Kubernetes with Containers and DevOps Workshop

Hands-on lab step-by-step

Aralık 2018

7.Operational Monitoring and Log Management

There are a number of monitoring solutions available today. Here is a quick, but not exhaustive list for reference purposes:

* Datadog
* Sysdig
* Elastic Stack
* Splunk
* Operations Management Suite
* Prometheus

For the purposes of this lab we will be focusing in on Prometheus and using Grafana to provide a visual Dashboard of our Azure Kubernetes Service Cluster.

We are going to be installing Prometheus and Grafana into our K8s cluster using Helm and Tiller. You can think of Helm as a package manager for Kubernetes with Tiller being the server-side component.

**Install Prometheus using Helm**

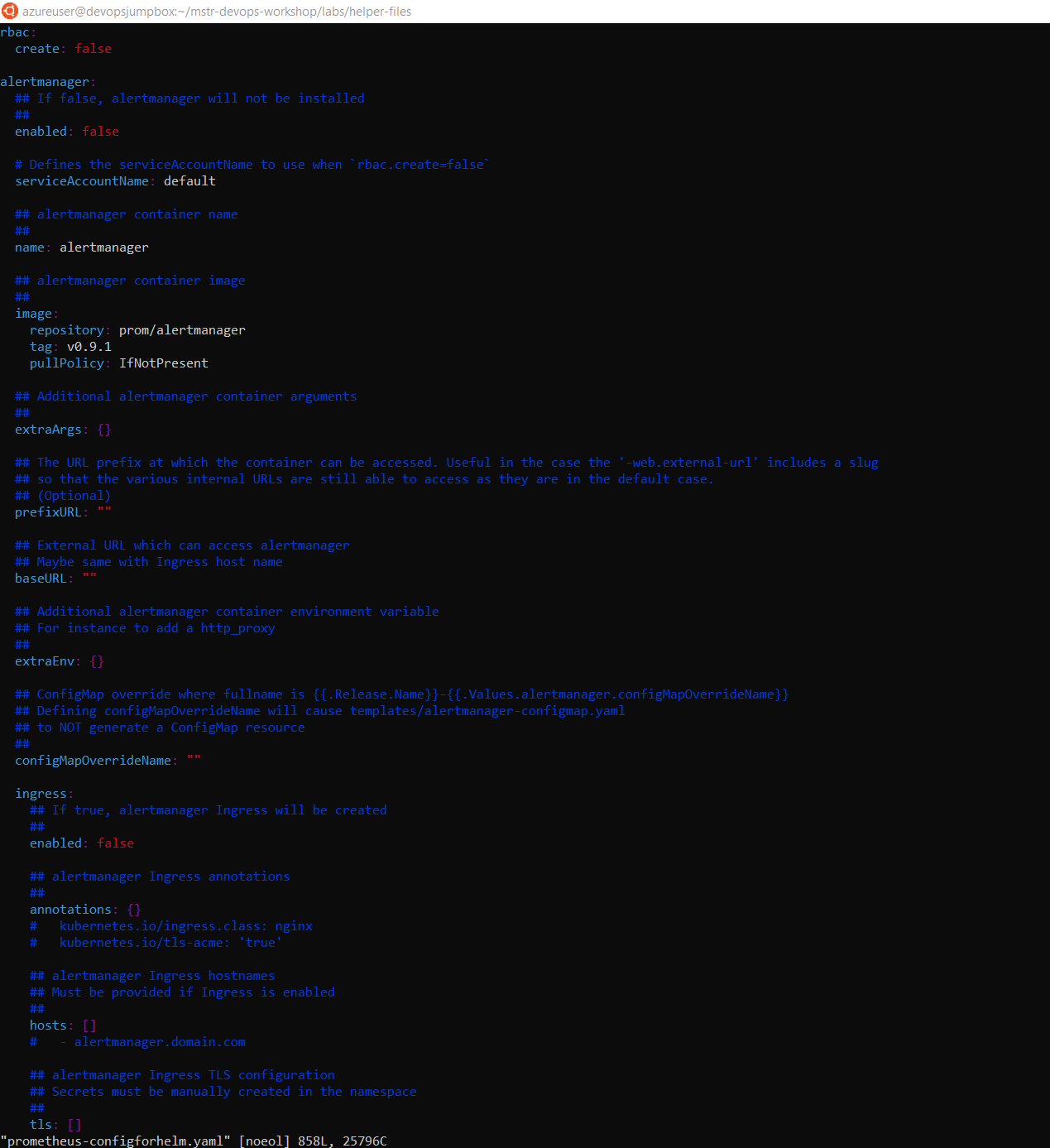
Prometheus is a Cloud Native Computing Foundation (CNCF) project used to collect and process metrics. It collects metrics from configured targets, in our case it is a Kubernetes Cluster.

1. Install Prometheus using Helm CLI

* Switch to the helper-files directory and view the prometheus-configforhelm.yaml file. This configures Helm to install Prometheus with our desired settings.

