**ContinuousEndToEndTestingAnsible**

**Table of Contents**

[Pre-requisites](#_Hlk528064603)2

[**Installing the Required software’s**](#_Hlk528064766)2

[Ansible](#_Hlk528064904) 2

[Terraform](#_Hlk528064964)2

[Kubectl](#_Hlk528065001)3

[Docker](#_Hlk528065039)3

[Service Account creation](#_Hlk528065056)3

[**Terraform files for cluster creation**](#_Hlk528065129)9

[Connections.tf](#_Hlk528065179)9

[Kubernetes.tf.](#_Hlk528065343)9

[Variables.tf](#_Hlk528065389)9

[terraform.tfvars](#_Hlk528065429)10

[**Zalenium files**](#_Hlk528065693)10

[plumbing.yaml](#_Hlk528077770)10

[pv.yaml](#_Hlk528077813)11

[zalenium.yaml](#_Hlk528077852)12

[Zalenium-service.yaml.](#_Hlk528077900)13

[Static IP creation.](#_Hlk528077960)14

[**File beat files**](#_Hlk528078386)14

[**Files for testcase deployment on cluster**](#_Hlk528145731)18

[**Dockerfile for image creation**](#_Hlk528145773)18

[nfs-pv.yaml](#_Hlk528145815)19

[nfs-pvc.yaml](#_Hlk528155109)19

[php-test.yaml](#_Hlk528155141)20

[**Ansible-playbook**[**Roles**](#_Hlk529181025)](#_Hlk528155248)20

[Terra](#_Hlk529179624)20

[Elk](#_Hlk529179677)21

[Zalenium](#_Hlk529179742)21

[Testcases](#_Hlk529180138)21

[Destroy](#_Hlk529180869)22

[**Final Ansible-playbook**](#_Hlk529181532)22

1. **Pre-requisites**: -
   1. Kubernetes
   2. Docker
   3. Ansible
   4. Terraform
   5. Elk
   6. Jenkins
   7. Basic of Google cloud
2. **Installing the Required software’s: -**  
   To start with creation of cluster on the fly In GCP, you first need to install Terraform, Ansible, Kubectl, gcloud and Docker in the Cloud VM (In this case VM is Ubuntu 18-lts). Follow the steps below to install this softwares.  
     
    1. Installing Ansible use the commands

|  |
| --- |
| $ sudo apt-get update $ sudo apt-add-repository ppa:ansible/ansible $ sudo apt-get install ansible $ansible –version |

2. Installing Terraform.  
First download terraform zip by using **wget**. Unzip the downloaded file by giving command **unzip terraform\_0.11.8\_linux\_amd64.zip.** Nowmove to directory where the executable file is stored and run the command **sudo mv terraform /usr/local/bin/.** Once this is done check you Terraform installation by giving the command **Terraform –version.** As done below.

$ wget <https://releases.hashicorp.com/terraform/0.11.8/terraform_0.11.8_linux_amd64.zip>  
$ unzip terraform\_0.11.8\_linux\_amd64.zip   
$ sudo mv terraform /usr/local/bin/  
$ terraform --version

