harishnshetty / **Kubernetes-From-scratch-Project**

Code   Issues   Pull requests   Actions   Projects   Wiki   Security   Insights   Settings

main ▾   Kubernetes-From-scratch-Project / **8.configmap** /

Go to file   t   Add file ▾   ···

harishnshetty config map readme added   5a641dc · now

Name	Name	Last commit date
..		
Readme.MD	config map readme added	now
confgmap.yml	config map readme added	now
secrets.yml	config map readme added	now

Readme.MD

🚀 Project Name

Example: `k8s-configmap-secret-demo`

> A Kubernetes application demonstrating best practices for ConfigMaps, Secrets, and Deployments using YAML manifests.

📌 Table of Contents

- Overview
- Architecture Diagram
- Folder Structure
- Prerequisites
- Kubernetes Manifests
    - ConfigMap
    - Secret
    - Deployment
- Deployment Steps
- Verification
- Cleanup

- Best Practices

🧩 Overview

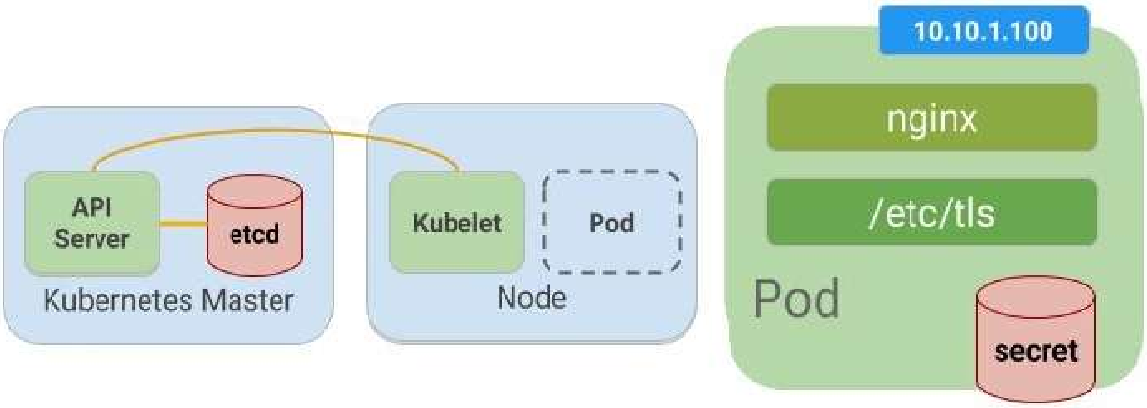
This project demonstrates:

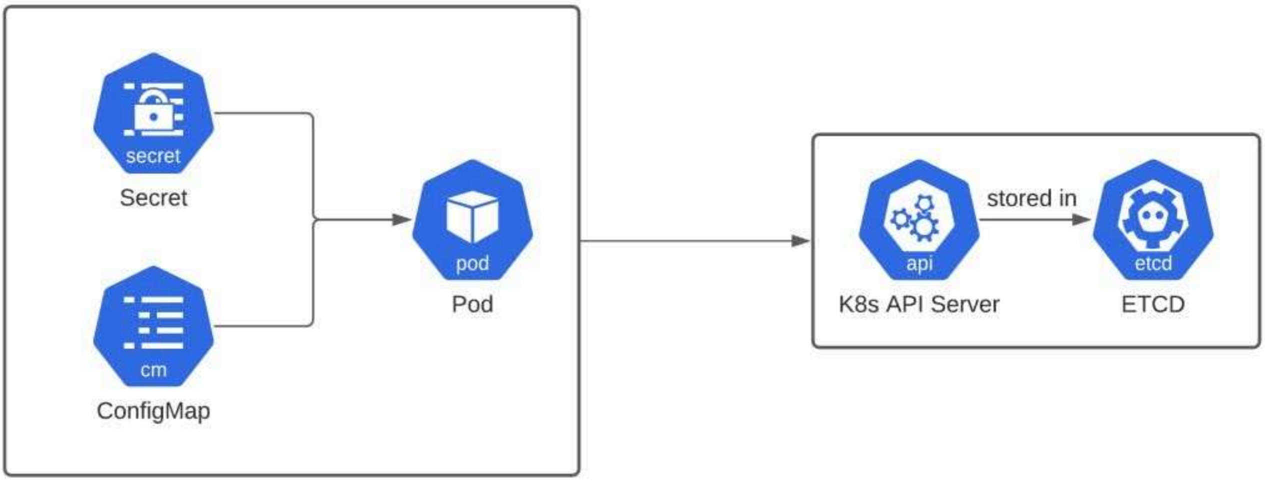
- Externalized configuration using **ConfigMaps**
- Secure credential management using **Secrets**
- Application deployment using **Deployments**
- Environment variable injection into containers