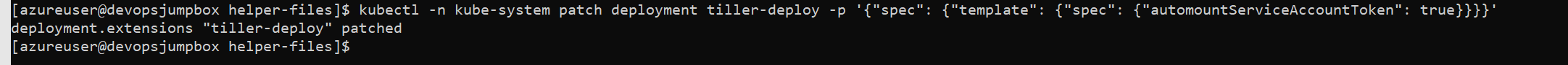
cd ~/cpx-oss-workshop/labs/helper-files/

cat prometheus-configforhelm.yaml



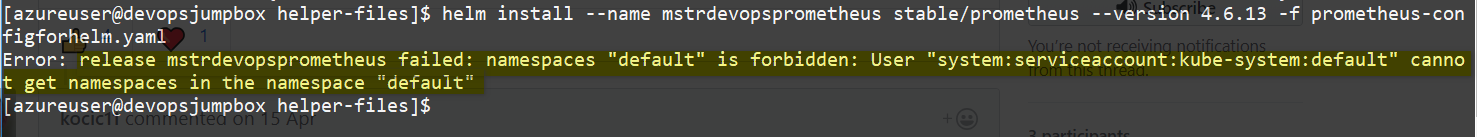
# The following command will install Prometheus into the K8s cluster using custom settings.

kubectl -n kube-system patch deployment tiller-deploy -p '{"spec": {"template": {"spec": {"automountServiceAccountToken": true}}}}'

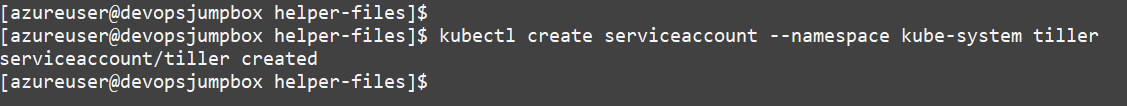


helm install --name mstrdevopsprometheus stable/prometheus --version 4.6.13 -f prometheus-configforhelm.yaml

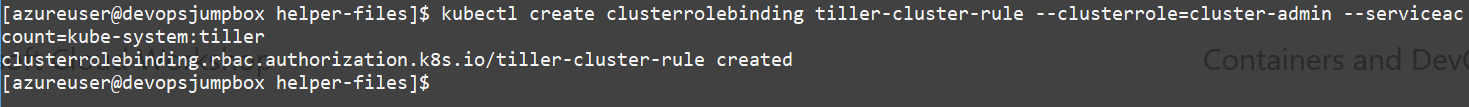
If you get an error, try the steps below.



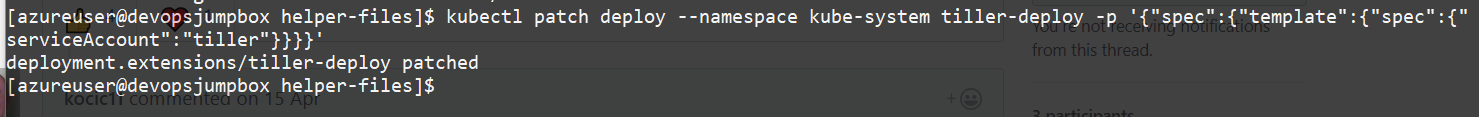
kubectl create serviceaccount --namespace kube-system tiller



kubectl create clusterrolebinding tiller-cluster-rule --clusterrole=cluster-admin --serviceaccount=kube-system:tiller

