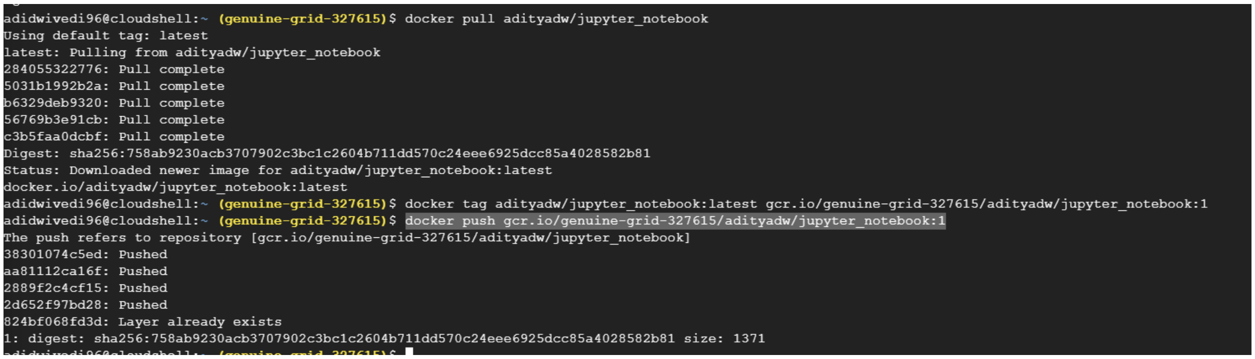
Deploying on Kubernetes-

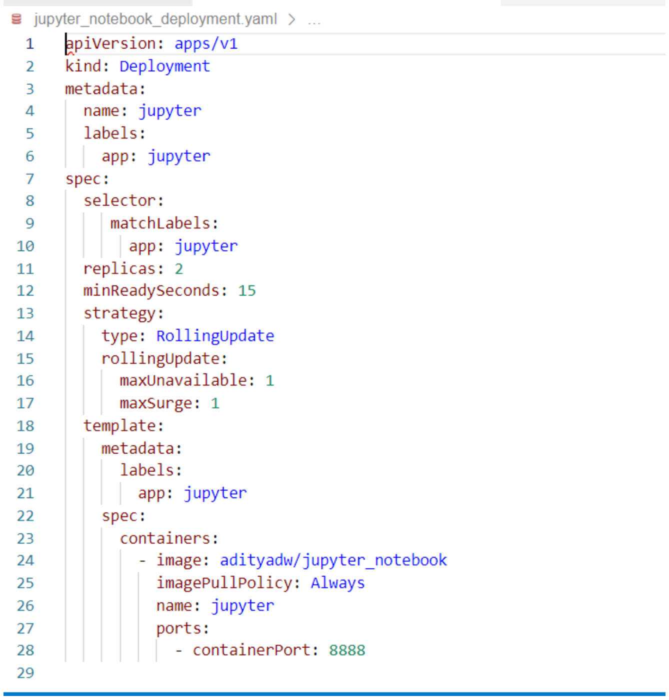
Jupyter Notebook

1. Create a Kubernetes cluster on GCP - gcloud container clusters create --machine-type n1-standard-2 --num-nodes 2 --zone us-central1-a --cluster-version latest adityadwkubernetescluster
2. docker pull adityadw/jupyter_notebook
3. docker tag adityadw/jupyter_notebook:latest gcr.io/genuine-grid-327615/adityadw/jupyter_notebook:1
4. docker push gcr.io/genuine-grid-327615/adityadw/jupyter_notebook:1



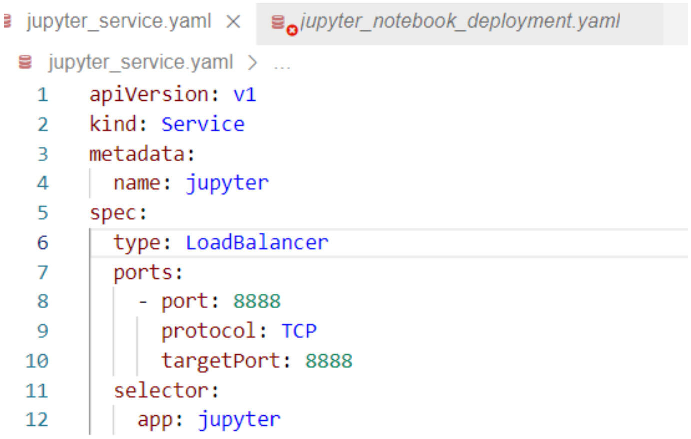
```
adidwivedi96@cloudshell:~ (genuine-grid-327615)$ docker pull adityadw/jupyter_notebook
Using default tag: latest
latest: Pulling from adityadw/jupyter_notebook
284055322776: Pull complete
5031b1992b2a: Pull complete
b6329deb9320: Pull complete
56769b3e91cb: Pull complete
c3b5faa0dcbf: Pull complete
Digest: sha256:758ab9230acb3707902c3bc1c2604b711dd570c24eee6925dcc85a4028582b81
Status: Downloaded newer image for adityadw/jupyter_notebook:latest
docker.io/adityadw/jupyter_notebook:latest
adidwivedi96@cloudshell:~ (genuine-grid-327615)$ docker tag adityadw/jupyter_notebook:latest gcr.io/genuine-grid-327615/adityadw/jupyter_notebook:1
adidwivedi96@cloudshell:~ (genuine-grid-327615)$ docker push gcr.io/genuine-grid-327615/adityadw/jupyter_notebook:1
The push refers to repository [gcr.io/genuine-grid-327615/adityadw/jupyter_notebook]
38301074c5ed: Pushed
aa81112ca16f: Pushed
2889f2c4cf15: Pushed
2d652f97bd28: Pushed
824bf068fd3d: Layer already exists
1: digest: sha256:758ab9230acb3707902c3bc1c2604b711dd570c24eee6925dcc85a4028582b81 size: 1371
adidwivedi96@cloudshell:~ (genuine-grid-327615)$
```

5. Create deployment file for jupyter notebook as follows, here the internal port on which service is running is 8888 -



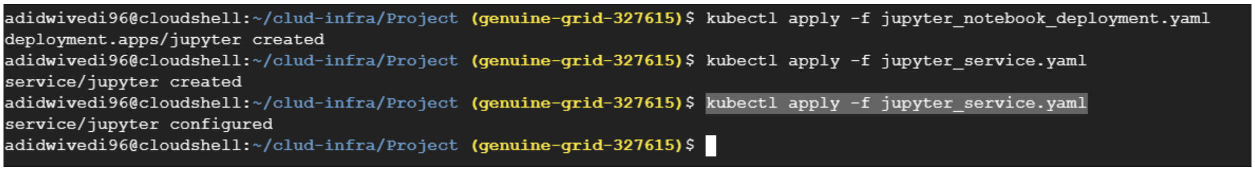
```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: jupyter
  labels:
    app: jupyter
spec:
  selector:
     matchLabels:
       app: jupyter
  replicas: 2
  minReadySeconds: 15
  strategy:
    type: RollingUpdate
    rollingUpdate:
      maxUnavailable: 1
      maxSurge: 1
  template:
    metadata:
      labels:
        app: jupyter
    spec:
      containers:
        - image: adityadw/jupyter_notebook
          imagePullPolicy: Always
          name: jupyter
          ports:
            - containerPort: 8888

```

6. kubectl apply -f jupyter_notebook_deployment.yaml
7. Create service file in order to deploy jupyter notebook as follows. Presently it has been set as loadbalancing service so that the testing can be done via browser -



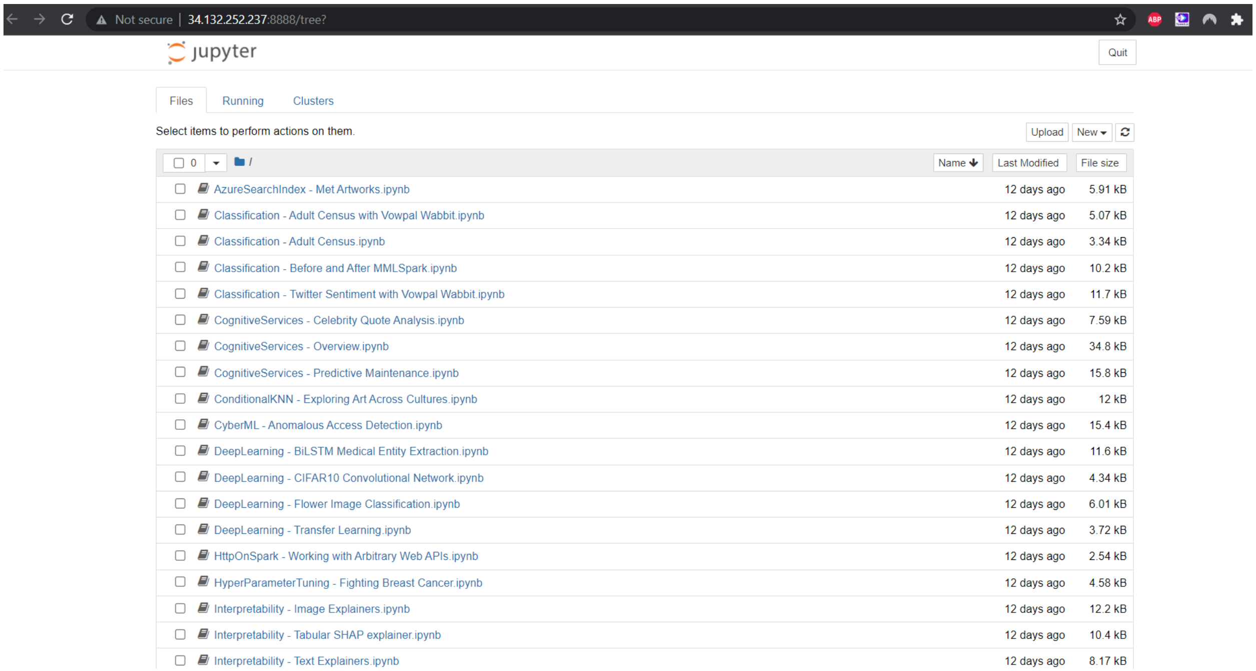
```yaml
apiVersion: v1
kind: Service
metadata:
  name: jupyter
spec:
  type: LoadBalancer
  ports:
    - port: 8888
      protocol: TCP
      targetPort: 8888
  selector:
    app: jupyter
```

