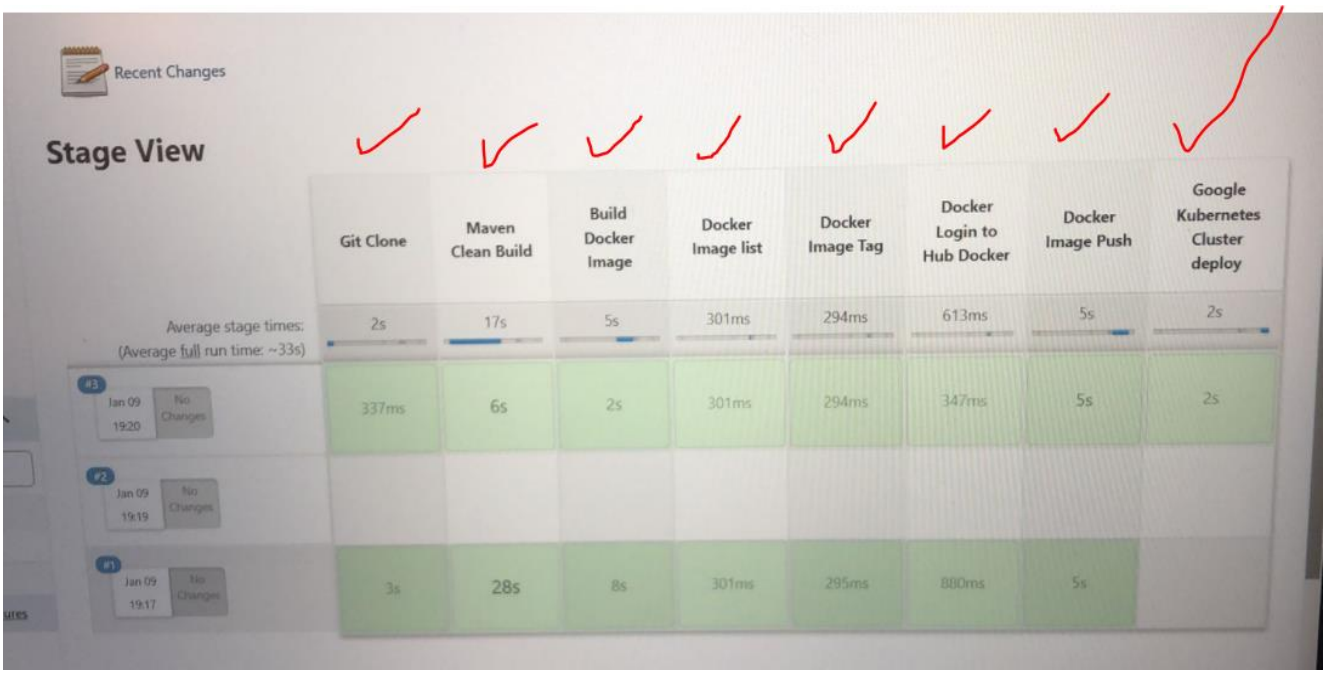
Project Number 1

This is what we did on Saturday. For more information, check on this link: https://github.com/joshking1/jenkins-buffet-CI-CD-Pipeline-PROJECT.git

