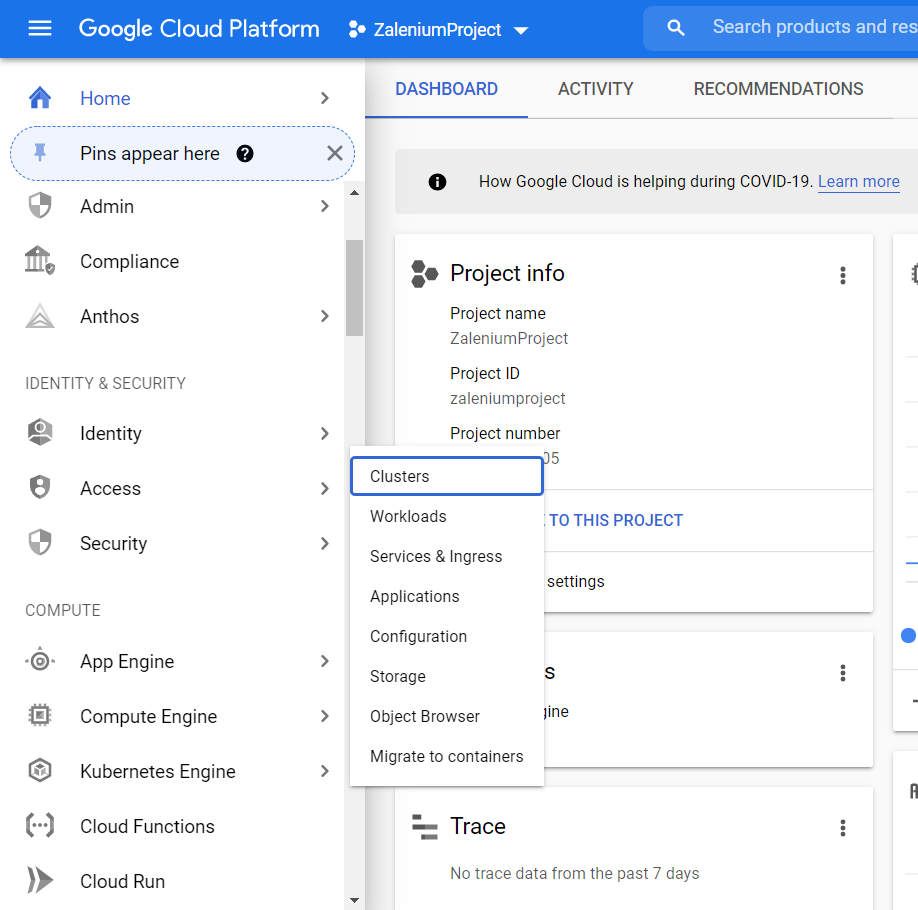
* **Setup to run script using Zalenium with kubernetes Engine**

**Zalenium -** A flexible and scalable container based Selenium Grid with video recording, live preview, basic auth & dashboard.

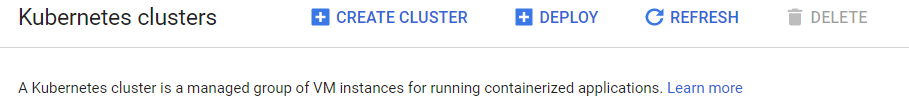
**Kubernetes** - Kubernetes is an open-source container-orchestration system for automating computer application deployment, scaling, and management.

