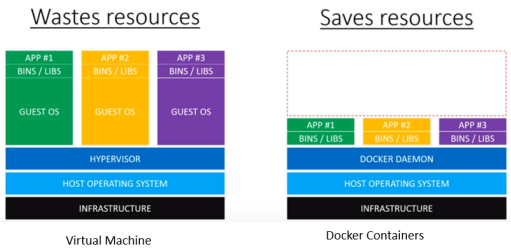
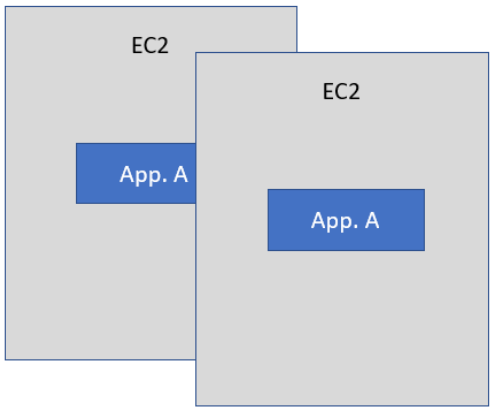
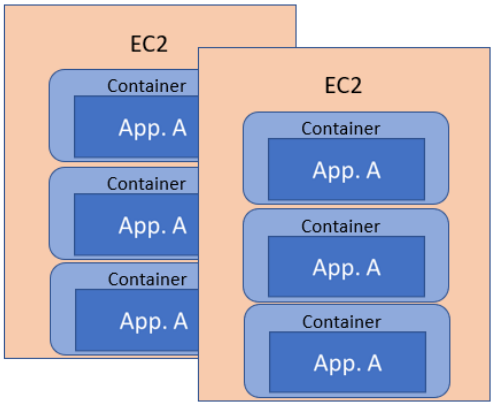
Name: PRADEEP MEDAGIRI

As you have learned, you can run an application in a virtual machine or in a container, and container is more efficient.





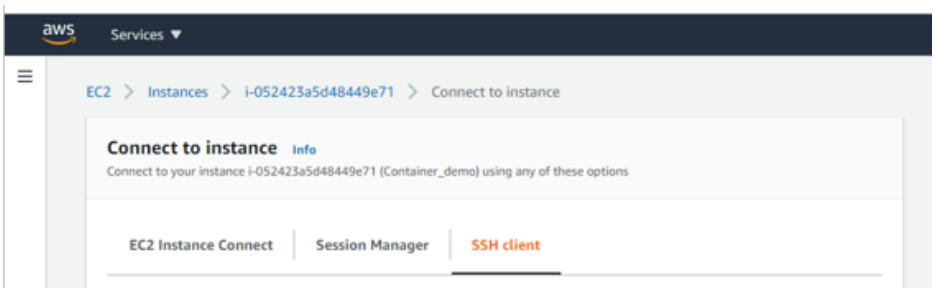


There are 2 parts in this homework:

> Part 1: create and deploy an nginx container manually (create Dockerfile, build docker image, run container).
>
> Part 2: Push the docker image created in part 1 to ECR.

Create an IAM user with AdministratorAccess policy (full access to AWS services). You should log-in using this IAM user account and work on the homework from it, not from the Root user. DON'T forget to save the information of this account, you can not retrieve them later, and you will need them to complete the homework. If you use your Root user, you will not have some of the needed information. We will use this account next week as well.

I recommend to use "ssh client" instead of EC2 Instance Connect.



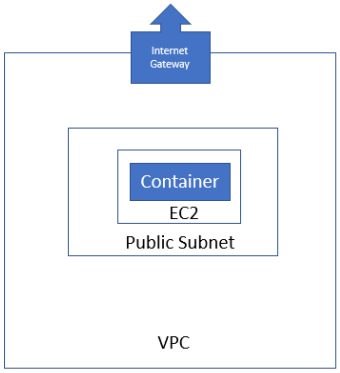
Part 1

From earlier chapters, you have learned how to deploy and run an application in a virtual machine (EC2). In this homework, you will deploy and run an application in a container.

You are to create a static website (**nginx**) that can be accessed from everywhere. You decide to use containers. Containers are to be deployed in an EC2 (default VPC is fine but you can create a customized VPC too). REMEMBER to set inbound rule of the Security Group appropriately.

