Kubernetes

Divyansh Rajesh Jain

UCDAVIS

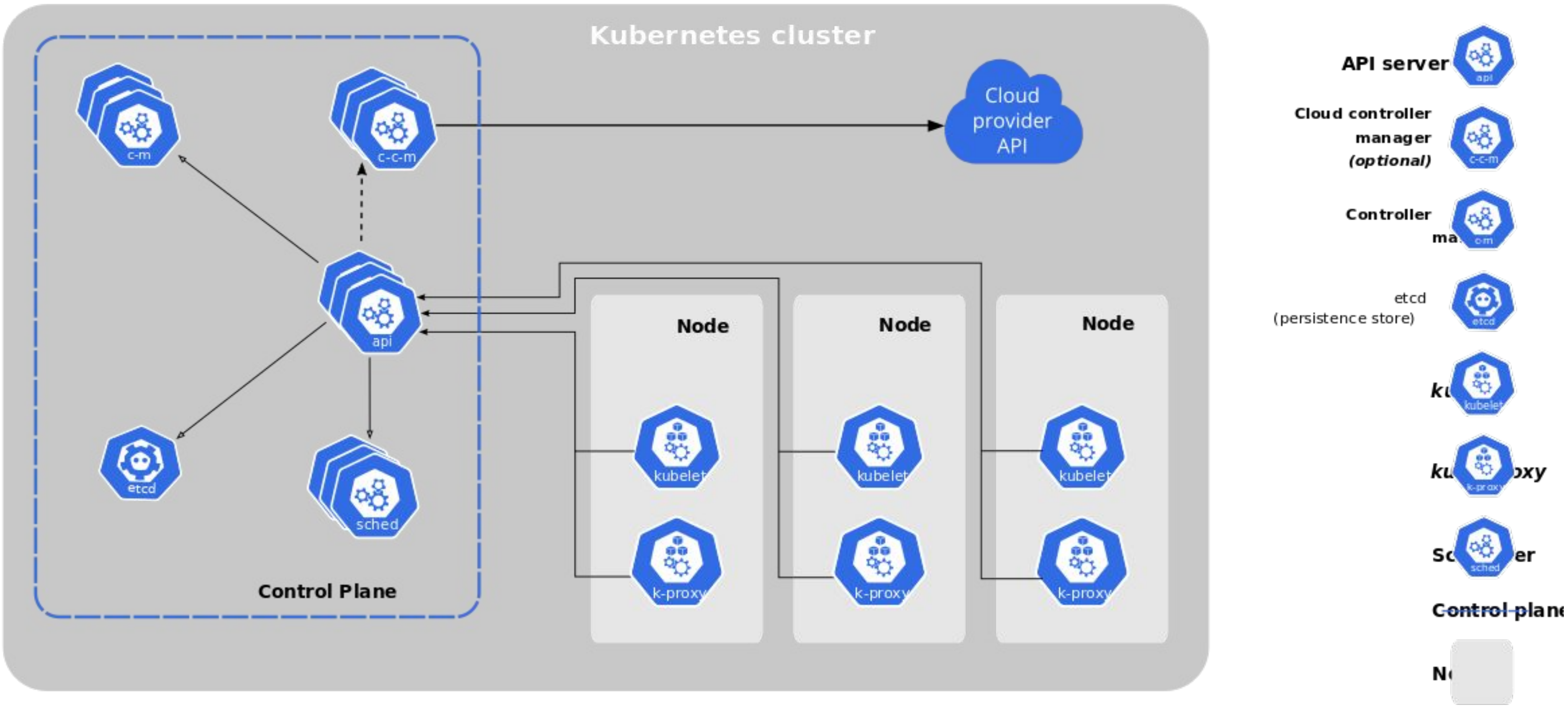
What is Kubernetes?

- Container Orchestration Platform
  - OCI Compliant Images
- Standardized Development and Deployment
  - Virtualizes them to look same
- Allows fast deployment
  - Tools for allow quick & “invisible” updates

High Level Features

- Allow for scalability
    - Deploy many containers of the same image
- Allow for logical grouping
    - Load Balancing for containers
- Self-healing and auto scaling
    - How does Kubernetes self-repair services?

High Level Architecture



Basic Components

- Pod → Smallest unit in Kubernetes
  - Represents a container in Kubernetes
- Deployment → Grouping of Pods
  - Used to start/update all pods originating from a specific image

Basic Components

- Persistent Volume → Define persistent storage for Deployment
  - Allows you to store data that outlives Deployment
- Persistent Volume Claim → Defines persistent storage mapping for cloud resources
  - Allows you to link to the same volume after every restart

Basic Components

- Secrets → Define any sort of “secret” information
  - Used to specific registry secrets to pull images
- ConfigMaps → Application level configuration
  - Key-value store that applications can read from and write to

Networking in Kubernetes

- Namespaces
  - Allow “logical” clusters of pods. Can communicate within themselves (Private Network)
- Service
  - Allow to expose a Deployment “externally”

Why Kubernetes?

