# Experiment No. 05

**Aim: To illustrate the Kubernetes Cluster Architecture, install and spin up a Kubernetes Cluster on linux machines or cloud platforms.**

**LOs:**

**Theory:**

This process involves understanding the components of a Kubernetes cluster and then using a tool like Minikube to build one on a cloud server.

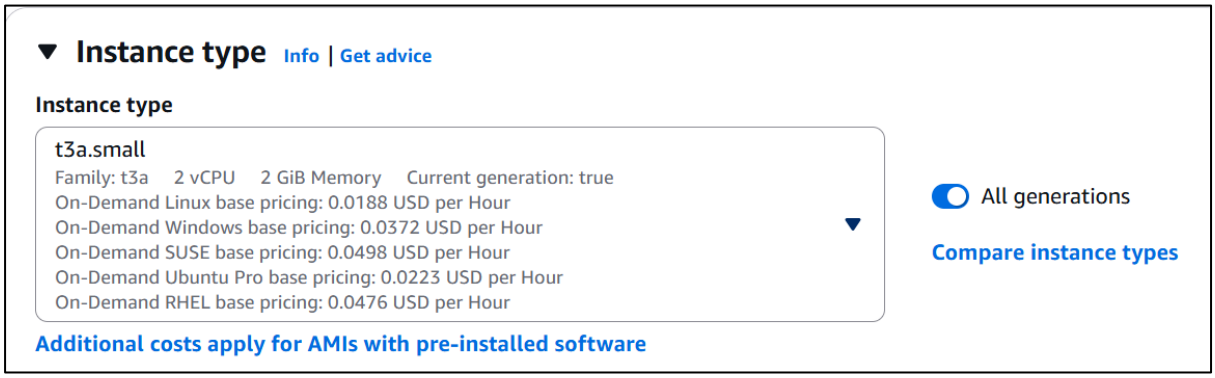
**Understanding the Architecture**

A Kubernetes cluster is composed of a **Control Plane** (the brain) and one or more **Worker Nodes** (the muscle).

- **Control Plane:** Manages the cluster and makes global decisions. Its key components are:
  - **API Server:** The front door for all communication. kubectl talks to this.
  - **etcd:** The cluster's memory; a reliable key-value store for all cluster data.
  - **Scheduler:** Decides which Worker Node will run a new application (Pod).
  - **Controller Manager:** A watchdog that ensures the cluster's desired state matches its actual state.
- **Worker Nodes:** These are the machines where your applications actually run. Each node has:
  - **Kubelet:** An agent that communicates with the Control Plane and ensures containers are running as they should be on its node.
  - **Kube-proxy:** Manages network rules on the node, allowing pods to communicate.
  - **Container Runtime:** The software that runs the containers (e.g., Docker).

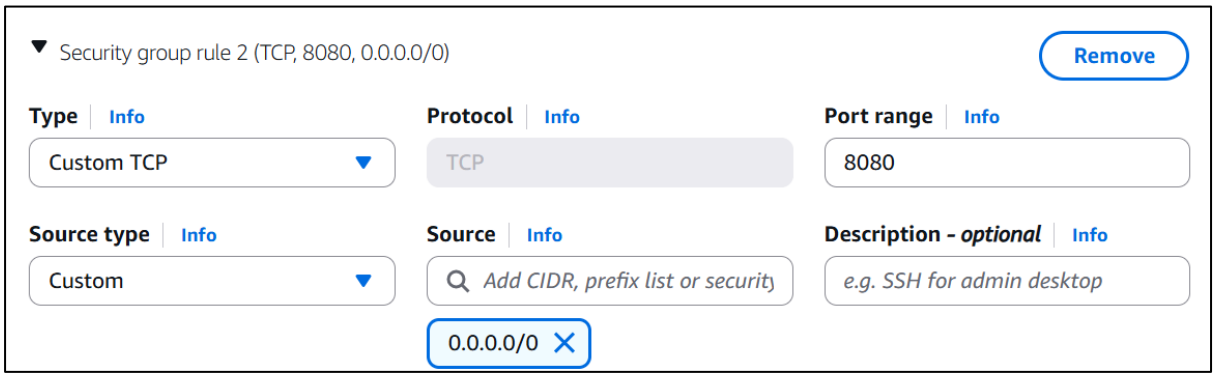
**Step-by-Step Installation and Cluster Creation**

Here are the commands to install the necessary tools and spin up a cluster on an Ubuntu server (like an AWS EC2 instance, **Instance type** should be **t3a.small or larger**).



**Steps for Network Settings**

- During the "Launch an instance" process, when you get to the **Network settings** section, click the **Edit** button.
- Click **Add security group rule**.




## In EC2 instance's terminal

1. **Prepare the Server:** First, update your system's package list.

   sudo apt-get update && sudo apt-get upgrade -y

```
ubuntu@ip-172-31-24-113:~$ sudo apt-get update && sudo apt-get upgrade -y
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Reading package lists... Done
Reading package lists... Done
Building dependency tree... Done
```

2. **Install a Container Runtime (Docker):** Kubernetes needs a container runtime to manage containers. We'll install Docker and grant your user permission to use it.

   sudo apt-get install docker.io -y
   sudo usermod -aG docker $USER
   newgrp docker

```
ubuntu@ip-172-31-24-113:~$ sudo apt-get install docker.io -y
sudo usermod -aG docker $USER
newgrp docker
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
docker.io is already the newest version (27.5.1-0ubuntu3~24.04.2).
0 upgraded, 0 newly installed, 0 to remove and 3 not upgraded.
```

3. **Install Minikube:** Minikube is the tool that will create a single-node Kubernetes cluster for you, perfect for learning.

   curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64
   sudo install minikube-linux-amd64 /usr/local/bin/minikube

```
ubuntu@ip-172-31-24-113:~$ curl -LO https://storage.googleapis.com
/minikube/releases/latest/minikube-linux-amd64
sudo install minikube-linux-amd64 /usr/local/bin/minikube
  % Total    % Received % Xferd  Average Speed   Time    Time
Time  Current
                                 Dload  Upload   Total   Spent
Left  Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --
 58  133M   58 78.3M    0     0   144M      0 --:--:-- --:--:-- --
100  133M  100  133M    0     0   165M      0 --:--:-- --:--:-- --
:--:--  165M
```

4. **Spin Up the Kubernetes Cluster:** This command downloads all the necessary Kubernetes components and starts your cluster.

   minikube start

```
ubuntu@ip-172-31-45-31:~$ minikube start
* minikube v1.37.0 on Ubuntu 24.04
* Automatically selected the docker driver. Other choices: none, s
sh

X The requested memory allocation of 1916MiB does not leave room f
or system overhead (total system memory: 1916MiB). You may face st
ability issues.
* Suggestion: Start minikube with less memory allocated: 'minikube
 start --memory=1916mb'
```

5. **Verify the Cluster:** Check that your cluster is running and ready.

   minikube status

```
ubuntu@ip-172-31-45-31:~$ minikube status
minikube
type: Control Plane
host: Running
kubelet: Running
apiserver: Running
kubeconfig: Configured
```