JCConf Taiwan 2021

Access Kubernetes API in Java

Matt Ho

Hi, I'm Matt 👋

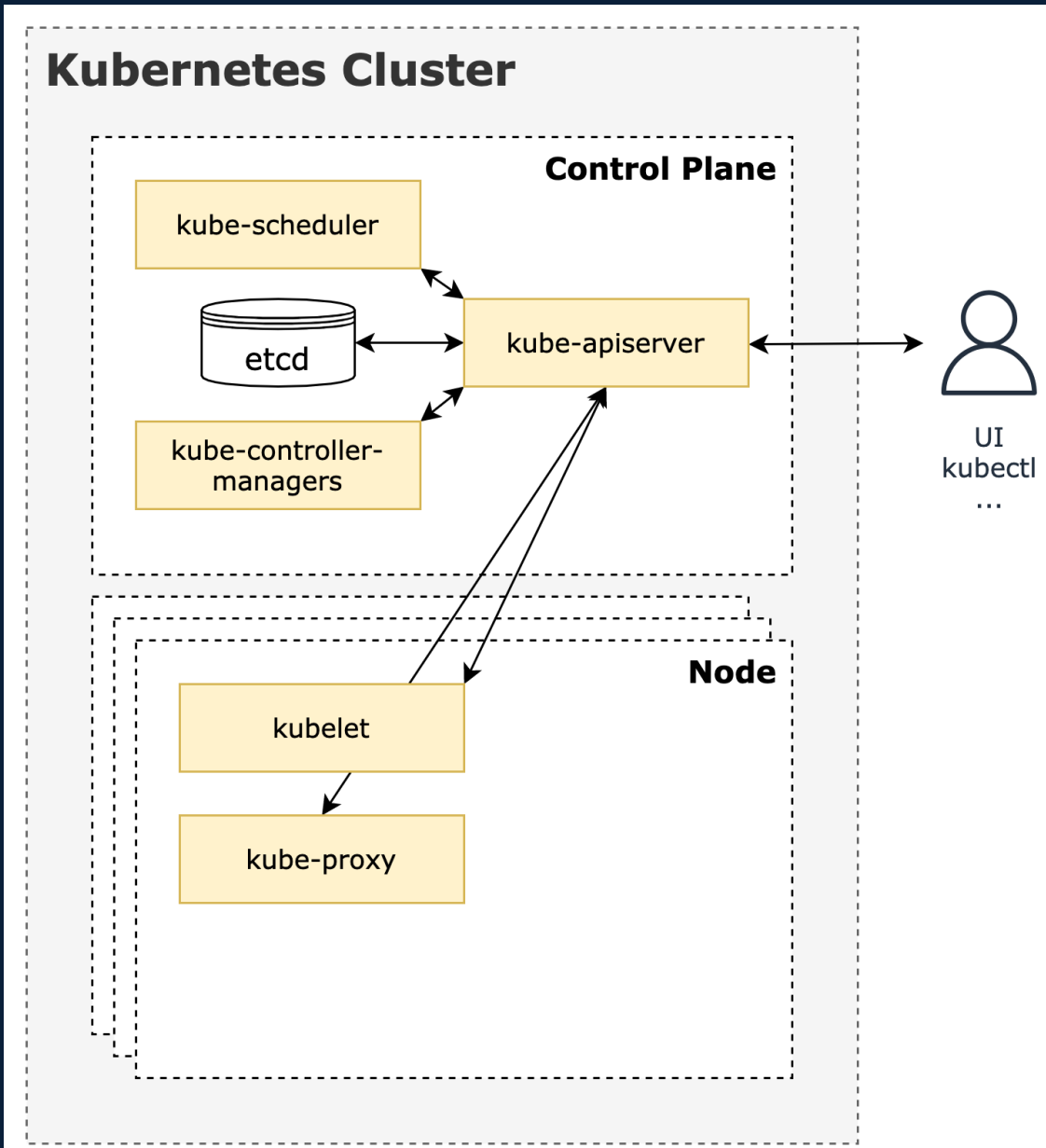
- https://github.com/shihyuho
- methodho@gmail.com
- @SoftLeader

