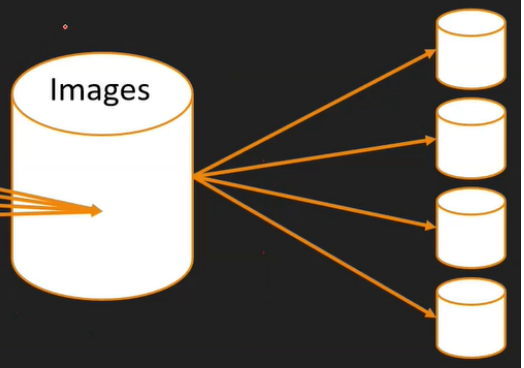
Container :

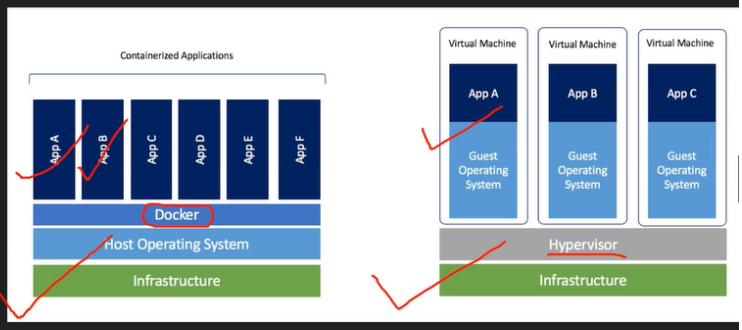
Container

- Software shipping technology
- Let's say building JAVA APP
  - java runtime (JRE)
  - All library dependency
  - Network configuration
  - Runtime DLL
- Combine all this thing into one single bucket & ship
- Compare with oops
  - Images like class (Blueprint)
  - Container like objects



VM vs Containers

- Container are lightweight
- Easily portable to any public cloud, VM, bare metal
- For Micro service deployment, lightweight containers are preferred
- Fast CI/CD cycle
- All major public cloud providers has services to deploy container
- In GCP
  - VM
  - Cloud Run
  - GKE



https://www.docker.com/blog/containers-replacing-virtual-machines/

which we are gonna see in the next video.

Docker

- Container are abstract concept.
- Docker is specific implementation of Container concept.
- Create Docker Images, from Images can create multiple containers
- Here you packaged app in images
- Container use image to start application
- Containers run on any operating system – prefer Linux based
- It works exactly same independent of OS, machine, Environment
- Lightweight compared to VM
- Easier to maintain & deploy
- Docker works with any language, runtime, OS

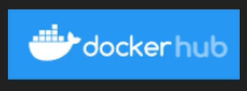
it doesn't care for what your bundle inside