8. kubectl apply -f jupyter_service.yaml



```
adidwivedi96@cloudshell:~/clud-infra/Project (genuine-grid-327615)$ kubectl apply -f jupyter_notebook_deployment.yaml
deployment.apps/jupyter created
adidwivedi96@cloudshell:~/clud-infra/Project (genuine-grid-327615)$ kubectl apply -f jupyter_service.yaml
service/jupyter created
adidwivedi96@cloudshell:~/clud-infra/Project (genuine-grid-327615)$ kubectl apply -f jupyter_service.yaml
service/jupyter configured
adidwivedi96@cloudshell:~/clud-infra/Project (genuine-grid-327615)$
```

9. We have the service running on public IP as follows -

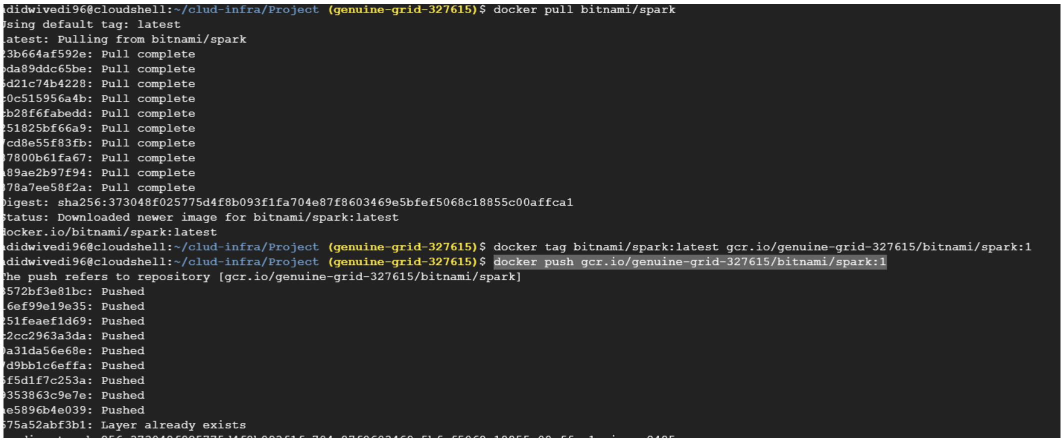


Select items to perform actions on them.

Name	Last Modified	File size
AzureSearchIndex - Met Artworks.ipynb	12 days ago	5.91 kB
Classification - Adult Census with Vowpal Wabbit.ipynb	12 days ago	5.07 kB
Classification - Adult Census.ipynb	12 days ago	3.34 kB
Classification - Before and After MMLSpark.ipynb	12 days ago	10.2 kB
Classification - Twitter Sentiment with Vowpal Wabbit.ipynb	12 days ago	11.7 kB
CognitiveServices - Celebrity Quote Analysis.ipynb	12 days ago	7.59 kB
CognitiveServices - Overview.ipynb	12 days ago	34.8 kB
CognitiveServices - Predictive Maintenance.ipynb	12 days ago	15.8 kB
ConditionalKNN - Exploring Art Across Cultures.ipynb	12 days ago	12 kB
CyberML - Anomalous Access Detection.ipynb	12 days ago	15.4 kB
DeepLearning - BiLSTM Medical Entity Extraction.ipynb	12 days ago	11.6 kB
DeepLearning - CIFAR10 Convolutional Network.ipynb	12 days ago	4.34 kB
DeepLearning - Flower Image Classification.ipynb	12 days ago	6.01 kB
DeepLearning - Transfer Learning.ipynb	12 days ago	3.72 kB
HttpOnSpark - Working with Arbitrary Web APIs.ipynb	12 days ago	2.54 kB
HyperParameterTuning - Fighting Breast Cancer.ipynb	12 days ago	4.58 kB
Interpretability - Image Explainers.ipynb	12 days ago	12.2 kB
Interpretability - Tabular SHAP explainer.ipynb	12 days ago	10.4 kB
Interpretability - Text Explainers.ipynb	12 days ago	8.17 kB

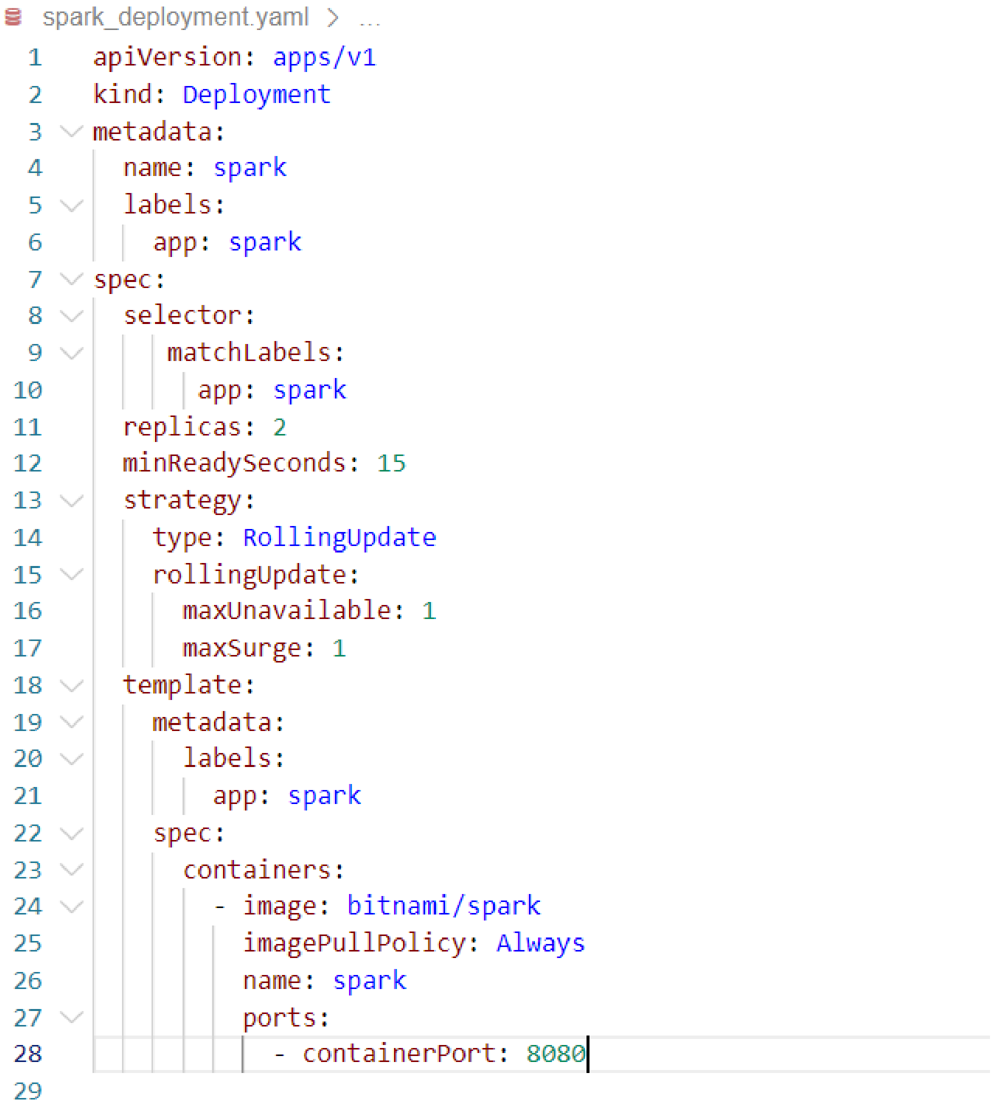
Apache Spark

1. Follow the steps as above to load the docker image from the third party to gcr - docker pull bitnami/spark
2. docker tag bitnami/spark:latest gcr.io/genuine-grid-327615/bitnami/spark:1
3. docker push gcr.io/genuine-grid-327615/bitnami/spark:1



```
adidwivedi96@cloudshell:~/clud-infra/Project (genuine-grid-327615)$ docker pull bitnami/spark
Using default tag: latest
latest: Pulling from bitnami/spark
23b664af592e: Pull complete
bda89ddc65be: Pull complete
5d21c74b4228: Pull complete
c0c515956a4b: Pull complete
cb28f6fabedd: Pull complete
251825bf66a9: Pull complete
7cd8e55f83fb: Pull complete
37800b61fa67: Pull complete
a89ae2b97f94: Pull complete
378a7ee58f2a: Pull complete
Digest: sha256:373048f025775d4f8b093f1fa704e87f8603469e5bfef5068c18855c00affca1
Status: Downloaded newer image for bitnami/spark:latest
docker.io/bitnami/spark:latest
adidwivedi96@cloudshell:~/clud-infra/Project (genuine-grid-327615)$ docker tag bitnami/spark:latest gcr.io/genuine-grid-327615/bitnami/spark:1
adidwivedi96@cloudshell:~/clud-infra/Project (genuine-grid-327615)$ docker push gcr.io/genuine-grid-327615/bitnami/spark:1
The push refers to repository [gcr.io/genuine-grid-327615/bitnami/spark]
3572bf3e81bc: Pushed
16ef99e19e35: Pushed
251feaef1d69: Pushed
c2cc2963a3da: Pushed
0a31da56e68e: Pushed
7d9bb1c6effa: Pushed
5f5d1f7c253a: Pushed
9353863c9e7e: Pushed
ae5896b4e039: Pushed
575a52abf3b1: Layer already exists
```

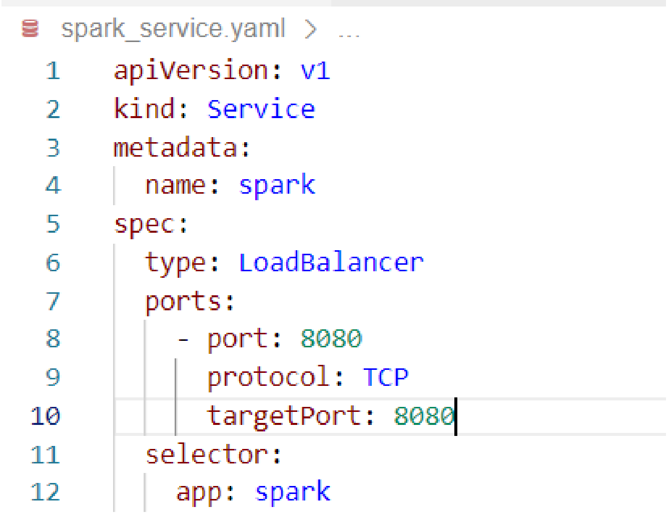
4. Create deployment file as follows -



```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: spark
  labels:
    app: spark
spec:
  selector:
     matchLabels:
       app: spark
  replicas: 2
  minReadySeconds: 15
  strategy:
    type: RollingUpdate
    rollingUpdate:
      maxUnavailable: 1
      maxSurge: 1
  template:
    metadata:
      labels:
        app: spark
    spec:
      containers:
        - image: bitnami/spark
          imagePullPolicy: Always
          name: spark
          ports:
            - containerPort: 8080

