**Selenium Hub deployment in OpenShift cluster (GCP)**

Table of Contents

Overview 3

Prerequisite3

Installation of OpenShift in Ubuntu 18.04 4

How to create OpenShift cluster using OC commands 5

Deployment of Selenium hub in OpenShift cluster 9

1. Deployment using OpenShift web UI 9

2. Deployment using YAMLs 16

**Overview**

This article gives the understanding of how to install OpenShift in Ubuntu virtual machine, how to create cluster and deploy selenium hub in the cluster.

**Prerequisite**

To work on OpenShift as per this article,

* **Google cloud console account**
* **Ubuntu virtual machine**
* **Docker**
* **Kubernetes**

**Installation of OpenShift in Ubuntu 18.04**

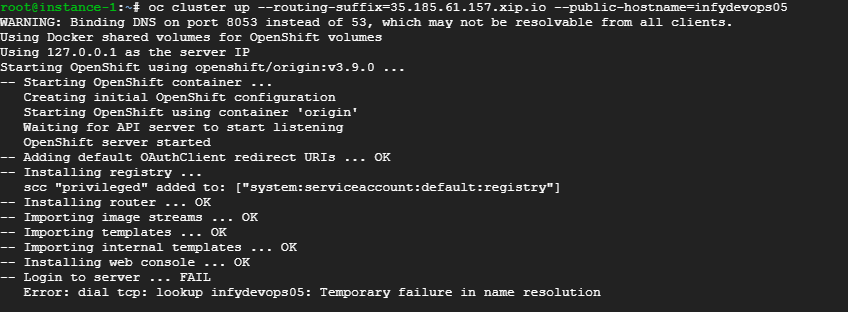
* Open your Ubuntu virtual machine in Google cloud platform.
* Before installing OpenShift into machine, make sure that Docker is installed in machine first. Because OpenShift basic requirement is
  + 1. Docker
    2. Kubernetes
* You have to install Docker externally by below steps.
* Execute below commands step by step.
  + **sudo apt update**
  + **sudo apt install apt-transport-https ca-certificates curl software-properties-common**
  + **curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add –**
  + **sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu bionic stable"**
  + **sudo apt update**
  + **sudo apt-cache policy docker-c0e**
  + **sudo apt install docker-ce**
  + **sudo systemctl status docker**
* By executing all the above command step by step you can install Docker and also check the status of Docker.
* Now that you have install Docker in your machine execute below command to install OpenShift. When you execute below commands it will install kubernetes as well.
* Make sure that Git is installed in your system. Then execute below command.
* **wget** [**https://github.com/openshift/origin/releases/download/v3.9.0/openshift-origin-client-tools-v3.9.0-191fece-linux-64bit.tar.gz**](https://github.com/openshift/origin/releases/download/v3.9.0/openshift-origin-client-tools-v3.9.0-191fece-linux-64bit.tar.gz)
* So above command will download package for you. So now you have to un-tar this package by below command.
* **tar -zvxf openshift-origin-client-tools-v3.9.0-191fece-linux-64bit.tar.gz**
* Then execute below command.
* **cd openshift-origin-client-tools-v3.9.0-191fece-linux-64bit**
* **sudo cp oc /usr/local/bin/**
* Check the OC version using below command.
* **oc version**
* Now before executing any oc command, make sure that you are adding below content to daemon.json file of Docker. Because OpenShift caches images locally for image re-use purposes. To add content execute below command.

**sudo su -  
cat << EOF > /etc/docker/daemon.json   
{  
 "insecure-registries" : [ "172.30.0.0/16" ]  
}  
EOF**

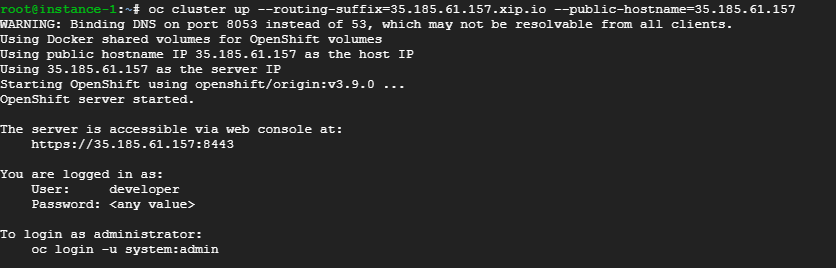
* Now restart Docker. To do that execute below command.
  + - **service docker restart**
* Now we have installed OpenShift and OC in our Ubuntu virtual machine. Now our requirement is to create OpenShift cluster and deploy selenium hub into it to execute our selenium test cases.

**How to create OpenShift cluster using OC commands**

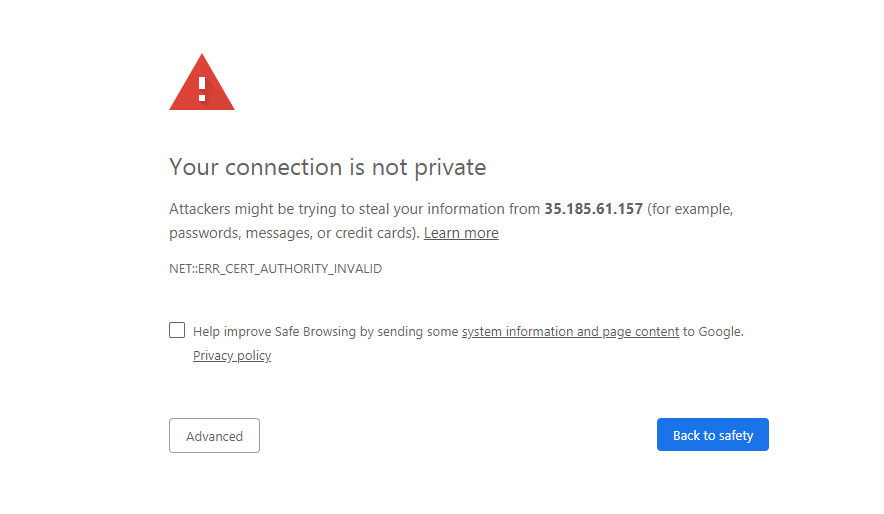
* So to create OpenShift cluster using OC command, execute below command.
  + - **oc cluster up --routing-suffix=<your public ubuntu IP address>.xip.io --public-hostname=<your ubuntu public DNS>**
* Now when you execute above command replace all the content with <> by public IP of your virtual machine. This is necessary.
* If you are specifying your virtual machine’s hostname in place of –public-hostname part then your cluster will get created on private ip. So you won’t be able to access the cluster web UI. This is not preferable. So make sure that you are using the public IP of your virtual machine. Which will create your OpenShift cluster on public IP and it will easily accessible.
* If you are not using hostname and using other name in the part of –public-hostname, in this case you will face below error.



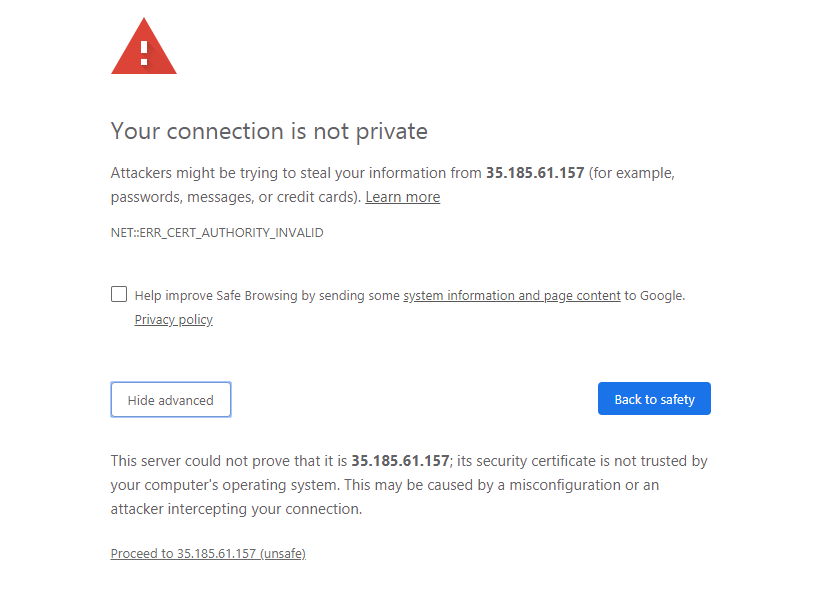
* So make sure that you are using the specific public IP in term of –public-hostname.
* So the final command to create OpenShift cluster is shown below.
  + - **oc cluster up --routing-suffix=35.185.61.157.xip.io --public-hostname=35.185.61.157**
    - **35.185.61.157 is public ip of my virtual machine.**
* So when you execute above command, you will get below output.