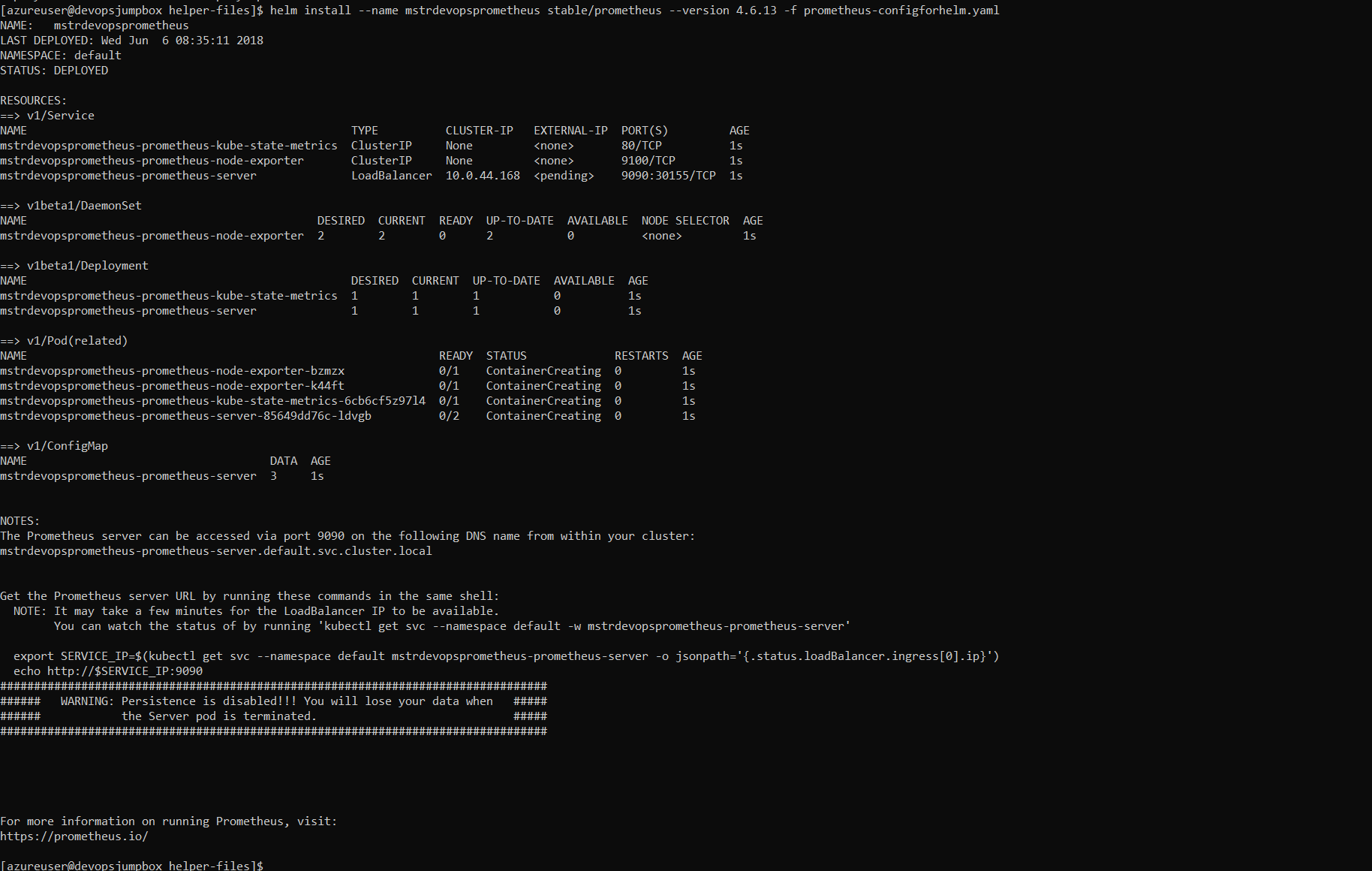


kubectl patch deploy --namespace kube-system tiller-deploy -p '{"spec":{"template":{"spec":{"serviceAccount":"tiller"}}}}'



try it now again!

helm install --name mstrdevopsprometheus2 stable/prometheus --version 4.6.13 -f prometheus-configforhelm.yaml

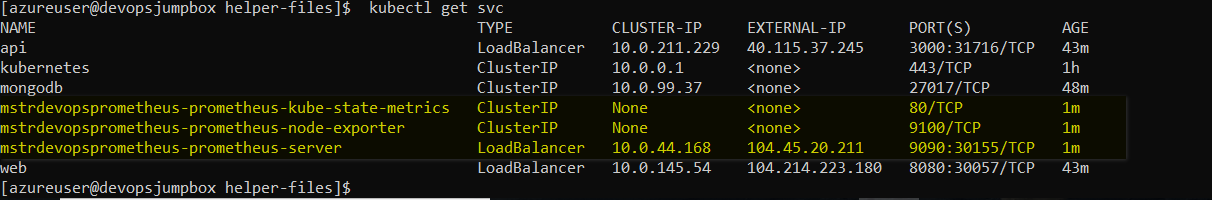


1. Validate that Prometheus was Installed

kubectl get pods



kubectl get svc



**Install Grafana**

Grafana is a dashboard visualization tool that can use all kinds of data sources. In our case, Prometheus will be used as the data source.

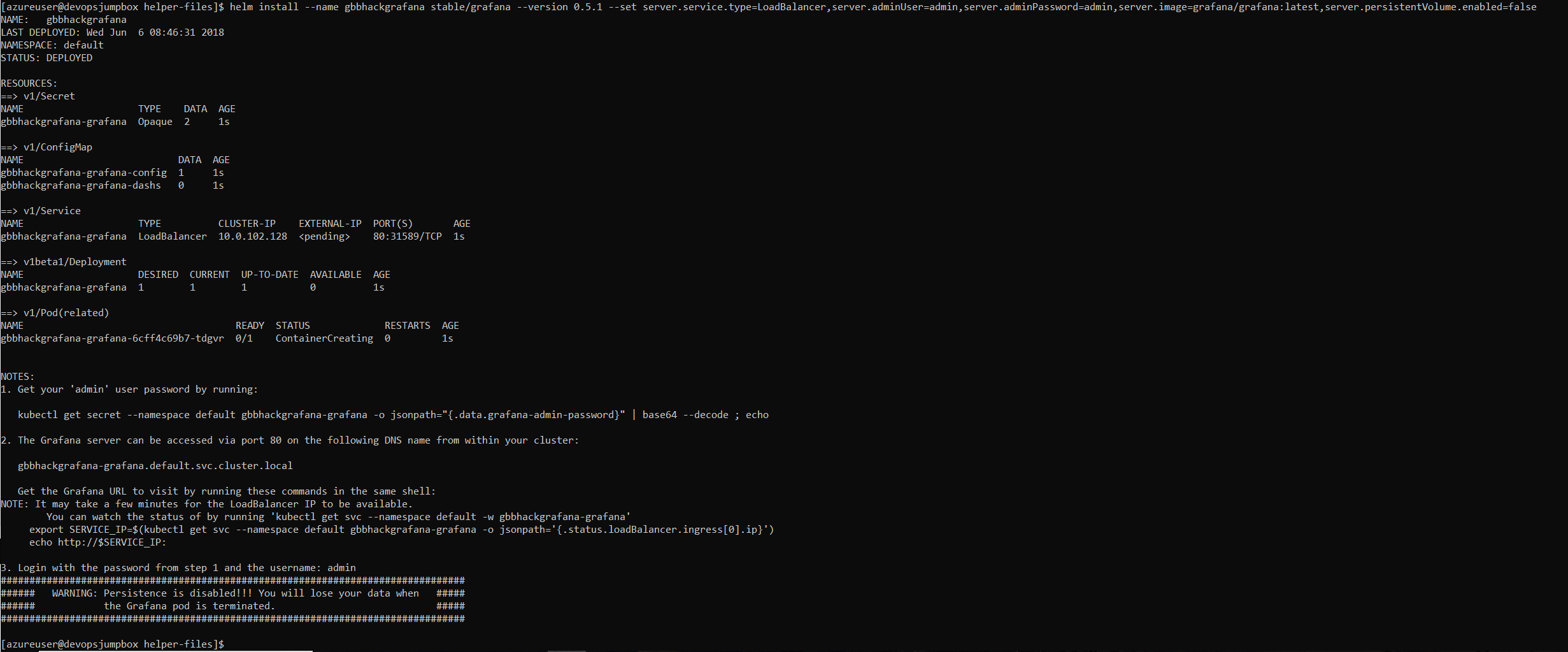
1. Install Grafana using Helm CLI

* The following command will install Grafana into the K8s cluster with a few custom settings to make it easier to access.