```

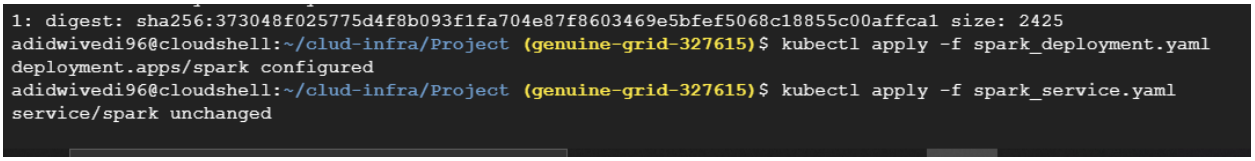
5. kubectl apply -f spark_deployment.yaml

6. Create service file as follows, again we are using loadbalancer service so that we get a public IP which can be tested in the browser



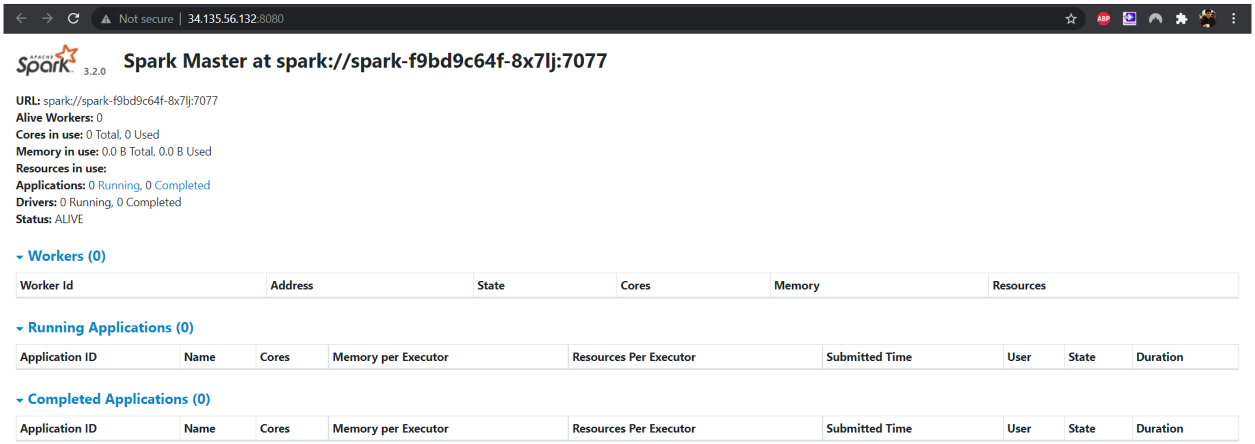
spark_service.yaml > ...

```yaml
apiVersion: v1
kind: Service
metadata:
  name: spark
spec:
  type: LoadBalancer
  ports:
    - port: 8080
      protocol: TCP
      targetPort: 8080
  selector:
    app: spark
```



```
1: digest: sha256:373048f025775d4f8b093f1fa704e87f8603469e5bfef5068c18855c00affca1 size: 2425
adidwivedi96@cloudshell:~/clud-infra/Project (genuine-grid-327615)$ kubectl apply -f spark_deployment.yaml
deployment.apps/spark configured
adidwivedi96@cloudshell:~/clud-infra/Project (genuine-grid-327615)$ kubectl apply -f spark_service.yaml
service/spark unchanged
```

7. We see the following output on the public IP



Spark 3.2.0 **Spark Master at spark://spark-f9bd9c64f-8x7lj:7077**

URL: spark://spark-f9bd9c64f-8x7lj:7077
Alive Workers: 0
Cores in use: 0 Total, 0 Used
Memory in use: 0.0 B Total, 0.0 B Used
Resources in use:
Applications: 0 Running, 0 Completed
Drivers: 0 Running, 0 Completed
Status: ALIVE

▾ Workers (0)

Worker Id	Address	State	Cores	Memory	Resources

▾ Running Applications (0)

Application ID	Name	Cores	Memory per Executor	Resources Per Executor	Submitted Time	User	State	Duration

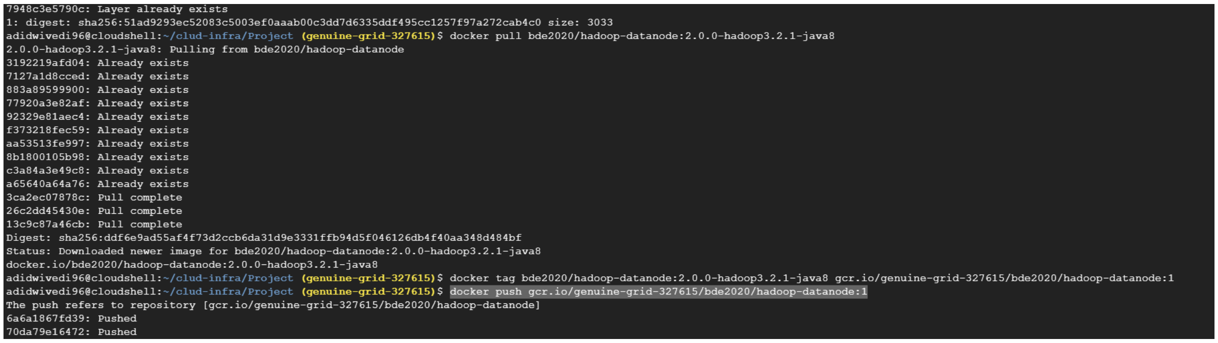
▾ Completed Applications (0)

Application ID	Name	Cores	Memory per Executor	Resources Per Executor	Submitted Time	User	State	Duration

Apache Hadoop

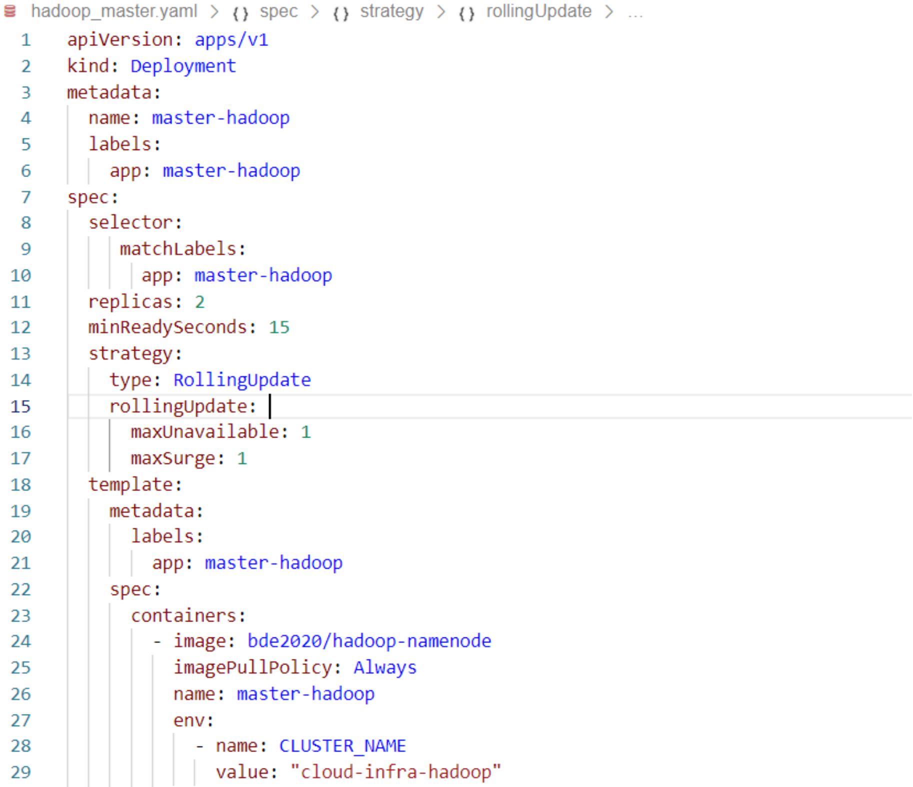
1. docker pull bde2020/hadoop-namenode:2.0.0-hadoop3.2.1-java8
2. docker tag bde2020/hadoop-namenode:2.0.0-hadoop3.2.1-java8 gcr.io/genuine-grid-327615/bde2020/hadoop-namenode:1
3. docker push gcr.io/genuine-grid-327615/bde2020/hadoop-namenode:1

4. docker pull bde2020/hadoop-datanode:2.0.0-hadoop3.2.1-java8
5. docker tag bde2020/hadoop-datanode:2.0.0-hadoop3.2.1-java8 gcr.io/genuine-grid-327615/bde2020/hadoop-datanode:1
6. docker push gcr.io/genuine-grid-327615/bde2020/hadoop-datanode:1



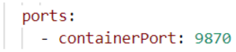
```
7948c3e5790c: Layer already exists
1: digest: sha256:51ad9293ec52083c5003ef0aaab00c3dd7d6335ddf495cc1257f97a272cab4c0 size: 3033
adidwivedi96@cloudshell:~/clud-infra/Project (genuine-grid-327615)$ docker pull bde2020/hadoop-datanode:2.0.0-hadoop3.2.1-java8
2.0.0-hadoop3.2.1-java8: Pulling from bde2020/hadoop-datanode
3192219afd04: Already exists
7127a1d8cced: Already exists
883a89599900: Already exists
77920a3e82af: Already exists
92329e81aec4: Already exists
f373218fec59: Already exists
aa53513fe997: Already exists
8b1800105b98: Already exists
c3a84a3e49c8: Already exists
a65640a64a76: Already exists
3ca2ec07878c: Pull complete
26c2dd45430e: Pull complete
13c9c87a46cb: Pull complete
Digest: sha256:ddf6e9ad55af4f73d2ccb6da31d9e3331ffb94d5f046126db4f40aa348d484bf
Status: Downloaded newer image for bde2020/hadoop-datanode:2.0.0-hadoop3.2.1-java8
docker.io/bde2020/hadoop-datanode:2.0.0-hadoop3.2.1-java8
adidwivedi96@cloudshell:~/clud-infra/Project (genuine-grid-327615)$ docker tag bde2020/hadoop-datanode:2.0.0-hadoop3.2.1-java8 gcr.io/genuine-grid-327615/bde2020/hadoop-datanode:1
adidwivedi96@cloudshell:~/clud-infra/Project (genuine-grid-327615)$ docker push gcr.io/genuine-grid-327615/bde2020/hadoop-datanode:1
The push refers to repository [gcr.io/genuine-grid-327615/bde2020/hadoop-datanode]
6a6a1867fd39: Pushed
70da79e16472: Pushed
```

7. Create deployment yaml for master apache hadoop node as follows. Here we have env section which sets our hadoop cluster’s name. The internal port to be exposed is 9870 -



hadoop_master.yaml > { } spec > { } strategy > { } rollingUpdate > ...