****

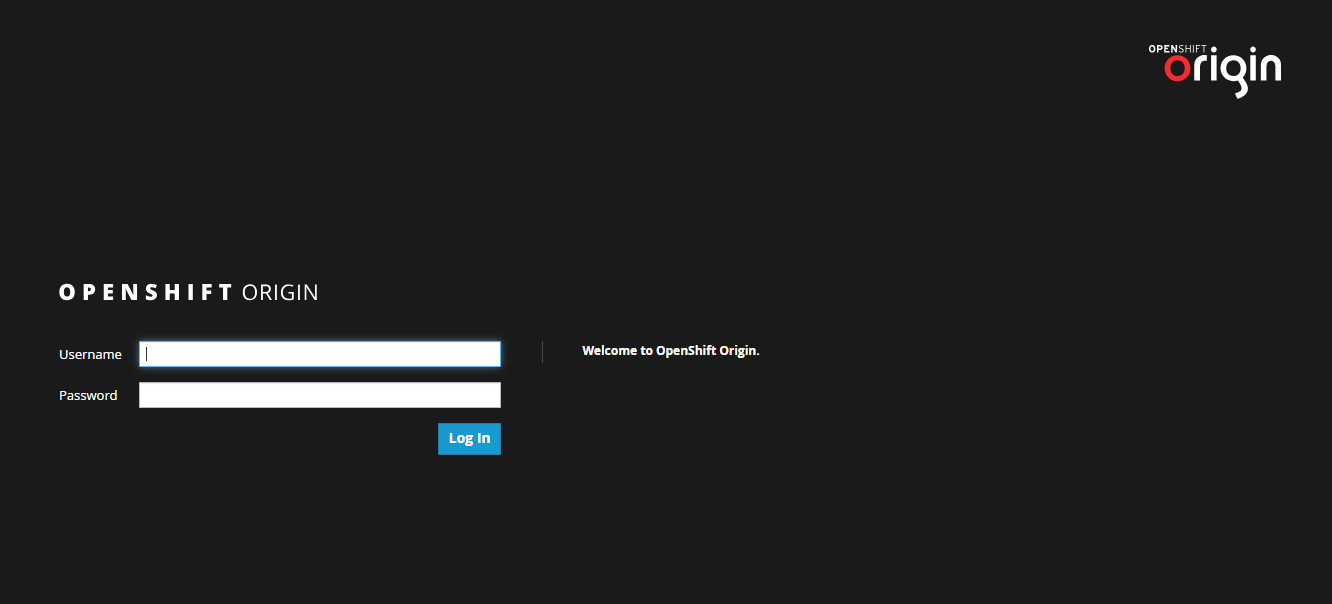
* As shown above, it is showing the server URL to access the web console.
* So use that URL to access the web UI.
* When you use that URL, in first step it will give you an error, which is SSL certificate error. As shown below.

****

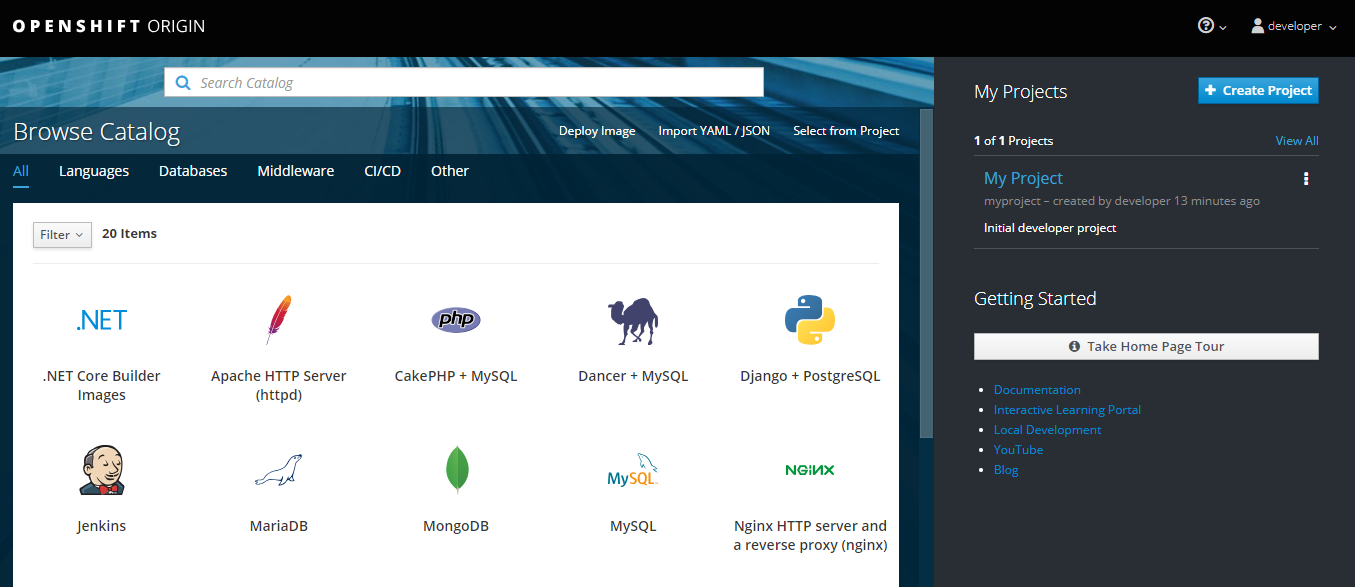
* To overcome this error click on that “Advanced ” button.
* After clicking on the “Advanced” button, it will give you below output.