Docker workflow

Docker Installation

Dockerfile + Source code

Docker Image



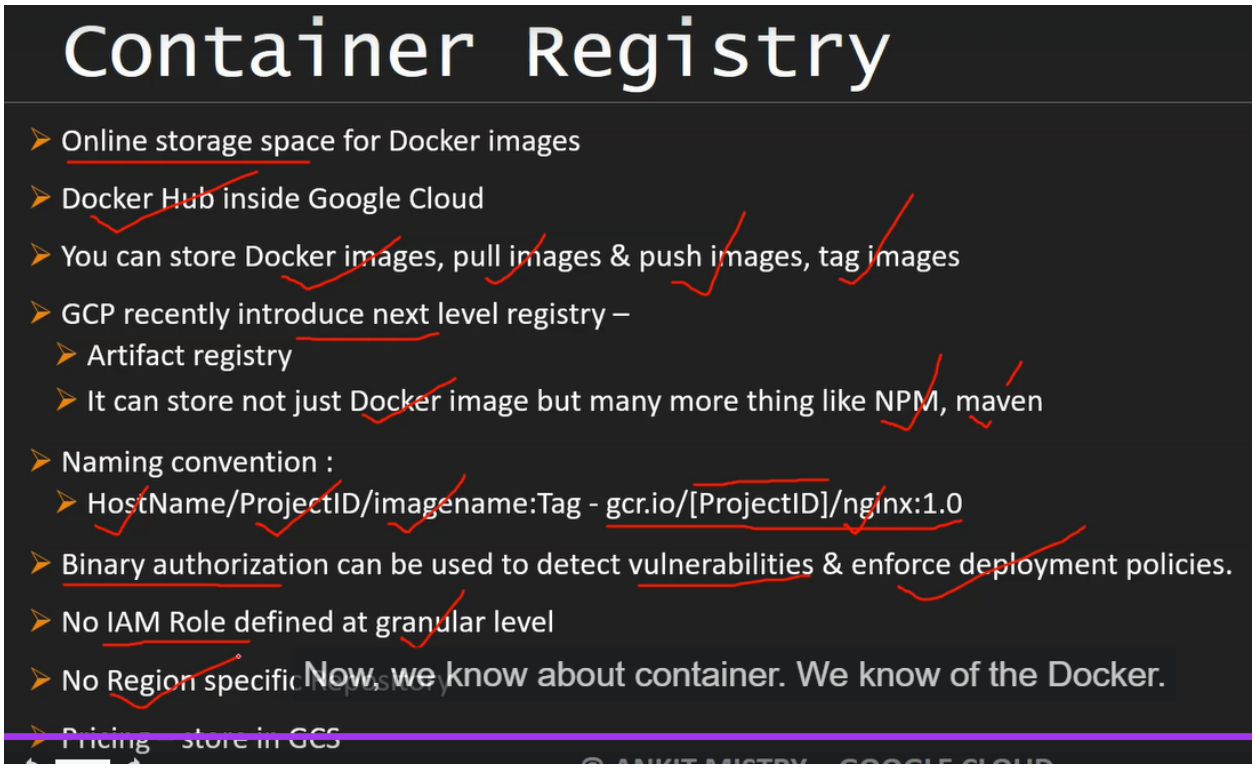
Container Registry

Artifact Registry

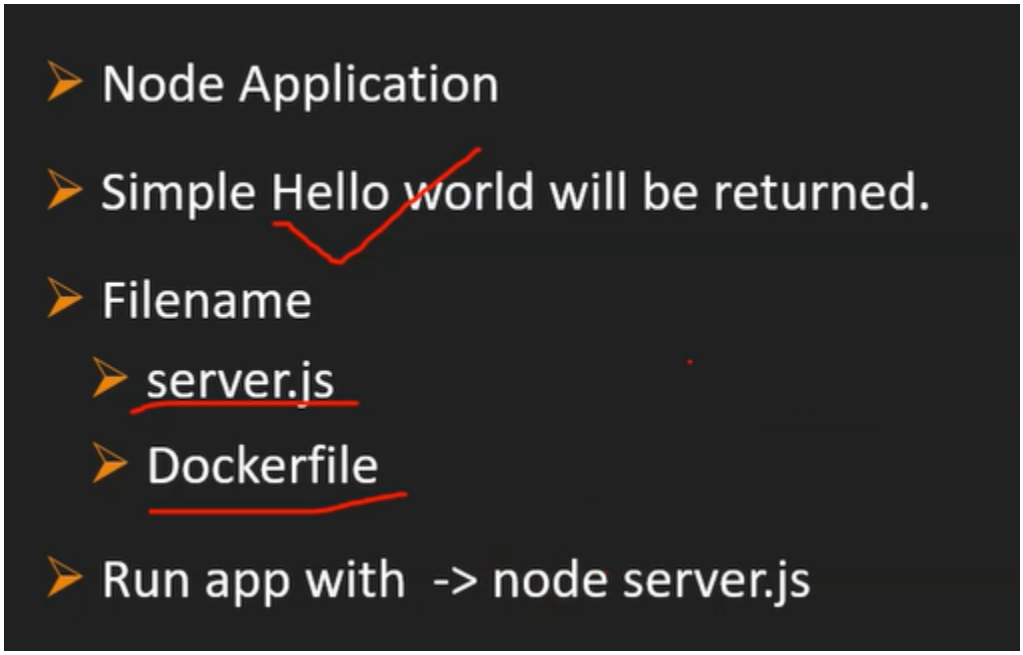
AWS ECR

Docker Container

Container registry :
When we push image to container registry naming convention is gcr.io or acr.gcr.io
Also images will be stored in specific locations only .
Eg : gcr.io - us location , acr.gcr.io - any location in asia .