**Steps to create kubernetes cluster in GCP**

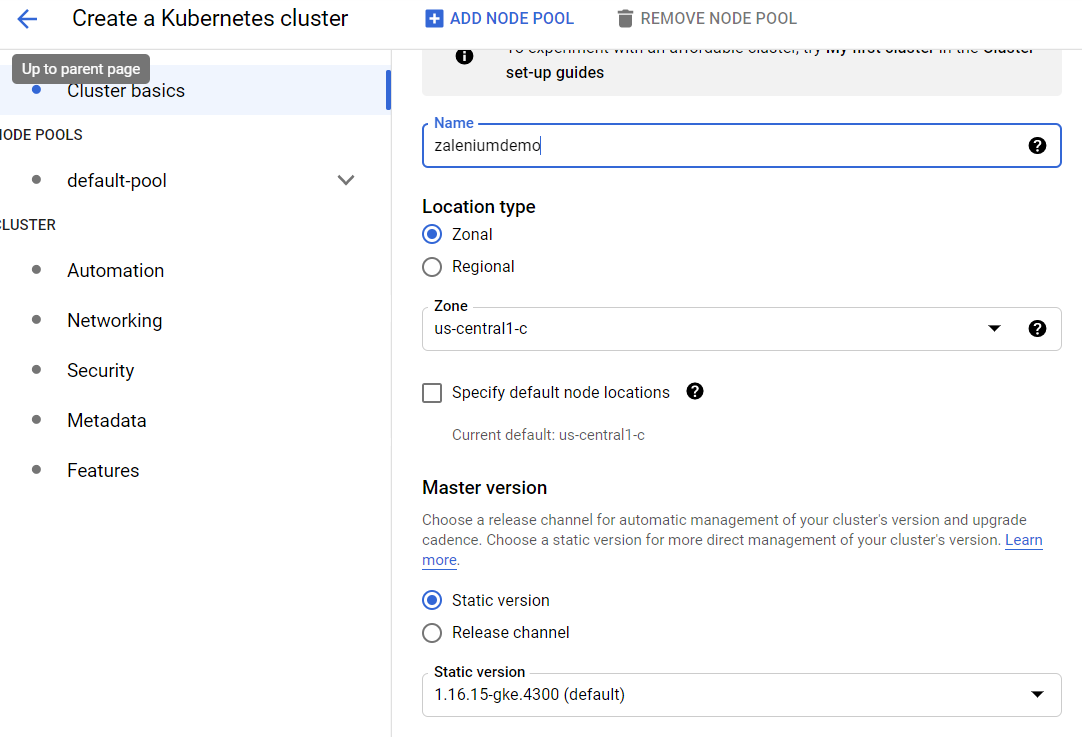
1. Login to the GCP Account and select cluster from kubernetes engine Service.