****

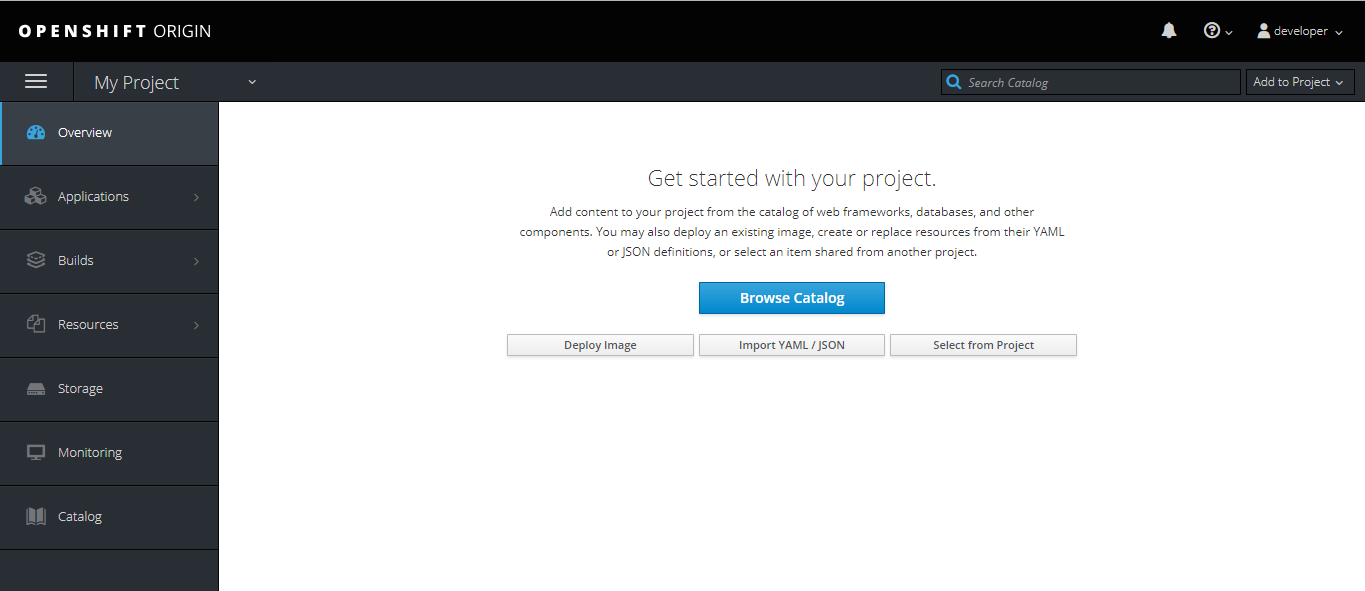
* So now click on the “Proceed to 35.185.61.157 (unsafe)” link. It will directly take you to the OpenShift web console.
* Where it will ask you username and password, as shown below.

****

* Add username and password as per your wish and click on Login and remember the username and password.
* After clicking on the Log in button. It will take you to the home of OpenShift web console.

****

* As you can see, it has already created on project which is “My Project”. Click on that project it will show you all the resources if there are any in your project.
* As I have just created the cluster, I don’t have any pods, services, deployments in my project. So, it will show you the overview as shown below.



* So currently, there is nothing the cluster.

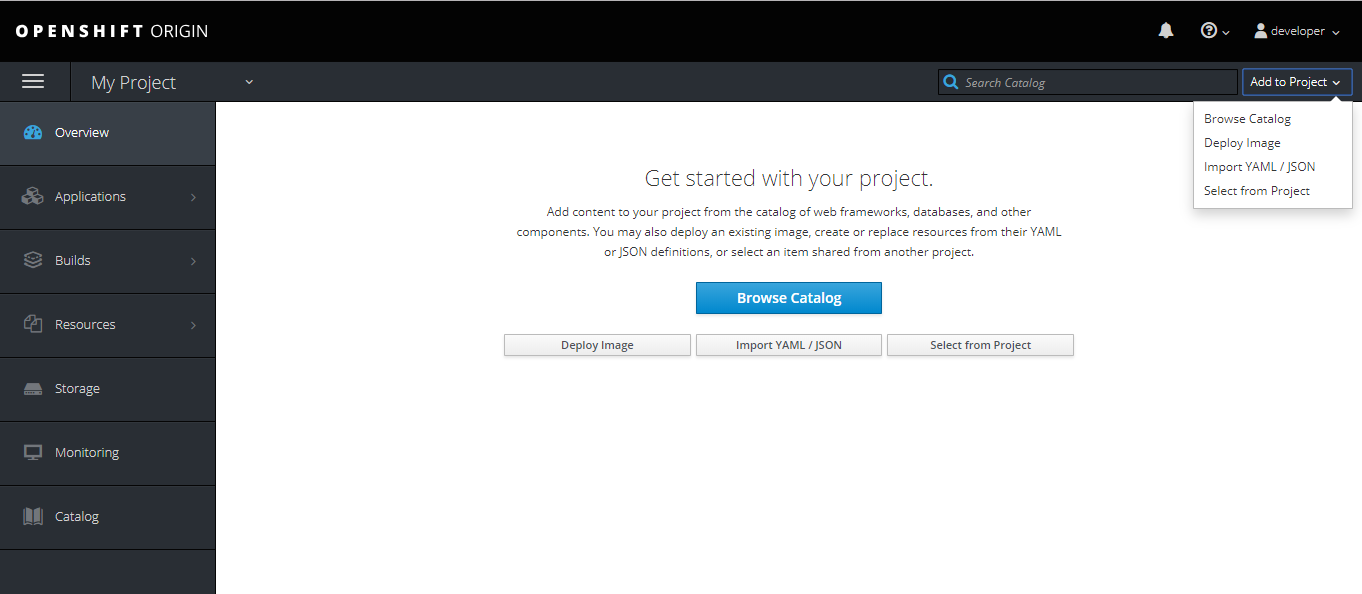
**Deployment of Selenium hub in OpenShift cluster**

* Now our requirement is to deploy Selenium-hub in the cluster.
* You can do this by 2 ways.

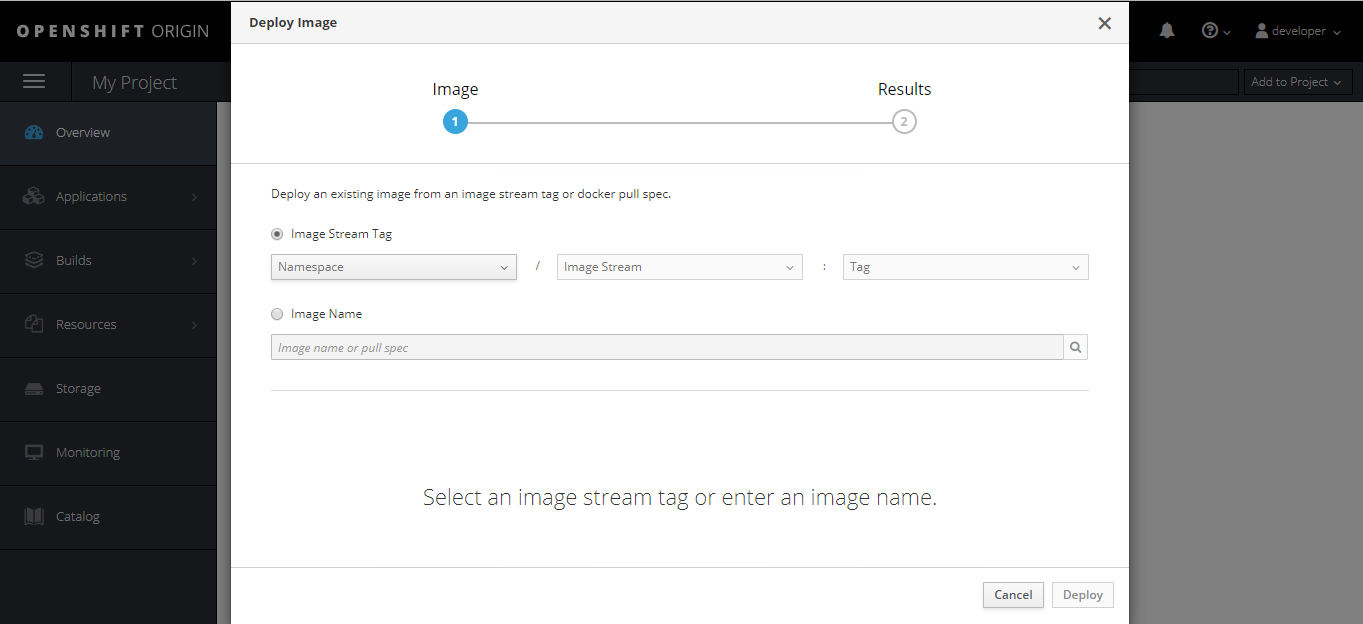
1. **By OpenShift web UI.**
2. **By executing Selenium hub yamls from Virtual machine using OC commands.**