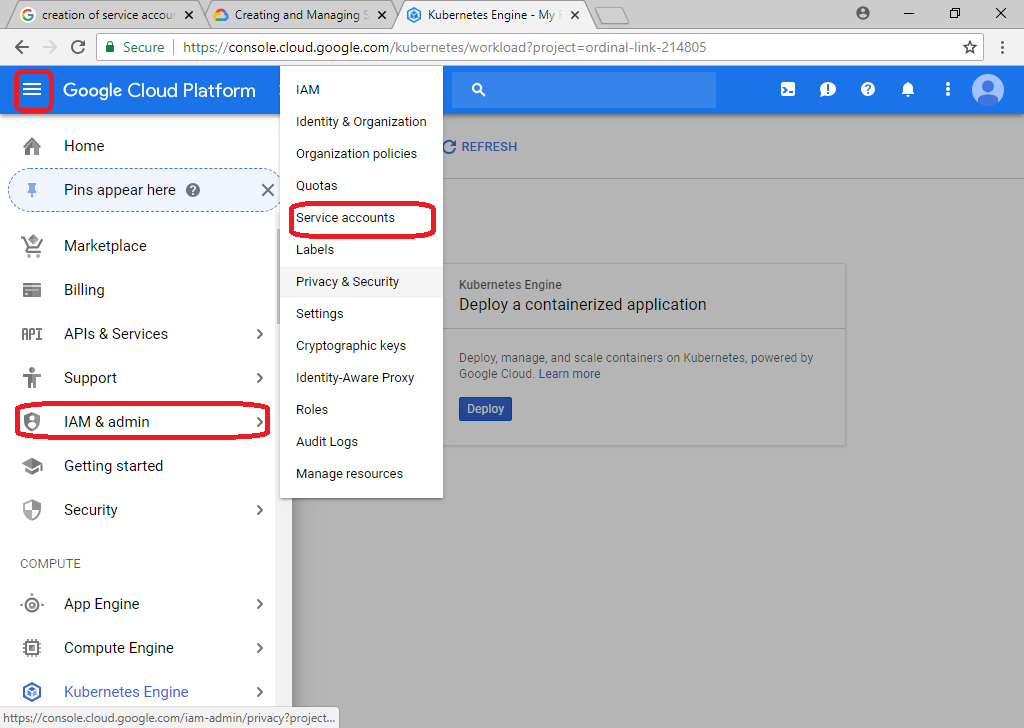
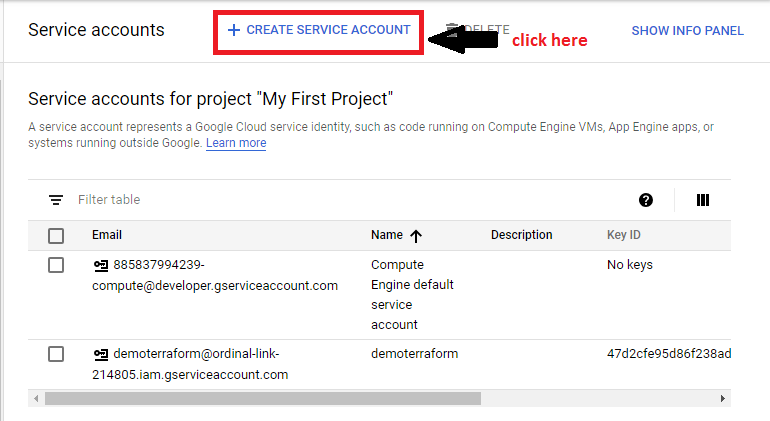
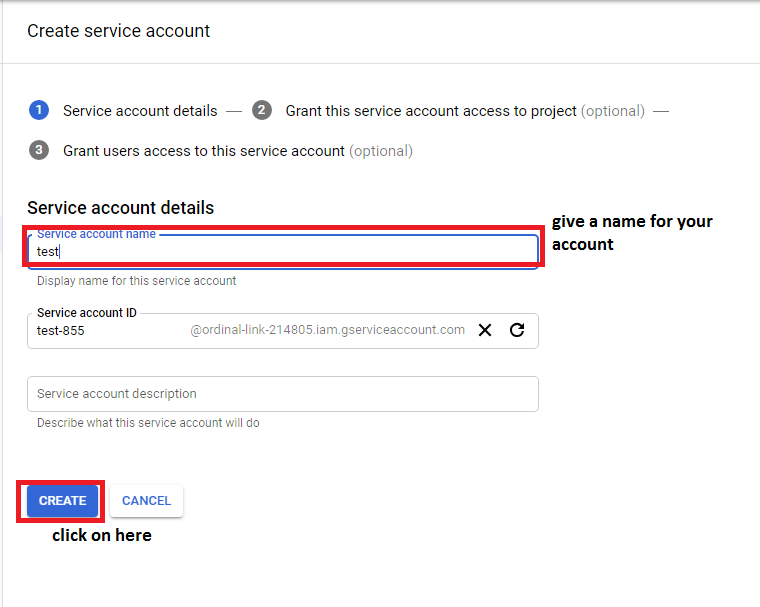
3. Installing Kubectl  
first set your gcloud and install Kubectl using gcloud as given below.  
  
  
  