# We are setting the default username and password to \*\*admin\*\* to make it easier to remember.

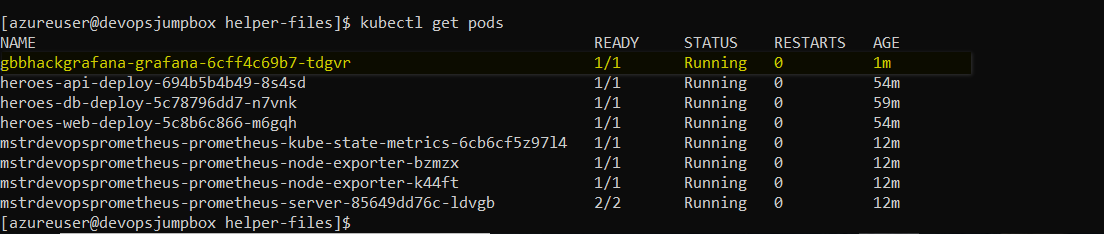
# We are also setting the service type to \*\*LoadBalancer\*\* to expose the service outside of the cluster and make it accessible via the Internet.

helm install --name mstrdevopsgrafana stable/grafana --version 0.5.1 --set server.service.type=LoadBalancer,server.adminUser=admin,server.adminPassword=admin,server.image=grafana/grafana:latest,server.persistentVolume.enabled=false