Create docker image :




```
var http = require('http');
var handleRequest = function(request, response) {
  response.writeHead(200);
  response.end("<h1> Welcome to Container world : Docker </h1>");
}
var www = http.createServer(handleRequest);
www.listen(8080);
```

```
student_00_038c145630ed@cloudshell:~ (qwiklabs-gcp-00-0d903f7a3981)$ cat server.js
var http = require('http');
var handleRequest = function(request, response) {
        response.writeHead(200);
        response.end("<h1>Welcome to container world:docker</h1>");
}
var www=http.createServer(handleRequest);
www.listen(8080);
student_00_038c145630ed@cloudshell:~ (qwiklabs-gcp-00-0d903f7a3981)$
```

Dockerfile :

We used a node image , exposed service on port 8080 .

```
Successfully tagged gcr.io/qwiklabs-gcp-00-0d903f7a3981/node:v.0.1
student_00_038c145630ed@cloudshell:~ (qwiklabs-gcp-00-0d903f7a3981)$ cat Dockerfile
FROM node
EXPOSE 8080
COPY server.js .
CMD node server.js
```

node server.js - to run node applications .
Bundle package to 1 image using docker build command .
docker build -t gcr.io/project-id/image-name:tag-version . (location of dockerfile)

```
Build an image from a Dockerfile
student_00_038c145630ed@cloudshell:~ (qwiklabs-gcp-00-0d903f7a3981)$ docker build -t gcr.io/qwiklabs-gcp-00-0d903f7a3981/node:v.0.1 .
Sending build context to Docker daemon  18.94kB
Step 1/4 : FROM node
latest: Pulling from library/node
bbeef03cda1f: Pull complete
f049f75f014e: Pull complete
56261d0e6b05: Pull complete
9bd150679dbd: Pull complete
5b282ee9da04: Pull complete
8bc43c905b24: Pull complete
3ce9e21024ae: Pull complete
36d91be124f1: Pull complete
7d6b04e90455: Pull complete
Digest: sha256:c1d6d7364e956b061d62241c362b3cd0856beba066ec60e25523a169e2137623
Status: Downloaded newer image for node:latest
 ---> 51bd6c84a7f2
Step 2/4 : EXPOSE 8080
 ---> Running in 8ee2e52af447
Removing intermediate container 8ee2e52af447
 ---> 92d8d7972c63
Step 3/4 : COPY server.js .
 ---> 72d963a60812
Step 4/4 : CMD node server.js
 ---> Running in c9581063f466
Removing intermediate container c9581063f466
```

OPTIMIZE DOCKER IMAGE & RUN DOCKER CONTAINER - V2

Previous image sizes were greater so we will use some light weight image of node (some alpine version) in docker file .
First delete old images .
docker rmi image img_id .

Update DOckerfile to use some alpine version node
FROM node:alpine_version

```
student_00_038c145630ed@cloudshell:~ (qwiklabs-gcp-00-0d903f7a3981)$ cat Dockerfile
FROM node:19-alpine3.16
EXPOSE 8080
COPY server.js .
CMD node server.js
```

