KubeTally Quick Start Guide

Table of Contents

Install KubeTally

This topic serves as a quick-start guide that assists you in installing KubeTally with the bare minimum configuration required to start using KubeTally.

KubeTally Registration

Go to https://avesha.io/kubeslice-registration/. You receive an email with login credentials to be used in the `topology.yaml` file for installation.

Cluster Authentication

To register your worker clusters with the KubeSlice Controller, it is necessary to authenticate with each cloud provider used in the installation.

Amazon EKS

```
Unset

aws eks update-kubeconfig --name <cluster-name> --region
<cluster-region>
```

Configure the Helm Repository

Add the helm repo using the following command:

```
Unset

helm repo add kubetally
https://kubeslice.aveshalabs.io/repository/kubetally-helm-e
nt-prod/
```

Update the repo using the following command:

```
Unset

helm repo update
```

Verify the repo using the following command:

```
Unset

helm search repo kubetally
```

KubeTally Installation through kubeslice-cli

Create the topology configuration file using the following template to install KubeTally on clusters.

Sample Topology Configuration File

The following is a minimal configuration file to install KubeTally on cloud clusters:

Create the topology configuration file using the following template:

```
Unset

configuration:
  cluster_configuration:
    profile: #{the KubeSlice Profile for the demo. Possible
values [full-demo, minimal-demo]}
    kube_config_path: #{specify the kube config file to use
for topology setup; for topology only}
```

```
    cluster_type: #{optional: specify the type of cluster.
Valid values are kind, cloud, data-center}
    controller:
      name: #{the user defined name of the controller
cluster}
      context_name: #{the name of the context to use from
kubeconfig file; for topology only}
      kube_config_path:#{the path to kube config file to
use for controller installation; for topology only.}
        #{This takes precedence over
configuration.cluster_configuration.kube_config_path}
      control_plane_address:#{the address of the control
plane kube-apiserver. kubeslice-cli determines the address
from kubeconfig}
        #{Override this flag if the address in kubeconfig
is not reachable by other clusters in topology}
      node_ip:#{the IP address of one of the node in this
cluster. kubeslice-cli determines this address from kubectl
get nodes}
        #{Override this flag to an address which is
discoverable by other clusters in the topology}
    workers: #{specify the list of worker clusters}
      - name: #{the user defined name of the worker
cluster}
        context_name: #{the name of the context to use from
the kubeconfig file; for topology only}
        kube_config_path:#{the path to kube config file to
use for worker installation; for topology only.}
          #{This takes precedence over
configuration.cluster_configuration.kube_config_path}
```

```
        control_plane_address:#{the address of the control
plane kube-apiserver. kubeslice-cli determines the address
from kubeconfig}
          #{Override this flag if the address in kubeconfig
is not reachable by other clusters in topology}
        node_ip:#{the IP address of one of the node in this
cluster. kubeslice-cli determines this address from kubectl
get nodes}
          #{Override this flag to an address which is
discoverable by other clusters in the topology}
      - name: #{the user defined name of the worker
cluster}
        context_name: #{the name of the context to use from
the kubeconfig file; for topology only}
        kube_config_path:#{the path to kube config file to
use for worker installation; for topology only.}
          #{This takes precedence over
configuration.cluster_configuration.kube_config_path}
        control_plane_address:#{the address of the control
plane kube-apiserver. kubeslice-cli determines the address
from kubeconfig}
          #{Override this flag if the address in kubeconfig
is not reachable by other clusters in topology}
        node_ip:#{the IP address of one of the node in this
cluster. kubeslice-cli determines this address from kubectl
get nodes}
          #{Override this flag to an address which is
discoverable by other clusters in the topology}
  kubeslice_configuration:
    project_name: #{the name of the KubeSlice Project}
```

```
    project_users: #{optional: specify KubeSlice Project
users with Readw-Write access. Default is admin}
  helm_chart_configuration:
    repo_alias: <file_path_to_helm_charts>
 #{The file path location of KubeSlice Charts}
use_local: true
    cert_manager_chart:
      chart_name: #{The name of the Cert Manager Chart}
      version: #{The version of the chart to use. Leave
blank for latest version}
    controller_chart:
      chart_name: #{The name of the Controller Chart}
      version: #{The version of the chart to use. Leave
blank for latest version}
      values: #(Values to be passed as --set arguments to
helm install)
    worker_chart:
      chart_name: #{The name of the Worker Chart}
      version: #{The version of the chart to use. Leave
blank for latest version}
      values: #(Values to be passed as --set arguments to
helm install)
    ui_chart:
      chart_name: #{The name of the UI/Enterprise Chart}
      version: #{The version of the chart to use. Leave
blank for latest version}
      values: #(Values to be passed as --set arguments to
helm install)
    prometheus_chart:
      chart_name: #{The name of the Prometheus Chart}
```

```
      version: #{The version of the chart to use. Leave
blank for latest version}
      values: #(Values to be passed as --set arguments to
helm install)
    helm_username: #{Helm Username if the repo is private}
    helm_password: #{Helm Password if the repo is private}
    image_pull_secret: #{The image pull secrets. Optional
for OpenSource, required for enterprise}
      registry: #{The endpoint of the OCI registry to use.
Default is `https://index.docker.io/v1/`}
      username: #{The username to authenticate against the
OCI registry}
      password: #{The password to authenticate against the
OCI registry}
      email: #{The email to authenticate against the OCI
registry}
```

