To check the cluster details :

```
$ kubectl cluster-info
```

To check recent events

```
$ kubectl get events
```

To deploy the dashboard do :

Run the following command to deploy the dashboard:

```
$ kubectl create -f https://raw.githubusercontent.com/kubernetes/dashboard/master/src/deploy/recommended/kubernetes-dashboard.yaml
```

Accessing Dashboard using the kubectl

```
$ kubectl proxy
```

It will proxy server between your machine and Kubernetes API server.
Now, to view the dashboard in the browser, navigate to the following address in the browser of your Master VM:

http://<master-ip>:8001/api/v1/namespaces/kube-system/services/https:kubernetes-dashboard:/proxy/

https://<master-ip>:<apiserver-port>/api/v1/namespaces/kube-system/services/https:kubernetes-dashboard:/proxy/

Create a Cluster Admin service account

In this step, we will create the service account for the dashboard and get its credentials.
***Note**: Run all these commands in a **new terminal**, otherwise your kubectl proxy command will stop.*
Run the following commands:
This command will create a service account for a dashboard in the default namespace

```
$ kubectl create serviceaccount dashboard -n default
```

Add the cluster binding rules to your dashboard account

```
$ kubectl create clusterrolebinding dashboard-admin -n default \ --clusterrole=cluster-admin \
--serviceaccount=default:dashboard
```

Copy the secret token required for your dashboard login using the below command:

```
$ kubectl get secret $(kubectl get serviceaccount dashboard -o jsonpath="{.secrets[0].name}")
-o jsonpath="{.data.token}" | base64 --decode
```

To get admin user's password::

**********************DAHSBOADS***********************

You can grant full admin privileges to Dashboard's Service Account by creating below ClusterRoleBinding. Copy the YAML file based on chosen installation method and save as, i.e. dashboard-admin.yaml. Use kubectl create -f dashboard-admin.yaml to deploy it. Afterwards you can use Skip option on login page to access Dashboard.

```
apiVersion: rbac.authorization.k8s.io/v1beta1
kind: ClusterRoleBinding
metadata:
  name: kubernetes-dashboard
  labels:
    k8s-app: kubernetes-dashboard
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: cluster-admin
subjects:
- kind: ServiceAccount
  name: kubernetes-dashboard
  namespace: kube-system
```

```
$ kubectl create -f dashboard-admin.yaml
```