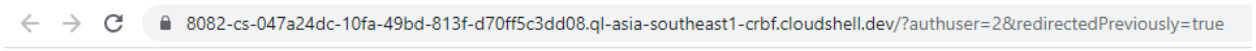
Once done we can build an image .

```
student_00_038c145630ed@cloudshell:~ (qwiklabs-gcp-00-0d903f7a3981)$ docker build -t gcr.io/qwiklabs-gcp-00-0d903f7a3981/node:v.0.2 .
Sending build context to Docker daemon  19.97kB
Step 1/4 : FROM node:19-alpine3.16
19-alpine3.16: Pulling from library/node
ca7dd9ec2225: Pull complete
d0cfa7ee7806: Pull complete
e363461a272b: Pull complete
4b27650c55ea: Pull complete
Digest: sha256:4ceb1b89ced8c3f8098ccc26039be8ebf43f102cd7c4f0d76b46a37c61b345e8
Status: Downloaded newer image for node:19-alpine3.16
 ---> 3f144b1a15b6
Step 2/4 : EXPOSE 8080
 ---> Running in 730256e7b2c7
Removing intermediate container 730256e7b2c7
 ---> 8d0bd0f81616
Step 3/4 : COPY server.js .
 ---> 057d28723310
Step 4/4 : CMD node server.js
 ---> Running in 590035d0cf42
Removing intermediate container 590035d0cf42
 ---> 68c06bbd22d2
Successfully built 68c06bbd22d2
Successfully tagged gcr.io/qwiklabs-gcp-00-0d903f7a3981/node:v.0.2
student_00_038c145630ed@cloudshell:~ (qwiklabs-gcp-00-0d903f7a3981)$ docker images
REPOSITORY                                   TAG             IMAGE ID       CREATED          SIZE
gcr.io/qwiklabs-gcp-00-0d903f7a3981/node   v.0.2           68c06bbd22d2   23 seconds ago   174MB
gcr.io/qwiklabs-gcp-00-0d903f7a3981/node   v.0.1           934d106f3b49   3 minutes ago    998MB
node                                         latest          51bd6c84a7f2   4 days ago       998MB
node                                         19-alpine3.16   3f144b1a15b6   8 days ago       174MB
```

Push image to container registry / docker hub / artifact registry.
Before we push an image we have to run the image as a container .
docker ps - to check all container running .
docker run -d(to run in background) -p 8082:8080 imageLversion

```
student_00_038c145630ed@cloudshell:~ (qwiklabs-gcp-00-0d903f7a3981)$ docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
student_00_038c145630ed@cloudshell:~ (qwiklabs-gcp-00-0d903f7a3981)$ docker run -d -p 8082:8080 gcr.io/qwiklabs-gcp-00-0d903f7a3981/node:v.0.2
04350b254cfd02bca6f0dad90ee1db2082793560d68f5dffbf1e9b39ede51b55
student_00_038c145630ed@cloudshell:~ (qwiklabs-gcp-00-0d903f7a3981)$
```

With this above command container will not be accessible on port 8080 , it will be accessible on 8082 which will redirect traffic to 8080 .
We have to check services on hostport 8082.



Welcome to container world:docker

docker stop container-id
docker rm container-id
docker ps -a

```
student_00_038c145630ed@cloudshell:~ (qwiklabs-gcp-00-0d903f7a3981)$ docker stop 04350b254cfd
04350b254cfd
student_00_038c145630ed@cloudshell:~ (qwiklabs-gcp-00-0d903f7a3981)$ docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
student_00_038c145630ed@cloudshell:~ (qwiklabs-gcp-00-0d903f7a3981)$ docker rm 04350b254cfd
04350b254cfd
student_00_038c145630ed@cloudshell:~ (qwiklabs-gcp-00-0d903f7a3981)$ docker ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
student_00_038c145630ed@cloudshell:~ (qwiklabs-gcp-00-0d903f7a3981)$
```

