

Kubernetes on Utho Cloud

1. Introduction to Kubernetes on Utho Cloud Platform

Kubernetes is a powerful, open-source platform designed to automate deploying, scaling, and managing containerized applications. Utho Cloud provides managed Kubernetes services, making it easy to set up, manage, and scale Kubernetes clusters on the cloud

2. Creating and Managing a Kubernetes Cluster on Utho Cloud Platform

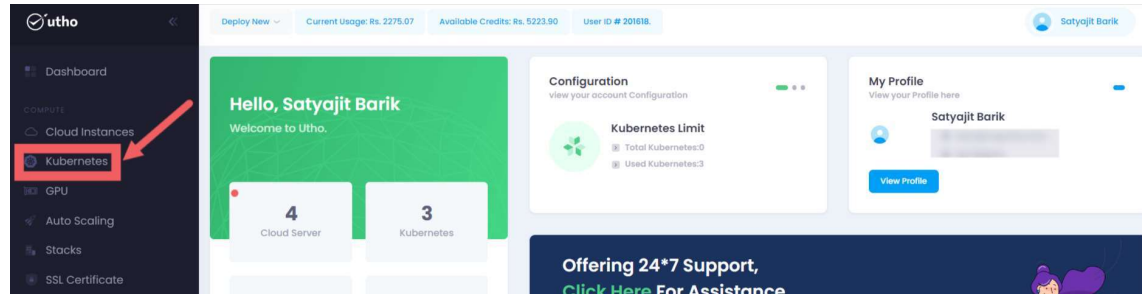
2.1 Creating a Kubernetes Cluster

To get started with Kubernetes on Utho Cloud, you first need to create a Kubernetes cluster.

1. Log in to Utho Cloud:

- Navigate to the [Utho Cloud Platform](#) dashboard and log in with your credentials.

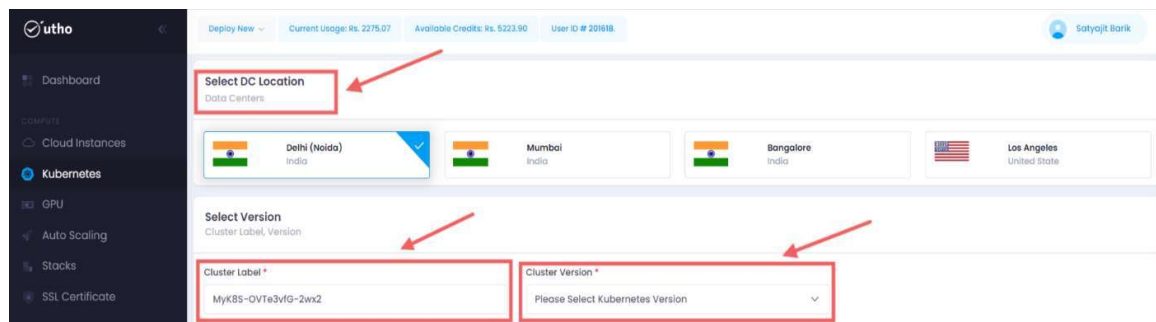
2. Navigate to Kubernetes Service:



- Go to **Cloud Services** → **Kubernetes** → Click on Kubernetes.

3. Create a New Kubernetes Cluster:

- Click **Create Cluster** and follow the on-screen prompts to configure the cluster. You will need to define:
 - **Data Centre:** Choose a region where you want to deploy your cluster.
 - **Select Version:** Fill Cluster Label.
 - **Kubernetes Cluster Version:** Choose the Kubernetes version you want to use.



Kubernetes on Utho Cloud

Utho Cloud interface showing the configuration for a Pool Worker (pool-KINYAX0L). The configuration includes fields for Pool Name, Node Size, Desired Count, and Total Cost. Below these, there is a VPC Network selection dropdown and an Endpoint access info section with radio button options for Public, Public and private, and Private access.

- **Manage Node Worker Pools:** Fill the Pool Name, Node Size, Desired Count & Total Cost accordingly.
- **VPC Network:** Choose VPC Network accordingly.
- **Endpoint access info:** Choose accordingly.

Utho Cloud interface showing the configuration for a Security Group and CPU Model. The Security Group section includes a search bar and a list of groups. The CPU Model section shows a selection between AMD and Intel. The Total Cost is displayed as Rs.0/mon. A Deploy Cluster button is visible at the bottom right.

- **Security Group:** Select Firewall (Optional).
- **CPU Model:** Chose CPU Model accordingly.

Once you have configured the cluster, click **Deploy Cluster**. Utho Cloud will provision your Kubernetes cluster, which may take several minutes.