The static website uses:

- Nginx (just like the in-class demo)
- Shall show "Hi there, this is *your first name and last name*" before the standard nginx print out.



Steps you need to do. If not specified, you can name/determine resource configurations whatever you want.

- Create an EC2 (with the latest ubuntu) and set the proper Security Group. Install the necessary software packages.
- Create index.html
- Create Dockerfile
- Build the docker image and name it yourlastname-nginx
- Create (run) a container.
- Test if the container created from the docker image will print the right print-out by calling the EC2 public IP address from a browser.

Part 2

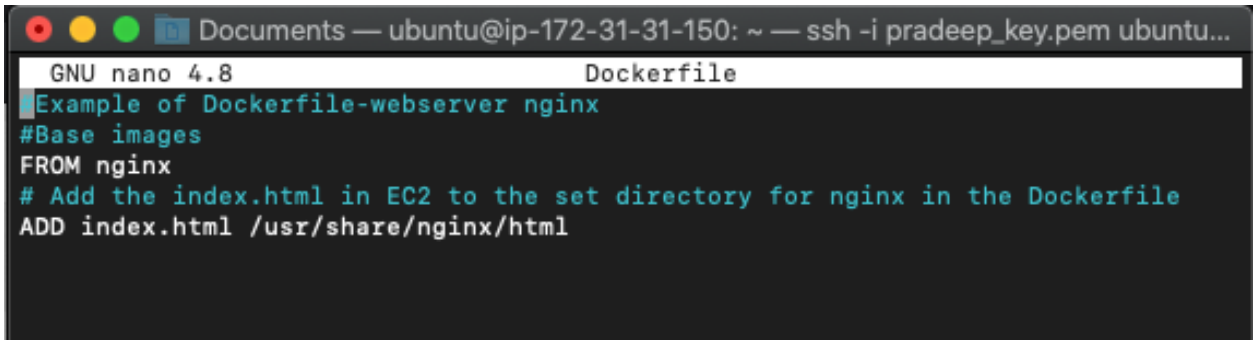
In this part, you will register the docker image you created in part 1 in ECR. NOTE: do not delete, you will use this in next week's Kubernetes homework.

What to submit (do all these after you have completed the homework):

1. Screenshot of the Dockerfile you created (entire file, ensure nothing is truncated). (20%)

```
latest: digest: sha256:91e89437f87aea76d686b6e45c32293697ec0513d862c43c5a2388523204937b size:
1569
[ubuntu@ip-172-31-31-150:~$ ls
Dockerfile  aws  awscliv2.zip  index.html
[ubuntu@ip-172-31-31-150:~$ cat Dockerfile
#Example of Dockerfile-webserver nginx
#Base images
FROM nginx
# Add the index.html in EC2 to the set directory for nginx in the Dockerfile
ADD index.html /usr/share/nginx/html
```

