# Instructions to setup Sandbox

## High level Workflow

[Register to get KubeSlice Enterprise repository tokens](#_9eyl765popr8)

[Install Kube Slice Controller and Manager UI](#_3njxp3471t7l)

[Register Worker Clusters to the Controller](#_cs3qevoh84g6) [[via UI](#_q5nexpuw2fam)]

[Install KubeSlice Worker operators](#_aw1gk4p7s5u)

[Setup KubeAccess Slice on AWS cluster](#_d80516ce5ia5)

## Register for Enterprise Tokens

**Please register to receive an email with the tokens to the enterprise docker repository**

<https://avesha.io/kubeslice-registration>

## Install KubeSlice Controller and Manager UI

**Please modify the controller.yaml file to change the following fields,**

customerName: <customer-name>

# username & password & email values for imagePullSecrets has to provided to create a secret

imagePullSecrets:

repository: https://index.docker.io/v1/

username: <user-name from the registration email>

password: <password from the registration email>

email: <email-id from the registration email>

**Please modify the kubeslice-ui.yaml file to change the following fields,**

# username & password & email values for imagePullSecrets has to provided to create a secret

imagePullSecrets:

repository: https://index.docker.io/v1/

username: <user-name from the registration email>

password: <password from the registration email>

email: <email-id from the registration email>

**Add helm repo add**

helm repo add kubeslice <https://kubeslice.aveshalabs.io/repository/kubeslice-helm-ent-prod/>

helm repo update

helm search repo kubeslice

**Install KubeSlice controller and ui on the cluster**

helm install kubeslice-controller kubeslice/kubeslice-controller -f controller.yaml --namespace kubeslice-controller --create-namespace

helm install kubeslice-ui kubeslice/kubeslice-ui -f kubeslice-ui.yaml -n kubeslice-controller

**Creating a project**

Please modify the project.yaml file with a project name and the serviceaccount name

kubectl apply -f project.yaml -n kubeslice-controller

**Accessing the KubeSlice ui**

kubectl get svc -n kubeslice-controller | grep kubeslice-ui-proxy

Copy the LoadBalancer url and open it in the browser, prefix “https://” to the url

**Retrieving the token to login to the UI**

kubectl describe secret <RW secret> -n <kubeslice-XXX>

<RW secret> and <kubeslice-xxx> based on project name and sa defined in the project.yaml file

**CI/CD notes**

1. Check-in the controller.yaml, kubeslice-ui.yaml, project.yaml file into the ArgoCD git repository.
2. Modify ArgoCD to use helm to install the KubeSlice components onto the KubeSlcie controller LKE cluster.

## **Instructions to Register Worker Clusters**

Repeat this for each worker cluster that needs to be used

**Reference:** <https://docs.avesha.io/documentation/enterprise/1.11.0/install-kubeslice/yaml/yaml-controller-install>

### **Prereq**

Label the nodes in the worker cluster and install metrics server and install prometheus

for node in $(kubectl get nodes -o name); do kubectl label $node [kubeslice.io/node-type=gateway](http://kubeslice.io/node-type=gateway) --overwrite done

kubectl create ns monitoring

helm install prometheus kubeslice/prometheus -n monitoring

### **Register Worker Clusters using yaml** [alternate easy path: [using UI](https://docs.google.com/document/d/1R9sWWrMFhQi6MDTtN3Q21KMnAY1cwl1nhj_RXib_FR0/edit#heading=h.pvcyihpljf7t)]

This step involves registering the worker clusters to the KubeSlice Controller and using the secrets from the controller cluster to install the slice operator on the worker clusters

### **Apply the Cluster Registration YAML File -** apply the yaml to the **controller** cluster

kubectl apply -f cluster-registration.yaml -n kubeslice-sandbox1

## Install the Kubeslice Worker Operator on Worker Cluster

## **[Instructions below for miami worker cluster, repeat for each worker cluster]**

Please use attached worker-operator-xxx.yaml template file to create the worker-operator-mia.yaml and edit the values for ControllerSecret and Cluster and imagePullSecrets

