

MONITORING

About Monitoring

To monitor kubernetes cluster, we are going to use open-source Prometheus, with database Grafana, and Alert manager stack. To configure Prometheus, we are going to use HELM. MONITORING will pull the data from Cadvisor, where it as help to get data in cluster.

Actual Readme

A monitoring system usually consists of a time-series database that houses metric data and a visualization layer. In addition, an alerting layer creates and manages alerts, handing them off to integrations and external services as necessary. Finally, one or more components generate or expose the metric data that will be stored, visualized, and processed for alerts by this monitoring stack.

One popular monitoring solution is the open-source Prometheus, Grafana, and Alertmanager stack: Prometheus is a time series database and monitoring tool that works by polling metrics endpoints and scraping and processing the data exposed by these endpoints. It allows you to query this data using PromQL, a time series data query language.

Grafana is a data visualization and analytics tool that allows you to build dashboards and graphs for your metrics data.

Alertmanager, usually deployed alongside Prometheus, forms the alerting layer of the stack, handling alerts generated by Prometheus and deduplicating, grouping, and routing them to integrations like email or PagerDuty.

Prerequisites

To follow this tutorial, you will need:

A Kubernetes clusters.

The kubectl command-line interface installed on your local machine and configured to connect to your cluster. You can read more about installing and configuring kubectl in its official documentation.

The Helm package manager (2.10+) installed on your local machine and Tiller installed on your cluster, as detailed in How To Install Software on Kubernetes Clusters with the Helm Package Manager.

<https://grafana.com/grafana/dashboards/10000>

<https://grafana.com/grafana/dashboards/6417>

<https://grafana.com/grafana/dashboards/315>

1860

<https://www.replex.io/blog/kubernetes-in-production-the-ultimate-guide-to-monitoring-resource-metrics-with-grafana>

##Install HELM##

:- Install helm incase if not installed, else go with next steps.

Note:- Just refer the Helm document.

##Install prometheus##

helm search Prometheus

helm fetch stable/Prometheus

ls -lrt |grep prometheus

```
[root@anskube manifest]# ls -lrt |grep prometheus
-rw-r--r--. 1 root      root      24108 Nov  8 07:40 prometheus-9.3.1.tgz
[root@anskube manifest]#
```

tar -zxvf prometheus-9.3.1.tgz

@Modify below value on values.yaml.

Note:- here we are changing these values for accessing via our system browser, in case if required, can do the same with ingress also.

#vim prometheus/values.yaml

```
## List of IP addresses at which the Prometheus server service is available
## Ref: https://kubernetes.io/docs/user-guide/services/#external-ips
##
externalIPs: []

loadBalancerIP: ""
loadBalancerSourceRanges: []
servicePort: 80
nodePort: 32100
type: NodePort
```

kubectl get namespaces

```
[root@anskube manifest]# kubectl get namespaces
NAME          STATUS    AGE
default       Active   7h31m
dev           Active   6h26m
kube-public   Active   7h30m
kube-system   Active   7h31m
[root@anskube manifest]#
```

helm install prometheus --name prometheus --namespace monitoring

```
[root@anskube manifest]# helm install prometheus --name prometheus --namespace monitoring
NAME: prometheus
LAST DEPLOYED: Fri Nov  8 07:47:44 2019
NAMESPACE: monitoring
STATUS: DEPLOYED
```

kubectl get namespaces |grep monitoring

```
[root@anskube manifest]# kubectl get namespaces |grep monitoring
monitoring    Active   7m25s
[root@anskube manifest]#
```

kubectl get all -n monitoring

```
[root@anskube manifest]# kubectl get all -n monitoring
```

NAME	READY	STATUS	RESTARTS	AGE
pod/prometheus-alertmanager-74ffdf8bd6-n9vv2	2/2	Running	0	8m7s
pod/prometheus-kube-state-metrics-77757854cf-mhvnv	1/1	Running	0	8m7s
pod/prometheus-node-exporter-4bv7c	1/1	Running	0	8m7s
pod/prometheus-node-exporter-8cb47	1/1	Running	0	8m7s
pod/prometheus-node-exporter-jtmfw	1/1	Running	0	8m7s
pod/prometheus-pushgateway-57688d8875-k72r4	1/1	Running	0	8m7s
pod/prometheus-server-5c8b68f5cd-pqbgp	2/2	Running	0	8m7s

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
service/prometheus-alertmanager	ClusterIP	10.36.1.213	<none>	80/TCP	8m7s
service/prometheus-kube-state-metrics	ClusterIP	None	<none>	80/TCP	8m7s
service/prometheus-node-exporter	ClusterIP	None	<none>	9100/TCP	8m7s
service/prometheus-pushgateway	ClusterIP	10.36.10.13	<none>	9091/TCP	8m7s
service/prometheus-server	NodePort	10.36.4.125	<none>	80:32100/TCP	8m7s

