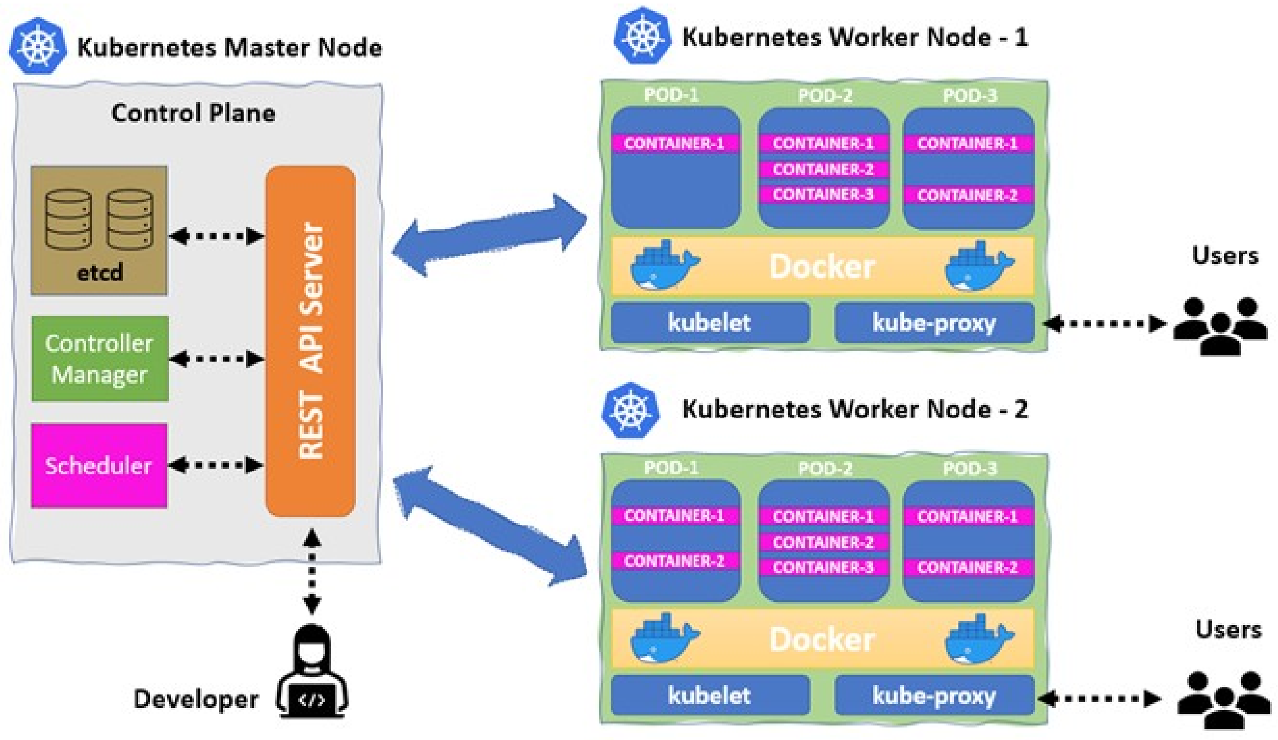
Setting up a Kubernetes cluster

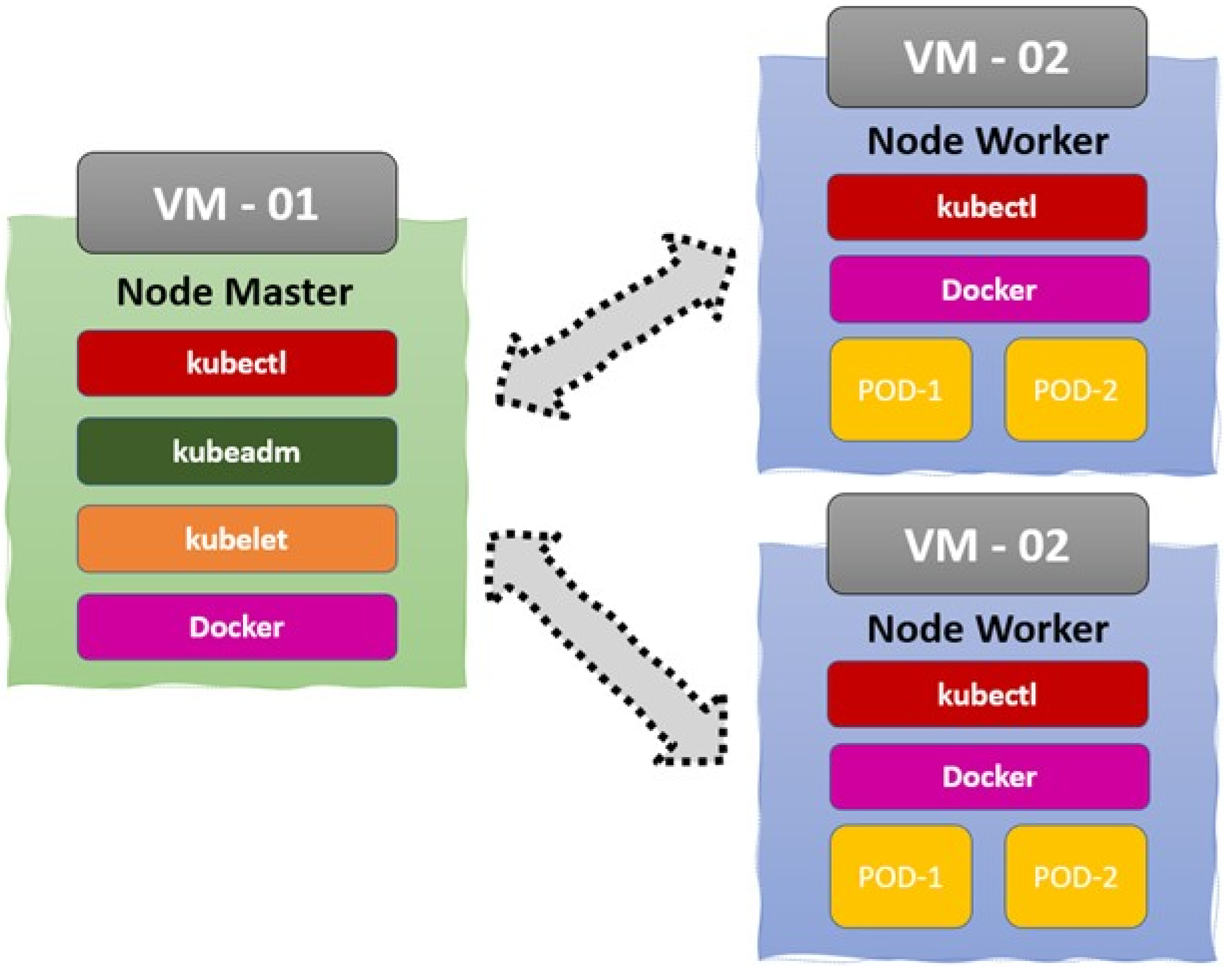
Kubernetes is an open-source platform for governing clusters of containerized application services. Kubernetes automates the vital aspects of container lifecycle management, including scaling, replication, monitoring, and scheduling.