**1. Deployment using OpenShift web UI**

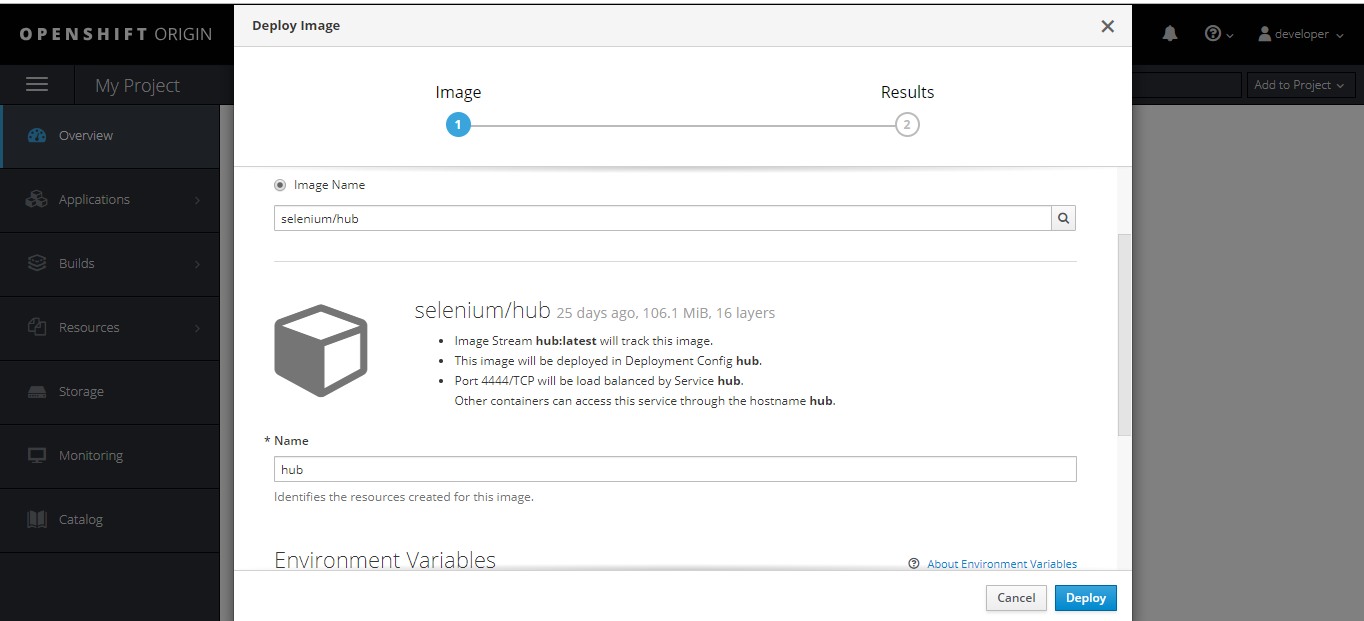
* So, let’s start with “By OpenShift web UI”.
* To deploy selenium hub from OpenShift web UI. You can either choose your own yamls or images. In terms of yamls part you have to copy past content or you can browse your yamls as well from your system.
* But in this case I will show you how you can deploy selenium-hub using docker images from OpenShift web UI.
* Please refer below screenshot for more understanding.

****

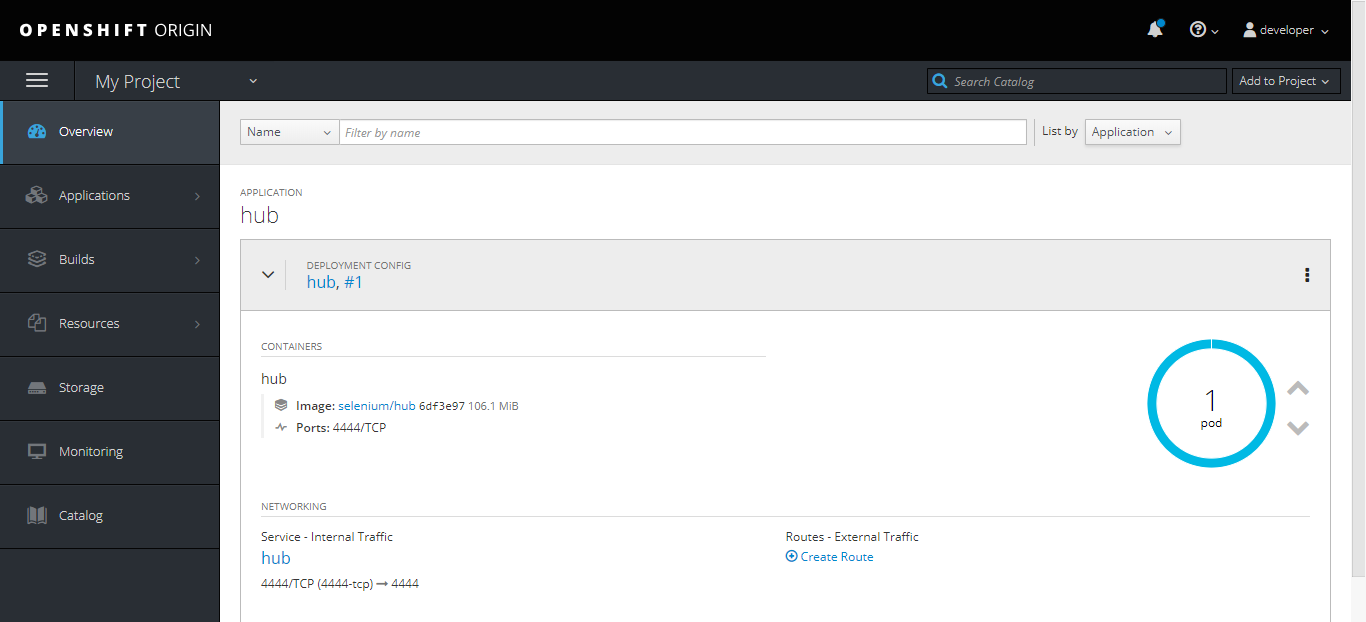
* As shown in the picture click on the “Add to Project” dropdown and it will give you list out of which select “Deploy Image”.
* It will directly take you to the “Deploy image” page. As shown below.

****

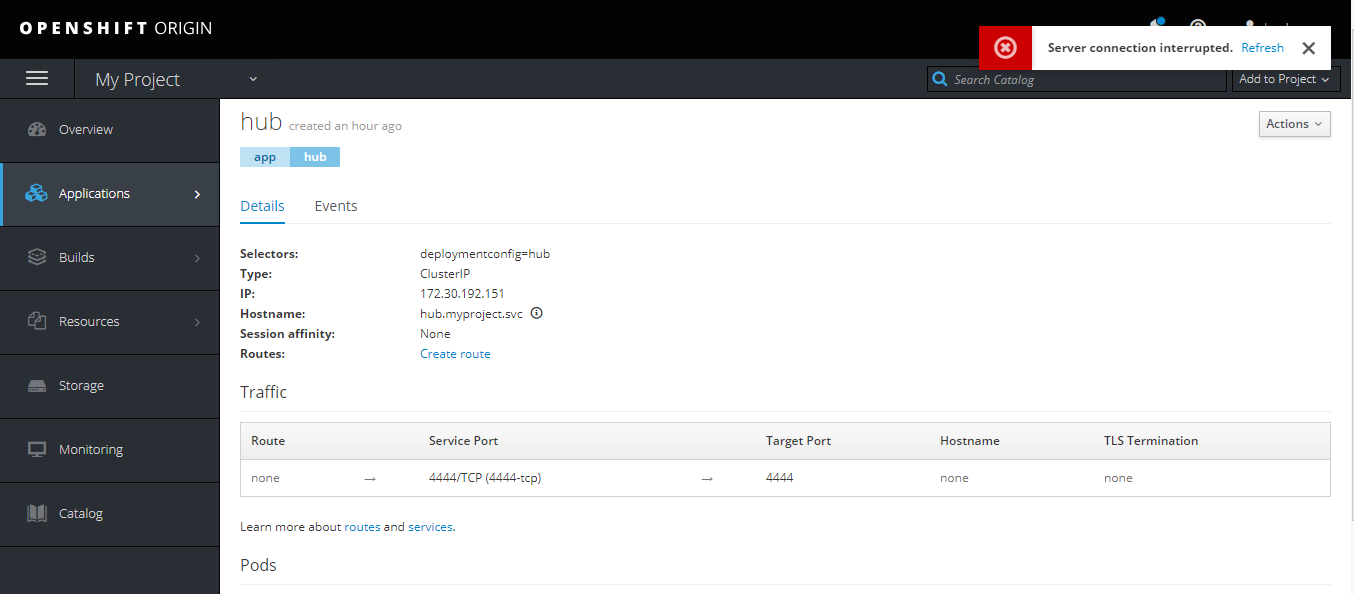
* In this page select “Image Name”.
* Then you have to type your image name which you want to deploy.
* So in our case we want to deploy selenium hub, so type selenium hub image name and click on the search option.
* It will show you below page.