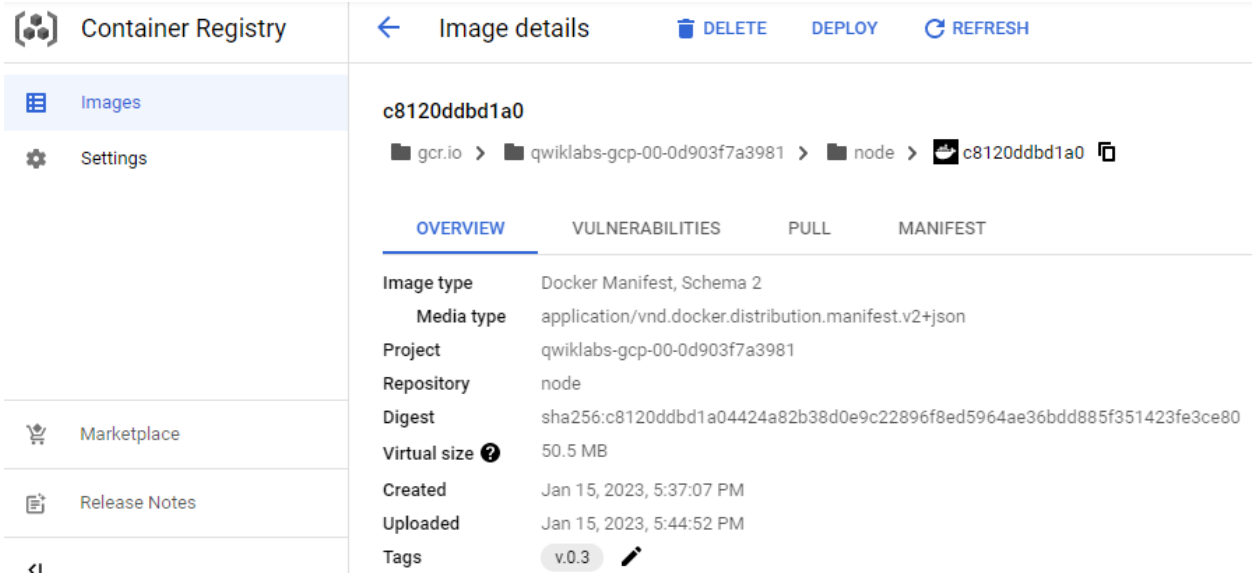
Push image to container registry - V2
Retag myapp image :
docker tag myapp:version gcr.io/project-id/myapp:version - we can use the same version that creates 1 more image having the same image id .
docker images - to check images

```
student_00_038c145630ed@cloudshell:~ (qwiklabs-gcp-00-0d903f7a3981)$ docker tag gcr.io/qwiklabs-gcp-00-0d903f7a3981/node:v.0.2 gcr.io/qwiklabs-gcp-00-0d903f7a3981/node:
v.0.3
student_00_038c145630ed@cloudshell:~ (qwiklabs-gcp-00-0d903f7a3981)$ docker images
REPOSITORY                                  TAG           IMAGE ID       CREATED          SIZE
gcr.io/qwiklabs-gcp-00-0d903f7a3981/node    v.0.2         68c06bbd22d2   6 minutes ago    174MB
gcr.io/qwiklabs-gcp-00-0d903f7a3981/node    v.0.3         68c06bbd22d2   6 minutes ago    174MB
gcr.io/qwiklabs-gcp-00-0d903f7a3981/node    v.0.1         934d106f3b49   10 minutes ago   998MB
node                                        latest        51bd6c84a7f2   4 days ago       998MB
node                                        19-alpine3.16 3f144b1a15b6   8 days ago       174MB
student_00_038c145630ed@cloudshell:~ (qwiklabs-gcp-00-0d903f7a3981)$
```