#nano



2. Screenshot of images created ("docker images"). Ensure all information are shown. (20%)

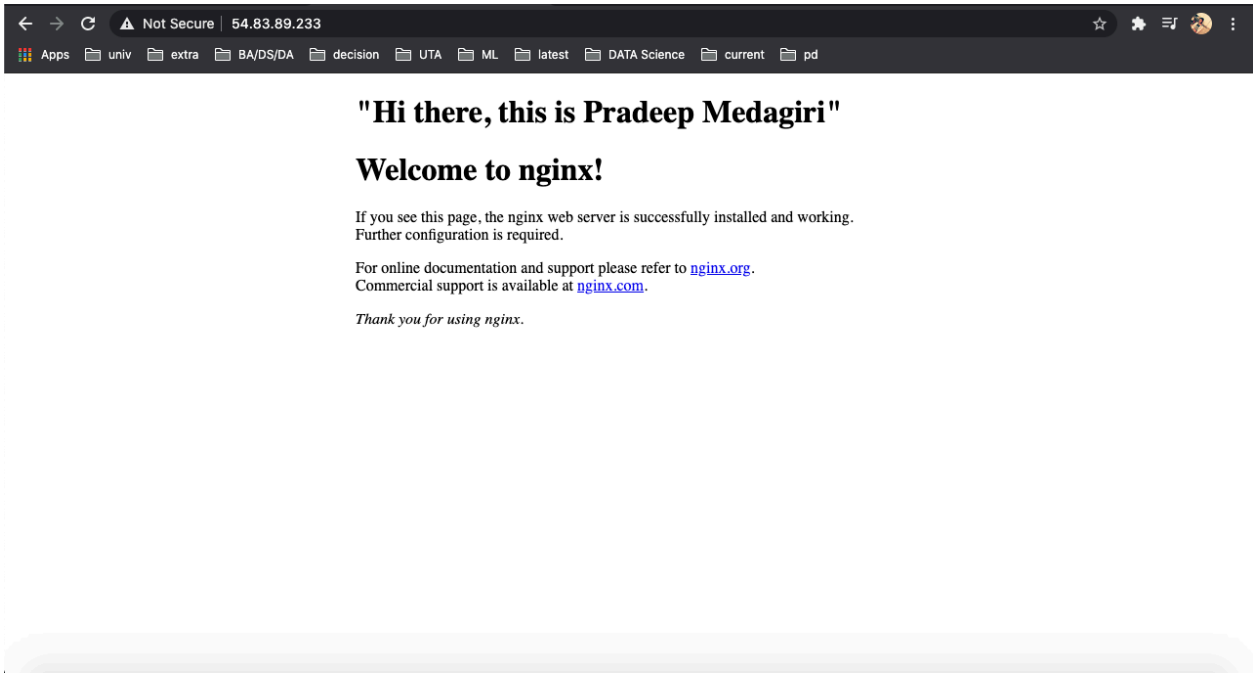
```
# Add the index.html in EC2 to the set directory for nginx in the Dockerfile
ADD index.html /usr/share/nginx/html


ubuntu@ip-172-31-31-150:~$ docker images
REPOSITORY                                                      TAG       IMAGE ID       CREATED          SIZE
medagiri_nginx                                                  latest    b78a8646e6a6   14 minutes ago   133MB
431429019981.dkr.ecr.us-east-1.amazonaws.com/medagiri_nginx     latest    b78a8646e6a6   14 minutes ago   133MB
nginx                                                           latest    c39a868aad02   5 days ago       133MB
ubuntu@ip-172-31-31-150:~$
```

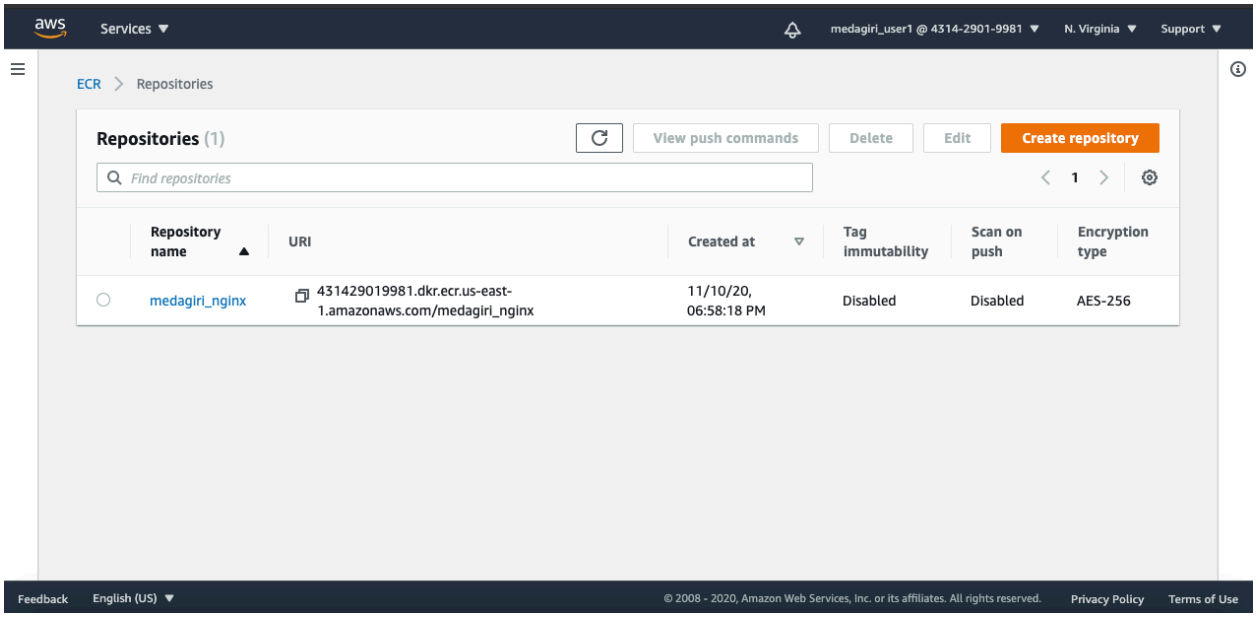
3. Screenshot of containers created ("docker ps -a"). Ensure all information are shown. (20%)