****

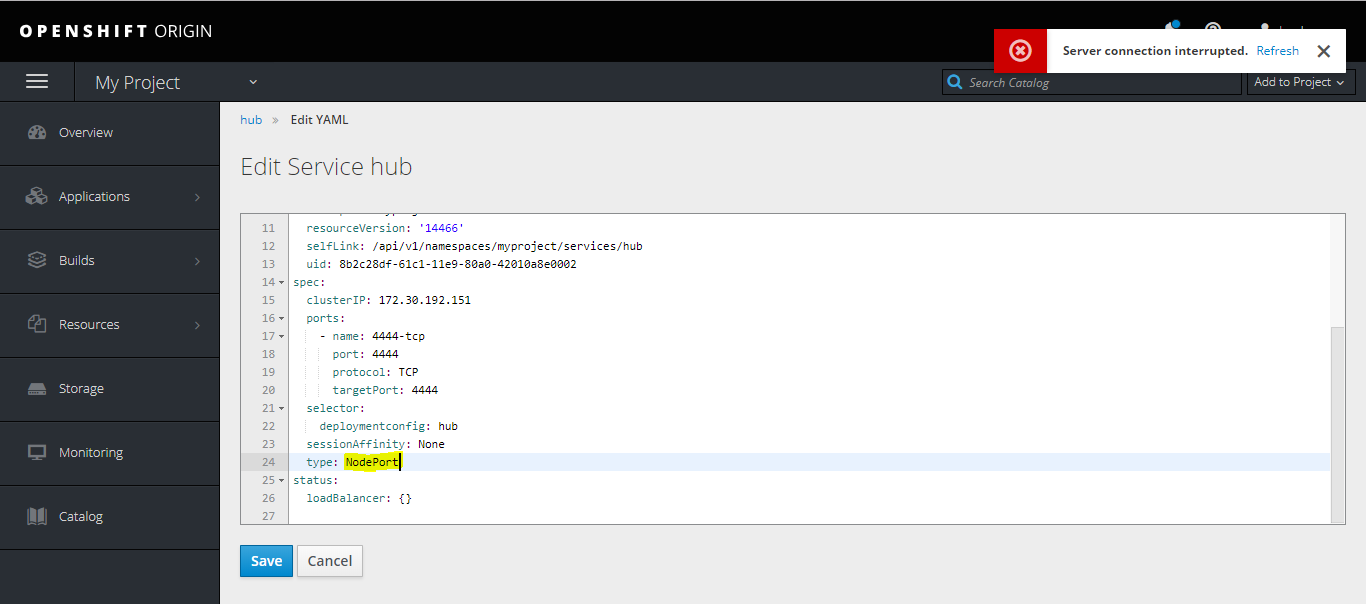
* So as shown in the picture, you can see that selenium hub image has been identified by the OpenShift.
* You can add your labels and environment variables as per the requirements.
* At the end click on the deploy button and check the selenium hub resource.
* You will get one selenium-hub running.
* As shown below.

****

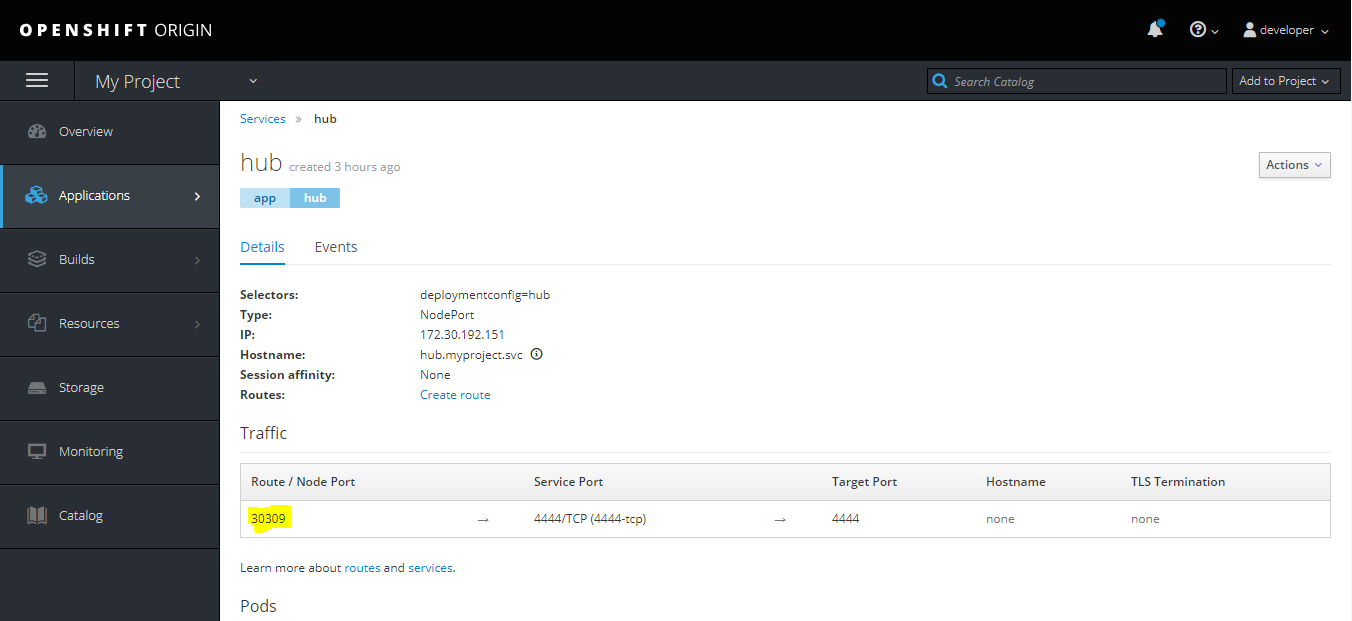
* You can see that one selenium hub pod has been created.
* One important part to which should be consider here is that when you check the services, you will have one service created which is not created by you, which is created by OpenShift itself and it is “Selenium-hub” service.
* But this service is created in “Cluster-IP” type. By which we can understand that the selenium-hub is accessible inside cluster only not outside the cluster.
* So you have to do some changes in the yaml of this service. You can do this from web UI only.
* Go to the “Applications” and then “Services” and then select the “hub” service.
* When you click on the selenium hub service. It will give whole information about the service.
* Please check the below screenshot.