**ControllerSecret** information from **controller** cluster:

kubectl get secrets kubeslice-rbac-worker-miam-lkei -o yaml -n kubeslice-sandbox1

**Cluster** information from **worker miami** cluster using

Kubectl cluster-info

**imagePullSecrets** from registration email

On the **miami-lke worker** cluster, apply the following command:

helm install kubeslice-worker kubeslice/kubeslice-worker -f worker-operator-mia-lke.yaml -n kubeslice-system --create-namespace

## Verify in the Smart Scaler UI that the worker cluster is successfully registered. This can take a few minutes to update.

## **Install the Slice Operator:**

## **Retrieve Registered Cluster Secrets:**

Create your secrets YAML file using the above output to install the Slice Operator on the worker cluster

kubectl get secrets -n kubeslice-<projectname>

kubectl get secrets <worker-cluster-secret-name> -o yaml -n kubeslice-<projectname>

### **Slice Worker Operator YAML Template**

# Base64 encoded secret values for the namespace, endpoint, ca.crt and token from the controller cluster.

controllerSecret:

namespace: <encoded\_namespace>

endpoint: <encoded\_endpoint>

ca.crt: <encoded\_ca.crt>

token: <encoded\_token>

# Provide the name and endpoint of the worker cluster.

cluster:

name: <worker cluster>

endpoint: <endpoint of control plane of the worker cluster>

# This parameter is optional. The default value is true.

kubesliceNetworking:

enabled: <boolean-value>

# Provide your username, password & email values from the KubeSlice Registration email under imagePullSecrets to create a secret.

imagePullSecrets:

repository: https://index.docker.io/v1/

username: <username from KubeSlice Registration email>

password: <accesstoken from KubeSlice Registration email>

email: <KubeSlice Registration email ID>

Apply the sliceoperator.yaml file on the kubeslice-system namespace using the following command:

helm install kubeslice-worker kubeslice/kubeslice-worker -f <full path of sliceoperator>.yaml -n kubeslice-system --create-namespace

**Reference:**

<https://docs.avesha.io/documentation/enterprise/1.11.0/install-kubeslice/yaml/yaml-register-worker-clusters>

## Setting up the KubeAccess Slice for managed redis

1. Register aws cluster
2. On the **controller** cluster apply awsslice yaml file to the project namespace

kubectl apply -f aws-vpc-slice.yaml -f kubeslice-sandbox1

1. On the aws cluster, service export managed redis on the “awsslice-vpc-access-gw-system

“ namespace

kubectl apply -f aws-vpc-serviceexport.yaml -n awsslice-vpc-access-gw-system

Now boutique application running in LKE cluster can operate with managed redis on aws vpc.

## Register Worker Clusters by using kubeslice-manager UI:

This topic describes how to use the KubeSlice Manager to register clusters with the KubeSlice Controller