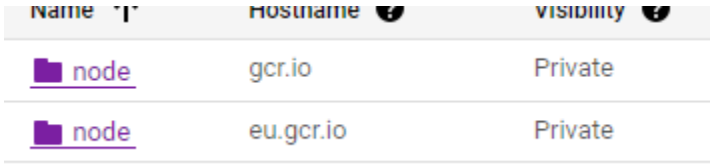
docker push gcr.io/projectid/myapp:version - it will push the image to the container registry .
docker pull - to pull image .

```
student_00_038c145630ed@cloudshell:~ (qwiklabs-gcp-00-0d903f7a3981)$ docker push gcr.io/qwiklabs-gcp-00-0d903f7a3981/node:v.0.3
The push refers to repository [gcr.io/qwiklabs-gcp-00-0d903f7a3981/node]
b4374af4a4cf: Pushed
0374324edcd7: Layer already exists
d0c8c4ce549a: Layer already exists
fe8fef27e38e: Layer already exists
e5e13b0c77cb: Layer already exists
v.0.3: digest: sha256:c8120ddbd1a04424a82b38d0e9c22896f8ed5964ae36bdd885f351423fe3ce80 size: 1365
student_00_038c145630ed@cloudshell:~ (qwiklabs-gcp-00-0d903f7a3981)$ docker pull gcr.io/qwiklabs-gcp-00-0d903f7a3981/node:v.0.3
v.0.3: Pulling from qwiklabs-gcp-00-0d903f7a3981/node
Digest: sha256:c8120ddbd1a04424a82b38d0e9c22896f8ed5964ae36bdd885f351423fe3ce80
Status: Image is up to date for gcr.io/qwiklabs-gcp-00-0d903f7a3981/node:v.0.3
gcr.io/qwiklabs-gcp-00-0d903f7a3981/node:v.0.3
```

After we push the image in gcp - 1 bucket will be created where container registry related data will be stored . - we can see the location will be us by default .



We can create 1 more tag image in eu.gcr.io and push it to eu.gcr.io - container registry .
Then the location will be changed to eu .
This is a limitation in the container registry that granular access is not there which is resolved in the artifact registry .

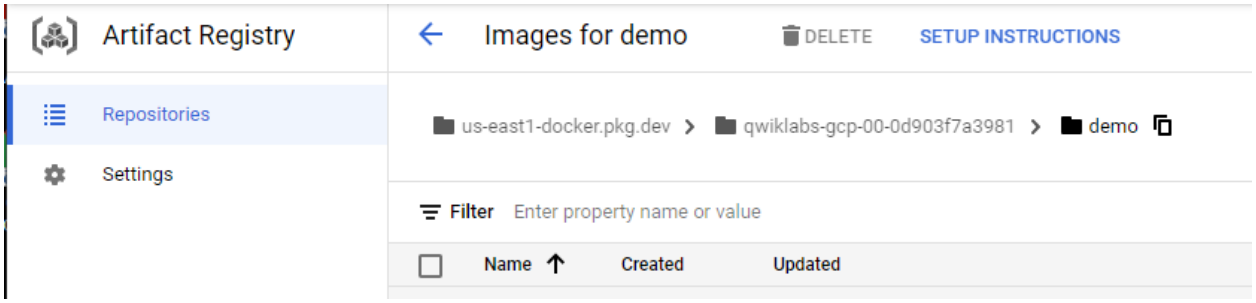


INTRODUCTION TO ARTIFACT REGISTRY - V2

Fine grained access control with cloud iam . we can manage access from cloud iam .
Check roles present for artifact registry in cloud iam . IAM > ROLES
Multiple repo per project . In the container registry we don't create any repo but in the artifact registry we can create multiple repo in the same or different location .

Regional and multi regional repo -
It can store DOCKER IMAGE ,NPM , MAVEN package python .
While creating a repo we can select what all we want to store .

ARTIFACT REGISTRY > Create repo > Format - select what u want to store > region >Select region where u want to store > Google managed encryption key . > create .



Now the repo is created in the artifact registry. We will push the docker image to the artifact registry .
1) Configure
gcloud auth configure-docker get-the-region-same-as-repo-created

```
student_00_038c145630ed@cloudshell:~ (qwiklabs-gcp-00-0d903f7a3981)$ gcloud auth configure-docker us-east1-docker.pkg.dev
WARNING: Your config file at [/home/student_00_038c145630ed/.docker/config.json] contains these credential helper entries:

{
  "credHelpers": {
    "gcr.io": "gcloud",
    "us.gcr.io": "gcloud",
    "eu.gcr.io": "gcloud",
    "asia.gcr.io": "gcloud",
    "staging-k8s.gcr.io": "gcloud",
    "marketplace.gcr.io": "gcloud"
  }
}
Adding credentials for: us-east1-docker.pkg.dev
After update, the following will be written to your Docker config file located at [/home/student_00_038c145630ed/.docker/config.json]:
 {
  "credHelpers": {
    "gcr.io": "gcloud",
    "us.gcr.io": "gcloud",
    "eu.gcr.io": "gcloud",
    "asia.gcr.io": "gcloud",
    "staging-k8s.gcr.io": "gcloud",
    "marketplace.gcr.io": "gcloud",
    "us-east1-docker.pkg.dev": "gcloud"
  }
}

Do you want to continue (Y/n)?  Y

Docker configuration file updated.
```