Requirements

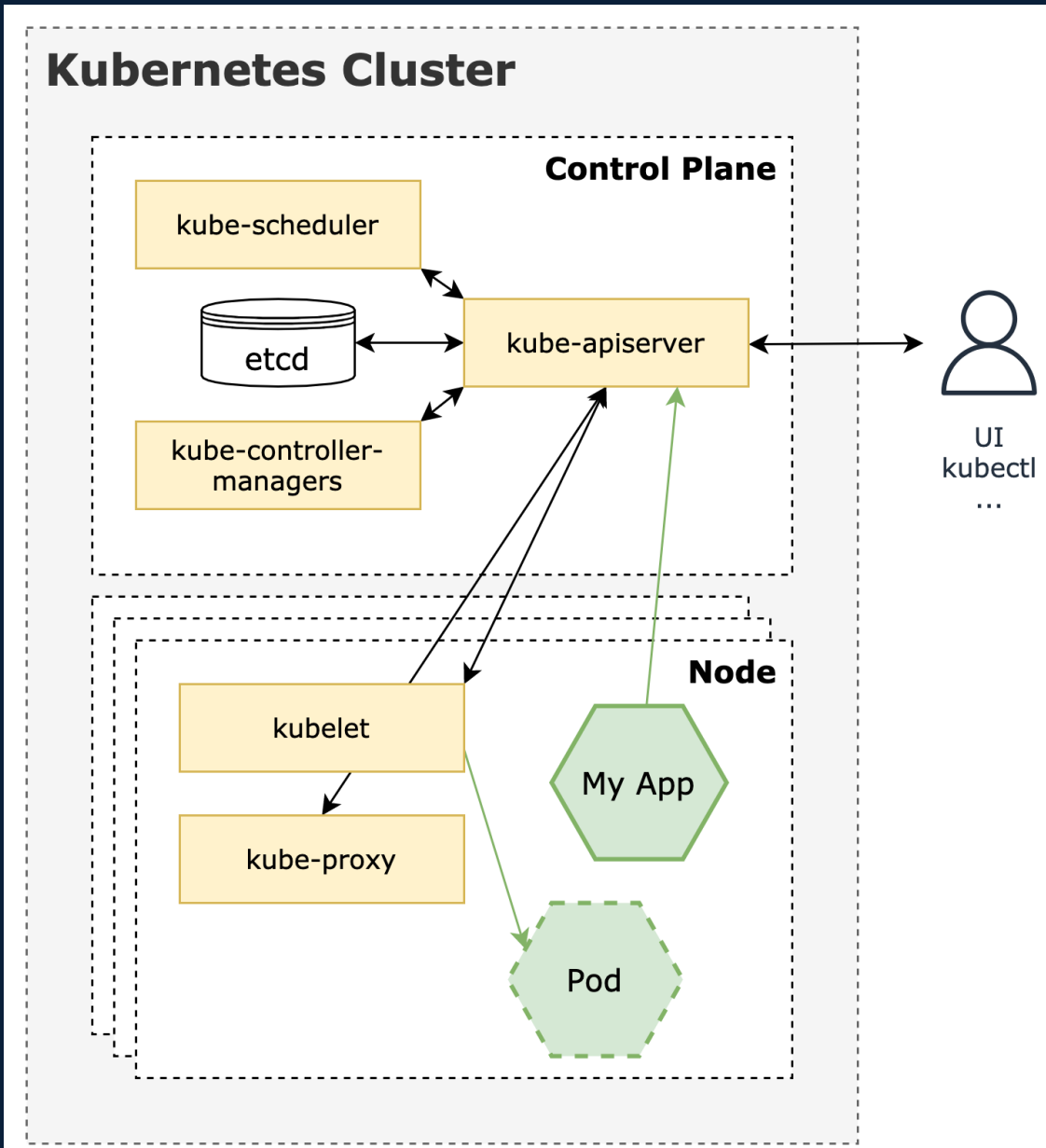
- Some experience with Java and Spring.
- Basic understanding of Kubernetes.
- YAML language.

你爲什麼會需要去跟 Kubernetes 互動?

Kubernetes Architecture



Kubernetes Architecture



環境準備

Local Kubernetes cluster 推薦: docker + minikube

```
# Start the cluster
$ minikube start

# Configure environment to use minikube’s Docker daemon
$ eval $(minikube docker-env)

# Halt the cluster
$ minikube stop
```

環境準備

A simple web app w/ Spring Boot

```
$ curl https://start.spring.io/starter.zip \
    -d dependencies=web,lombok,devtools \
    -d bootVersion=2.5.7 \
    -o demo.zip
```