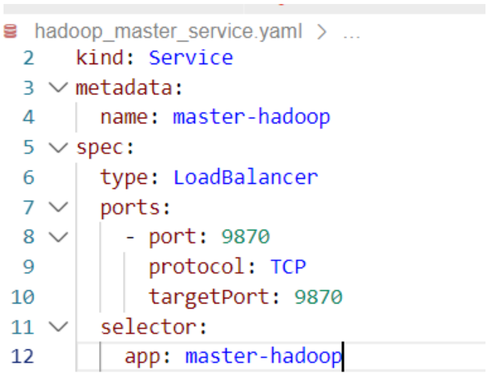
```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: master-hadoop
  labels:
    app: master-hadoop
spec:
  selector:
     matchLabels:
       app: master-hadoop
  replicas: 2
  minReadySeconds: 15
  strategy:
    type: RollingUpdate
    rollingUpdate: 
      maxUnavailable: 1
      maxSurge: 1
  template:
    metadata:
      labels:
        app: master-hadoop
    spec:
      containers:
        - image: bde2020/hadoop-namenode
          imagePullPolicy: Always
          name: master-hadoop
          env:
            - name: CLUSTER_NAME
              value: "cloud-infra-hadoop"
```



```yaml
ports:
  - containerPort: 9870
```

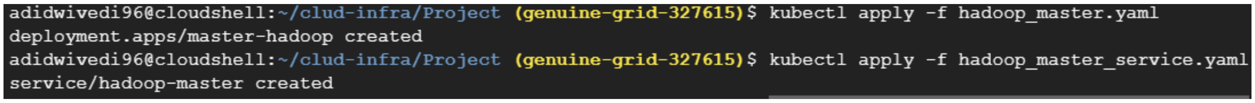
(tabbed space not shown)

8. kubectl apply -f hadoop_master.yaml
9. Create the service file for hadoop master as follows. Here again we have a loadbalancer service which maps internal port 9870 to external 9870 -



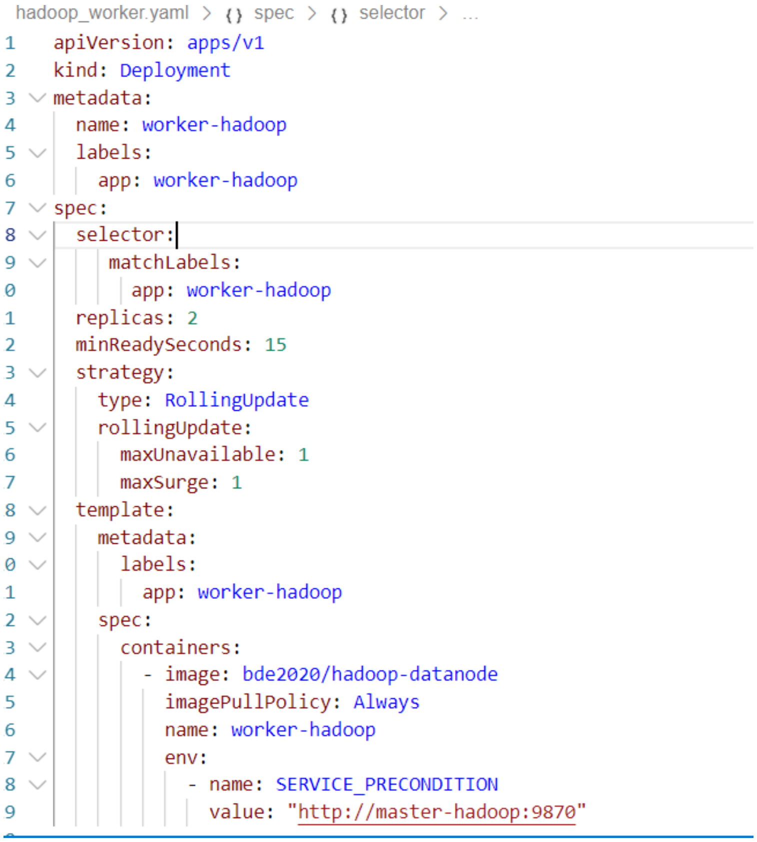
```yaml
kind: Service
metadata:
  name: master-hadoop
spec:
  type: LoadBalancer
  ports:
    - port: 9870
      protocol: TCP
      targetPort: 9870
  selector:
    app: master-hadoop
```

10. kubectl apply -f hadoop_master_service.yaml



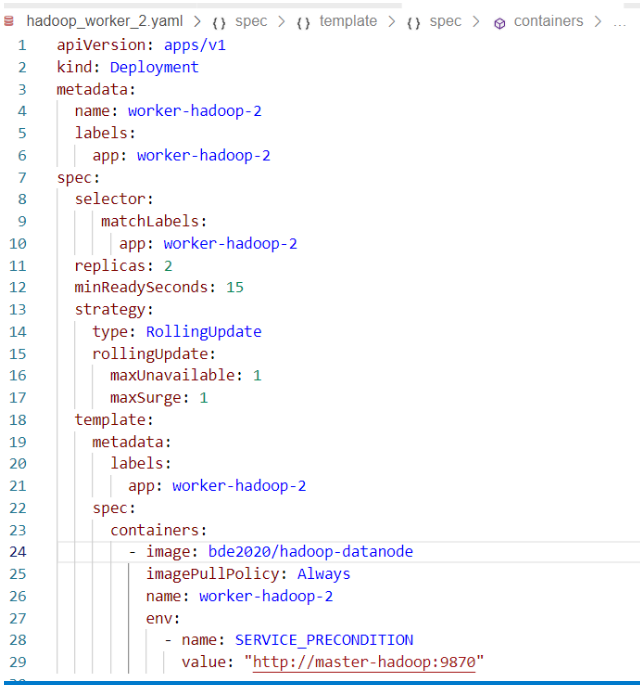
```
adidwivedi96@cloudshell:~/clud-infra/Project (genuine-grid-327615)$ kubectl apply -f hadoop_master.yaml
deployment.apps/master-hadoop created
adidwivedi96@cloudshell:~/clud-infra/Project (genuine-grid-327615)$ kubectl apply -f hadoop_master_service.yaml
service/hadoop-master created
```

11. Create worker hadoop deployment as follows. Here there is an environment variable which points the hadoop worker to master using kubedns.



```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: worker-hadoop
  labels:
    app: worker-hadoop
spec:
  selector:
    matchLabels:
      app: worker-hadoop
  replicas: 2
  minReadySeconds: 15
  strategy:
    type: RollingUpdate
    rollingUpdate:
      maxUnavailable: 1
      maxSurge: 1
  template:
    metadata:
      labels:
        app: worker-hadoop
    spec:
      containers:
        - image: bde2020/hadoop-datanode
          imagePullPolicy: Always
          name: worker-hadoop
          env:
            - name: SERVICE_PRECONDITION
              value: "http://master-hadoop:9870"
```

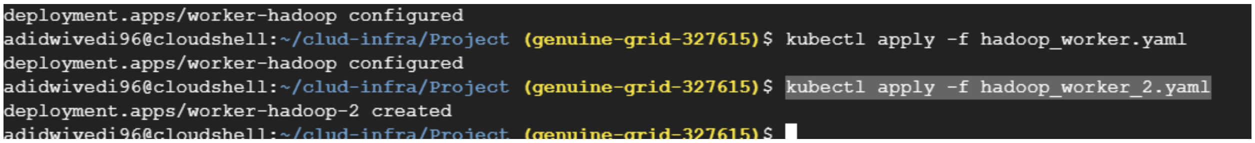
11. Replicate the same deployment file for 2nd hadoop worker as well with the change in name and labels as follows -



hadoop_worker_2.yaml > {} spec > {} template > {} spec > containers > ...

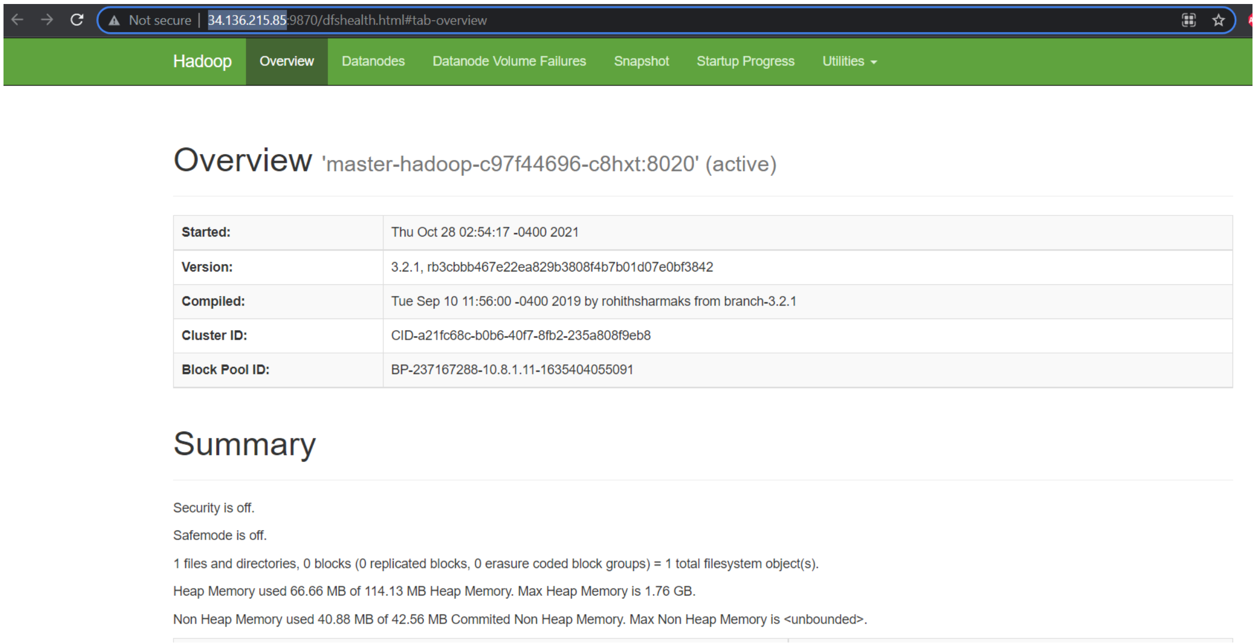
```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: worker-hadoop-2
  labels:
    app: worker-hadoop-2
spec:
  selector:
    matchLabels:
      app: worker-hadoop-2
  replicas: 2
  minReadySeconds: 15
  strategy:
    type: RollingUpdate
    rollingUpdate:
      maxUnavailable: 1
      maxSurge: 1
  template:
    metadata:
      labels:
        app: worker-hadoop-2
    spec:
      containers:
      - image: bde2020/hadoop-datanode
        imagePullPolicy: Always
        name: worker-hadoop-2
        env:
          - name: SERVICE_PRECONDITION
            value: "http://master-hadoop:9870"
```