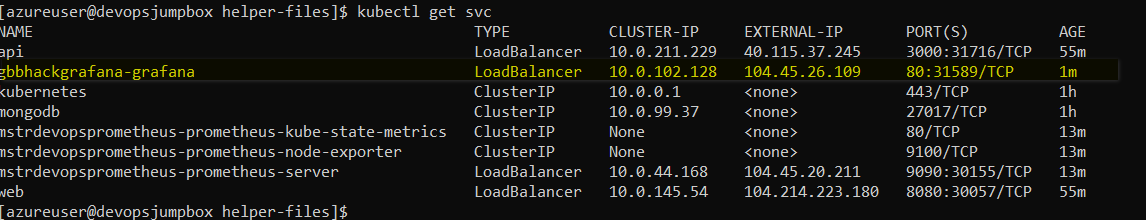


1. Validate that Grafana was Installed

kubectl get pods

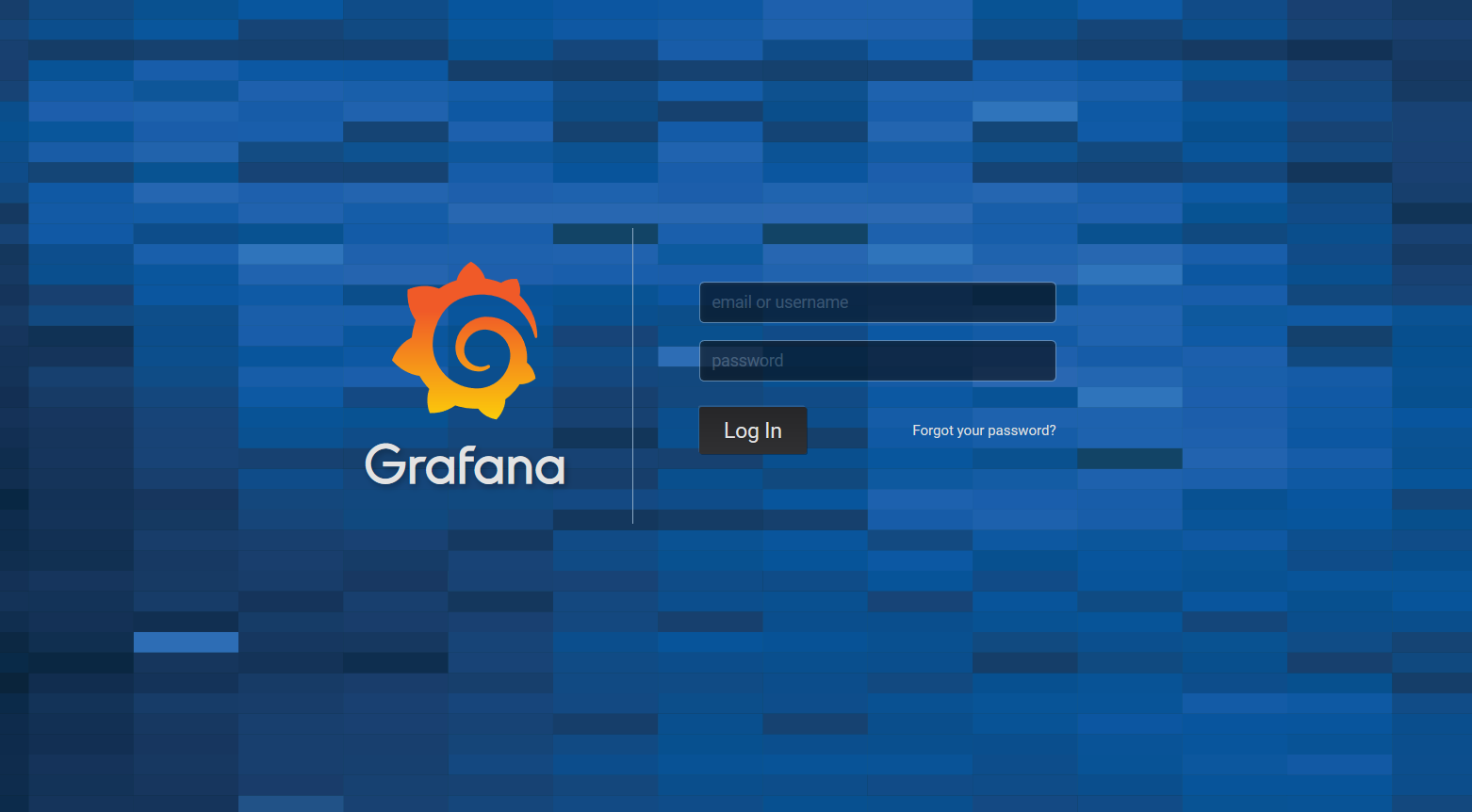


kubectl get svc



Test Grafana UI Comes Up Use the EXTERNAL-IP value from the previous step and put that into your browser:

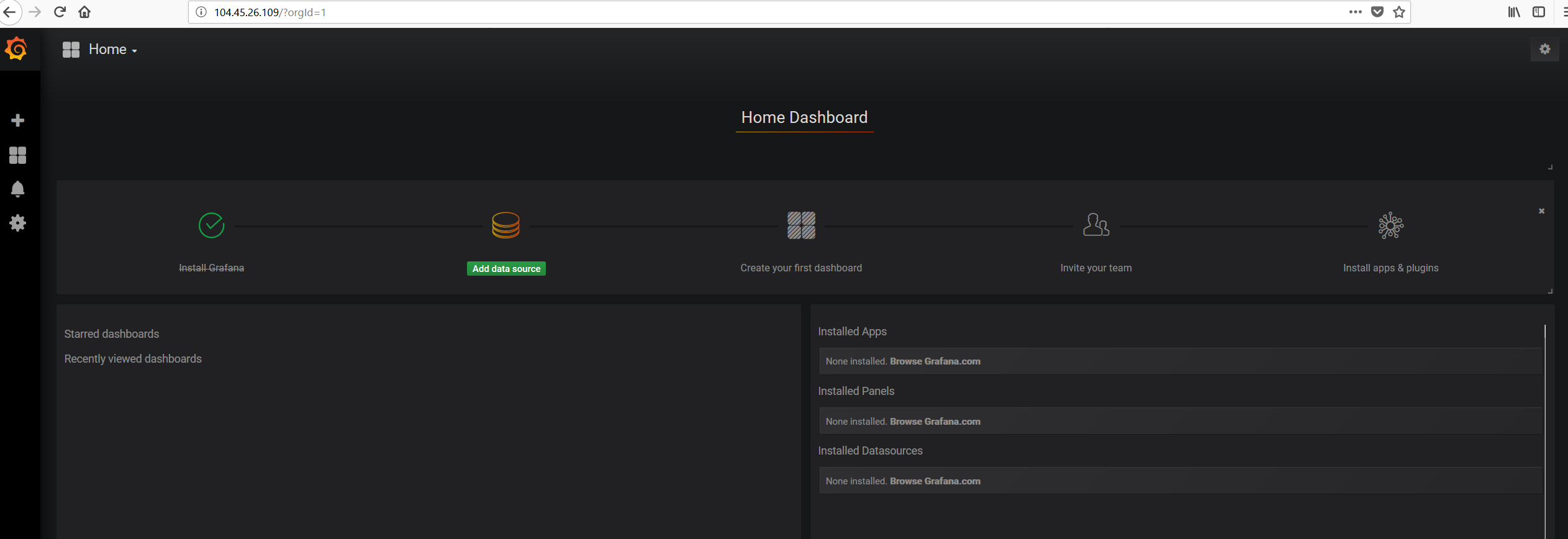
* eg. [http://52.226.75.38](http://52.226.75.38/), EXTERNAL-IP column from above. You should see something like the following come up, be patient it will take a moment or two:



**Setting up Grafana**

1. Log into Grafana Dashboard using **admin** for the username and password

* You should see something like the following:



1. Add Prometheus as a Data Source

* If you recall from above, we exposed a number of K8s services, one of those services was the Prometheus Server. We are going to use that Service endpoint in our Data Service configuration. The Add Data Source screen should look something like the below screen shot.