The central component of Kubernetes is a cluster, which is itself made up of multiple physical or virtual machines. Each cluster component performs a specific function as either a master or a worker—a master controls and manages the containers in the nodes, while a worker hosts the groups of one or more containers.



This feature-rich cluster contains six key components, namely an API server, scheduler, controller, etcd, kubelet, and kube-proxy. Through this lens, the first four components run on the master, whereas the rest of the functions run on the worker.

kubeadm builds a minimum viable, production-ready Kubernetes cluster that conforms to best practices. It also allows us to choose the container runtime, though it has Docker by default. This solution requires a minimum of two VMs to run the master and worker.



Now, let’s delve into the installation steps. They’ll be illustrated from the root user-space, so if you’re trying them from non-root user-space, add sudo before every command.

Prerequisites

1. 2 or more Linux Server running Ubuntu 18.04 (Min Req. 2GB of RAM and 2 CPUs.)
2. Access to a user account on each system with **sudo** or root privileges
3. Open Inbound and Outbound Rules on both Machines

Container Runtime: Docker Installation

Kubernetes Requires an Existing Docker installation. If you already have Docker skip to the next step.

Step 1: Update the package list with the Command

```
$ sudo apt-get update
```

Step 2: Install Docekr with Command:

```
$ sudo apt install docker.io
```

Step 3: Now check Docker Status

```
$ service docker status
```

Install Kubernetes

Before installing Kubernetes on Ubuntu, you should first run through a few prerequisite tasks to ensure the installation goes smoothly.

1. **Install transport-https and curl package using apt-get install the command. Transport-https package allows the use of repositories accessed via the HTTP Secure protocol, and curl allows you to transfer data to or from a server or download, etc.**

```
$ sudo apt-get update
$ sudo apt-get install -y apt-transport-https ca-certificates curl
```

2. **Add the GPG key for the official Kubernetes repository to your system using `curl` command.**

```
$ sudo curl -fsSLo /usr/share/keyrings/kubernetes-archive-keyring.gpg
   https://packages.cloud.google.com/apt/doc/apt-key.gpg
```

Note:- If you get an erroe Curl not installed you can install it using

```
$ sudo apt-get install curl
```

3. **Add the Kubernetes repository to APT sources and update the system.**

```
$ echo "deb [signed-by=/usr/share/keyrings/kubernetes-archive-keyring.gpg]
   https://apt.kubernetes.io/ kubernetes-xenial main" | sudo tee
   /etc/apt/sources.list.d/kubernetes.list
```

Note:- Repeat this step on both the machines

4. **Kubernetes Installation Tools**

   Now Install kubectl (which manages cluster), kubeadm(which starts cluster), and

   kubelt (which manages Pods and Container) on both the Machines

```
$ sudo apt-get update
$ sudo apt-get install -y kubelet kubeadm kubectl
$ sudo apt-mark hold kubelet kubeadm kubectl
```

5. **Initialize the cluster (run only on the master):**

```
$ sudo kubeadm init --pod-network-cidr=10.244.0.0/16
```

6. **Set up local `kubeconfig`**

```
 $ mkdir -p $HOME/.kube
 $ sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
 $ sudo chown $(id -u):$(id -g) $HOME/.kube/config
```

7. **`Apply Flannel CNI network overlay (or any other network as per need)`**

```
$ kubectl apply -f
$ https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-
flannel.yml
```

8. **`Join the worker nodes to the cluster`**

```
$ kubeadm join [your unique string from the kubeadm init command]
```

9. **`Verify the worker nodes have joined the cluster successfully`**

```
$ kubectl get nodes
```