12. kubectl apply -f hadoop_worker.yaml
13. kubectl apply -f hadoop_worker_2.yaml



```
deployment.apps/worker-hadoop configured
adidwivedi96@cloudshell:~/clud-infra/Project (genuine-grid-327615)$ kubectl apply -f hadoop_worker.yaml
deployment.apps/worker-hadoop configured
adidwivedi96@cloudshell:~/clud-infra/Project (genuine-grid-327615)$ kubectl apply -f hadoop_worker_2.yaml
deployment.apps/worker-hadoop-2 created
adidwivedi96@cloudshell:~/clud-infra/Project (genuine-grid-327615)$
```

14. We get the following hadoop master view on its public IP



Overview 'master-hadoop-c97f44696-c8hxt:8020' (active)

Started:	Thu Oct 28 02:54:17 -0400 2021
Version:	3.2.1, rb3cbbb467e22ea829b3808f4b7b01d07e0bf3842
Compiled:	Tue Sep 10 11:56:00 -0400 2019 by rohithsharmaks from branch-3.2.1
Cluster ID:	CID-a21fc68c-b0b6-40f7-8fb2-235a808f9eb8
Block Pool ID:	BP-237167288-10.8.1.11-1635404055091

Summary

Security is off.

Safemode is off.

1 files and directories, 0 blocks (0 replicated blocks, 0 erasure coded block groups) = 1 total filesystem object(s).

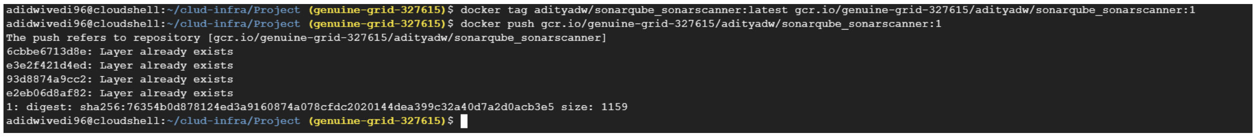
Heap Memory used 66.66 MB of 114.13 MB Heap Memory. Max Heap Memory is 1.76 GB.

Non Heap Memory used 40.88 MB of 42.56 MB Commited Non Heap Memory. Max Non Heap Memory is <unbounded>.

Sonarqube and Sonar-Scanner

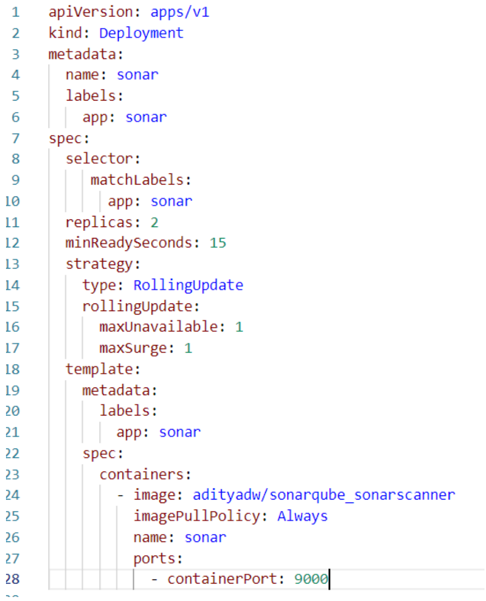
1. docker pull adityadw/sonarqube_sonarscanner

2. docker tag adityadw/sonarqube_sonarscanner:latest gcr.io/genuine-grid-327615/adityadw/sonarqube_sonarscanner:1
3. docker push gcr.io/genuine-grid-327615/adityadw/sonarqube_sonarscanner:1



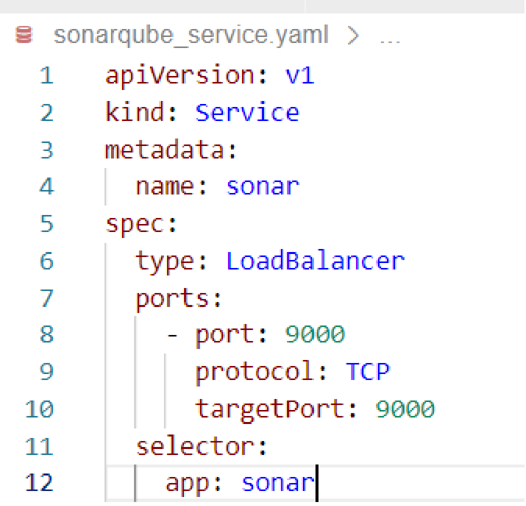
```
adidwivedi96@cloudshell:~/clud-infra/Project (genuine-grid-327615)$ docker tag adityadw/sonarqube_sonarscanner:latest gcr.io/genuine-grid-327615/adityadw/sonarqube_sonarscanner:1
adidwivedi96@cloudshell:~/clud-infra/Project (genuine-grid-327615)$ docker push gcr.io/genuine-grid-327615/adityadw/sonarqube_sonarscanner:1
The push refers to repository [gcr.io/genuine-grid-327615/adityadw/sonarqube_sonarscanner]
6cbbe6713d8e: Layer already exists
e3e2f421d4ed: Layer already exists
93d8874a9cc2: Layer already exists
e2eb06d8af82: Layer already exists
1: digest: sha256:76354b0d878124ed3a9160874a078cfdc2020144dea399c32a40d7a2d0acb3e5 size: 1159
adidwivedi96@cloudshell:~/clud-infra/Project (genuine-grid-327615)$
```

4. Create the sonarqube deployment as follows, exposing internal port 9000



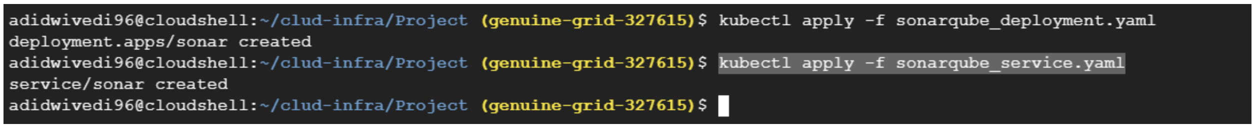
```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: sonar
  labels:
    app: sonar
spec:
  selector:
     matchLabels:
       app: sonar
  replicas: 2
  minReadySeconds: 15
  strategy:
    type: RollingUpdate
    rollingUpdate:
      maxUnavailable: 1
      maxSurge: 1
  template:
    metadata:
      labels:
        app: sonar
    spec:
      containers:
        - image: adityadw/sonarqube_sonarscanner
          imagePullPolicy: Always
          name: sonar
          ports:
            - containerPort: 9000
```

5. kubectl apply -f sonarqube_deployment.yaml
6. Create sonarqube service as follows, mapping internal port 9000 to external port 9000. It's of type LoadBalancer so that it can be accessed via public IP on browser for testing



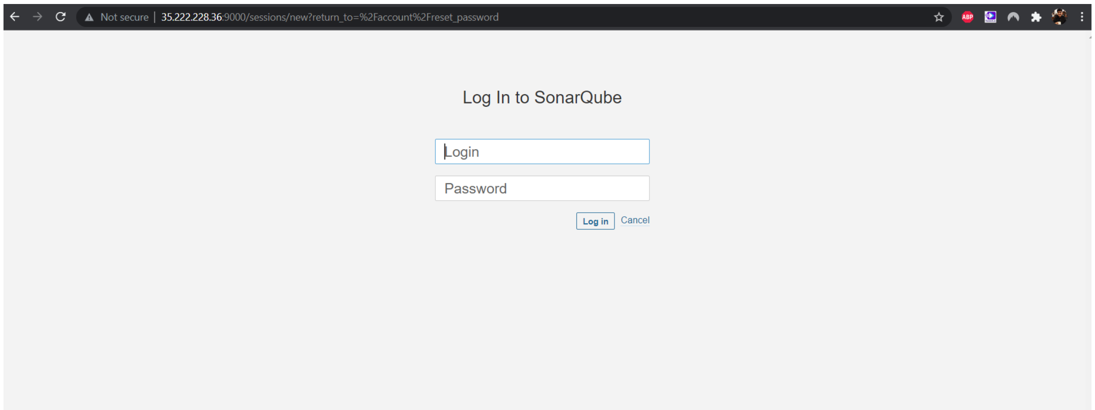
```yaml
apiVersion: v1
kind: Service
metadata:
  name: sonar
spec:
  type: LoadBalancer
  ports:
    - port: 9000
      protocol: TCP
      targetPort: 9000
  selector:
    app: sonar
```

7. kubectl apply -f sonarqube_service.yaml

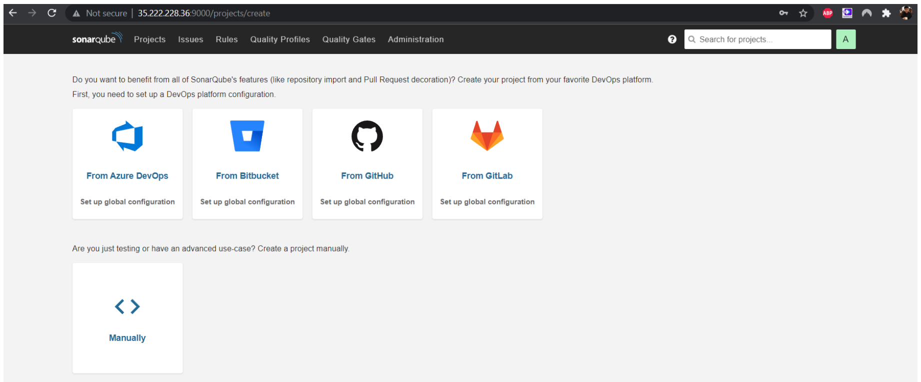


```
adidwivedi96@cloudshell:~/clud-infra/Project (genuine-grid-327615)$ kubectl apply -f sonarqube_deployment.yaml
deployment.apps/sonar created
adidwivedi96@cloudshell:~/clud-infra/Project (genuine-grid-327615)$ kubectl apply -f sonarqube_service.yaml
service/sonar created
adidwivedi96@cloudshell:~/clud-infra/Project (genuine-grid-327615)$
```