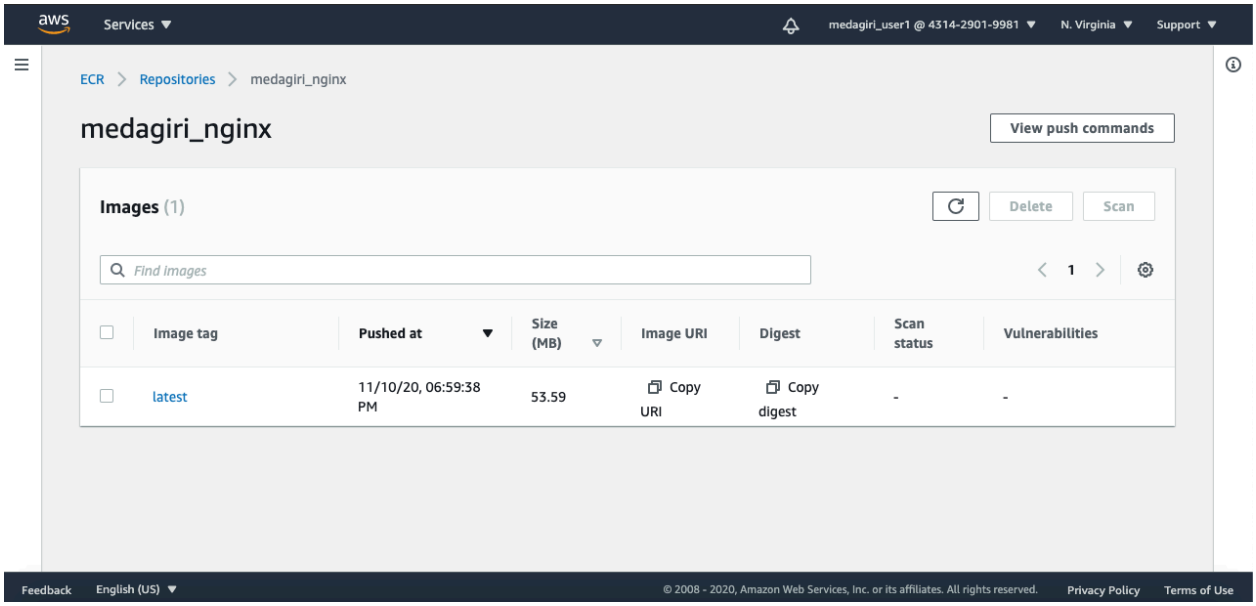
```
nginx                                                           latest    c39a868aad02   5 days ago       133MB
[ubuntu@ip-172-31-31-150:~$ docker ps -a
CONTAINER ID   IMAGE              COMMAND                  CREATED          STATUS          PORTS                NAMES
09c6df671604   medagiri_nginx     "/docker-entrypoint.…"   12 minutes ago   Up 12 minutes   0.0.0.0:80->80/tcp   crazy
_chebyshev
ubuntu@ip-172-31-31-150:~$
```

4. Screenshot when you call the EC2 IP address from a browser. (20%)



5. Show the screenshot of the ECR with the image you pushed. (20%)




aws
Services
medagiri_user1 @ 4314-2901-9981
N. Virginia
Support
ECR
Repositories
medagiri_nginx
medagiri_nginx
View push commands
Images (1)
Delete
Scan
Find images
1
Image tag
Pushed at
Size (MB)
Image URI
Digest
Scan status
Vulnerabilities
latest
11/10/20, 06:59:38 PM
53.59
Copy URI
Copy digest
-
-
Feedback
English (US)
© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use