NAME	DESIRED	CURRENT	READY	UP-TO-DATE	AVAILABLE	NODE SELECTOR	AGE
daemonset.apps/prometheus-node-exporter	3	3	3	3	3	<none>	8m7s

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
deployment.apps/prometheus-alertmanager	1/1	1	1	8m7s
deployment.apps/prometheus-kube-state-metrics	1/1	1	1	8m7s
deployment.apps/prometheus-pushgateway	1/1	1	1	8m7s
deployment.apps/prometheus-server	1/1	1	1	8m7s

NAME	DESIRED	CURRENT	READY	AGE
replicaset.apps/prometheus-alertmanager-74ffdf8bd6	1	1	1	8m7s
replicaset.apps/prometheus-kube-state-metrics-77757854cf	1	1	1	8m7s
replicaset.apps/prometheus-pushgateway-57688d8875	1	1	1	8m7s
replicaset.apps/prometheus-server-5c8b68f5cd	1	1	1	8m7s

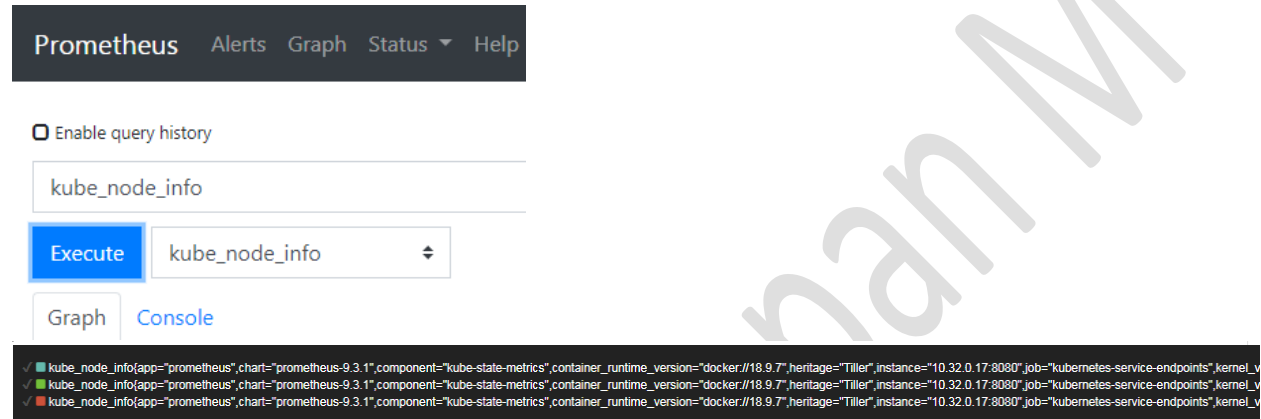
```
[root@anskube manifest]#
```

@@Now try to access from outside with node port@@

kubectl get node -o wide

```
[root@anskube manifest]# kubectl get node -o wide
NAME                                STATUS    ROLES    AGE   VERSION   INTERNAL-IP   EXTERNAL-IP   OS-IMAGE
KERNEL-VERSION   CONTAINER-RUNTIME
gke-robo-default-pool-dfc31c9c-hz0j Ready    <none>    7h40m v1.13.11-gke.9  10.128.0.50   104.197.141.14 Container-Optimized OS from Google
4.14.145+        docker://18.9.7
gke-robo-default-pool-dfc31c9c-mtjv Ready    <none>    7h40m v1.13.11-gke.9  10.128.0.52   34.66.126.120 Container-Optimized OS from Google
4.14.145+        docker://18.9.7
gke-robo-default-pool-dfc31c9c-tw87 Ready    <none>    7h40m v1.13.11-gke.9  10.128.0.51   35.232.168.36 Container-Optimized OS from Google
4.14.145+        docker://18.9.7
[root@anskube manifest]#
```

<http://104.197.141.14:32100/graph>



The screenshot shows the Prometheus web interface. At the top, there are tabs for 'Prometheus', 'Alerts', 'Graph', 'Status', and 'Help'. Below these, there is a search bar with 'kube_node_info' entered. To the left of the search bar is a checkbox labeled 'Enable query history'. Below the search bar is a blue 'Execute' button and a dropdown menu showing 'kube_node_info'. Below the search bar, there are two tabs: 'Graph' (selected) and 'Console'. The 'Graph' tab displays three query results, each with a green checkmark icon and the following text: 'kube_node_info[app="prometheus",chart="prometheus-9.3.1",component="kube-state-metrics",container_runtime_version="docker://18.9.7",heritage="Tiller",instance="10.32.0.17:8080",job="kubernetes-service-endpoints",kernel_v...'