Now this package will be added in our local area .
.docker/config.json will be created after this we can add this json for each region while automating tasks .
NOw tag our image to the name of the repo created . We can use the same app and version name .

```
student_00_038c145630ed@cloudshell:~ (qwiklabs-gcp-00-0d903f7a3981)$ docker images
REPOSITORY                                      TAG             IMAGE ID       CREATED          SIZE
eu.gcr.io/qwiklabs-gcp-00-0d903f7a3981/node     v.0.3           68c06bbd22d2   21 minutes ago   174MB
gcr.io/qwiklabs-gcp-00-0d903f7a3981/node        v.0.2           68c06bbd22d2   21 minutes ago   174MB
gcr.io/qwiklabs-gcp-00-0d903f7a3981/node        v.0.3           68c06bbd22d2   21 minutes ago   174MB
gcr.io/qwiklabs-gcp-00-0d903f7a3981/node        v.0.1           934d106f3b49   25 minutes ago   998MB
node                                            latest          51bd6c84a7f2   4 days ago       998MB
node                                            19-alpine3.16   3f144b1a15b6   8 days ago       174MB
student_00_038c145630ed@cloudshell:~ (qwiklabs-gcp-00-0d903f7a3981)$ docker tag gcr.io/qwiklabs-gcp-00-0d903f7a3981/node:v.0.2 us-east1-docker.pkg.dev/qwiklabs-gcp-00-0
d903f7a3981/demo/node:v.0.1
student_00_038c145630ed@cloudshell:~ (qwiklabs-gcp-00-0d903f7a3981)$ docker images
REPOSITORY                                                TAG             IMAGE ID       CREATED          SIZE
eu.gcr.io/qwiklabs-gcp-00-0d903f7a3981/node               v.0.3           68c06bbd22d2   23 minutes ago   174MB
gcr.io/qwiklabs-gcp-00-0d903f7a3981/node                  v.0.2           68c06bbd22d2   23 minutes ago   174MB
gcr.io/qwiklabs-gcp-00-0d903f7a3981/node                  v.0.3           68c06bbd22d2   23 minutes ago   174MB
us-east1-docker.pkg.dev/qwiklabs-gcp-00-0d903f7a3981/demo/node   v.0.1           68c06bbd22d2   23 minutes ago   174MB
gcr.io/qwiklabs-gcp-00-0d903f7a3981/node                  v.0.1           934d106f3b49   26 minutes ago   998MB
node                                                      latest          51bd6c84a7f2   4 days ago       998MB
node                                                      19-alpine3.16   3f144b1a15b6   8 days ago       174MB
```

2) Push image to artifact registry .
docker push repo_name_and_app_version_anme .

```
student_00_038c145630ed@cloudshell:~ (qwiklabs-gcp-00-0d903f7a3981)$ docker push us-east1-docker.pkg.dev/qwiklabs-gcp-00-0d903f7a3981/demo/node:v.0.1
The push refers to repository [us-east1-docker.pkg.dev/qwiklabs-gcp-00-0d903f7a3981/demo/node]
b4374af4a4cf: Pushed
0374324edcd7: Pushed
d0c8c4ce549a: Pushed
fe8fef27e38e: Pushed
e5e13b0c77cb: Pushed
v.0.1: digest: sha256:c8120ddbd1a04424a82b38d0e9c22896f8ed5964ae36bdd885f351423fe3ce80 size: 1365
student_00_038c145630ed@cloudshell:~ (qwiklabs-gcp-00-0d903f7a3981)$
```

This is how we create images and push them to the container registry and artifact registry .