4. Installing Docker  
Install docker using the following commands.

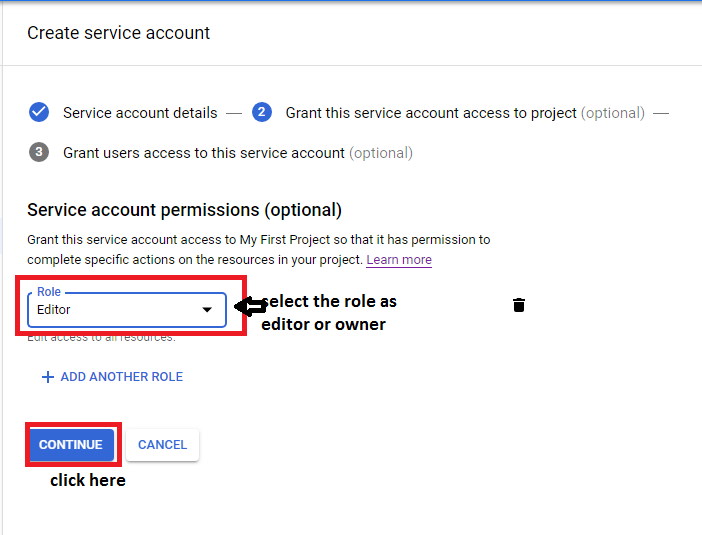
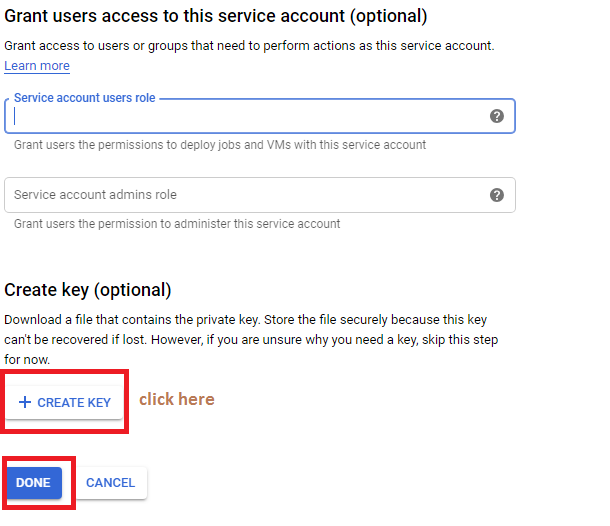
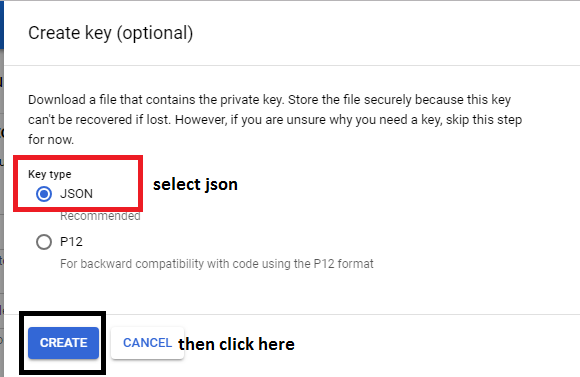
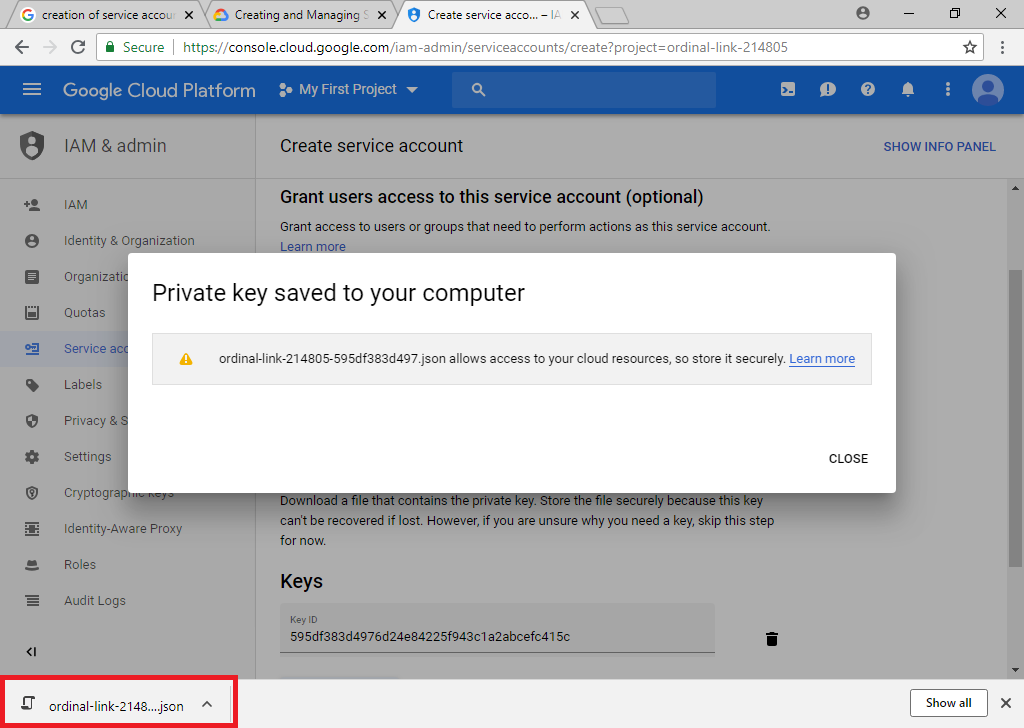
$ curl -LO https://storage.googleapis.com/kubernetes-release/release/v1.11.0/bin/linux/amd64/kubectl  
$ cp kubectl /usr/bin  
$ cd /usr/bin  
$ chmod +x kubectl  
$ Kubectl version