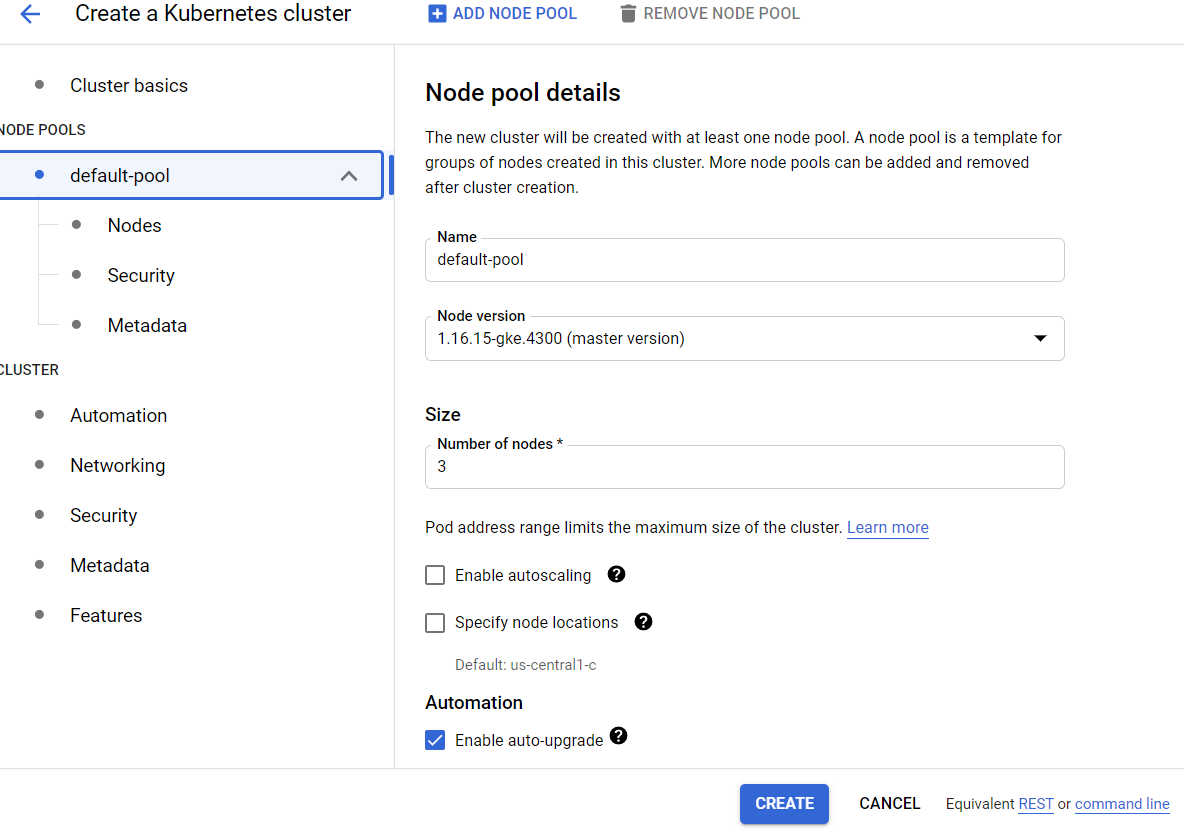
1. Click on create cluster



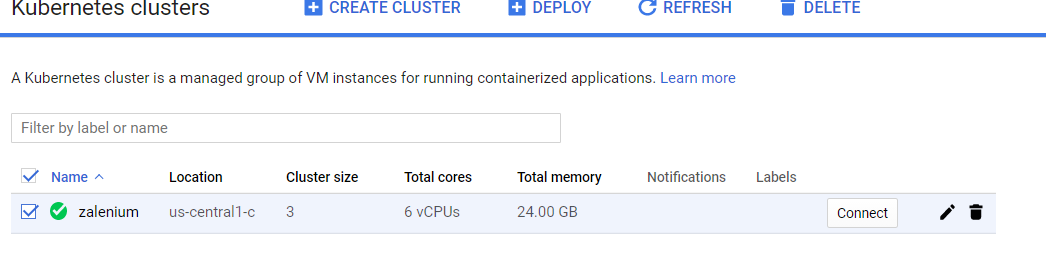
1. Give a name to the cluster, select zone, select master version and click on default pool.



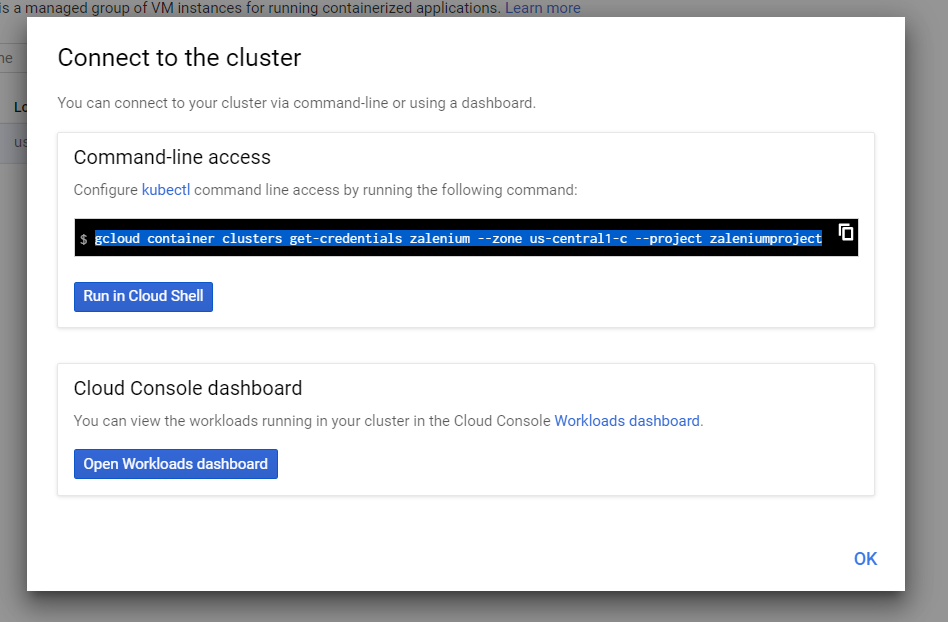
1. Give a pool name and select no of nodes and then click on create.