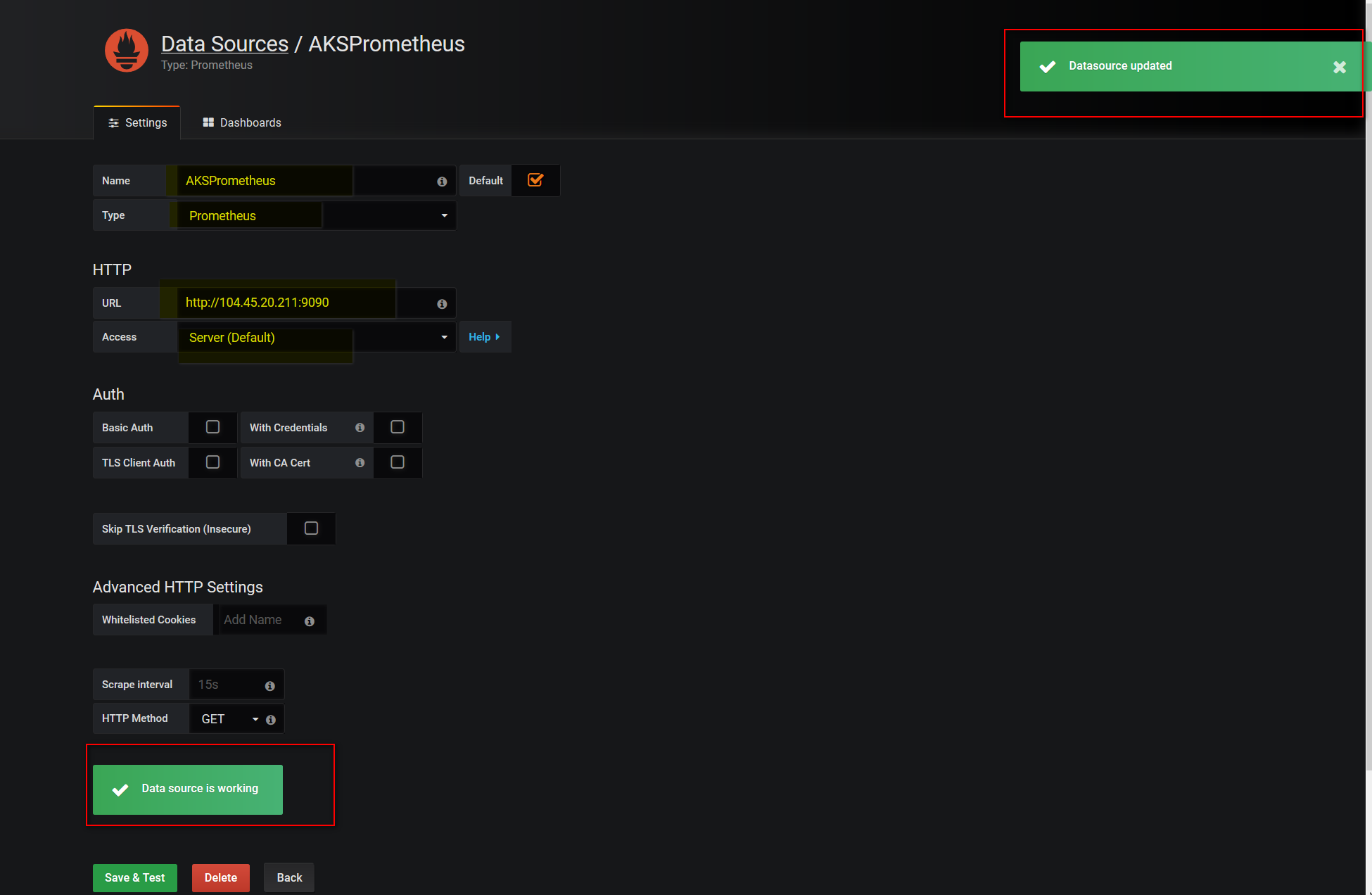
Use http://{PROMETHEUS\_PUBLIC\_IPADDRESS}:9090 for the URL in the HTTP settings.

1. Validate Prometheus Data Source

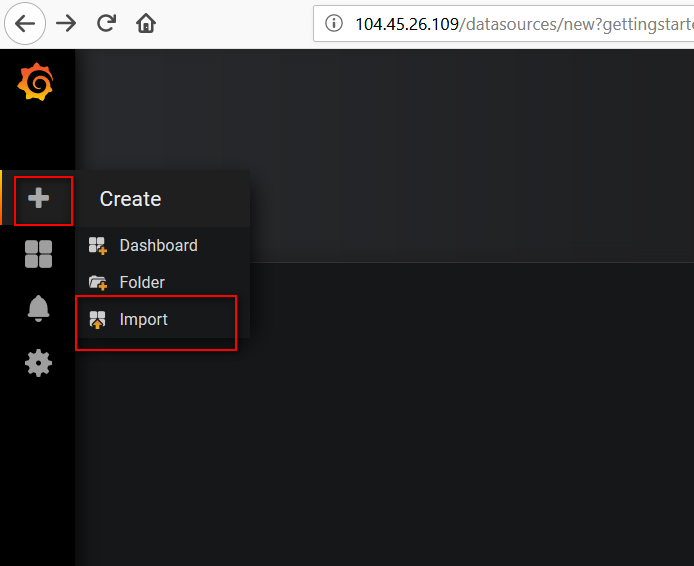
* Once you have filled in the values similar to the screenshot above, click the **Add** button and ensure no errors come back.