8. We get the following output when we visit the sonarqube



Not secure | 35.222.228.36:9000/sessions/new?return_to=%2Faccount%2Freset_password
Log In to SonarQube
Login
Password
Log in
Cancel



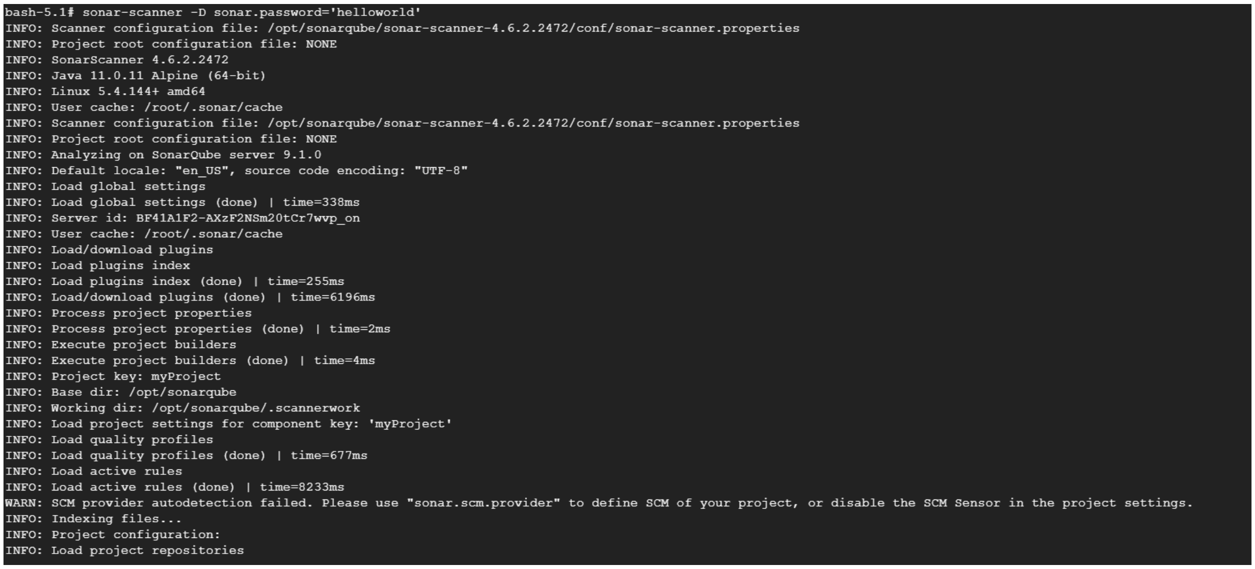
Not secure | 35.222.228.36:9000/projects/create
sonarqube
Projects
Issues
Rules
Quality Profiles
Quality Gates
Administration
Search for projects...
A

Do you want to benefit from all of SonarQube's features (like repository import and Pull Request decoration)? Create your project from your favorite DevOps platform.
First, you need to set up a DevOps platform configuration.

From Azure DevOps
Set up global configuration
From Bitbucket
Set up global configuration
From GitHub
Set up global configuration
From GitLab
Set up global configuration

Are you just testing or have an advanced use-case? Create a project manually.

9. We can run sonar-scanner commands within the pod and get results as seen -

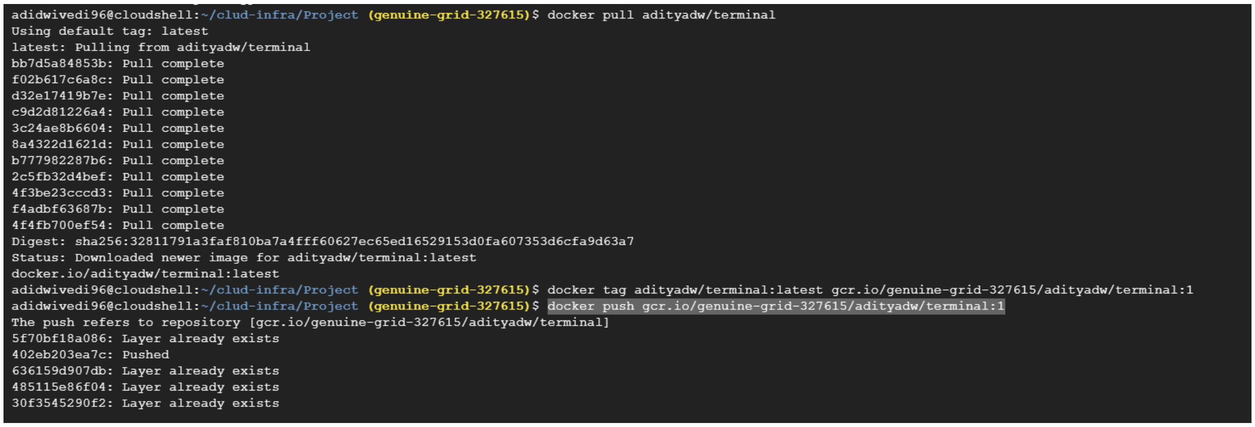


```
bash-5.1# sonar-scanner -D sonar.password='helloworld'
INFO: Scanner configuration file: /opt/sonarqube/sonar-scanner-4.6.2.2472/conf/sonar-scanner.properties
INFO: Project root configuration file: NONE
INFO: SonarScanner 4.6.2.2472
INFO: Java 11.0.11 Alpine (64-bit)
INFO: Linux 5.4.144+ amd64
INFO: User cache: /root/.sonar/cache
INFO: Scanner configuration file: /opt/sonarqube/sonar-scanner-4.6.2.2472/conf/sonar-scanner.properties
INFO: Project root configuration file: NONE
INFO: Analyzing on SonarQube server 9.1.0
INFO: Default locale: "en_US", source code encoding: "UTF-8"
INFO: Load global settings
INFO: Load global settings (done) | time=338ms
INFO: Server id: BF41A1F2-AXzF2NSm20tCr7wvp_on
INFO: User cache: /root/.sonar/cache
INFO: Load/download plugins
INFO: Load plugins index
INFO: Load plugins index (done) | time=255ms
INFO: Load/download plugins (done) | time=6196ms
INFO: Process project properties
INFO: Process project properties (done) | time=2ms
INFO: Execute project builders
INFO: Execute project builders (done) | time=4ms
INFO: Project key: myProject
INFO: Base dir: /opt/sonarqube
INFO: Working dir: /opt/sonarqube/.scannerwork
INFO: Load project settings for component key: 'myProject'
INFO: Load quality profiles
INFO: Load quality profiles (done) | time=677ms
INFO: Load active rules
INFO: Load active rules (done) | time=8233ms
WARN: SCM provider autodetection failed. Please use "sonar.scm.provider" to define SCM of your project, or disable the SCM Sensor in the project settings.
INFO: Indexing files...
INFO: Project configuration:
INFO: Load project repositories
```

10. To do so the commands used are -
    a. kubectl exec --stdin --tty sonar-56b9b86486-t82gv -- /bin/bash
    b. sonar-scanner -D sonar.password='password'

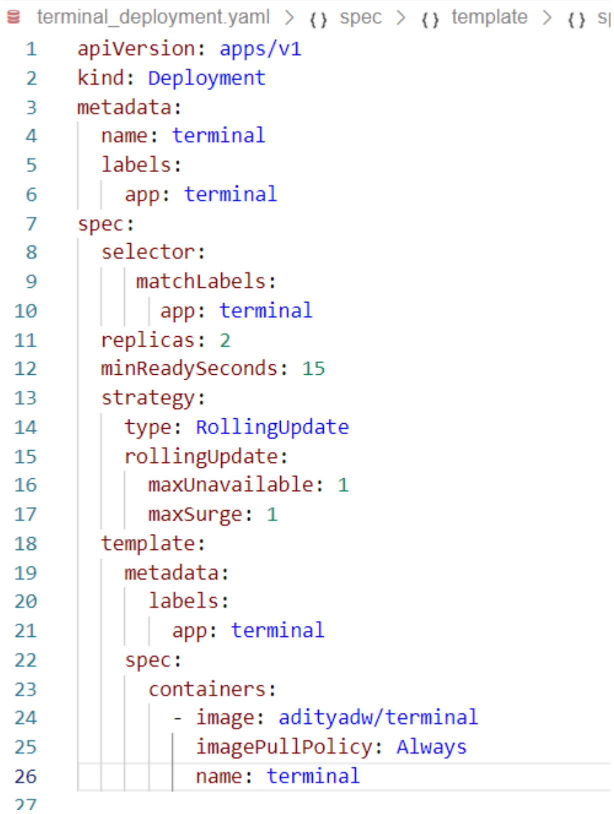
Terminal

1. docker pull adityadw/terminal
2. docker tag adityadw/terminal:latest gcr.io/genuine-grid-327615/adityadw/terminal:1
3. docker push gcr.io/genuine-grid-327615/adityadw/terminal:1



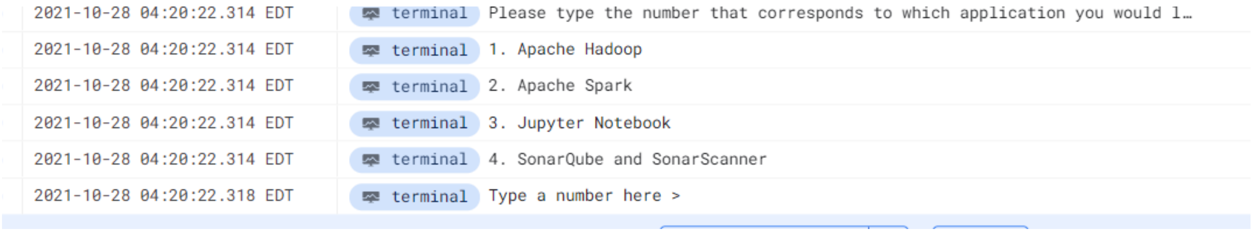
```
adidwivedi96@cloudshell:~/clud-infra/Project (genuine-grid-327615)$ docker pull adityadw/terminal
Using default tag: latest
latest: Pulling from adityadw/terminal
bb7d5a84853b: Pull complete
f02b617c6a8c: Pull complete
d32e17419b7e: Pull complete
c9d2d81226a4: Pull complete
3c24ae8b6604: Pull complete
8a4322d1621d: Pull complete
b777982287b6: Pull complete
2c5fb32d4bef: Pull complete
4f3be23cccd3: Pull complete
f4adbf63687b: Pull complete
4f4fb700ef54: Pull complete
Digest: sha256:32811791a3faf810ba7a4fff60627ec65ed16529153d0fa607353d6cfa9d63a7
Status: Downloaded newer image for adityadw/terminal:latest
docker.io/adityadw/terminal:latest
adidwivedi96@cloudshell:~/clud-infra/Project (genuine-grid-327615)$ docker tag adityadw/terminal:latest gcr.io/genuine-grid-327615/adityadw/terminal:1
adidwivedi96@cloudshell:~/clud-infra/Project (genuine-grid-327615)$ docker push gcr.io/genuine-grid-327615/adityadw/terminal:1
The push refers to repository [gcr.io/genuine-grid-327615/adityadw/terminal]
5f70bf18a086: Layer already exists
402eb203ea7c: Pushed
636159d907db: Layer already exists
485115e86f04: Layer already exists
30f3545290f2: Layer already exists
```