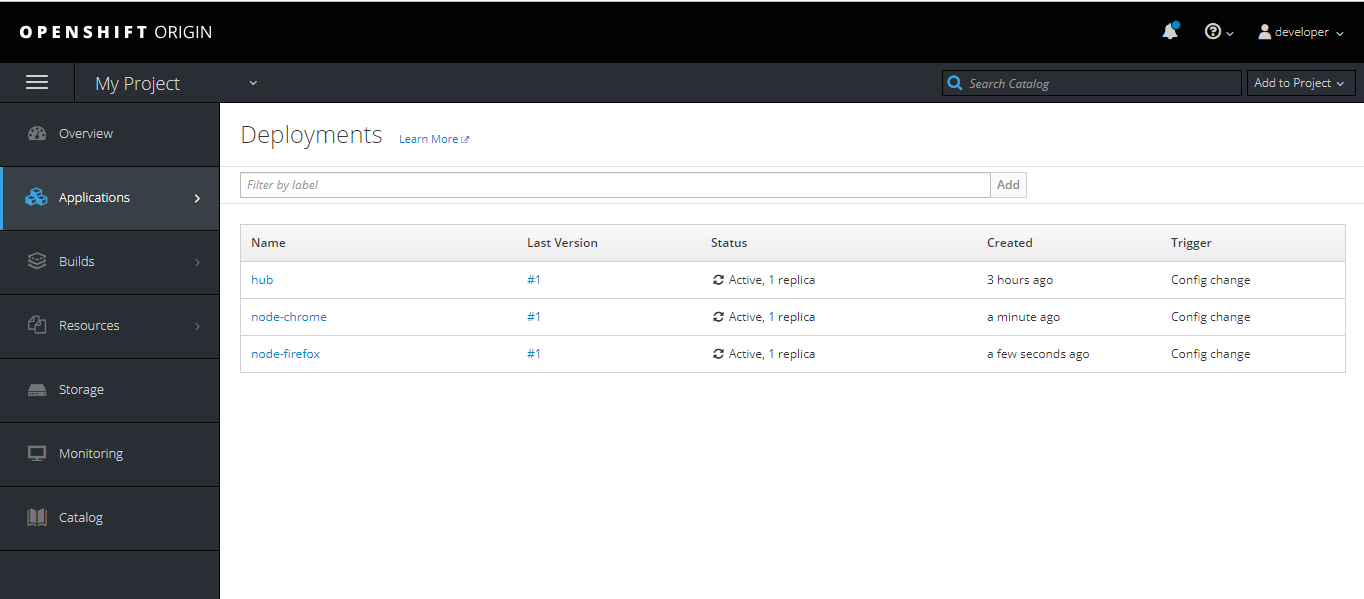
* In this click on the “Actions” button and then click on the “Edit Yaml.”
* Then change the service type to “NodePort” and save it. As shown below.



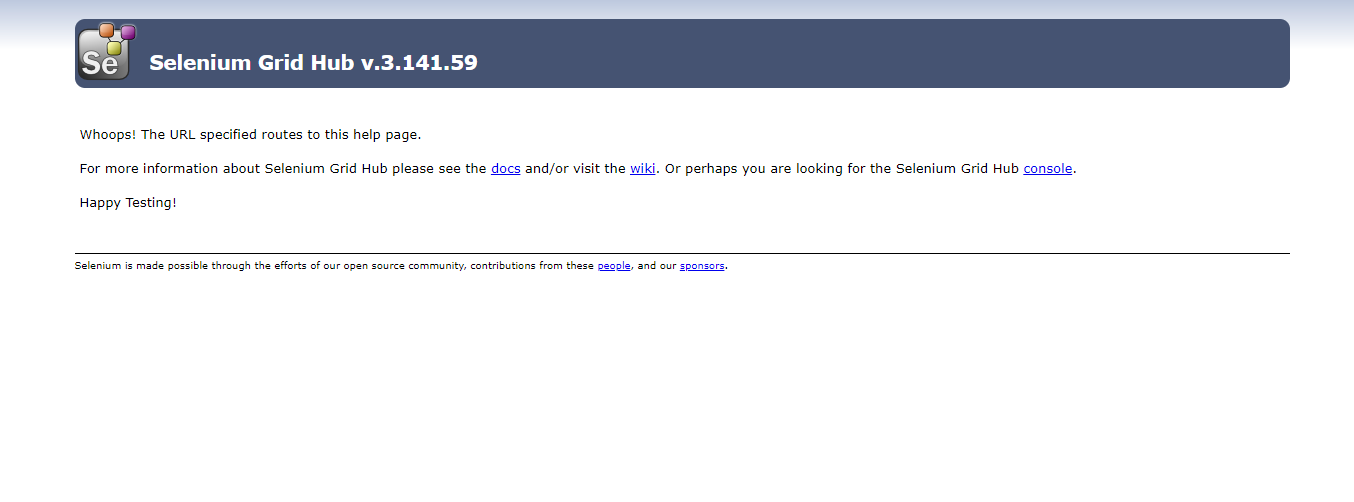
* This will create nodeport service and will give you “port” as well. As shown below.



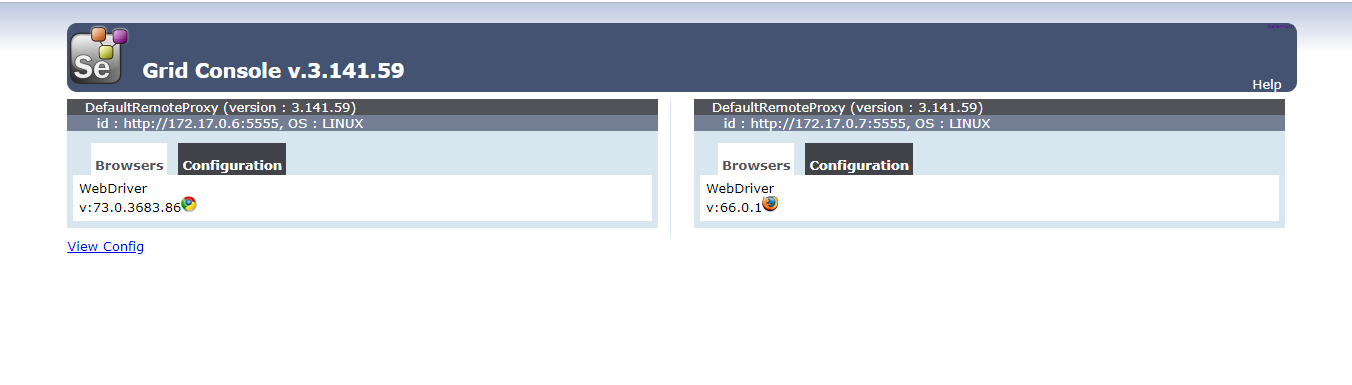
* Add this port number to your GCP firewall.
* Now deploy chrome node in the OpenShift. Same way as Selenium-hub deployment from OpenShift web UI.
* You can also deploy firefox node same way as Selenium-hub deployment from OpenShift web UI.
* After doing that you can check the list of pods in pods section. It will look like this.