1. Once cluster is setup click on connect to the cluster.

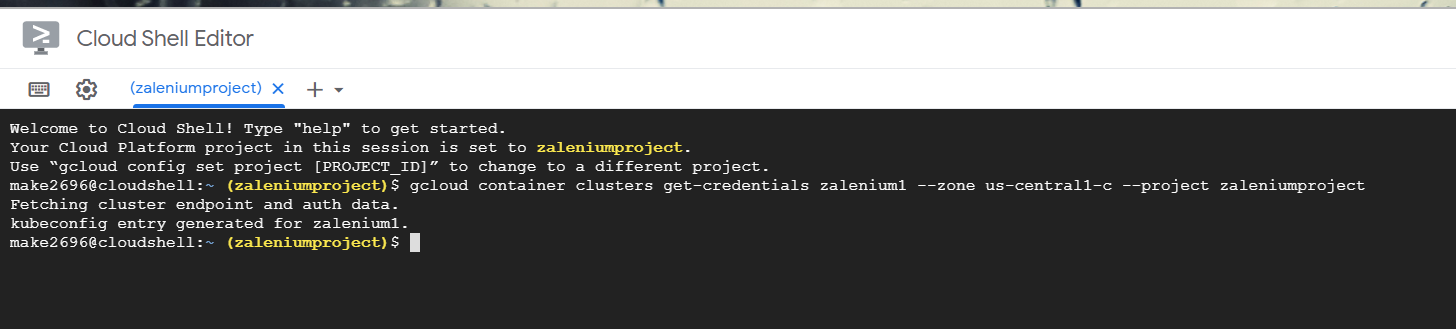


1. Copy command and click on run in cloud shell.

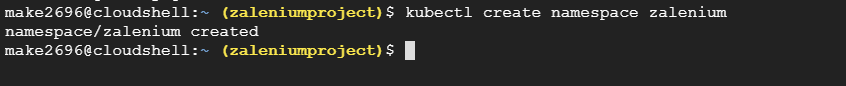


1. Run the below command in cloud shell to connect to the cluster.

**gcloud container clusters get-credentials zalenium1 --zone us-central1-c --project zaleniumproject**



1. Create a name space with below command.



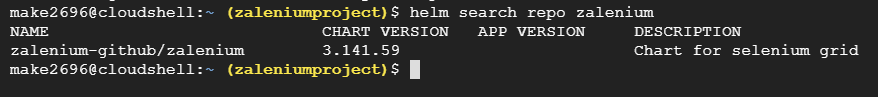
1. Clone Zalenium repo using below command.

**helm repo add zalenium-github** [**https://raw.githubusercontent.com/zalando/zalenium/master/charts/zalenium**](https://raw.githubusercontent.com/zalando/zalenium/master/charts/zalenium)



1. To check the repo is present or not enter below common.

**helm search repo zalenium**



1. To create a instance run below command.

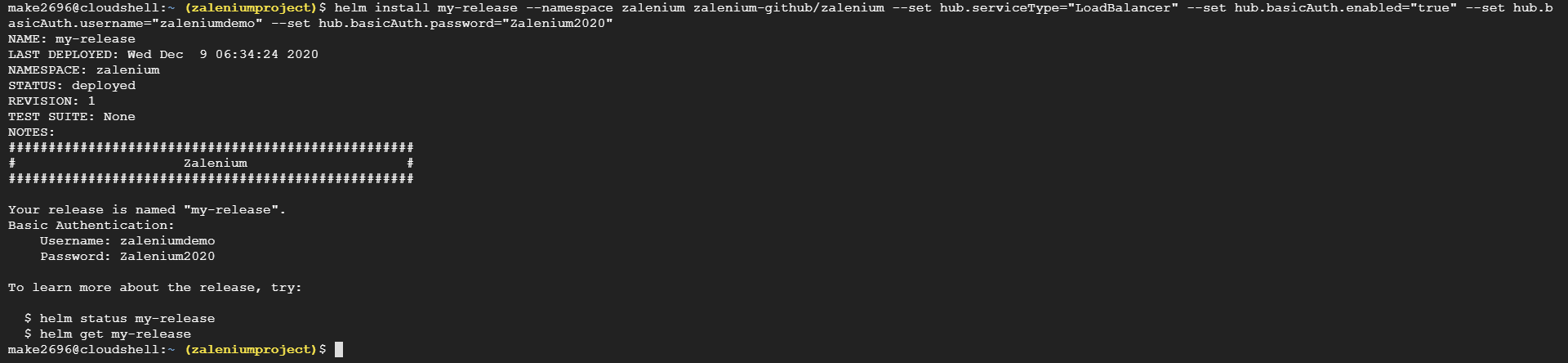
**helm install my-release --namespace zalenium zalenium-github/zalenium --set hub.serviceType="LoadBalancer" --set hub.basicAuth.enabled="true" --set hub.basicAuth.username="zaleniumdemo" --set hub.basicAuth.password="Zalenium2020"**