This is what we accomplished

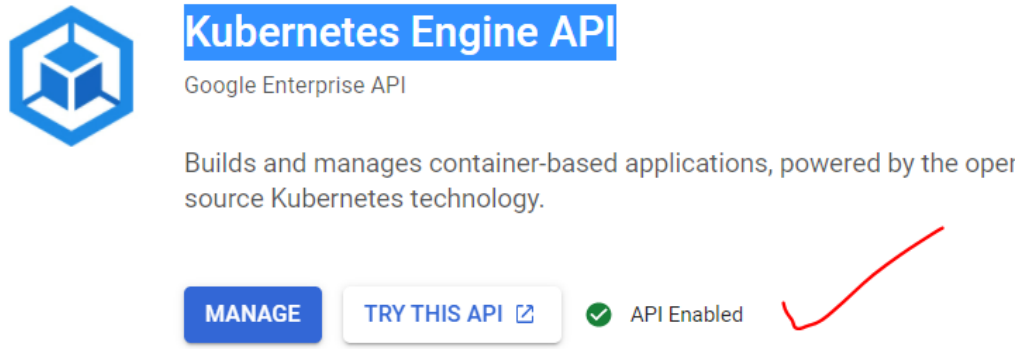


Project Number 2

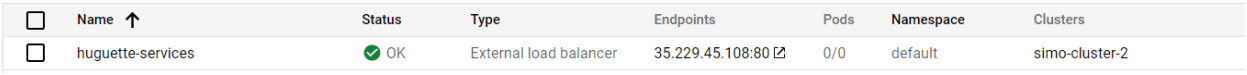
GKC – Google Kubernetes Cluster Project

1. Enable Kubernetes Engine API

   It should look like this



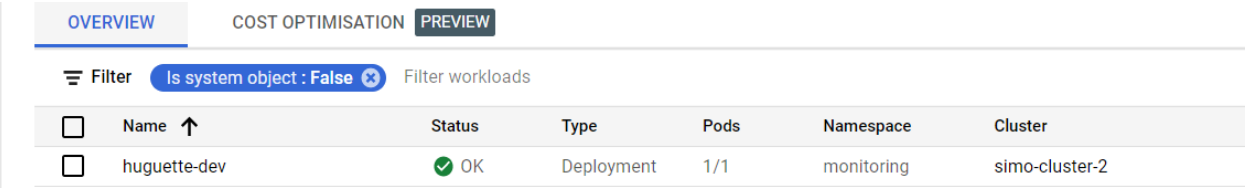
2. Ensure your service and Ingress are provisioned as Load balancer with the name of the APP



Name	Status	Type	Endpoints	Pods	Namespace	Clusters
huguette-services	OK	External load balancer	35.229.45.108:80	0/0	default	simo-cluster-2

You should only have one load balancer as shown above

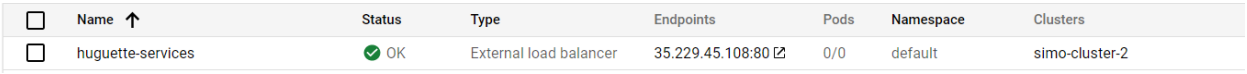
3. Ensure your deployment is provisioned



Name	Status	Type	Pods	Namespace	Cluster
huguette-dev	OK	Deployment	1/1	monitoring	simo-cluster-2

You should have one deployment with the name of the APP

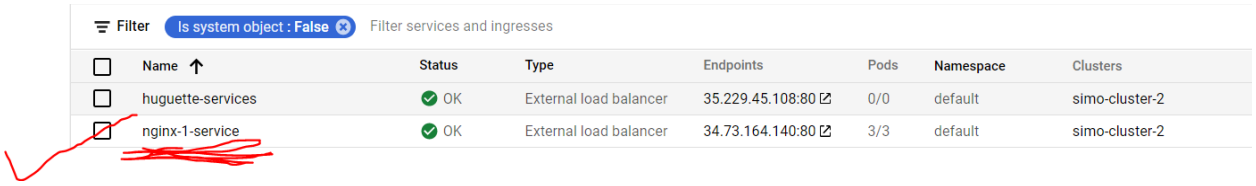
4. Click on the endpoints of the load balancer - it will not work



Name	Status	Type	Endpoints	Pods	Namespace	Clusters
huguette-services	OK	External load balancer	35.229.45.108:80	0/0	default	simo-cluster-2

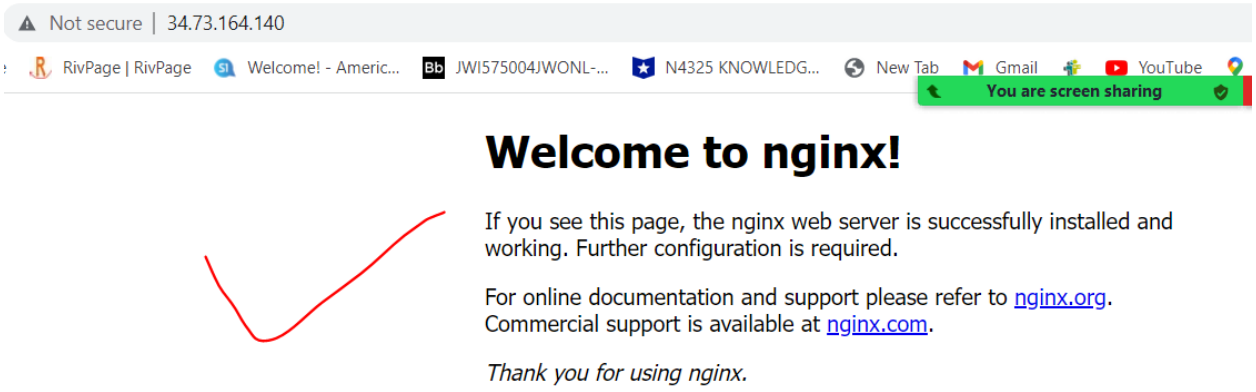
5. Enable the load balancer and ingress rule provisioning for NGINX image to work.