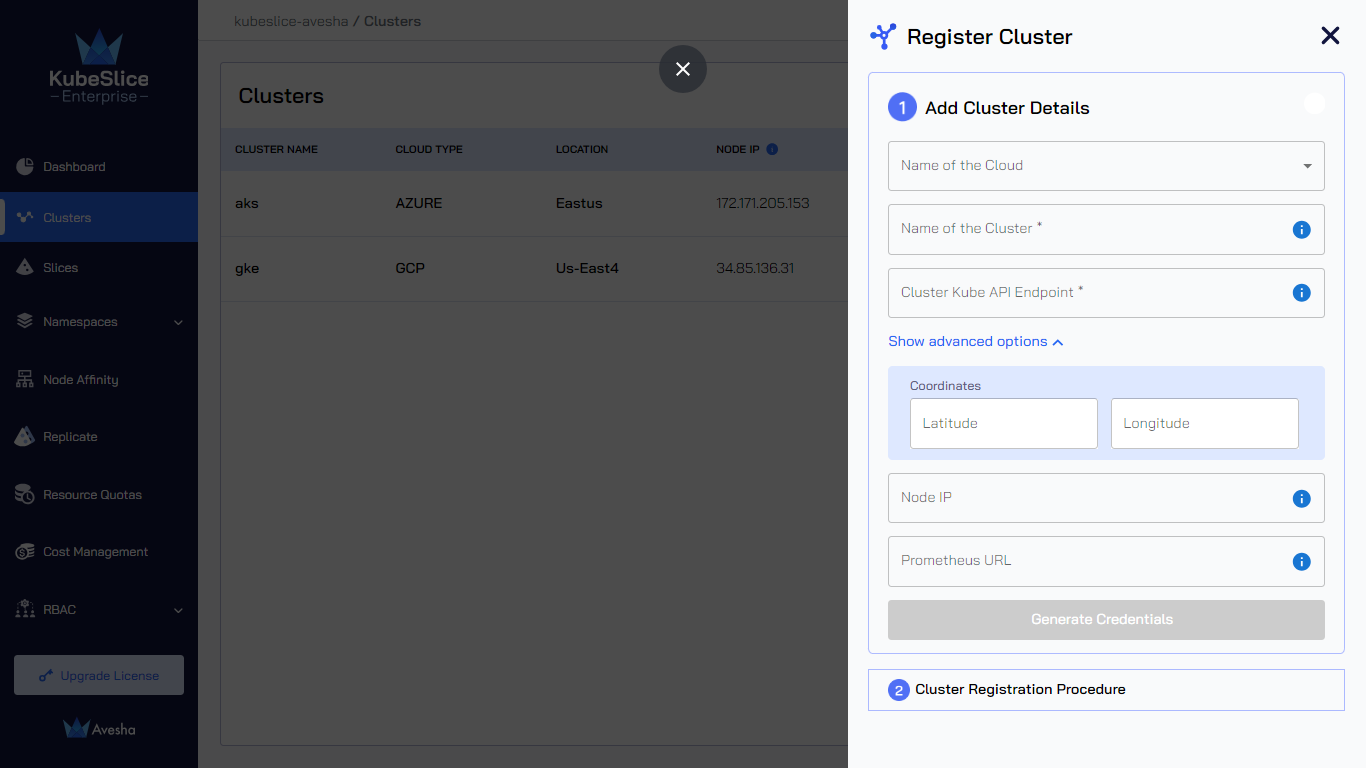
## **Manual Method -**

Register a new worker cluster with the KubeSlice Controller by entering the cluster name, the cloud name, and the cluster Kube API endpoint parameters during cluster registration. The Node IP and the Prometheus URL are optional parameters.



**To register a cluster:**

1. Under Register Clusters, select the Manual mode.
2. click Next to add the cluster details.
3. In the Add Cluster Details section, enter the following information:
   * Select the cloud from the Name of the Cloud drop-down list. The saved value is immutable.
   * Enter a name for a worker cluster in the Name of the cluster text box. The saved value is immutable.
   * Enter the control plane's kube-apiserver endpoint of the cluster in the Cluster Kube API Endpoint text box. Run this command on the cluster to get the endpoint: kubectl cluster-info.

