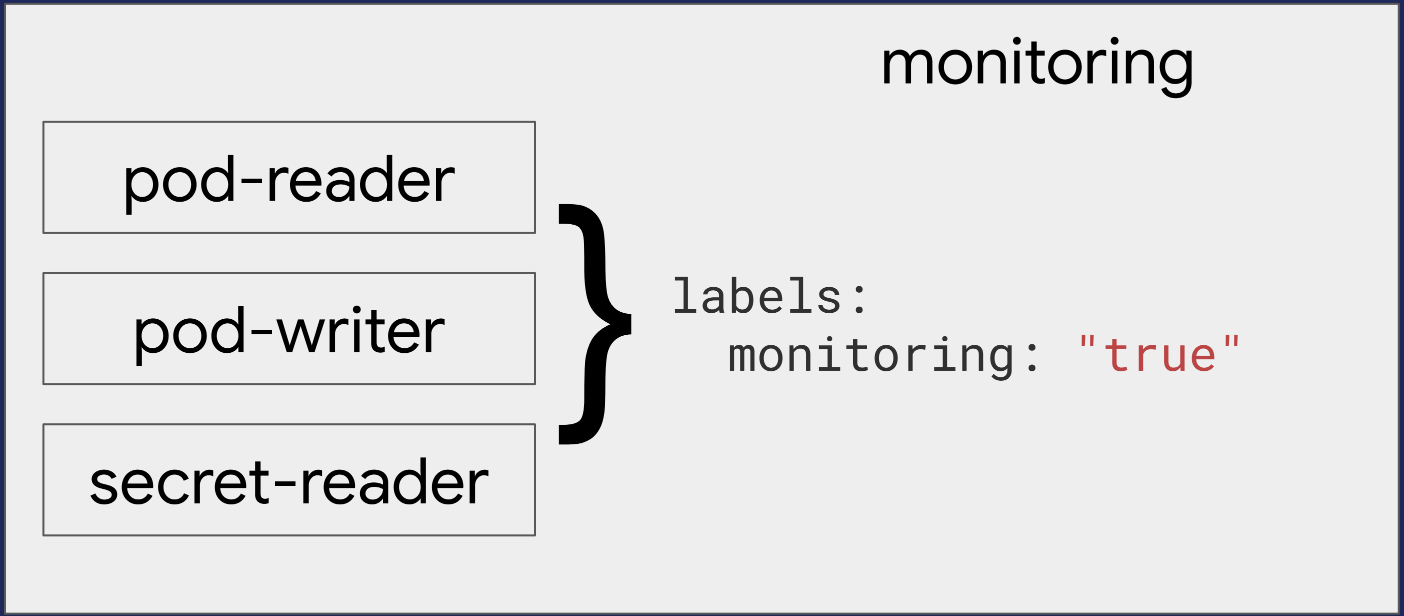
```yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: read-secrets-global

subjects:
- kind: Group
  name: managers@google.com
  apiGroup: rbac.authorization.k8s.io

roleRef:
  kind: ClusterRole
  name: secret-reader
  apiGroup: rbac.authorization.k8s.io
```

roleRef CANT BE EDITED ONCE IT IS DEPLOYED! MUST BE DELETED FIRST!

Aggregated ClusterRole



```
labels:
  monitoring: "true"
```

Aggregated ClusterRole

```yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: monitoring
aggregationRule:
  clusterRoleSelectors:
  - matchLabels:
      monitoring: "true"
rules: [] # Rules are automatically filled in by the controller
manager.
```

Aggregated ClusterRole (cont’d)

```yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: monitoring-endpoints
  labels:
    monitoring: "true"
rules:
- apiGroups: [""]
  resources: ["services", "endpoints", "pods"]
  verbs: ["get", "list", "watch"]
```

More on Kubernetes Docs!