- Portable Configuration
    - Can change cloud providers to any Kubernete Complaint Cloud
- Allow for containerization deployment
    - What is the advantage of this?
- Resource allocation is done optimally (Why?)

Common Use Cases

- Microservice Orchestration
    - Deploy Microservices effectively with namespaces
- Multi-Cloud Deployment
    - Abstract Cloud Resources, so Multi-Cloud is very possible
- AI/ML Deployment
    - Dynamic Resource Allocation → Very important for AI

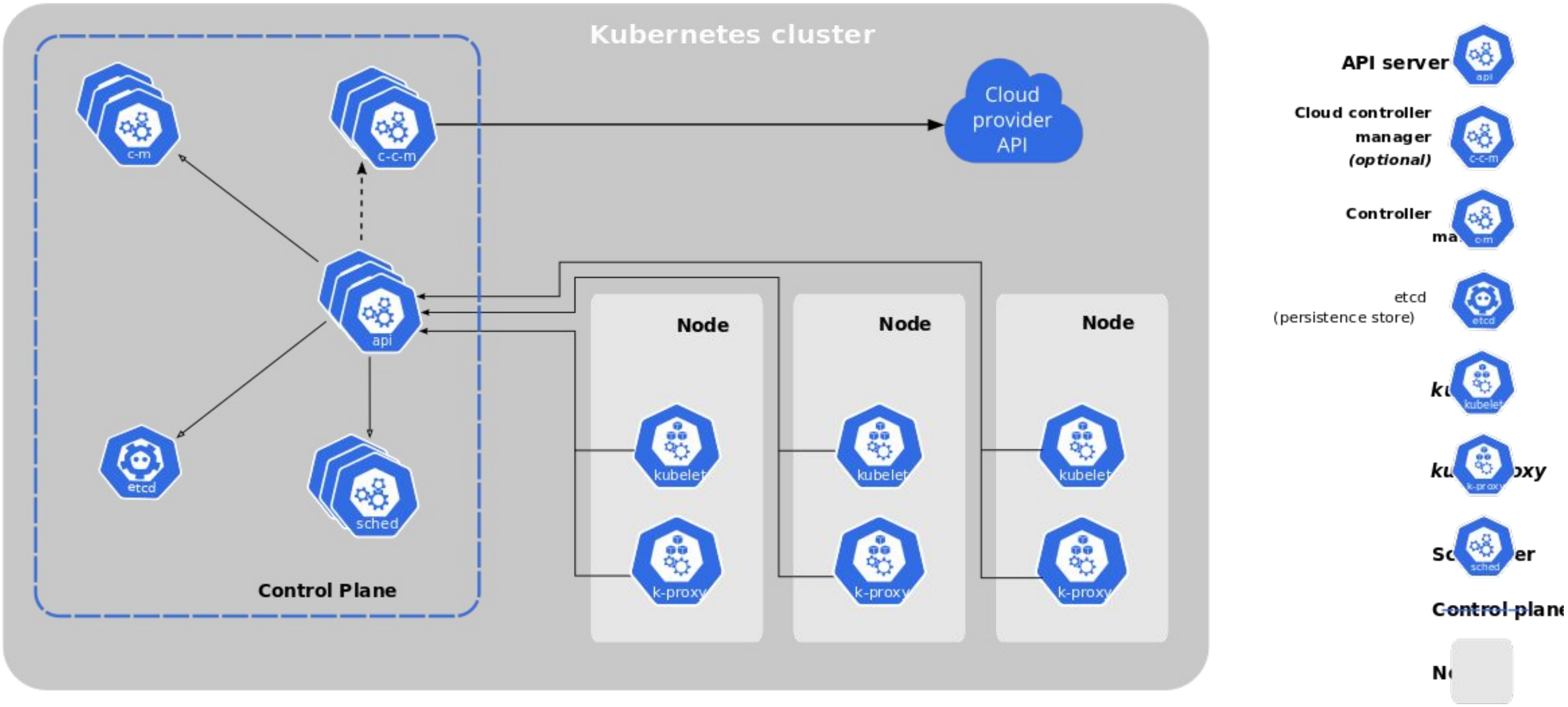
Challenges of Kubernetes

- Very complex to set up
  - Explicitly configure every detail
- kompose → Tool for converting Docker Compose configuration to Kubernetes Configuration
  - Requires very little modification after!

Important Kubernetes Tools

- Kubectl → CLI Client for Kubernetes
  - Interacts with Control Plane for you
- Usually some custom tools are used
  - Automated deployment using Kubernetes API at the end of CI/CD pipelines

Tying back everything



Thank You!

Next Time: Intro to Structured Development