4. Create the following deployment yaml

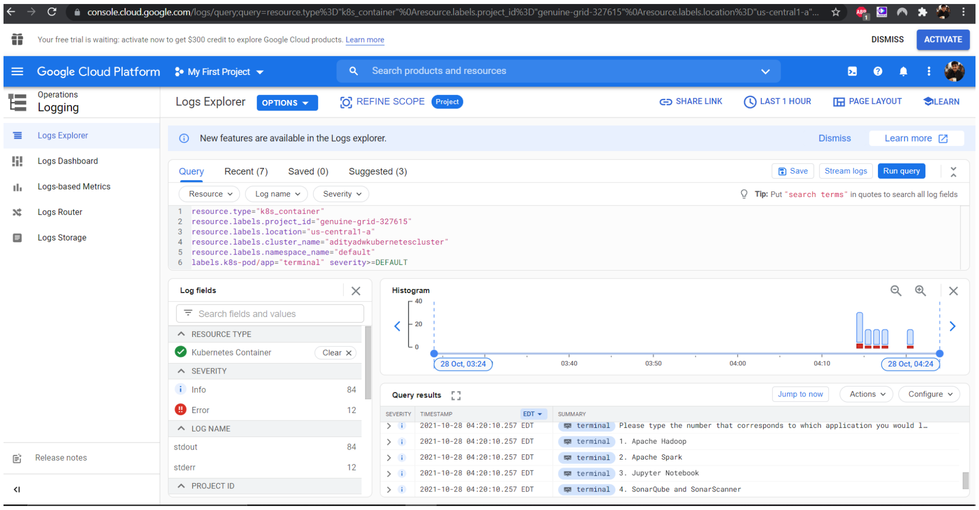


```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: terminal
  labels:
    app: terminal
spec:
  selector:
    matchLabels:
      app: terminal
  replicas: 2
  minReadySeconds: 15
  strategy:
    type: RollingUpdate
    rollingUpdate:
      maxUnavailable: 1
      maxSurge: 1
  template:
    metadata:
      labels:
        app: terminal
    spec:
      containers:
        - image: adityadw/terminal
          imagePullPolicy: Always
          name: terminal
```

The logs show the following output -



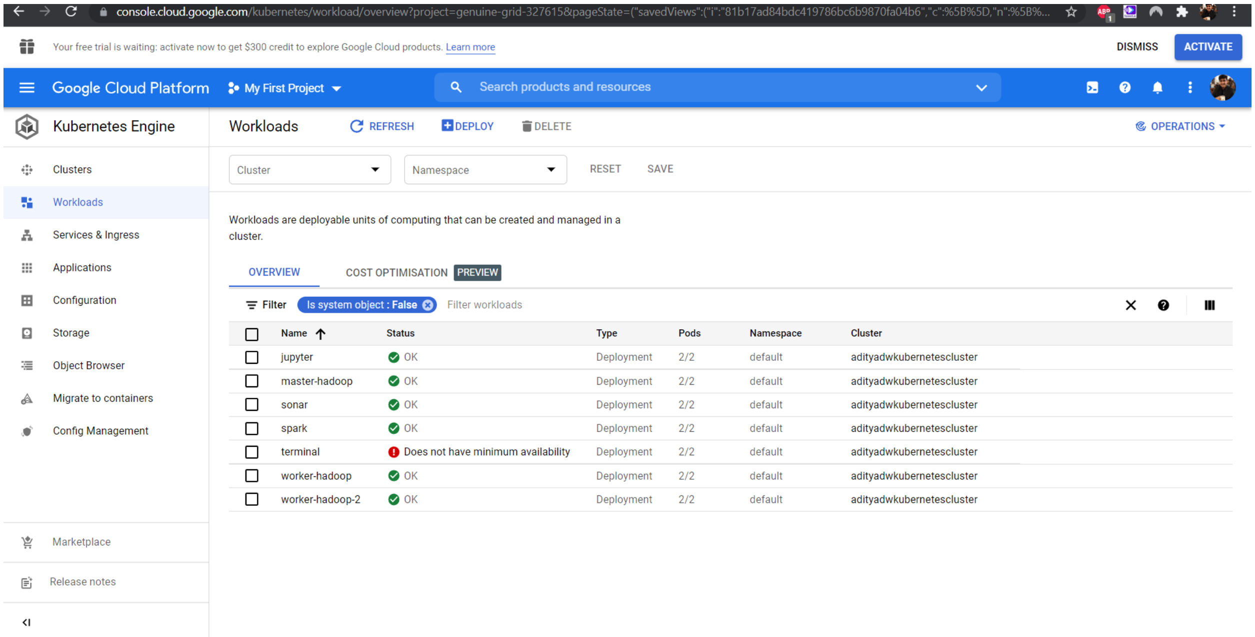
```
2021-10-28 04:20:22.314 EDT   terminal   Please type the number that corresponds to which application you would l…
2021-10-28 04:20:22.314 EDT   terminal   1. Apache Hadoop
2021-10-28 04:20:22.314 EDT   terminal   2. Apache Spark
2021-10-28 04:20:22.314 EDT   terminal   3. Jupyter Notebook
2021-10-28 04:20:22.314 EDT   terminal   4. SonarQube and SonarScanner
2021-10-28 04:20:22.318 EDT   terminal   Type a number here >
```



```
resource.type="k8s_container"
resource.labels.project_id="genuine-grid-327615"
resource.labels.location="us-central1-a"
resource.labels.cluster_name="adityadwkubernetescluster"
resource.labels.namespace_name="default"
labels.k8s-pod/app="terminal" severity>=DEFAULT
```

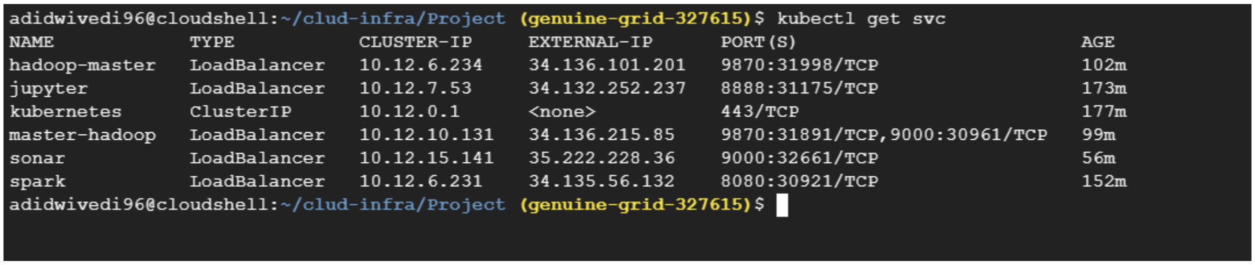
```
SEVERITY  TIMESTAMP                     EDT  SUMMARY
i         2021-10-28 04:20:10.257 EDT        terminal  Please type the number that corresponds to which application you would l…
i         2021-10-28 04:20:10.257 EDT        terminal  1. Apache Hadoop
i         2021-10-28 04:20:10.257 EDT        terminal  2. Apache Spark
i         2021-10-28 04:20:10.257 EDT        terminal  3. Jupyter Notebook
i         2021-10-28 04:20:10.257 EDT        terminal  4. SonarQube and SonarScanner
```

Deployment output



Workloads are deployable units of computing that can be created and managed in a cluster.

	Name	Status	Type	Pods	Namespace	Cluster
□	jupyter	✓ OK	Deployment	2/2	default	adityadwkubernetescluster
□	master-hadoop	✓ OK	Deployment	2/2	default	adityadwkubernetescluster
□	sonar	✓ OK	Deployment	2/2	default	adityadwkubernetescluster
□	spark	✓ OK	Deployment	2/2	default	adityadwkubernetescluster
□	terminal	! Does not have minimum availability	Deployment	2/2	default	adityadwkubernetescluster
□	worker-hadoop	✓ OK	Deployment	2/2	default	adityadwkubernetescluster
□	worker-hadoop-2	✓ OK	Deployment	2/2	default	adityadwkubernetescluster



```
adidwivedi96@cloudshell:~/clud-infra/Project (genuine-grid-327615)$ kubectl get svc
NAME           TYPE           CLUSTER-IP     EXTERNAL-IP      PORT(S)                          AGE
hadoop-master  LoadBalancer   10.12.6.234    34.136.101.201   9870:31998/TCP                   102m
jupyter        LoadBalancer   10.12.7.53     34.132.252.237   8888:31175/TCP                   173m
kubernetes     ClusterIP      10.12.0.1      <none>           443/TCP                          177m
master-hadoop  LoadBalancer   10.12.10.131   34.136.215.85    9870:31891/TCP,9000:30961/TCP    99m
sonar          LoadBalancer   10.12.15.141   35.222.228.36    9000:32661/TCP                   56m
spark          LoadBalancer   10.12.6.231    34.135.56.132    8080:30921/TCP                   152m
adidwivedi96@cloudshell:~/clud-infra/Project (genuine-grid-327615)$
```

Building docker images

Jupyter Notebook

1. docker pull mcr.microsoft.com/mmlspark/release
2. docker build -t adityadw/jupyter_notebook . → runs the jupyter notebook command
3. docker run -it -p 8888:8888 adityadw/jupyter_notebook

