

Helm Setup for Local Minikube

Prerequisite:

- Minikube
- Kubectl
- Helm

Step 1: Install Minikube.

Link: <https://kubernetes.io/docs/tasks/tools/install-minikube/>

Step 2: Check Minikube is properly installed or not by using:

```
$ minikube status
```

Output:

```
host: Running
kubelet: Running
api server: Running
kubectl: correctly configured: Pointing to minikube VM at 192.168.10.100
```

Step 3: Now Install and Set Up Kubectl.

link: <https://kubernetes.io/docs/tasks/tools/install-kubectl/#install-kubectl-on-linux/>

Step 4: Install helm in your system and config it's tiller.

link: https://helm.sh/docs/using_helm/#installing-helm

For configuring your tiller:

```
$ kubectl create serviceaccount --namespace kube-system tiller
```

```
$ kubectl create clusterrolebinding tiller-cluster-rule
--clusterrole=cluster-admin --
```

```
serviceaccount=kube-system:tiller
```

```
$ helm init --service-account tiller --upgrade
```

Step 4.1: You can download helm version according to your need for this click on below links

link: <https://github.com/helm/helm/releases>

Step 5: Now follow the steps README file for the deployment of helm chart.

Step 6: You can check the status using:

\$ helm status name_of_the_release This command show status of Helm chart installed.

Output:

```
LAST DEPLOYED: Tue Jun 11 17:06:58 2019
NAMESPACE: nexaops
STATUS: DEPLOYED

RESOURCES:
==> v1/ConfigMap
NAME      DATA  AGE
prometheus 1     3h31m

==> v1/Pod(related)
NAME                                             READY  STATUS   RESTARTS  AGE
testing-stuart-microservice-stack-database-cd9d46947-w4t4n  1/1    Running   0         3h31m
testing-stuart-microservice-stack-grafana-7644fd944f-7cfwf  1/1    Running   0         3h31m
testing-stuart-microservice-stack-microservice-api-689d795lphlz  1/1    Running   0         3h31m
testing-stuart-microservice-stack-prometheus-75f965cff9-slgb1  1/1    Running   0         3h31m

==> v1/Service
NAME                                             TYPE        CLUSTER-IP    EXTERNAL-IP  PORT(S)          AGE
testing-stuart-microservice-stack-database      NodePort    10.102.17.132 <none>       5432:32459/TCP  3h31m
testing-stuart-microservice-stack-grafana       NodePort    10.104.227.61 <none>       3000:31188/TCP  3h31m
testing-stuart-microservice-stack-microservice-api NodePort    10.98.20.40   <none>       5000:32583/TCP  3h31m
testing-stuart-microservice-stack-prometheus    NodePort    10.111.189.141 <none>       9090:30718/TCP  3h31m

==> v1beta2/Deployment
NAME                                             READY  UP-TO-DATE  AVAILABLE  AGE
testing-stuart-microservice-stack-database      1/1    1           1          3h31m
testing-stuart-microservice-stack-grafana       1/1    1           1          3h31m
testing-stuart-microservice-stack-microservice-api 1/1    1           1          3h31m
testing-stuart-microservice-stack-prometheus    1/1    1           1          3h31m
```

Step 7: Now check all pods are running or not by using:

\$ kubectl get pods This command will show you all the pods.

Output:

```
testing-stuart-microservice-stack-database-cd9d46947-w4t4n    1/1    Running   0    108m
testing-stuart-microservice-stack-grafana-7644fd944f-7cfwf    1/1    Running   0    108m
testing-stuart-microservice-stack-microservice-api-689d795lphlz 1/1    Running   0    108m
testing-stuart-microservice-stack-prometheus-75f965cff9-slgb1 1/1    Running   0    108m
```

Step 8: If all the pods are in the Running state then check your all services by using:

\$ kubectl get svc This command will show you all the services.

Output:

testing-stuart-microservice-stack-database	NodePort	10.102.17.132	<none>	5432:32459/TCP	137m
testing-stuart-microservice-stack-grafana	NodePort	10.104.227.61	<none>	3000:31188/TCP	137m
testing-stuart-microservice-stack-microservice-api	NodePort	10.98.20.40	<none>	5000:32583/TCP	137m
testing-stuart-microservice-stack-prometheus	NodePort	10.111.189.141	<none>	9090:30718/TCP	137m

Step 9: When all the pods are in the running state then you can check the output in your web browser.

Screenshots:

Microservices-api: URL: <NodeIP:Portno/swagger>

The screenshot displays the Swagger UI for the **MicroService.WebAPI**. The top navigation bar is green with the Swagger logo, a search bar containing `http://example.com/api`, and a dropdown menu showing **MicroService.WebAPI**. Below the header, the title **MicroService.WebAPI** is followed by the version **MicroService.WebAPI**. The main content area is titled **TestData** and includes links for [Show/Hide](#), [List Operations](#), and [Expand Operations](#). Two API endpoints are listed: **GET /api/TestData** (labeled [Get Test Data Set Dump.](#)) and **GET /api/TestData/percentile** (labeled [Get Test Data Percentile.](#)). The selected endpoint, **GET /api/TestData/percentile**, is expanded to show details. It indicates a **Response Class (Status 200)** of **double**. The **Response Content Type** is set to **application/json**. A **Try it out!** button is present, along with a [Hide Response](#) link. The **Curl** section shows the command: `curl -X GET --header 'Accept: application/json' 'http://172.16.200.10:32583/api/TestData/percentile'`. The **Request URL** is `http://172.16.200.10:32583/api/TestData/percentile`. The **Response Body** is `9949.956379714422`. The **Response Code** is `200`. The **Response Headers** are: `{ "content-type": "application/json; charset=utf-8", "date": "Tue, 11 Jun 2019 11:41:22 GMT", "server": "Kestrel", "transfer-encoding": "chunked" }`.

Prometheus: URL: <NodeIP:Portno>

Prometheus

Alerts

Graph

Status ▾

Help

☐ Enable query history

Expression (press Shift+Enter for newlines)

Execute

- insert metric at cursor ▾

Graph

Console

◀

Moment

▶

Element	Value
no data	

Remove Graph

Add Graph

Grafana: URL: <NodeIP:Portno>

Home ▾

+

Home Dashboard

Install Grafana

Add data source

New dashboard

Add Users

Explore plugin repository

Starred dashboards

Recently viewed dashboards

Installed Apps

None installed. [Browse Grafana.com](#)

Installed Panels

None installed. [Browse Grafana.com](#)

Installed Datasources

None installed. [Browse Grafana.com](#)