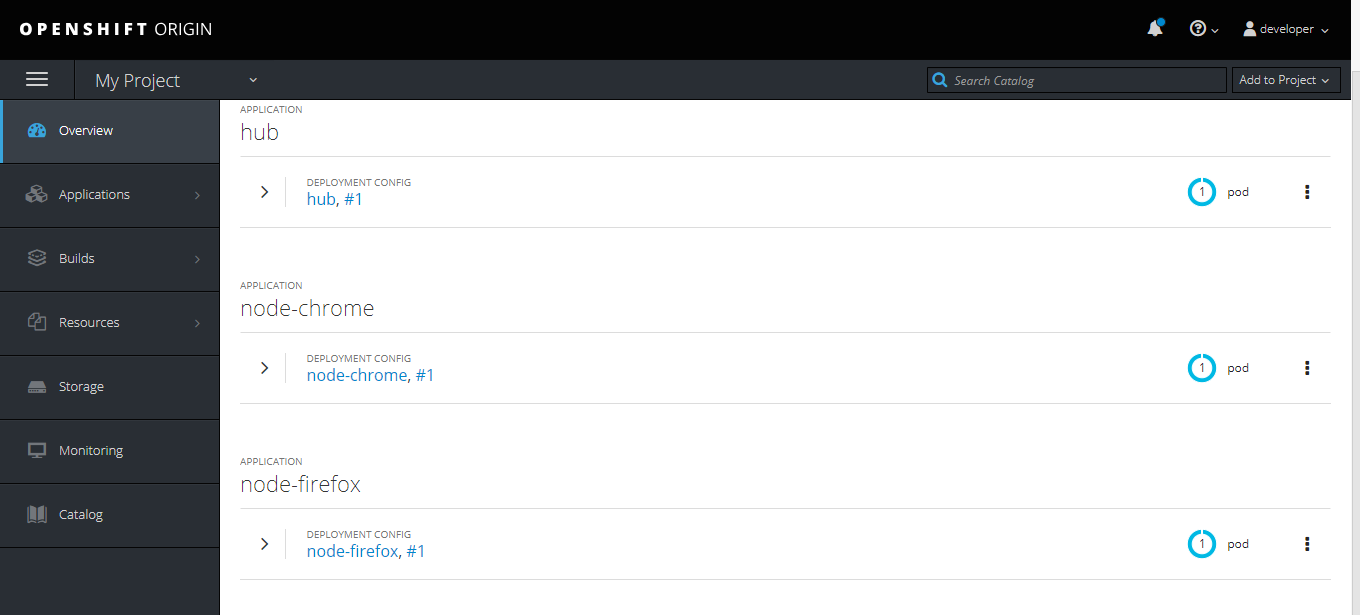
* So as you have deployed selenium hub, chrome node and firefox node. Also you have created the nodeport service for selenium hub. You can now access the selenium hub web UI.
* To access the selenium hub make sure you are using below URL.
  + - http://<public\_IP of virtual machine>:nodeport
    - In my case it is look like this “”
* Selenium hub web UI will look like this.



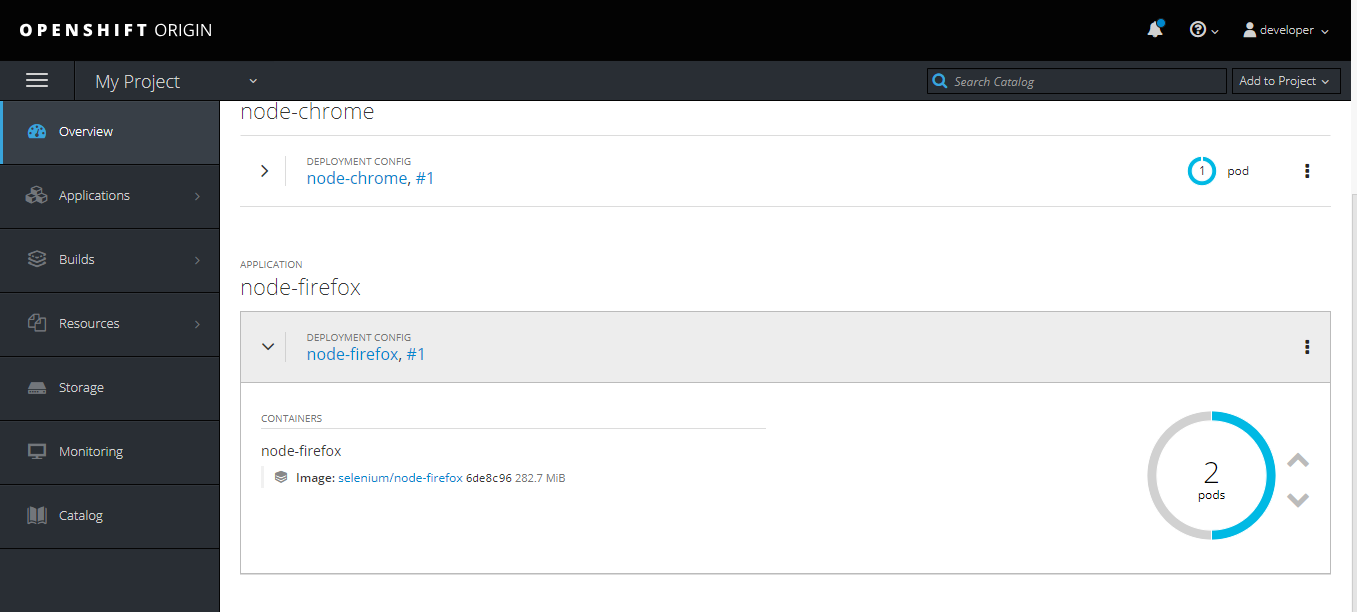
* Now you can click on the console link, which shows you all the chrome and firefox nodes in the selenium hub.



* So this is how you can deploy selenium hub, chrome node and firefox node in the OpenShift cluster.
* One more useful advantage of OpenShift web UI. When you click on the “Overview”.
* It will give you all the pods which are currently running in your cluster.
* Currently, it will look like this.



* So as you can see that there are selenium hub, chrome and firefox pods are currently running in the OpenShift cluster.
* If you want to scale up your chrome or firefox pods then you can easily do that from web UI.
* As shown below.



* As shown in the firefox pod. When you click on the “^” beside the number pods. It will scale up the pods.
* So currently it is scaling up to 2 pods.
* After that you can also check the selenium hub console, in which you can see that there are two firefox nodes available. As shown below.



* So this is how you can scale up your number of pods from web UI.

1. **Deployment using YAMLs**

* So in this case you can use OC commands to execute your YAMLs.
* So currently, I am having my selenium hub, chrome and firefox YAMLs in my virtual machine.
* But in this case you have to crate service as well using YAML. Because OpenShift will not create service as it creates in the web UI deployment.
* So you have to execute selenium hub, chrome, firefox and service YAML from virtual machine using OC commands.
* To execute all this YAMLs execute below commands,
  + - **oc create –f “YAML name”**
* So after executing this command you can check the deployment, list of pods, list of services.
* So to check the list of pods, services and deployment execute below command.
  + - oc get pods
    - oc get svc
    - oc get deployment
* After doing all this you can check the pods, services and deployments in OpenShift web UI as well.

**OpenShift cluster creation using Ansible**

Above I have created OpenShift cluster of one machine only. But in real we have to create cluster of multiple nodes. So in this case we can use OpenShift-Ansible solution.

To use this solution we have followed this github repository :-<https://github.com/openshift/openshift-ansible>**.**

**Prerequisite:-**

Make sure that the all the machines are Linux vms with the OS as fedora, CentOS or RHEL.

Install the following softwares in the controller machine.

Ansible

Docker

Controller machine should be able to connect to other machine through SSH.

So first to create OpenShift cluster, we have to execute node scaleup playbook.