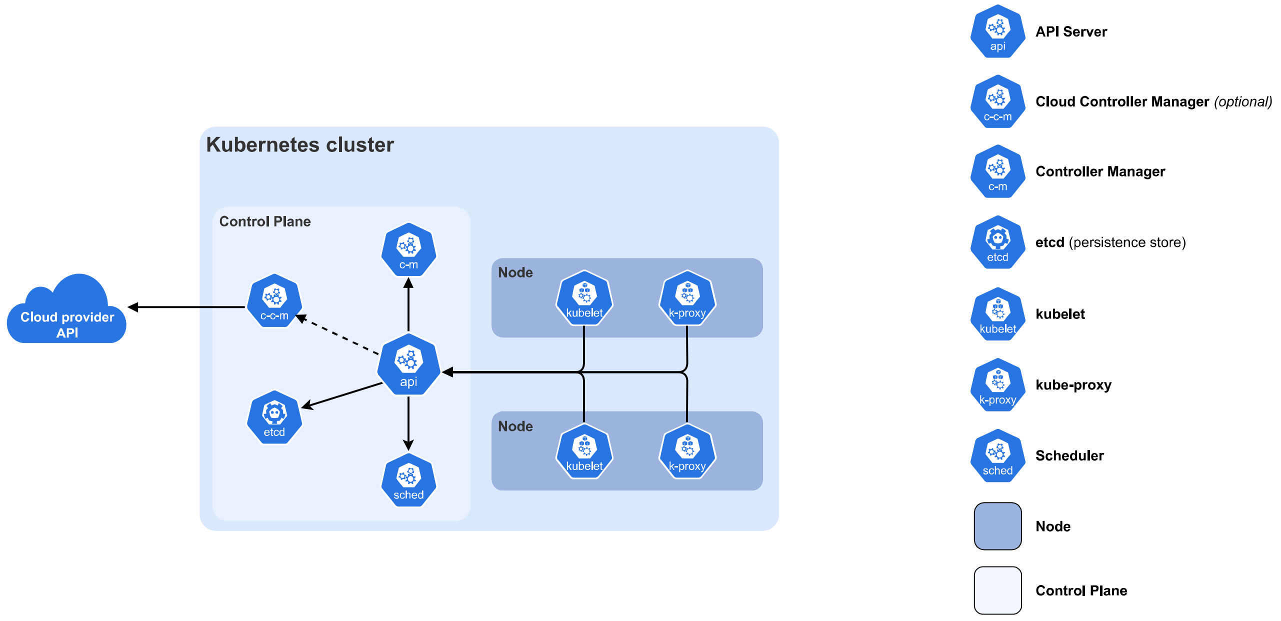
Use case:

> Ideal for **two-tier or microservice apps** running on Kubernetes clusters (Minikube, EKS, AKS, GKE).

🏗 Architecture Diagram







Architecture Flow

```
┌─────────────┐
│  User/App   │
└──────┬──────┘
       │
┌──────▼──────────────────┐
│ Kubernetes Deployment   │
│ ┌──────────────────────┐ │
│ │ Pod                  │ │
│ │  ├─ ConfigMap        │ │
│ │  ├─ Secret           │ │
│ │  └─ Container App    │ │
│ └──────────────────────┘ │
└─────────────────────────┘
```

📁 Folder Structure

```
.
├── README.md
├── k8s
│   ├── configmap.yml
│   ├── secret.yml
│   └── deployment.yml
```

⚙️ Prerequisites

- Kubernetes Cluster (Minikube / EKS / Kind)
- kubectl configured
- Docker image available

Check:

```
kubectl get nodes
```

📦 Kubernetes Manifests

🔹 ConfigMap (`configmap.yml`)

```yaml
apiVersion: v1
kind: ConfigMap
metadata:
  name: app-config
data:
  APP_ENV: "production"
  APP_PORT: "8080"
```

🔹 Secret (`secret.yml`)

```yaml
apiVersion: v1
kind: Secret
metadata:
  name: app-secret
type: Opaque
data:
  DB_USERNAME: YWRtaW4=     # admin
  DB_PASSWORD: cGFzc3dvcmQ= # password
```

> 🔐 Secrets must be base64 encoded:

```bash
echo -n admin | base64
```

🔹 Deployment (`deployment.yml`)

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: demo-app
spec:
  replicas: 2
  selector:
    matchLabels:
      app: demo
  template:
    metadata:
      labels:
        app: demo
    spec:
      containers:
      - name: demo-container
        image: nginx:latest
        ports:
        - containerPort: 80
        env:
        - name: APP_ENV
          valueFrom:
            configMapKeyRef:
              name: app-config
              key: APP_ENV
        - name: DB_USERNAME
          valueFrom:
            secretKeyRef:
              name: app-secret
              key: DB_USERNAME
```

🚀 Deployment Steps

Apply resources in order:

```
kubectl apply -f configmap.yml
kubectl apply -f secret.yml
kubectl apply -f deployment.yml
```

🔍 Verification

```
kubectl get pods
kubectl describe pod <pod-name>
```

Check environment variables:

```
kubectl exec -it <pod-name> -- env
```

🧹 Cleanup

```
kubectl delete -f deployment.yml
kubectl delete -f configmap.yml
kubectl delete -f secret.yml
```

✅ Best Practices

✔️ Never commit plaintext secrets ✔️ Use External Secrets in production (AWS Secrets Manager, Vault) ✔️ Separate manifests per environment (dev/stage/prod) ✔️ Use Helm or Kustomize for scaling ✔️ Enable RBAC and Network Policies