

A node is a working machine in Kubernetes cluster which is also known as a minion. They are working units which can be physical, VM, or a cloud instance.

Each node has all the required configuration required to run a pod on it such as the proxy service and kubelet service along with the Docker, which is used to run the Docker containers on the pod created on the node.

They are not created by Kubernetes but they are created externally either by the cloud service provider or the Kubernetes cluster manager on physical or VM machines.

The key component of Kubernetes to handle multiple nodes is the controller manager, which runs multiple kind of controllers to manage nodes. To manage nodes, Kubernetes creates an object of kind node which will validate that the object which is created is a valid node.

Service with Selector

```
apiVersion: v1
kind: node
metadata:
  name: < ip address of the node>
  labels:
    name: <lable name>
```

In JSON format the actual object is created which looks as follows –

```
{
  Kind: node
  apiVersion: v1
  "metadata":
  {
    "name": "10.01.1.10",
    "labels"
    {
      "name": "cluster 1 node"
    }
  }
}
```

Node Controller

They are the collection of services which run in the Kubernetes master and continuously monitor the node in the cluster on the basis of metadata.name. If all the required services are running, then the node is validated and a newly created pod will be assigned to that node by the controller. If it is not valid, then the master will not assign any pod to it and will wait until it becomes valid.

Kubernetes master registers the node automatically, if **–register-node** flag is true.

```
–register-node = true
```

However, if the cluster administrator wants to manage it manually then it could be done by turning the flag of –

```
–register-node = false
```