**my-release : enter release name**

**--namespace repository name : Zalenium**

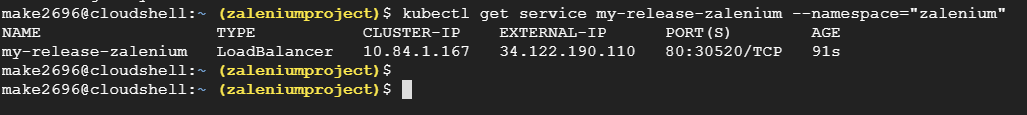
**Hub.serviceType : “Load Balancer” because we are using multiple node**

**Then basic auth for authentication.**



1. To get the service details enter below command.

**kubectl get service my-release-zalenium --namespace="zalenium"**



1. Use the external ip .

External-IP : 34.122.190.110

**Zalenium Dashboard** – http:// 34.122.190.110/dashboard/#

Zalenium Live Preview – http:// 34.122.190.110/grid/admin/live

Zalenium Grid Console – http:// 34.122.190.110/console

Zalenium Hub – http:// 34.122.190.110/wd/hub

1. Copy command and click on run in cloud shell.

**Content in docker-compose.yml file**

**Version 3:** version of the compose file format.

**Services:** Defines all the services you want to deploy.

We are using three services 1 for selenium hub and two for nodes chrome and Firefox.

**Image:** defines docker image e.g. selenium/hub, selenium/node-chrome, selenium/node-firefox.

**Ports:** Ports mentioned in docker-compose.yml will be shared among different services started by the docker-compose.

**Environment:** It is used to set environment variables.

**MAX\_SESSION:** This means how many browsers can run in parallel at a time in remote system.

**MAX\_INSTANCES:** This means how many instances of the same browser can run over the Remote System.

**HUB\_PORT\_4444\_TCP\_ADDR:** This represents the address of the hub.

**HUB\_PORT\_4444\_TCP\_PORT:** This represents the port number which we are using for hub e.g. 4444.

* **To See the execution**

1. Replace the image files for selenium/node-chrome and selenium/node-firefox with selenium/node-chrome-debug and selenium/node-firefox-debug in docker-compose.ym file as below attached file.