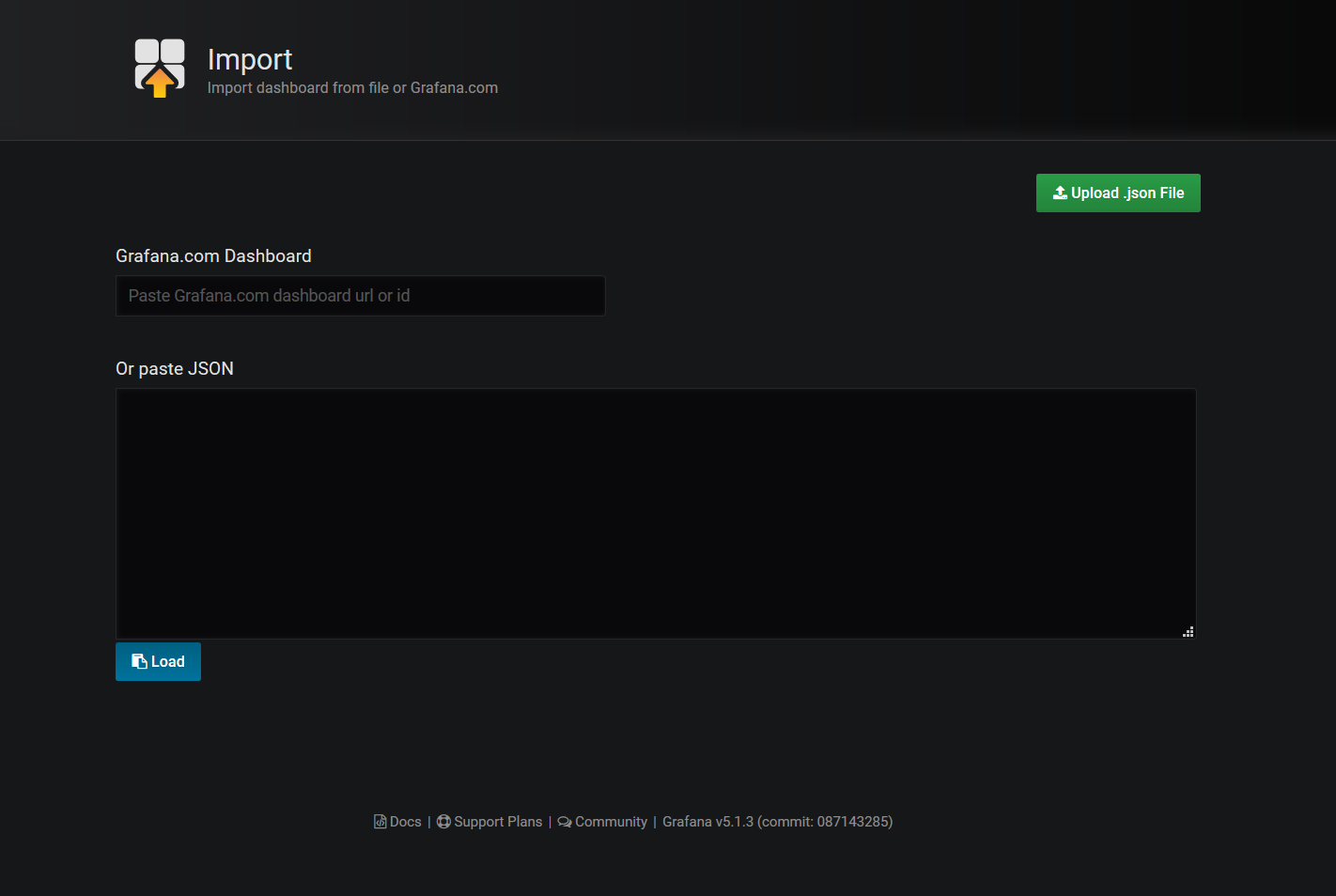
1. Add K8s Monitoring Dashboard to Grafana

* After the datasource has been added, it is now time to add a dashboard. Grafana dashboards can be shared on Grafana.com. Go to import dashboards viam the menu in the top left.