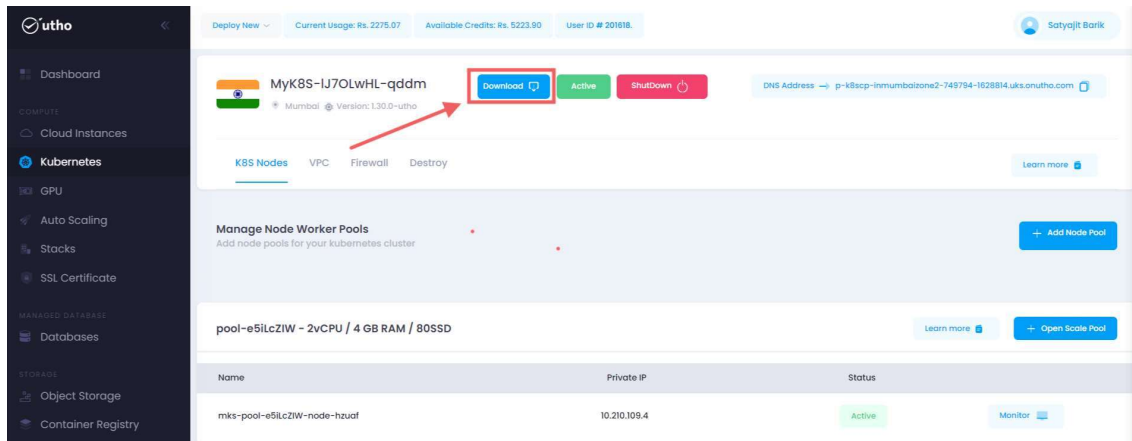
Utho Cloud interface showing the Kubernetes cluster management page. The table lists clusters with columns for Name, Control Plane IP, Workers, Status, and Created_at. A Manage button is visible for each cluster entry.

Name	Control Plane IP	Workers	Status	Created_at	Manage
MyK8S-lJ7OLwHt-qddm Mumbai	157.20.215.207	1	Running	2024-11-07 15:47:18	Manage
MyK8S-qeInnPD9-qjagh Delhi (Noida)	103.208.145.145	1	Running	2024-11-06 13:19:30	Manage
MyK8S-egWWXBZ7-si6h Delhi (Noida)	103.208.145.205	1	Running	2024-11-01 20:36:03	Manage

Congratulations your Kubernetes Cluster has been created Successfully. Click on Manage to View Details of Kubernetes Clusters

Kubernetes on Utho Cloud

- After the cluster is ready, download the kubeconfig file from Utho Cloud to connect your local machine to the cluster



- After the cluster is ready, download the kubeconfig file from Utho Cloud to connect your local machine to the cluster.

2.2 Accessing the Kubernetes Cluster

1. Install kubectl:

- If you haven't installed kubectl, install it by following the instructions from the [Official Kubernetes documentation](#).

2. Set up kubectl Configuration:

- After downloading the kubeconfig file, set up your kubectl configuration:

```
export KUBECONFIG=/path/to/kubeconfig
```

i.e

```
root@cloudserver-gNihdm4V:~# ls
helm-v3.12.0-linux-amd64.tar.gz  kubeconfig_mks.yaml  main.tf  satya.yaml  uthoctl  utho.txt
LICENSE  README.md  snap  uthoctl_0.1.5_linux_amd64.tar.gz
root@cloudserver-gNihdm4V:~# export KUBECONFIG=/root/kubeconfig_mks.yaml
```

3. Verify Access:

- Run the following command to verify that you can access the cluster:

```
kubectl get nodes
```

4. List all the pod:

- TO list all the pods in all namespaces within your Kubernetes cluster.

```
kubectl get pods --all-namespaces
```

```
root@cloudserver-gNihdm4V:~# kubectl get pods --all-namespaces
NAMESPACE   NAME                                     READY   STATUS    RESTARTS   AGE
default     nginx-pod                              1/1     Running   0           41h
default     utho-app-operator-fd64fc8d5-qf5jj      2/2     Running   0           42h
ingress-nginx  my-nginx-ingress-controller-665d858599-v8jqh  1/1     Running   0           41h
kube-system   coredns-85c69f454c-4c2sz               1/1     Running   0           2d14h
kube-system   connectivity-agent-8rmz2               1/1     Running   0           2d14h
kube-system   kube-proxy-zf9r2                      1/1     Running   0           2d14h
kube-system   kube-router-2nk5q                     1/1     Running   0           2d14h
kube-system   metrics-server-5cd4986bbc-8d5sh        1/1     Running   0           2d14h
openebs      openebs-localpv-provisioner-bb8f85f5f-7h6b9  1/1     Running   0           2d14h
openebs      openebs-ndm-operator-5b8bd59566-1581v    1/1     Running   0           2d14h
openebs      openebs-ndm-qtqfv                     1/1     Running   0           2d14h
```

~Thank You~