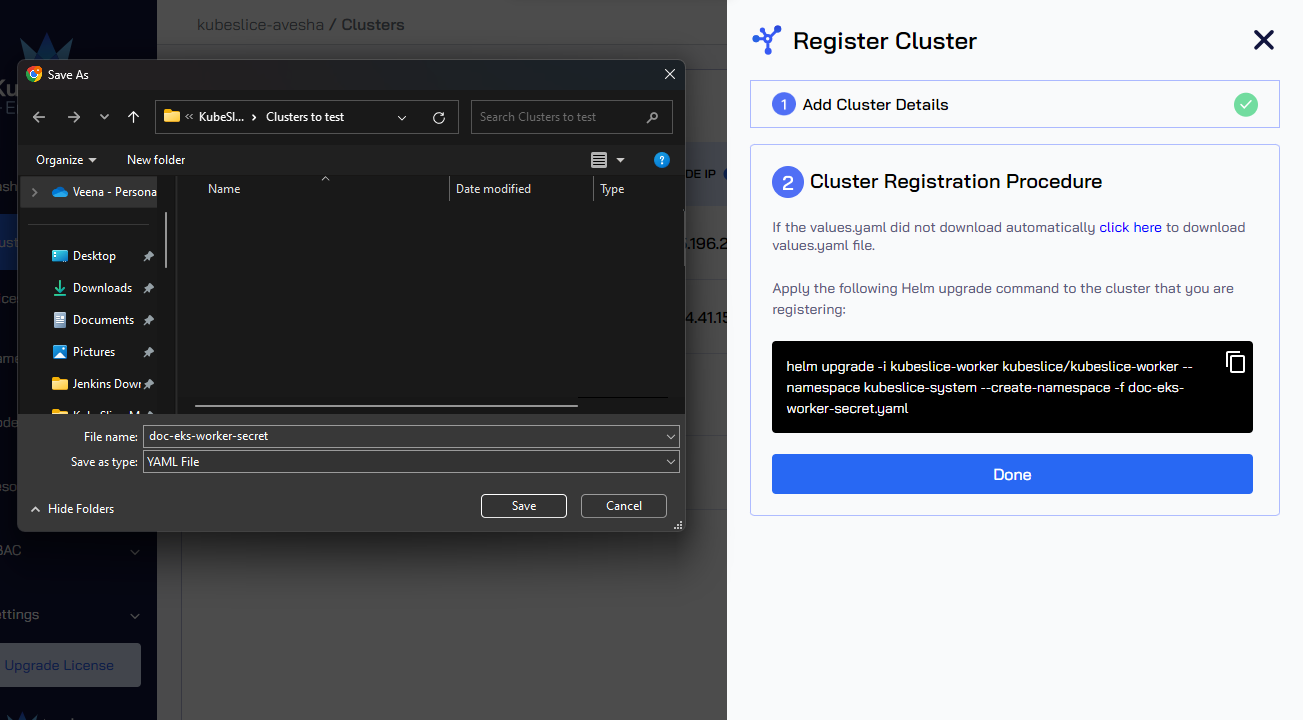


4. (*Optional*) Click Show advanced options and enter the following information:

* + KubeSlice detects the latitude and longitude coordinates of the cluster's geographical location.
  + KubeSlice detects the Node IP for nodes labeled kubeslice.io/node-type=gateway. You can leave it blank. You can enter the specific Node IP during cluster registration, but make sure KubeSlice has access to it.
  + Enter the URL of Prometheus that is installed on your cluster in the Prometheus URL text box.

### 5. Download the Slice Operator Values File

Click Generate Credentials to generate the values file. The values file is downloaded automatically. Save the file for later use



6. Edit the values file you downloaded file of register a worker cluster . Fill in the image pull secrets

7. Install the Slice Operator

Switch to the the worker cluster

helm upgrade -i kubeslice-worker kubeslice/kubeslice-worker --namespace kubeslice-system --create-namespace -f /<path-to-file>-secret.yaml

**Reference:**

[**https://docs.avesha.io/documentation/enterprise/1.11.0/kubeslice-manager/kubeslice-manager-cluster-operations#manual-method-1**](https://docs.avesha.io/documentation/enterprise/1.11.0/kubeslice-manager/kubeslice-manager-cluster-operations#manual-method-1)

**MISC**

**spec:**

**clusterProperty:**

**telemetry:**

**enabled: true**

**telemetryProvider: "prometheus"**

**endpoint: "**[**http://35.198.88.149:32700**](http://35.198.88.149:32700/)**"**