# OracleCloudInstanceInitialSetup

1. Create Oracle Compute Instance: select either oracle Linux or CentOS.

2. Availability zone and capacity similar like AWS.

3. Boot volume similar like EBS is a secondary volume to run instance.

4. Make sure Add SSH keys: Open Putty Gen and create private key and save it with (.ppk). Add public key while SSH key field. Private key is unique and associate with its public key. Like private key is key and public key is locker. Such as to open any locker, unique is requires. If you have more than one instances, then generate each of its private and public keys.

5. Virtual Cloud Network similar like Private and public network (VPN). Make sure create all private and public subnet with port number.

6. Ingress Rules – Add CIDR and protocols like TCP or all.

7. How to connect: Use MobaXTerm, provide IP address, username: blank, port: 22 , add private key and click connect. Similar way left username: blank on putty also.

8. Connect from console: sudo ssh opc@public-id-address -i /etc/rsa-private-key.ppk

9. Setup docker: https://technology.amis.nl/2020/02/06/run-always-free-docker-container-on-oracle-cloud-infrastructure/

Please note: Always run with sudo, if yum not work, install it.

sudo yum-config-manager --enable ol7\_addons

sudo yum install docker-engine -y

sudo systemctl start docker

sudo systemctl enable docker

**10. Install and connect Alpine on docker:**

# docker pull alpine

Usually, an Alpine Linux image doesn't contain bash, Instead you can use /bin/ash, /bin/sh, ash or only sh.

**/bin/ash**

docker run -it --rm alpine /bin/ash

**/bin/sh**

docker run -it --rm alpine /bin/sh

**ash**

docker run -it --rm alpine ash

**sh**

docker run -it --rm alpine sh

**11. Setup nginx container image with this command:**

sudo docker run -d \

--restart always \

--hostname nginx.lucasjellema.com \

-p 3456:443 \

-p 3457:80 \

--name my-nginx \

nginx

(So nginx is rinning on docker!)

12. curl http://publicIPAddress:3457

**13. Setup Apache HTTP Server(This is not in docker):** https://scriptingmysql.wordpress.com/2019/11/19/oracles-always-free-cloud-instance-adding-a-web-server-to-your-free-mysql-compute-instance-with-zero-monthly-charges/

sudo yum install httpd -y

sudo apachectl start

sudo systemctl enable httpd

sudo apachectl configtest

sudo firewall-cmd --permanent --zone=public --add-service=http

sudo firewall-cmd –reload

sudo bash -c 'echo This is my new web page running on Oracle Cloud Always Free compute instance > /var/www/html/index.html'

curl <http://publicIPAddress:80>

To setup new react application: build the app, rename the project folder to html and replace the html file at the VM location: /var/www/. Thats it!

14. **Setup Apache HTTP Server(This is in docker):**

sudo docker pull httpd:latest

sudo docker run -dit --name <image name: httpd> -p 8080:80 -v /var/www/html/:/usr/local/apache2/htdocs/ <httpd:2.4 or httpd:latest>

Example: sudo docker run -dit --name httpd -p 8080:80 -v /var/www/html/:/usr/local/apache2/htdocs/ httpd:2.4

curl <http://publicIPAddress:80>80

We can get an interactive shell session to the container using the 'docker exec' command.

docker exec -ti docker-httpd /bin/bash

To setup new react application: build the app, rename the project folder to html and replace the html file at the VM location: /var/www/. Thats it!

**15. Oracle Autonomous Database detail instructions:**

https://docs.oracle.com/en/cloud/paas/autonomous-data-warehouse-cloud/tutorial-getting-started-autonomous-db/index.html

**16. Oracle Autonomous Data Warehouse detail instructions:**

https://docs.oracle.com/en/cloud/paas/autonomous-data-warehouse-cloud/tutorial-getting-started-autonomous-db-adw/index.html