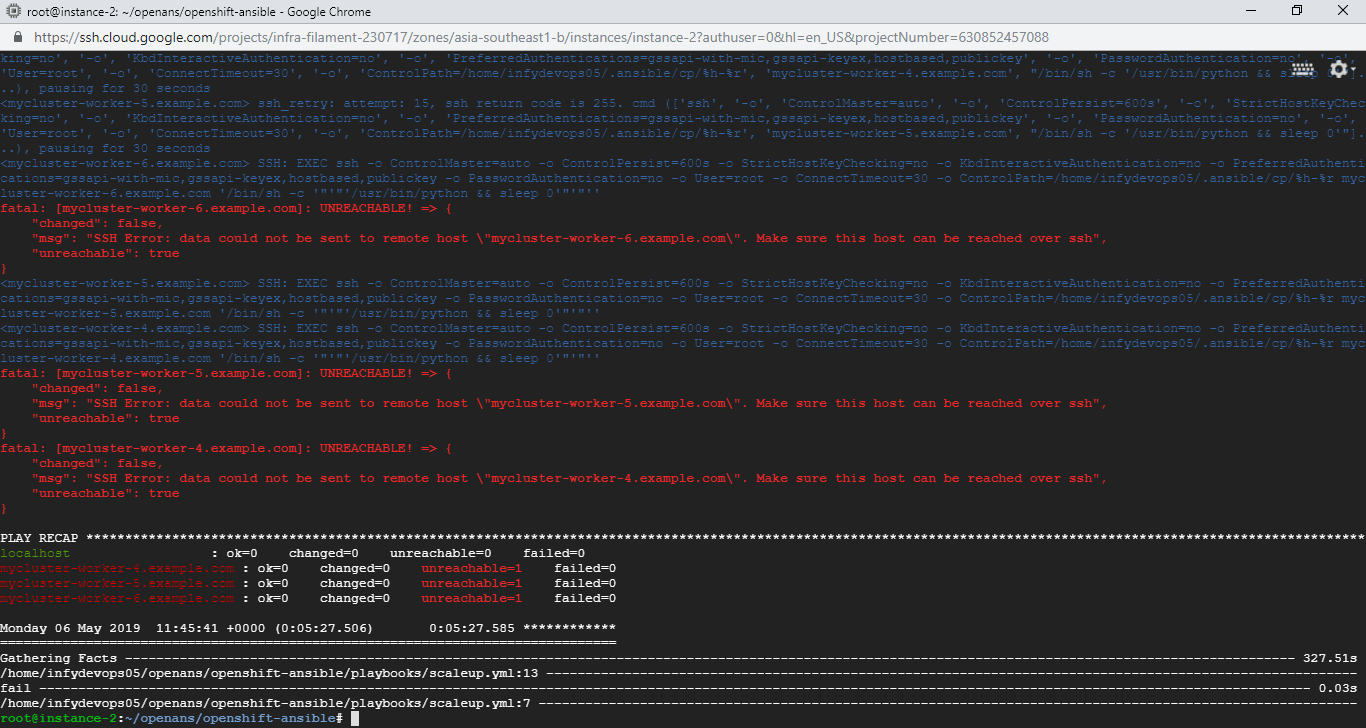
So to execute this playbook you have to execute below command.

ansible-playbook -i inventory/hosts.example playbooks/scaleup.yml

Make sure that while you execute this command you are inside the openshift-ansible folder otherwise it will give you an error that playbook not found.

So now go inside the openshift-ansible folder and execute command.

While you execute this command in result you may face below error.

****

This is because we thought that the pleaybook will scaleup and provision all the GCP machines at the time of execution and all machine will be able to communicate with each other. Which is not working so in this case you have create on virtual machine. As mention above you can create virtual machine of fedora, centOS or RHEL.

In our case we had create centOs virtual machine. After doing that we had tried to create SSH key and tried to ssh into centOS virtual machine from Ubuntu virtual machine.

On first you may face the error of this is not working, you may face permission denied error.

Now to ssh into centOS machine, you have to do some changes in the SSHD\_CONFIG file which in the etc folder.

In the sshd\_config file, find PERMIT\_ROOT\_LOGIN parameter. Make sure that the value of this parameter is YES in the file.

If it is NO then make it YES and then tried to SSH into centOS machie by using “ssh ‘centOS vm plubilc IP’”. Now you will be able to ssh into this vm.

So after doing that we have to execute the playbook again. But this time you may face the error of CRI-TOOL package not available. So to overcome that add the repo where CRI-TOOL package is available. Run the following command.

cat <<EOF > /etc/yum.repos.d/kubernetes.repo

> [kubernetes]

> name=Kubernetes

> baseurl=https://packages.cloud.google.com/yum/repos/kubernetes-el7-x86\_64

> enabled=1

> gpgcheck=1

> repo\_gpgcheck=1

> gpgkey=https://packages.cloud.google.com/yum/doc/yum-key.gpg

> https://packages.cloud.google.com/yum/doc/rpm-package-key.gpg

> EOF

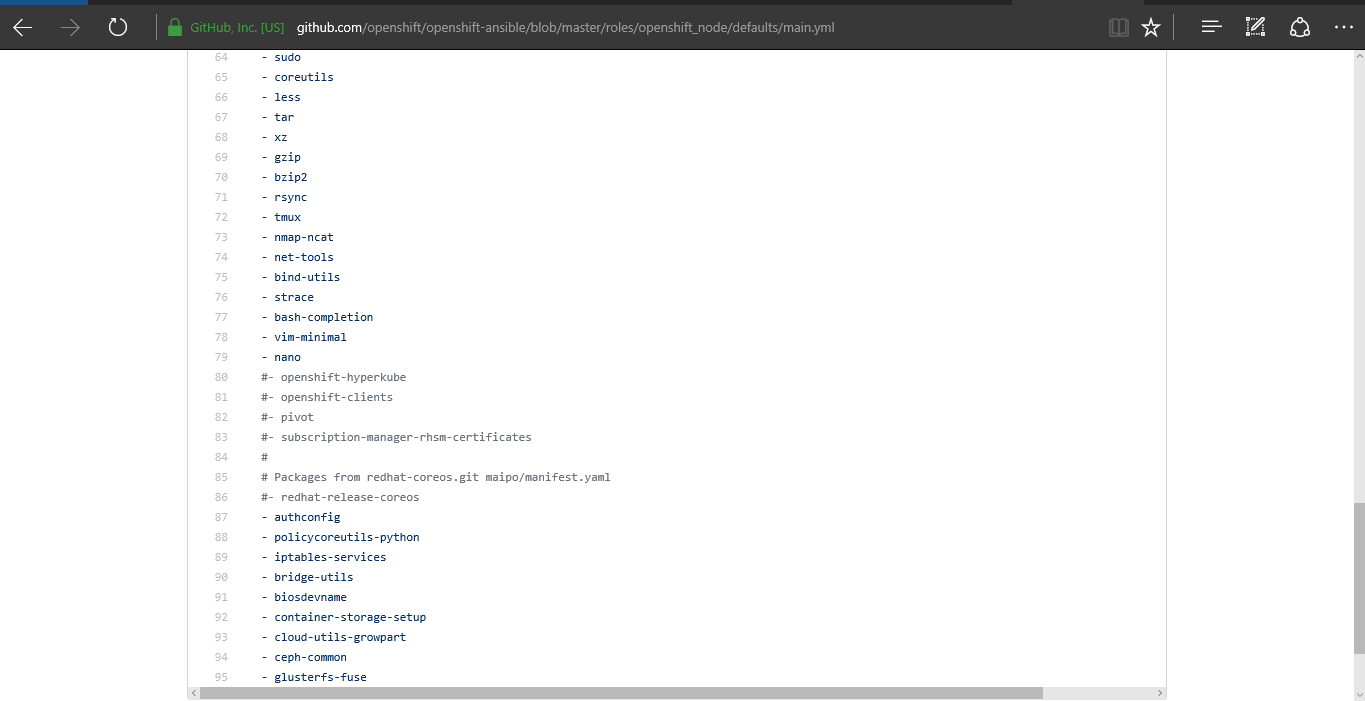
After doing above step again run the playbook which we had executed earlier.

So while you execute playbook in this case you may face the error of package conflict.

To overcome this error, please make sure that go inside the below location.

[openshift-ansible](https://github.com/openshift/openshift-ansible)/[roles](https://github.com/openshift/openshift-ansible/tree/master/roles)/[openshift\_node](https://github.com/openshift/openshift-ansible/tree/master/roles/openshift_node)/[defaults](https://github.com/openshift/openshift-ansible/tree/master/roles/openshift_node/defaults)/**main.yml**

**Then as per the error you are facing for specific package, make sure that you are commenting out the package name in the main.yaml which looks like this.**



For example, we got conflict error of gce-storage and cloud-utils-growpart, so commented that package. Now, run the anisble playbook again.