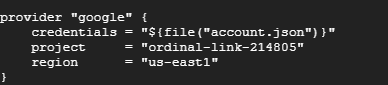
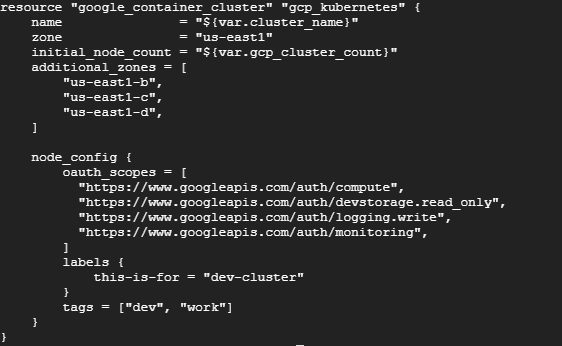
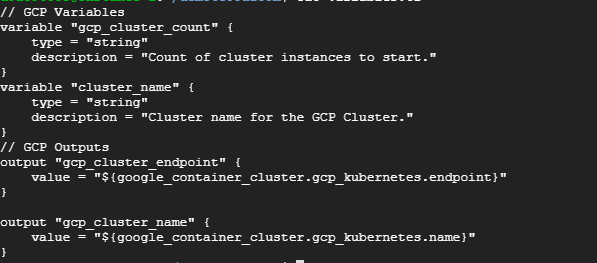
$ sudo apt-get update  
$ sudo apt-get docker  
$ sudo snap install docker  
$ sudo docker version

5. Service Account creation  
Also when you working In GCP for Cluster creation, you need to create a service account. Follow the steps below.