環境準備

Add the following dependency to your `pom.xml` file:

```
<dependency>
    <groupId>io.fabric8</groupId>
    <artifactId>kubernetes-client</artifactId>
    <version>5.10.1</version>
</dependency>

<!-- Optional -->
<dependency>
    <groupId>org.springdoc</groupId>
    <artifactId>springdoc-openapi-ui</artifactId>
    <version>1.5.12</version>
</dependency>
```

Kubernetes Java Client

- Officially-supported - kubernetes-client/java
- Community-maintained - fabric8io/kubernetes-client

起手式

```
try (var client = new DefaultKubernetesClient()) {
    client.{apiGroup}.{apiVersion}.{resource}.{verb}...
}
```

```
kubectl get pod -n default
kubectl get service -A
kubectl get deploy -l my=label
kubectl get cronjob myjob
```

```
client.pods().inNamespace("default")
client.services().inAnyNamespace()
client.apps().deployments().withLabel("my", "label")
client.batch().v1beta1().cronjobs().withName("myjob")
```

A Hello Pod

```
apiVersion: v1
kind: Pod
metadata:
  name: hello
spec:
  containers:
  - name: hello
    image: busybox
    imagePullPolicy: IfNotPresent
    command: ["sh", "-c", "echo Hello JCConf Taiwan; sleep 2"]
  restartPolicy: Never
```

Builder Pattern

```
new {Resource}Builder()
    .withNew{FieldObject}
        .with{Field}(...)
        .with{Field}(...)
    .end{FieldObject}
.build();
```

Packing Image

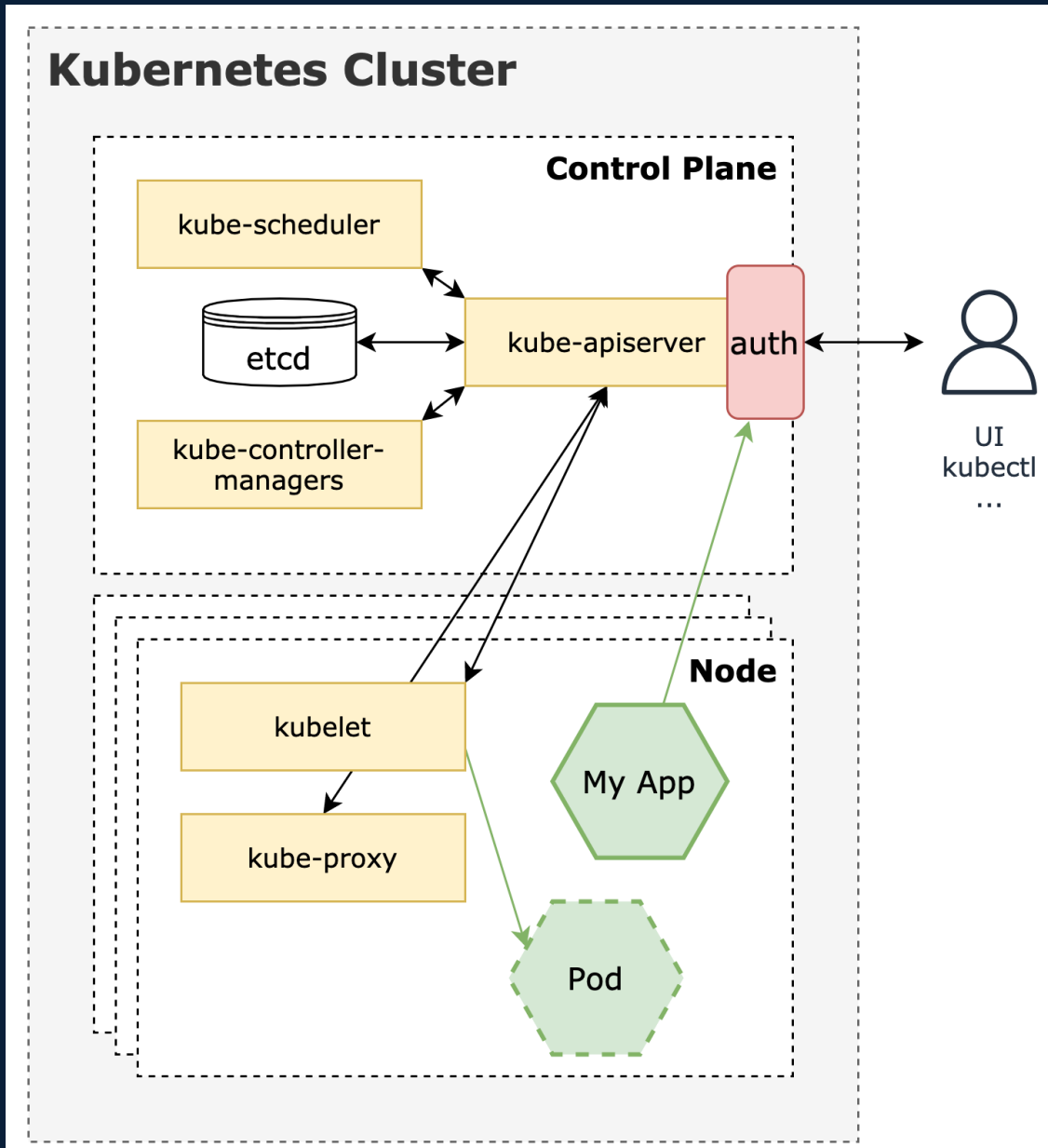
```
mvn compile com.google.cloud.tools:jib-maven-plugin:3.1.4:dockerBuild -Djib.to.image=demo:1.0.0

# mvn spring-boot:build-image -Dspring-boot.build-image.imageName=demo:1.0.0
```