Click on the endpoints, and a load balancer with the name of the image should appear as follow

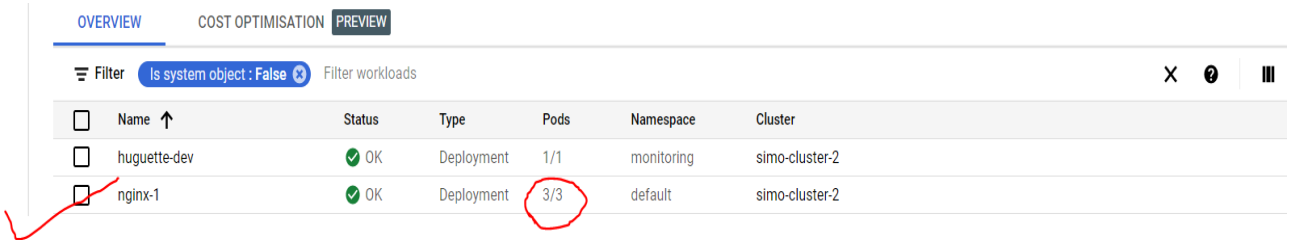


nginx-1-service load balancer will be provisioned externally

Click on the endpoints of nginx-1-service, and the load balancer should display the following

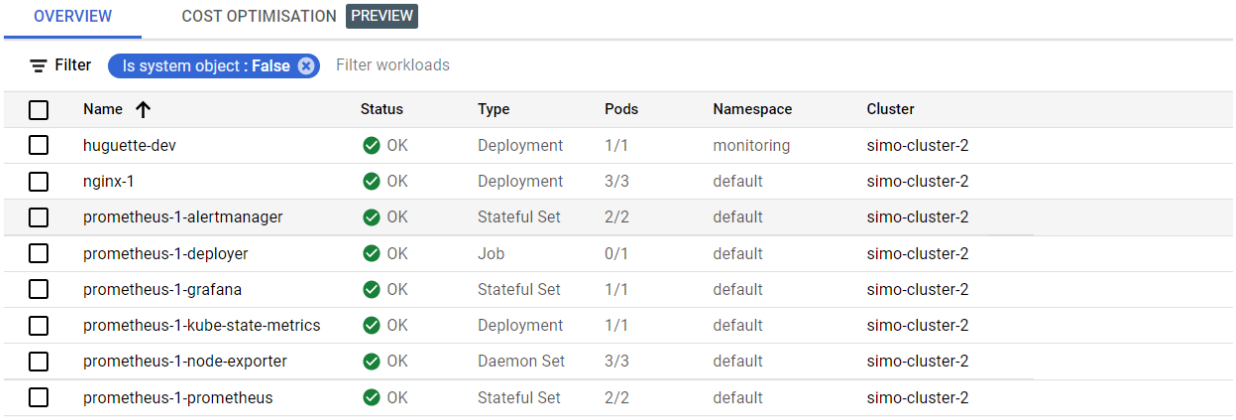


6. Deploy the nginx image to become a container of three pods as shown below

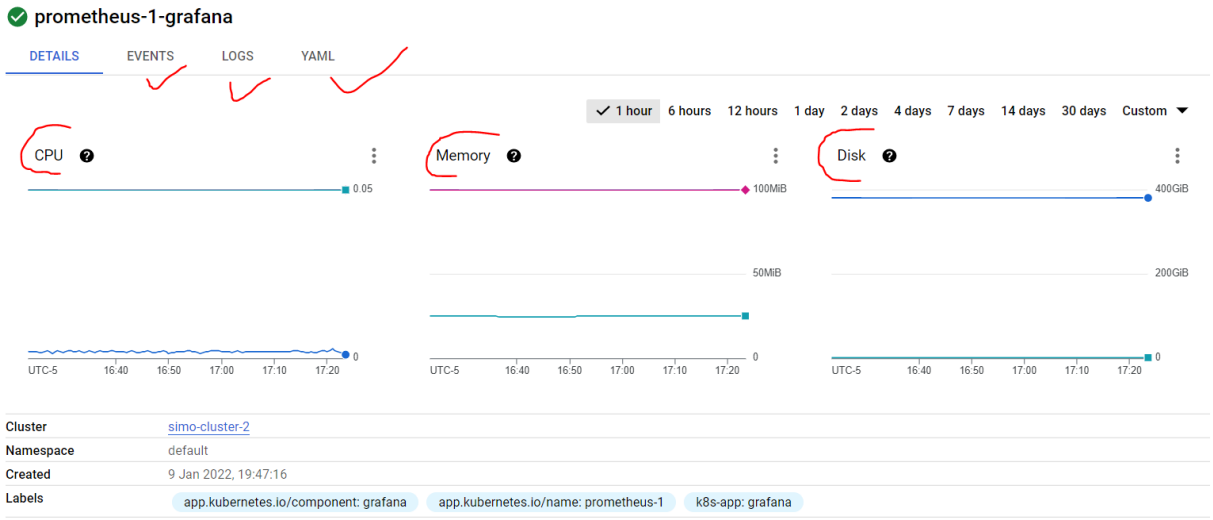


Remember nginx stay as an image until it is deployed to become a nginx container

7. Go to application and add Prometheus + Grafana to monitor your cluster and you should have something like this



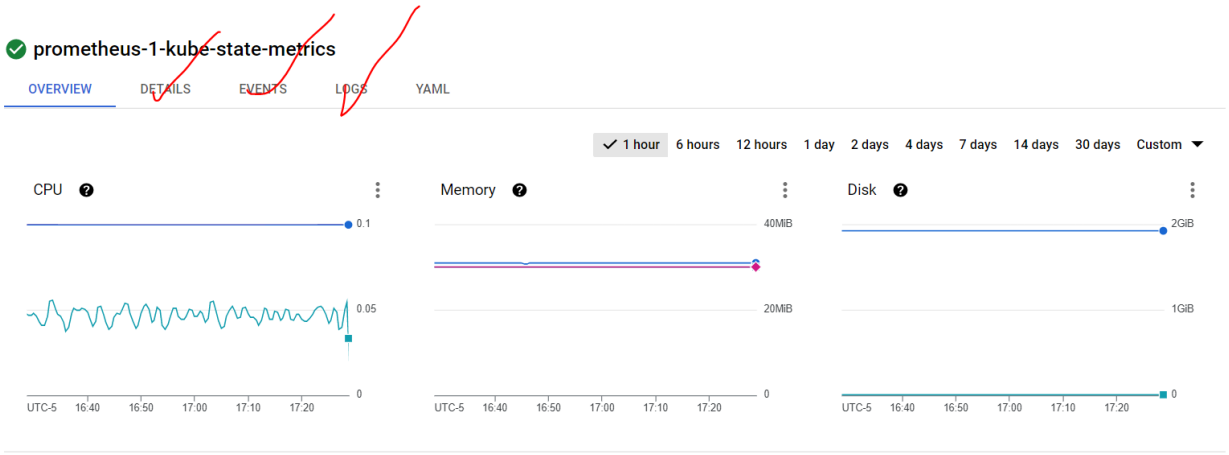
Click on Prometheus + Grafana and should display the following



Check the logs of your cluster and events

8. Go to service and Ingress – (traffic protocol) TCP or UDP

   Click on the Prometheus + Grafana and you should have the following



9. Check the config map and the secret

You should have something close to this

Name	Type	Namespace	Cluster
default-token-d8m6g	Secret	monitoring	simo-cluster-2
default-token-pfcrr	Secret	kube-node-lease	simo-cluster-2
kube-root-ca.crt	Config Map	default	simo-cluster-2
kube-root-ca.crt	Config Map	kube-node-lease	simo-cluster-2
kube-root-ca.crt	Config Map	monitoring	simo-cluster-2
kube-root-ca.crt	Config Map	kube-public	simo-cluster-2