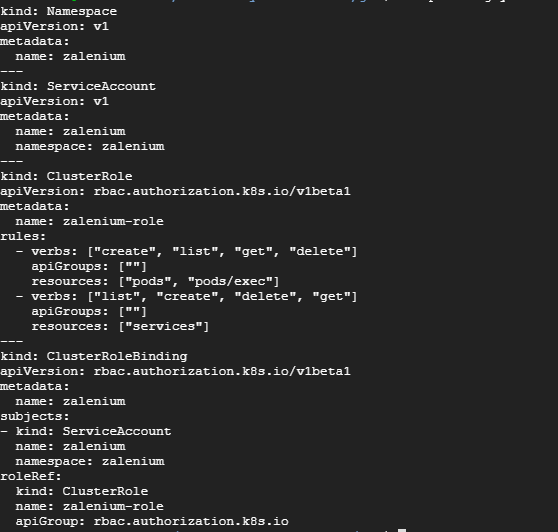
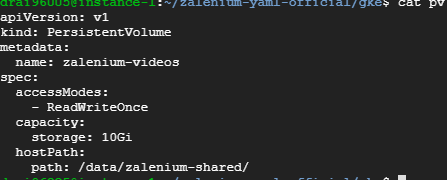
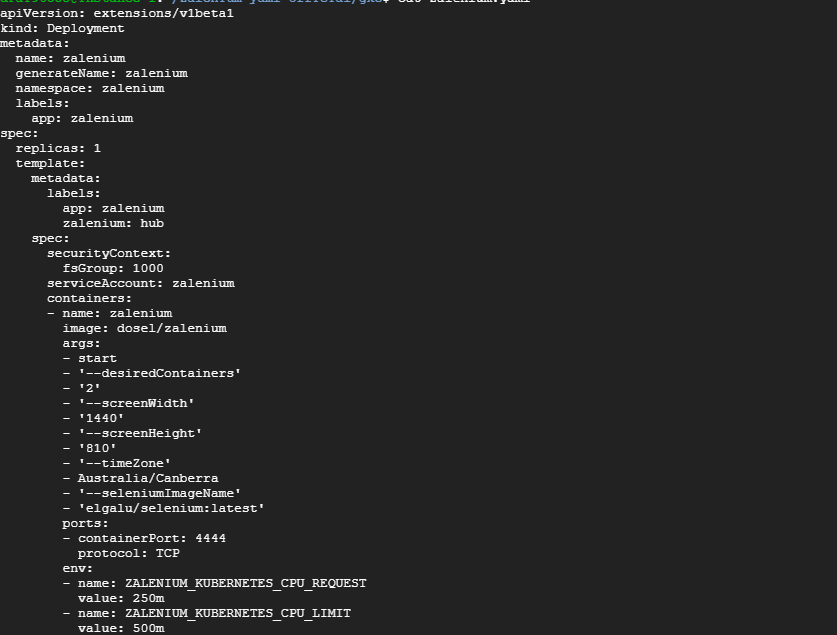
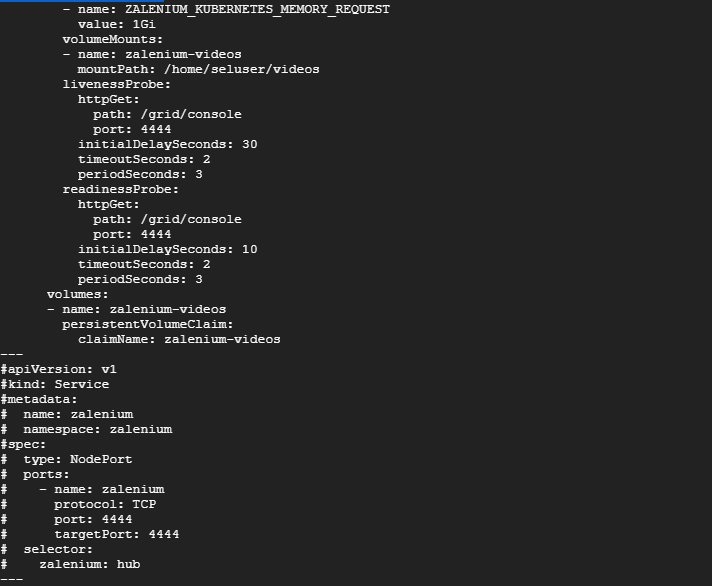
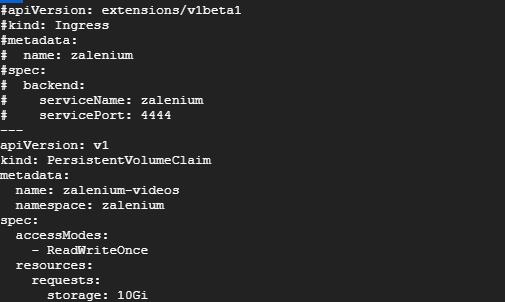
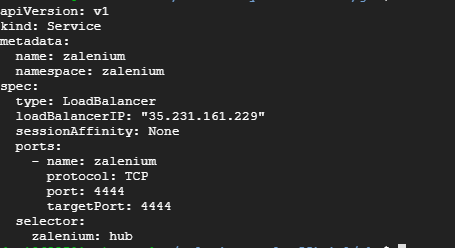
Fist on your console page click on left hand side menu, there you select IAM in that click on the service account as shown below.

  
   
After clicking on create service account do the following.  