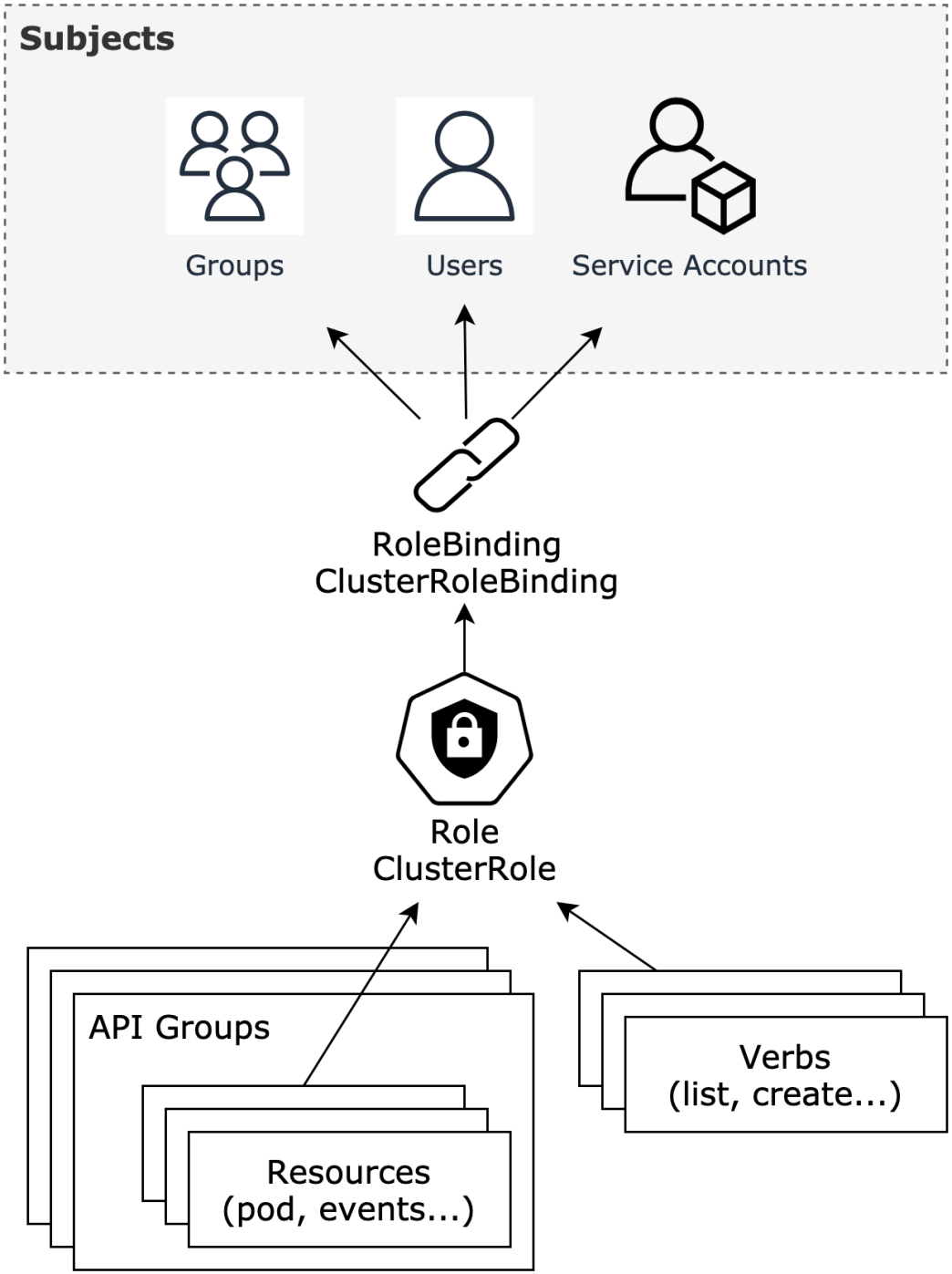
Deploy

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: demo
spec:
  selector:
    matchLabels:
      app: demo
  template:
    metadata:
      labels:
        app: demo
    spec:
      containers:
      - name: demo
        image: demo:1.0.0
```

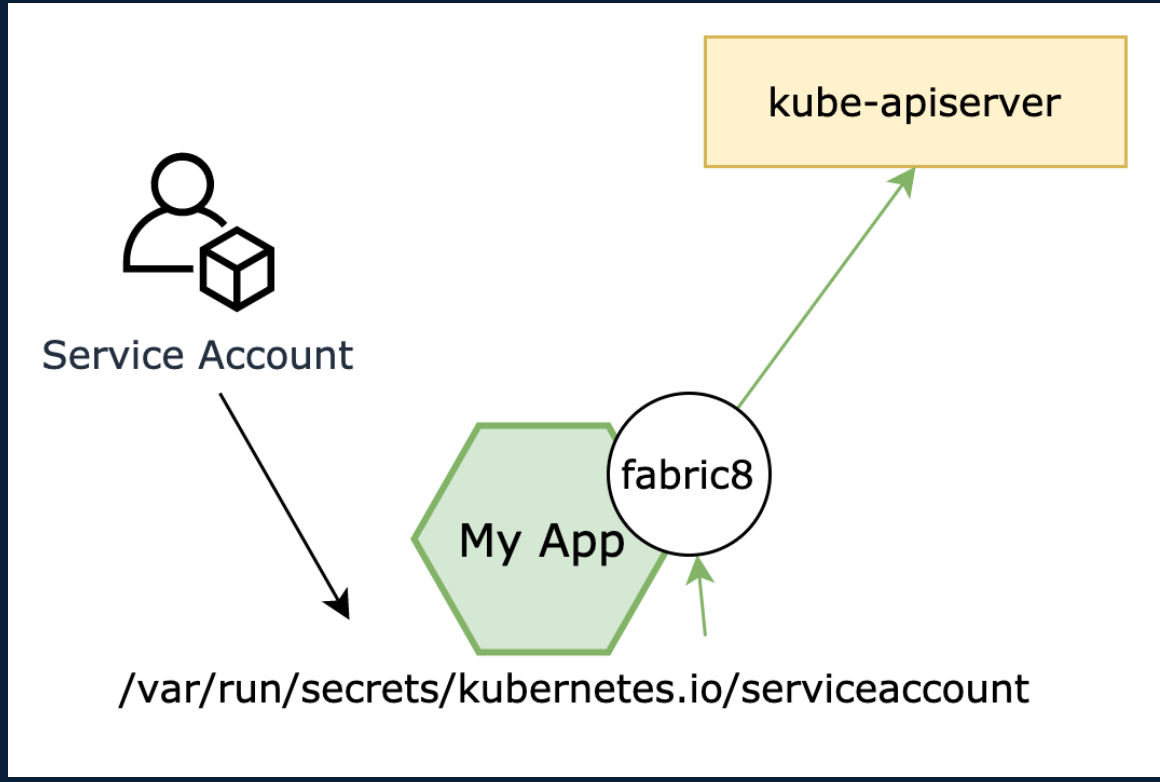
Kubernetes RBAC



Kubernetes RBAC



Kubernetes RBAC



```yaml
apiVersion: v1
kind: ServiceAccount
metadata:
  name: demo
  namespace: default
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: demo
rules:
  - apiGroups: [ "" ]
    resources: [ "pods" ]
    verbs: [ "get", "list", "watch", "create", "update", "patch", "delete" ]
  - apiGroups: [ "" ]
    resources: [ "events" ]
    verbs: [ "list" ]
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: demo
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: demo
subjects:
  - kind: ServiceAccount
    name: demo
    namespace: default
```

Recap

- Basic understanding of Kubernetes API.
- How to access Kubernetes API in Java.
- How to configure access control to the app.
- Demo code.

Thank you 🙌