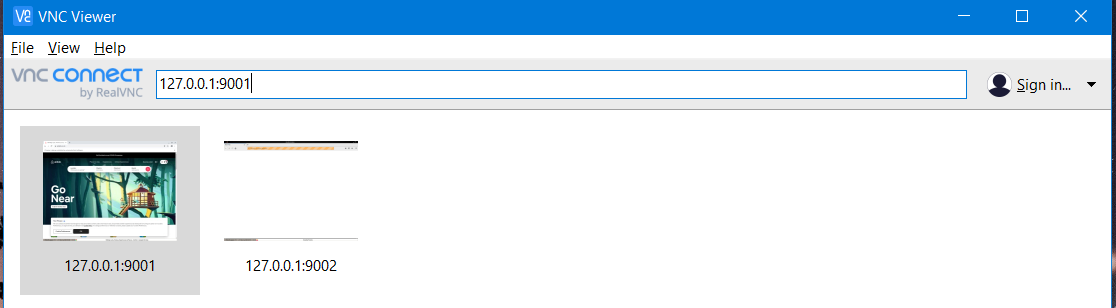
****

1. Download VNC Viewer from below url.

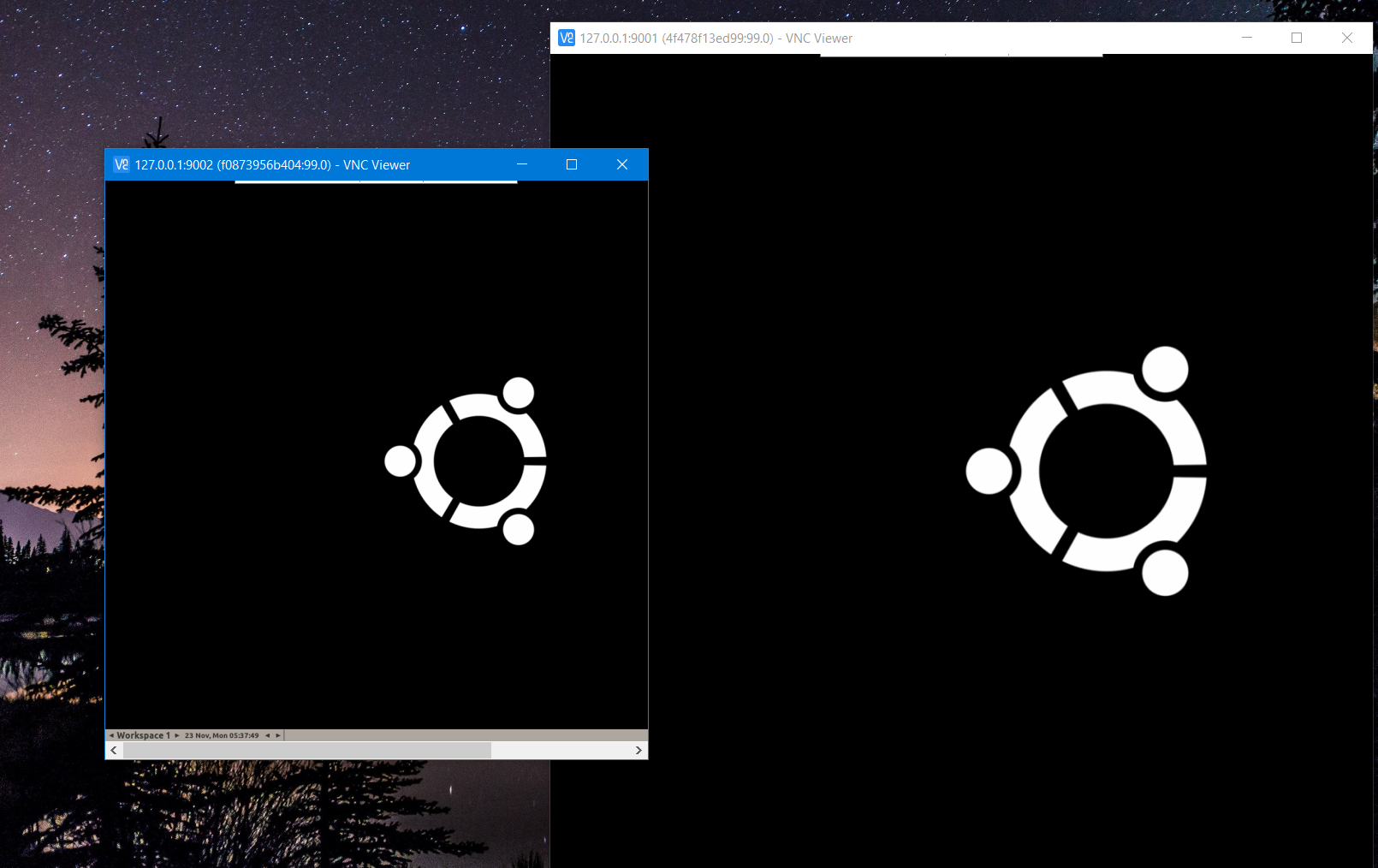
<https://www.realvnc.com/en/connect/download/viewer/>

1. After starting server and nodes open VNC viewer.
2. Enter hub address followed by port number in address bar of VNC viewer as below screenshot.

For local machine give 127.0.0.1 followed by port number.



1. Enter password as **secret** in VNC viewer and click on continue.



1. When you start the execution it’ll show in VNC viewer.