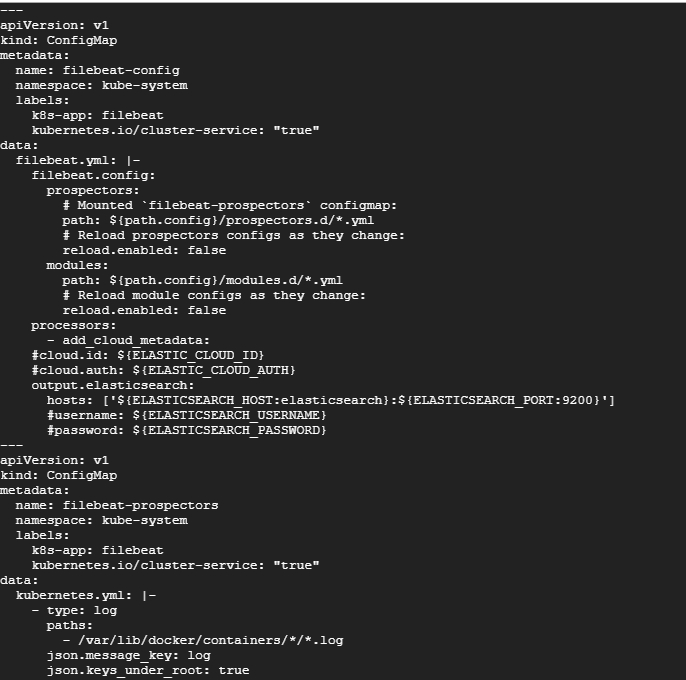
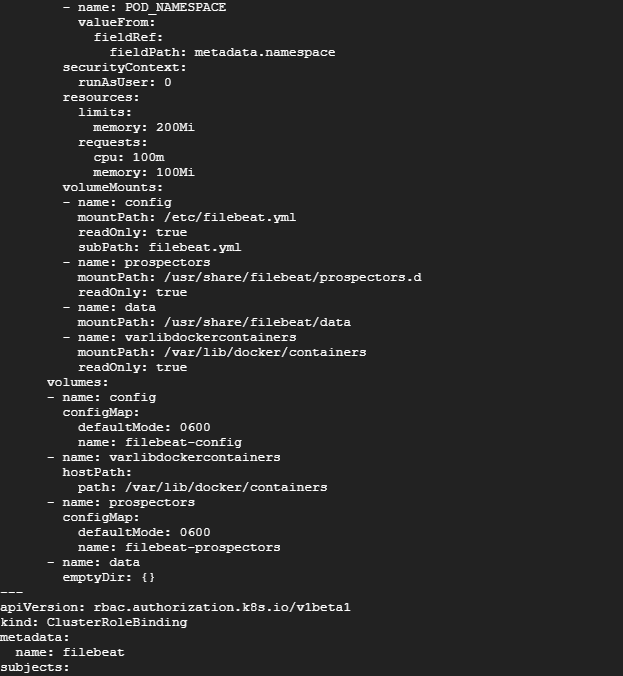
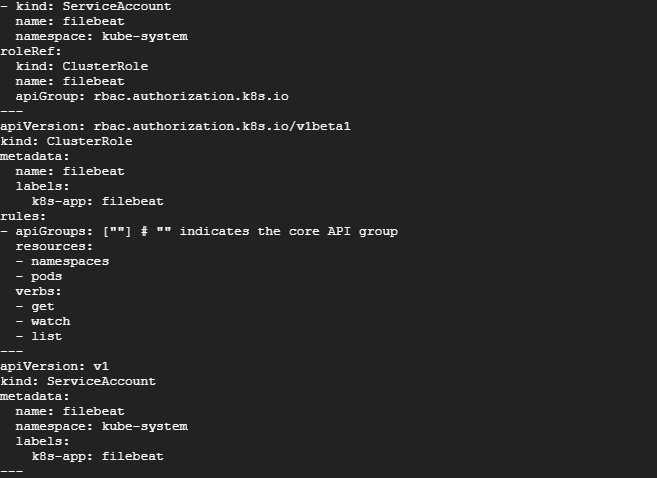
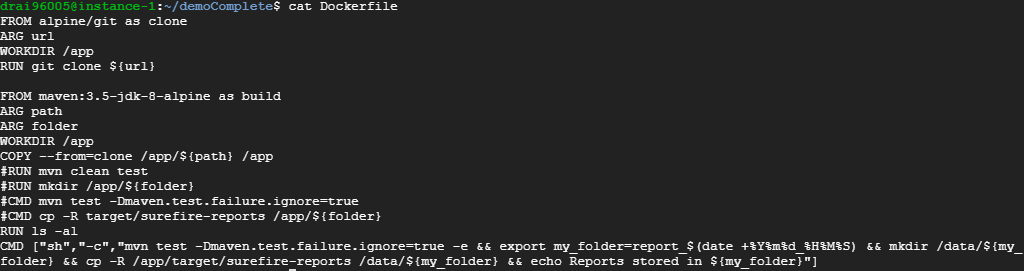
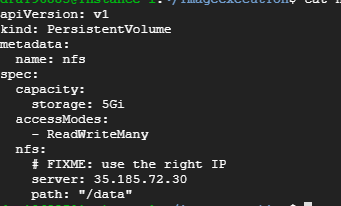
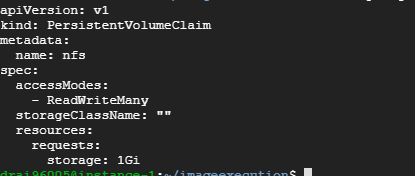
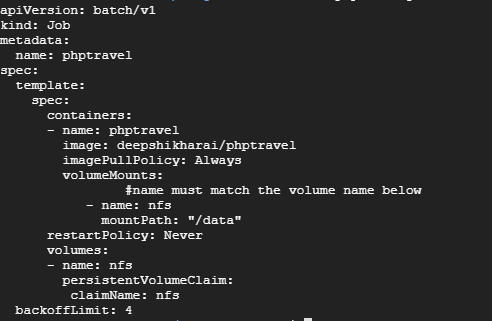
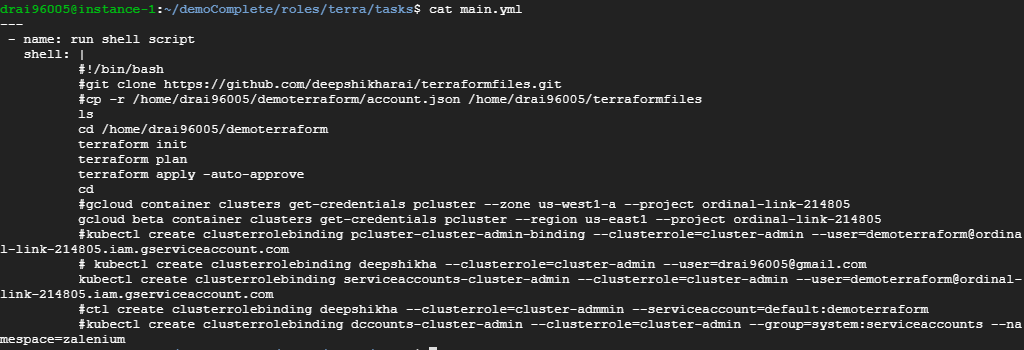
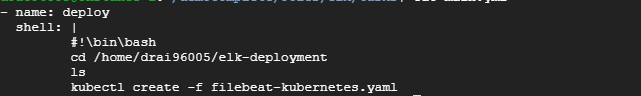
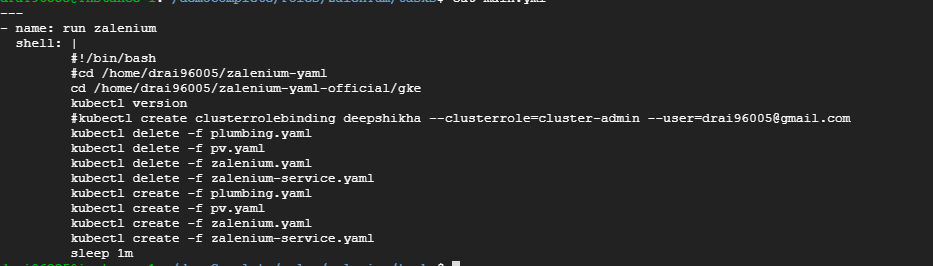
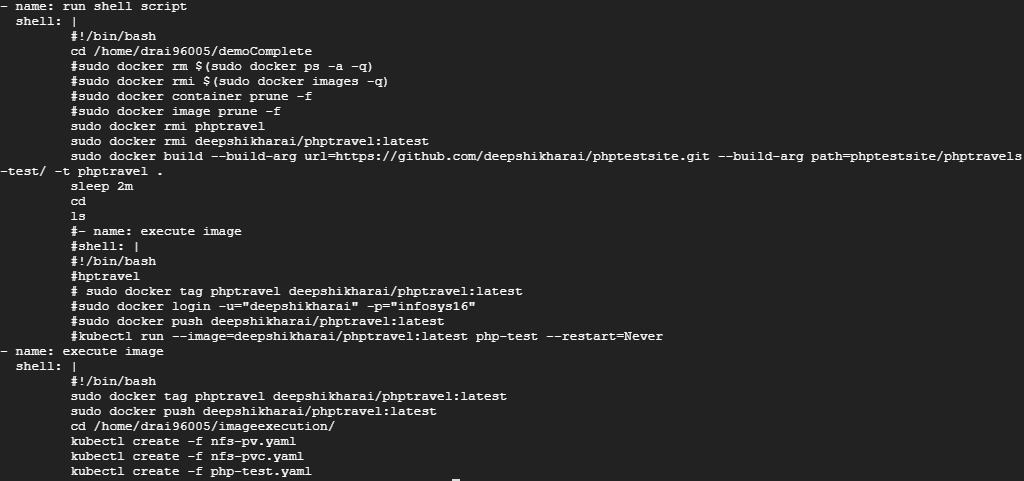
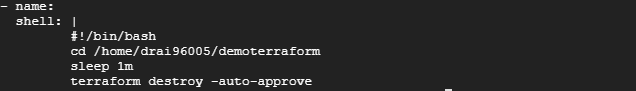
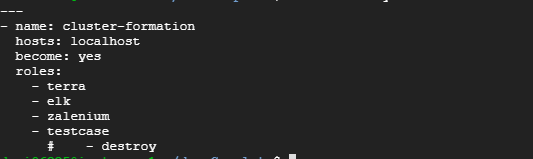
1. **Terraform files for cluster creation: -**  
   Once the installation of softwares are done. I have created my .tf files which will be used to create cluster. I have a folder (demoterraform) in which I have stored the .tf files. Also I have my account.json stored here as it will be required while creating the cluster, so the structure looks as seen below.  
   C:\Users\deepshikha.rai\Desktop\screenshots\terra1.PNG  
     
