🧪 Module 2 Lab – Kubernetes Fundamentals: Cluster & API Exploration

This lab helps you discover how to interact with a Kubernetes cluster using `kubectl`, identify cluster components, and inspect available API resources.

🌟 Objectives

- Create a local Kubernetes cluster with `kind`
- Explore the cluster with `kubectl`
- Identify control plane and worker components
- Use the Kubernetes API to discover resource definitions and usage

---

🚀 Step 0: Create a Kind Cluster

Before starting, if you don't have already created the cluster in the previous module, make sure Docker is running and the following configuration is used to create your Kind cluster:

`kind-config.yaml`

```yaml
kind: Cluster
apiVersion: kind.x-k8s.io/v1alpha4
name: cka-lab
nodes:
  - role: control-plane
    image: kindest/node:v1.32.0
```

Then create the cluster:

```
kind create cluster --config kind-config.yaml
```

> 📝 *This configuration uses `containerd` as the runtime and a basic default CNI setup.*

🔍 Step 1: Verify Cluster Access

```
kubectl cluster-info
kubectl get nodes -o wide
kubectl version
```

> *Make sure you have a running cluster and `kubectl` is configured correctly.*

🔍 Step 2 – Explore API Resources via kubectl proxy

Start the proxy:

```
kubectl proxy
```

Then open your browser and explore:

- http://localhost:8001/
- http://localhost:8001/api/v1
- http://localhost:8001/apis

Use curl for raw access:

```
curl http://localhost:8001/apis/apps/v1
```

> *These endpoints let you browse live Kubernetes API structure and resources.*

✅ Checklist

- □ Verified cluster access with `kubectl`
- □ Explored API resources via `kubectl proxy` and HTTP APIs
- □ Cluster cleaned up after completion

💬 What’s Next?

You're now familiar with your Kubernetes environment and the API. In the next module, you'll begin deploying workloads and managing cluster objects.

> *Note: CNI, CRI, and CSI interfaces will be covered in a **dedicated lab**.*