Apply the Topology Configuration YAML

To install KubeSlice using the topology YAML file, use the following command:

```
Unset

kubeslice-cli --config <path-to-the-topology.yaml> install
```

Retrieve the KubeTally Endpoint

Use the following command to retrieve the KubeTally endpoint:

```
Unset

kubeslice-cli get ui-endpoint -c
<path-to-custom-topology-file>
```

Output format:

```
Unset
https://<Node-IP>:<Node-Port>
```

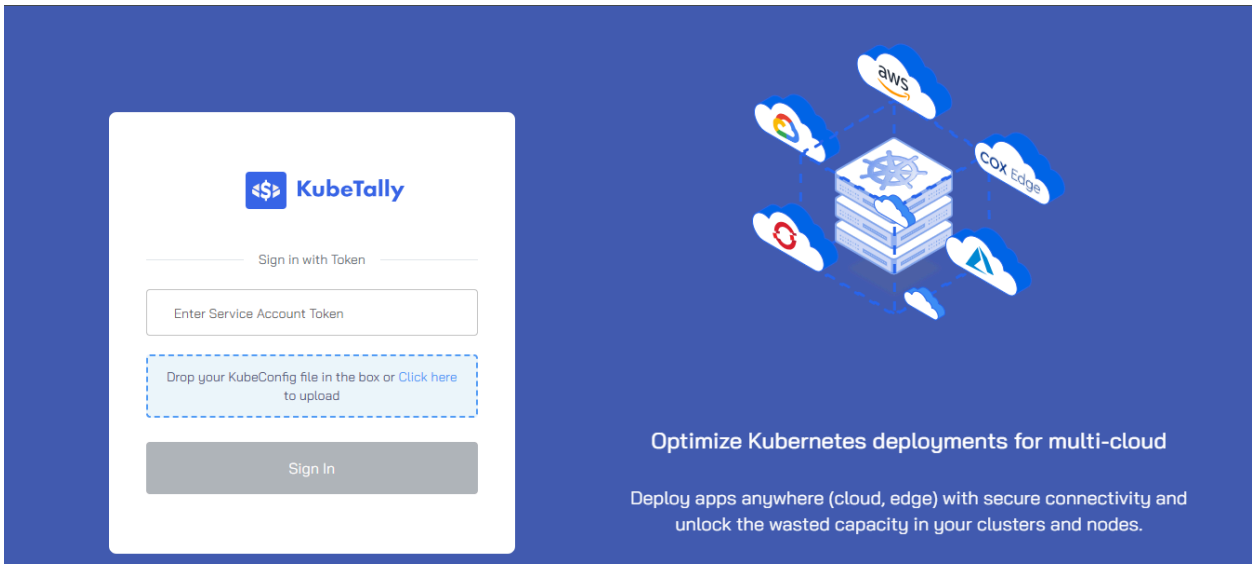
When LoadBalancer is used:

```
https://<LoadBalancer-IP>
```

Copy the endpoint/URL from the output to use it to log in to KubeTally.

KubeTally UI Login

Use the endpoint/URL that you copied from the command output earlier into your browser window to access the KubeTally.



You must create a service-account token to log in to KubeTally. Create a service-account token using the following command:

```
Unset
kubectl get secret kubeslice-rbac-rw-admin -o
jsonpath="{.data.token}" -n kubeslice-<project> | base64
--decode
```

Access KubeTally using the Token

📜**Note**: *We recommend using a managed Postgres instance for better management of data.*

Go to the URL that you have retrieved using the command in the KubeTally UI Login.

On the login page, for **Enter Service Account Token**, copy the token from the command output above and paste it in the text box.

Click **Sign in**. After a successful authentication, you see the dashboard of the KubeTally as the landing page.