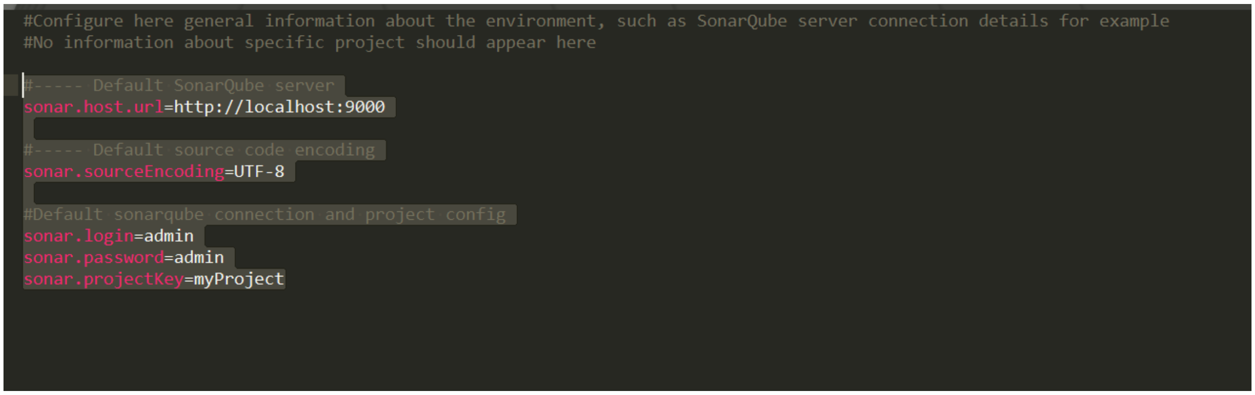
Apache Spark

1. docker pull bitnami/spark
URL - https://hub.docker.com/r/bitnami/spark/

Apache Hadoop
1. docker pull bde2020/hadoop-namenode:2.0.0-hadoop3.2.1-java8
2. docker pull bde2020/hadoop-datanode:2.0.0-hadoop3.2.1-java8

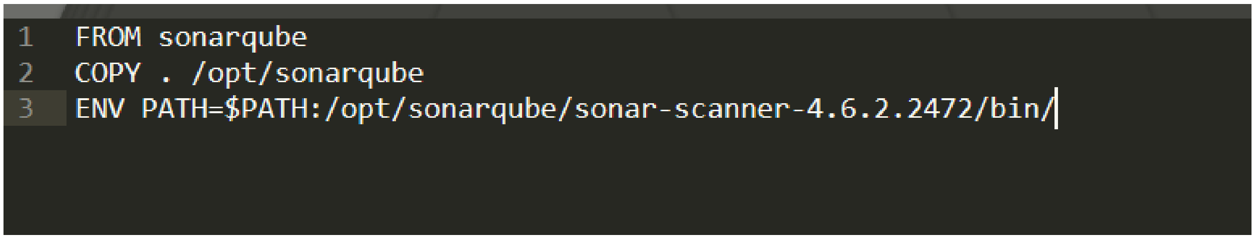
URLs - https://hub.docker.com/r/bde2020/hadoop-namenode,
https://hub.docker.com/r/bde2020/hadoop-datanode

SonarQube and Sonarscanner
1. docker pull sonarqube
2. Download sonar-scanner for any platform which requires pre-installed JVM from here https://binaries.sonarsource.com/Distribution/sonar-scanner-cli/sonar-scanner-cli-4.6.2.2472.zip
3. Edit sonar-scanner configuration file located under config folder as follows -



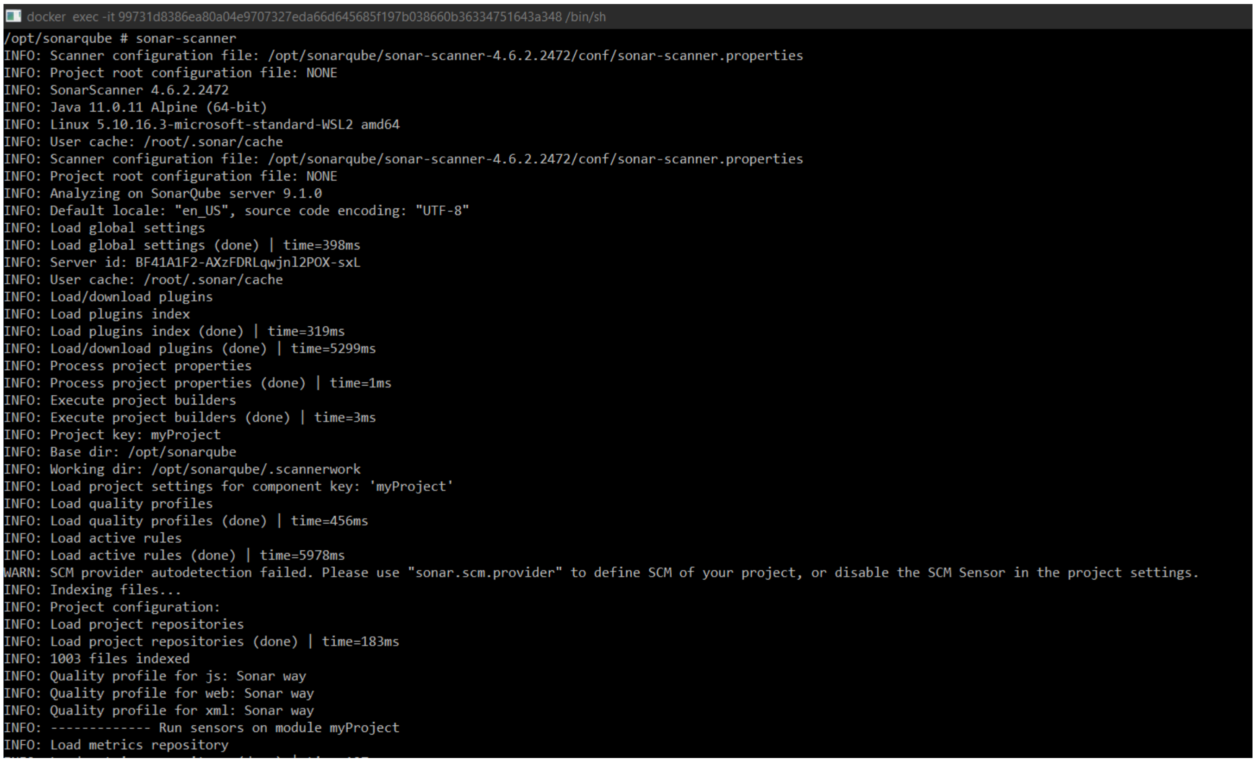
```
#Configure here general information about the environment, such as SonarQube server connection details for example
#No information about specific project should appear here

#----- Default SonarQube server
sonar.host.url=http://localhost:9000

#----- Default source code encoding
sonar.sourceEncoding=UTF-8

#Default sonarqube connection and project config
sonar.login=admin
sonar.password=admin
sonar.projectKey=myProject
```

4. Copy the sonar-scanner directory to container and set the PATH variable to bin directory of sonar-scanner as follows



```
FROM sonarqube
COPY . /opt/sonarqube
ENV PATH=$PATH:/opt/sonarqube/sonar-scanner-4.6.2.2472/bin/
```

5. Sonar-scanner command works in container as follows -


docker exec -it 99731d8386ea80a04e9707327eda66d645685f197b038660b36334751643a348 /bin/sh

```
/opt/sonarqube # sonar-scanner
INFO: Scanner configuration file: /opt/sonarqube/sonar-scanner-4.6.2.2472/conf/sonar-scanner.properties
INFO: Project root configuration file: NONE
INFO: SonarScanner 4.6.2.2472
INFO: Java 11.0.11 Alpine (64-bit)
INFO: Linux 5.10.16.3-microsoft-standard-WSL2 amd64
INFO: User cache: /root/.sonar/cache
INFO: Scanner configuration file: /opt/sonarqube/sonar-scanner-4.6.2.2472/conf/sonar-scanner.properties
INFO: Project root configuration file: NONE
INFO: Analyzing on SonarQube server 9.1.0
INFO: Default locale: "en_US", source code encoding: "UTF-8"
INFO: Load global settings
INFO: Load global settings (done) | time=398ms
INFO: Server id: BF41A1F2-AXzFDRLqwjnl2POX-sxL
INFO: User cache: /root/.sonar/cache
INFO: Load/download plugins
INFO: Load plugins index
INFO: Load plugins index (done) | time=319ms
INFO: Load/download plugins (done) | time=5299ms
INFO: Process project properties
INFO: Process project properties (done) | time=1ms
INFO: Execute project builders
INFO: Execute project builders (done) | time=3ms
INFO: Project key: myProject
INFO: Base dir: /opt/sonarqube
INFO: Working dir: /opt/sonarqube/.scannerwork
INFO: Load project settings for component key: 'myProject'
INFO: Load quality profiles
INFO: Load quality profiles (done) | time=456ms
INFO: Load active rules
INFO: Load active rules (done) | time=5978ms
WARN: SCM provider autodetection failed. Please use "sonar.scm.provider" to define SCM of your project, or disable the SCM Sensor in the project settings.
INFO: Indexing files...
INFO: Project configuration:
INFO: Load project repositories
INFO: Load project repositories (done) | time=183ms
INFO: 1003 files indexed
INFO: Quality profile for js: Sonar way
INFO: Quality profile for web: Sonar way
INFO: Quality profile for xml: Sonar way
INFO: ------------- Run sensors on module myProject
INFO: Load metrics repository
```

6. docker build -t adityadw/sonarqube_sonarscanner .
7. docker push adityadw/sonarqube_sonarscanner

Terminal

Created python program to simulate the terminal
Created a dockerfile from python base image to run the script
docker build -t adityadw/terminal .
docker push adityadw/terminal

#references - https://hub.docker.com/_/microsoft-mmlspark-release, https://github.com/big-data-europe/docker-hadoop/blob/master/docker-compose.yml, https://www.datamechanics.co/blog-post/optimized-spark-docker-images-now-available, https://computingforgeeks.com/how-to-install-apache-spark-on-ubuntu-debian/,
https://docs.sonarqube.org/latest/analysis/scan/sonarscanner/
https://www.geeksforgeeks.org/python-3-input-function/
https://stackoverflow.com/questions/27093612/in-a-dockerfile-how-to-update-path-environment-variable
https://docs.sonarqube.org/latest/analysis/scan/sonarscanner/
https://www.edureka.co/community/7415/copy-is-not-working-in-docker
https://github.com/mohamedfarag/14-848-extra-credit-project
https://github.com/rinormaloku/k8s-mastery/tree/master/resource-manifests
https://kubernetes.io/docs/tasks/debug-application-cluster/get-shell-running-container/