prometheus-1-alertmanager-config	Config Map	default	simo-cluster-2
prometheus-1-alertmanagerserviceaccount-4054-token-bjgxf	Secret	default	simo-cluster-2
prometheus-1-dashboards	Config Map	default	simo-cluster-2
prometheus-1-deployer-config	Config Map	default	simo-cluster-2
prometheus-1-grafana	Secret	default	simo-cluster-2
prometheus-1-grafana-dashboardproviders	Config Map	default	simo-cluster-2
prometheus-1-grafana-datasources	Config Map	default	simo-cluster-2
prometheus-1-grafana-ini	Config Map	default	simo-cluster-2
prometheus-1-grafanaserviceaccount-2229-token-hfm58	Secret	default	simo-cluster-2
prometheus-1-kubestatemetricsserviceaccount-608f-token-85xgp	Secret	default	simo-cluster-2

An example of confirmed details

```
"groups":
- "name": "kubernetes-absent"
  "rules":
  - "alert": "AlertmanagerDown"
```

```
    "annotations":
      "message": "Alertmanager has disappeared from Prometheus target
discovery."
      "runbook_url": "https://github.com/kubernetes-monitoring/kubernetes-
mixin/tree/master/runbook.md#alert-name-alertmanagerdown"
    "expr": |
      absent(up{job="alertmanager"} == 1)
    "for": "15m"
    "labels":
      "severity": "critical"

apiVersion: 1
datasources:
- access: proxy
  isDefault: true
  name: prometheus
  type: prometheus
  url: http://prometheus-1-prometheus:9090
  version: 1
```

The cluster will access the links and to do so, secret must be available