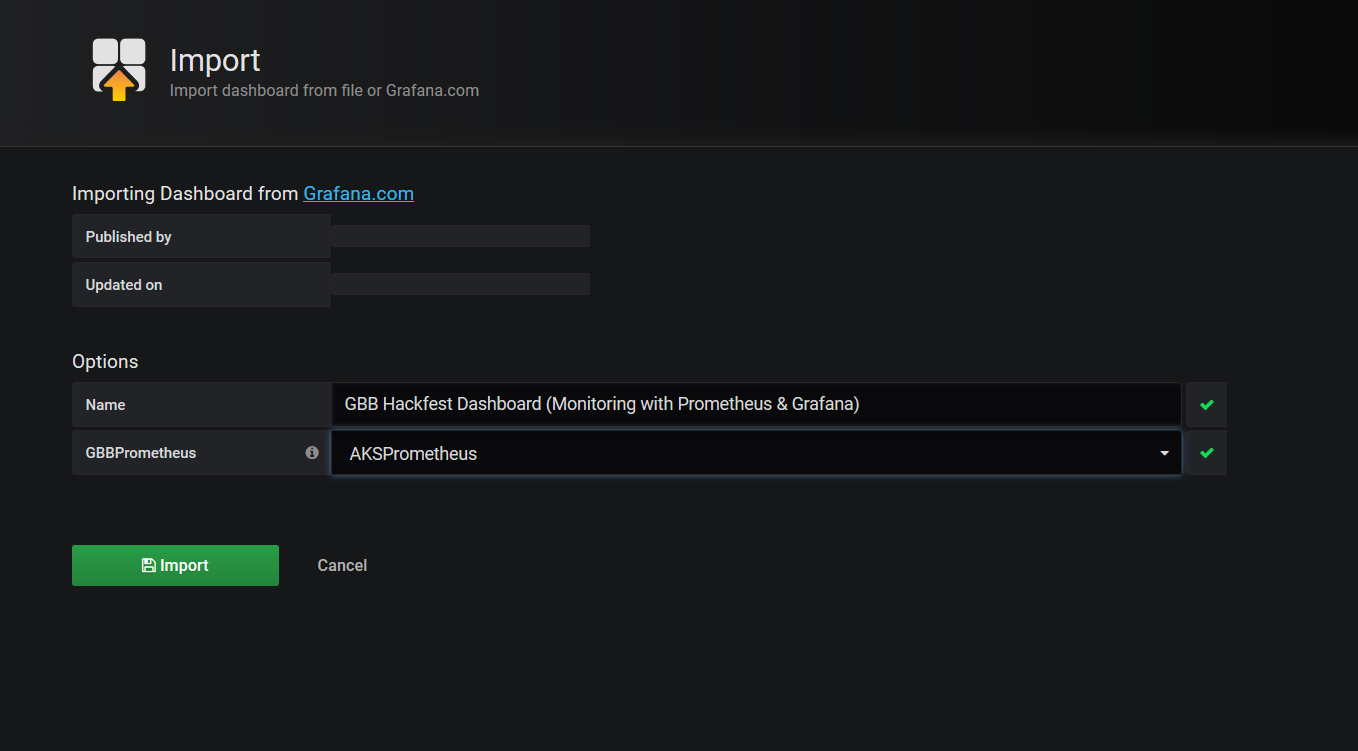


* Click on the **Upload File** button and browse to the **grafana-dashboard.json** in the **cpx-oss-workshop/labs/helper-files/** directory. You can also paste the contents of the json into the text box.

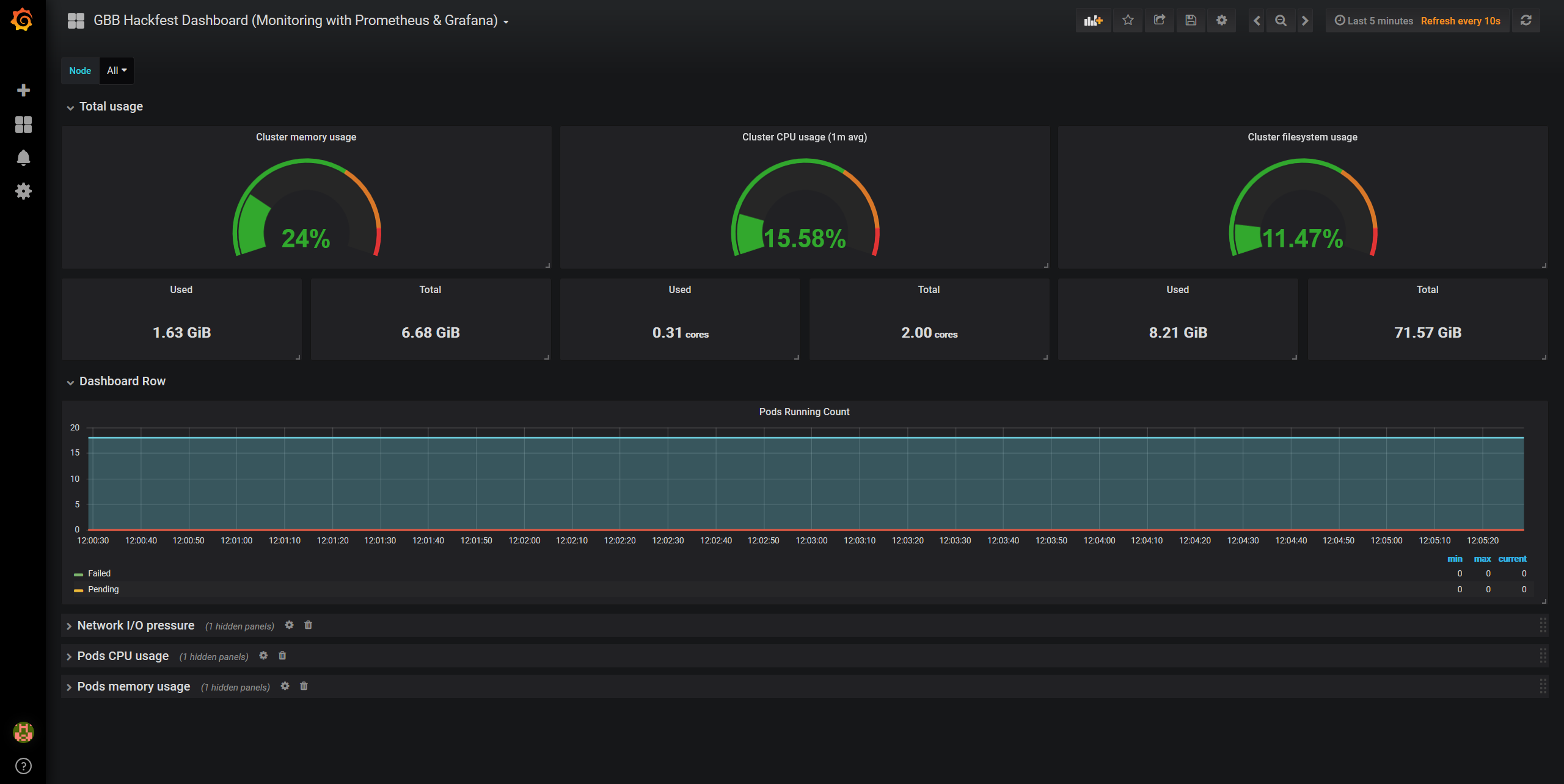




* Set the datasource dropdown to the "AKSPrometheus" that was created in the previous step.



* Click the **Import** button.



You should now have Prometheus and Grafana running in your Azure Kubernetes Service cluster and be able to see the Grafana Dashboard.