@Now try to configure Alertmanager.

:-Delete the existing setup

helm del --purge Prometheus

vim prometheus/values.yaml

```
## List of IP addresses at which the alertmanager service is available
## Ref: https://kubernetes.io/docs/user-guide/services/#external-ips
##
externalIPs: []

loadBalancerIP: ""
loadBalancerSourceRanges: []
servicePort: 80
nodePort: 32101
type: NodePort
```

@@On top of above, we are going to configure Grafana@@

helm search Grafana

ls -lrt | grep Grafana

tar -zxvf grafana-4.0.1.tgz

@Add credentials and node ports details on below file

cat grafana/values.yaml

```
# Administrator credentials when not using an existing secret (see below)
adminUser: admin
adminPassword: admin123
# adminPassword: strongpassword
```

```
## Expose the grafana service to be accessed from outside the cluster (LoadBalancer service).
## or access it from within the cluster (ClusterIP service). Set the service type and the port to serve it.
## ref: http://kubernetes.io/docs/user-guide/services/
##
service:
  type: NodePort
  nodePort: 32102
  port: 80
  targetPort: 3000
```

helm install grafana --name grafana --namespace monitoring

```
[root@anskube manifest]# kubectl get all -n monitoring
```

NAME	READY	STATUS	RESTARTS	AGE
pod/grafana-7cd8c96c74-4l6wn	1/1	Running	0	28s
pod/prometheus-alertmanager-74ffdf8bd6-74lkm	2/2	Running	0	10m
pod/prometheus-kube-state-metrics-77757854cf-6648m	1/1	Running	0	10m
pod/prometheus-node-exporter-6t8xk	1/1	Running	0	10m
pod/prometheus-node-exporter-hnx2t	1/1	Running	0	10m
pod/prometheus-node-exporter-mgjsp	1/1	Running	0	10m
pod/prometheus-pushgateway-57688d8875-ldtlm	1/1	Running	0	10m
pod/prometheus-server-5c8b68f5cd-krmvg	2/2	Running	0	10m

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
service/grafana	NodePort	10.36.7.80	<none>	80:32102/TCP	28s
service/prometheus-alertmanager	NodePort	10.36.10.71	<none>	80:32101/TCP	10m
service/prometheus-kube-state-metrics	ClusterIP	None	<none>	80/TCP	10m
service/prometheus-node-exporter	ClusterIP	None	<none>	9100/TCP	10m
service/prometheus-pushgateway	ClusterIP	10.36.0.48	<none>	9091/TCP	10m
service/prometheus-server	NodePort	10.36.1.23	<none>	80:32100/TCP	10m

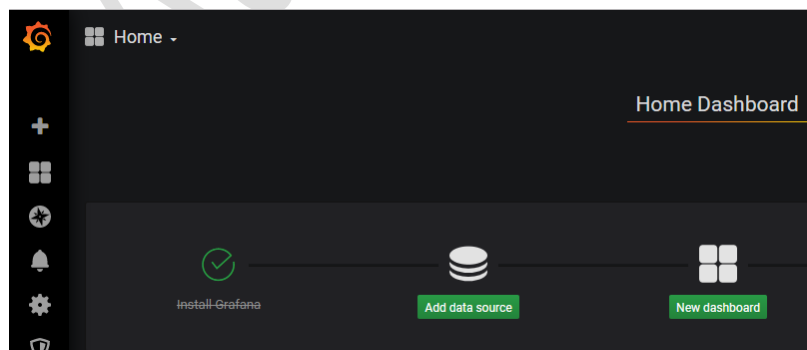
NAME	DESIRED	CURRENT	READY	UP-TO-DATE	AVAILABLE	NODE SELECTOR
daemonset.apps/prometheus-node-exporter	3	3	3	3	3	<none>

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
deployment.apps/grafana	1/1	1	1	28s
deployment.apps/prometheus-alertmanager	1/1	1	1	10m
deployment.apps/prometheus-kube-state-metrics	1/1	1	1	10m
deployment.apps/prometheus-pushgateway	1/1	1	1	10m
deployment.apps/prometheus-server	1/1	1	1	10m