   1. Connections.tf  
     
   Here I have selected google as my provider as I am working in Google Cloud, you can user AWS and AZURE as your preference. In credentials I have provided my account.json which is the credentials file for my service account of GCP, the project is the project id of my Project in GCP and then the last is the region of my cluster to be made.  
     
   2. Kubernetes.tf.  
     
   In Kubernetes.tf we have specified the zone in which can make our cluster, and allowed the APIs of cloud that can be used further so we don’t get any error of blocking the API.  
     
   3. Variables.tf.  
     
   In variables.tf we define all the variables that we will using in different .tf files. For example cluster\_name etc.  
     
   4. terraform.tfvars.  
   In this we specify the name of the cluster and count i.e no of cluster we want to make. You can also refer the link for this files <https://github.com/deepshikharai/terraformfiles> .  
     
     
     
     
   Once the files are created you can use the above command and create cluster in Google cloud.

$ terraform init  
$ terraform plan   
$ terraform apply

1. **Zalenium files: -**for better understanding follow the document **Zalenium\_deployment\_in\_kubernetes\_cluster\_of\_GCP.docx.**We have to create 3 YAML files for zalenium deployment: plumbing.yaml, pv.yaml and zalenium.yaml. Create YAML files by following the zalenium official gihub repository <https://github.com/zalando/zalenium/blob/master/docs/k8s/gke/>   
     
   1. plumbing.yaml  
     
     
   In this YAML file, it will create a different namespace “zalenium” where our pods and services will be deployed instead of default one. Then, it will create service account to more access. Also, cluster roles are defined so that it can create and destroy the containers on demand with the help of service account.  
     
   2. pv.yaml  
     
     
     
   This YAML file will create a permanent volume to store the test case videos captured by zalenium in the specified path.  
     
   3. zalenium.yaml  
     
     
     
   This YAML file creates deploys the zalenium container with the specified parameters and created a service of it. Also, it defines the volume mounting of the captures logs and videos which will be linked with the persistent volume defined in pv.yaml file.  
     
     
   4. Zalenium-service.yaml.  
     
   In this Yaml file we define our load balancing for which we use a static IP.  
     
   4. Static IP creation.  
   Use the below commands to create a static IP in Google cloud VM.  
     
     
   In the above command helloweb-ip is the name for the static IP, also select the same region as that of cluster.

$ gcloud compute addresses create helloweb-ip --region us-central1s

$ gcloud compute addresses describe helloweb-ip --region us-central1

1. **File beat files: -**To connect the project to ELK we also need to deploy filebeat on the cluster. For   
   that we create a filebeat-kubernetes.yaml as below.  
     
     
     
   
2. **Files for testcase deployment on cluster**as we are deplying our test cases on cluster, we first need to create the image of our test cases using Dockerfile and then create three different Yaml files and deploy it on the cluster. Also we have created a nfs-server to store our test reports in that for that refer NFS server document.  
     
   1. **Dockerfile for image creation:-**  
   For dockerfile refer Dockerfinal.docx  
   ****  
   this dockerfile will create the image after execution .  
     
   2. nfs-pv.yaml  
     
   This is the yaml file that contains the information like where to store the reporta of the test cases.  
     
   3. nfs-pvc.yaml  
     
     
     
   4. php-test.yaml  
     
   This is yaml file that contains the details of the image to be deployed on the cluster.
3. **Roles in Ansible-playbook for end-to-end testing.**To understand ansible in better way refer document Ansible.docx. the structure of my playbook is as below: -  
   **C:\Users\deepshikha.rai\Desktop\screenshots\p1.PNG**  
   In this ansible playbook we have role for each tasks to be performed.  
     
   1. Terra: -  
   In this roles we are going inside the terraform files folder and executing the given commands to create the cluster also we are giving the command to esstablish connection with the cluster.so the main.yml looks as below  
     
   2.Elk  
     
   In this roles we are deploying the filebeat-kubernetes.yaml on the cluster.  
   3. Zalenium  
     
     
   this the yaml file to deploy Zalenium on cluster.  
     
   4. Testcases  
     
     
   This is the yaml file to create the image of the test cases and deploy the image on the cluster.  
     
   5. Destroy  
     
   this the yaml file to destroy the cluster.
4. **Final Ansible-playbook  
   **this is the ansible-playbook we will be executing where we have define all the roles to be executed**.**