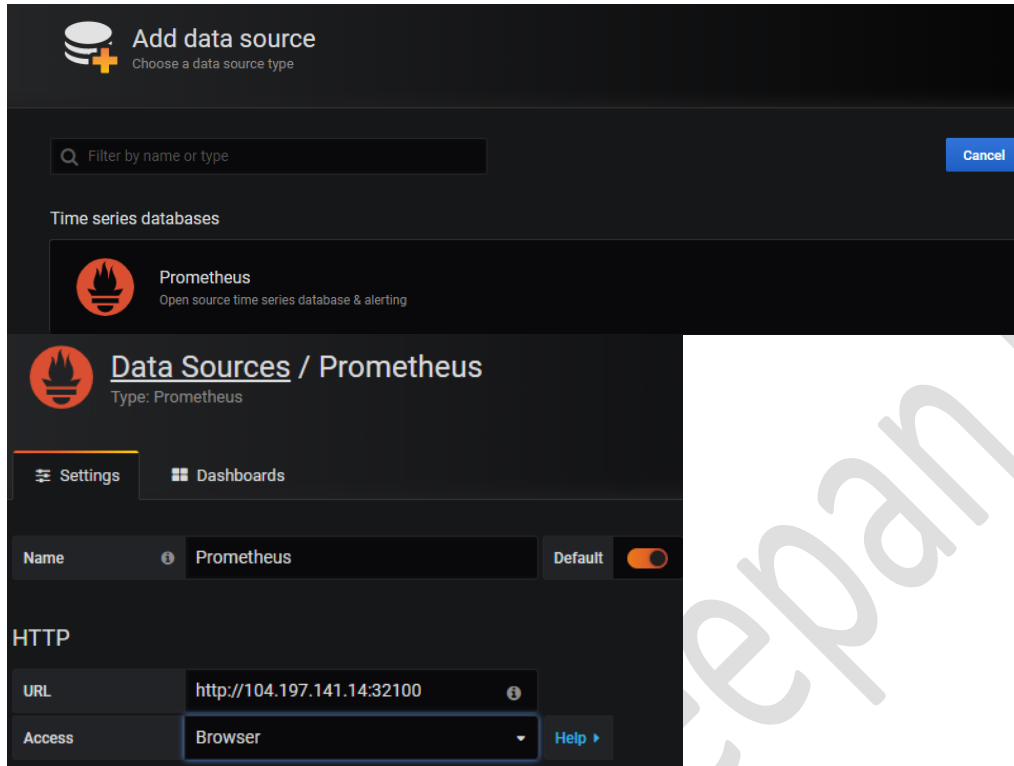
NAME	DESIRED	CURRENT	READY	AGE
replicaset.apps/grafana-7cd8c96c74	1	1	1	28s
replicaset.apps/prometheus-alertmanager-74ffdf8bd6	1	1	1	10m
replicaset.apps/prometheus-kube-state-metrics-77757854cf	1	1	1	10m
replicaset.apps/prometheus-pushgateway-57688d8875	1	1	1	10m
replicaset.apps/prometheus-server-5c8b68f5cd	1	1	1	10m

```
[root@anskube manifest]#
```

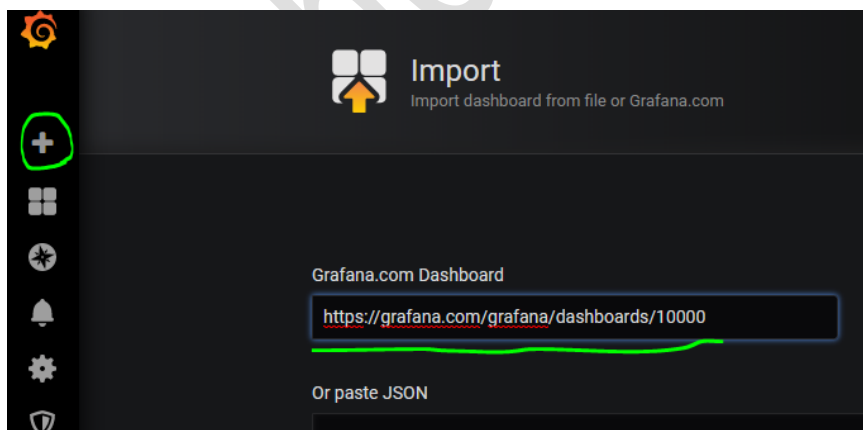
@Now open your node external ip with Grafana port via browser --> <http://104.197.141.14:32102/login>




:- Click Add datastore --> select prometheus --> type URL `http://35.223.44.107:32100` | Access -- Browser --> click savetest



:- Import dashboard Click + --> select import --> copy the dashboard id from <https://grafana.com/grafana/dashboards> --> paste it on your dashboard import option --> select load→
<https://grafana.com/grafana/dashboards/10000>
<https://grafana.com/grafana/dashboards/6417>
<https://grafana.com/grafana/dashboards/315>





Import

Import dashboard from file or Grafana.com

Importing Dashboard from [Grafana.com](#)

Published by	Pivotal Observability
Updated on	2019-08-09 00:52:07

Options

Name	Cluster Monitoring for Kubernetes	✓
Folder	General	▼
Unique identifier (uid)	value set	change
Prometheus	Prometheus	▼ ✓

[Import](#)
[Cancel](#)

