

CS 4740- Cloud Computing

Programming Assignment 4

Goal of this PA

- Gain hand-on experience with the Docker
- Understand the terminology of Docker
- Know how to create image and create container based on image
- Know how to use container service on Amazon AWS
- Know how to build multi-container environment

Comparison of platform virtualization software

Hardware virtualization (hypervisors)	Native	Adeos • CP/CMS • Hyper-V • KVM (Red Hat Enterprise Virtualization) • LDom​s / Oracle VM Server for SPARC • LynxSecure • PikeOS • Proxmox VE • SIMMON • vCloud) • VMware Infrastructure • Xen (Oracle VM Server for x86 • XenClient) • XtratuM • z/VM	
	Hosted	Specialized	Basilisk II • bhyve • Bochs • Cooperative Linux • DOSBox • DOSEMU • PCem • PikeOS • SheepShaver • SIMH • Windows on Windows (VirtualBox)
		Independent	Microsoft Virtual Server • Parallels Workstation • Parallels Desktop for Mac • Parallels Server for Mac • PearPC • QEMU • VirtualBox • Virtual Machine Monitor • VMware Server • VMware Workstation • <u>Windows Virtual PC</u>
	Tools	Ganeti • oVirt • Virtual Machine Manager	
OS-level virtualization	cgroups-based (lmcftfy • Linux-VServer • LXC • Docker • OpenVZ) • FreeBSD jail • iCore Virtual Accounts • Kubernetes • Linux namespaces • Solaris Containers • Workload		
Desktop virtualization	Citrix XenApp • Citrix XenDesktop • Remote Desktop Services • VMware Horizon View • Uteo Open Virtual Desktop		
Application virtualization	Ceedo • Citrix XenApp • Dalvik • InstallFree • Microsoft App-V • Remote Desktop Services • Spoon • Symantec Workspace Virtualization • VMware ThinApp • ZeroVM		
Network virtualization	Distributed Overlay Virtual Ethernet (DOVE) • NVGRE • Open vSwitch • Virtual security switch • Virtual Extensible LAN (VXLAN)		

See also: List of emulators

Prerequisite

- Do NOT use Windows OS for this PA. If you have Windows PC only, install VirtualBox and create a VM with Linux (e.g., Ubuntu 14.04) installed (20GB disk size and 3GB memory are enough).

Virtualbox: <https://www.virtualbox.org/wiki/Downloads>

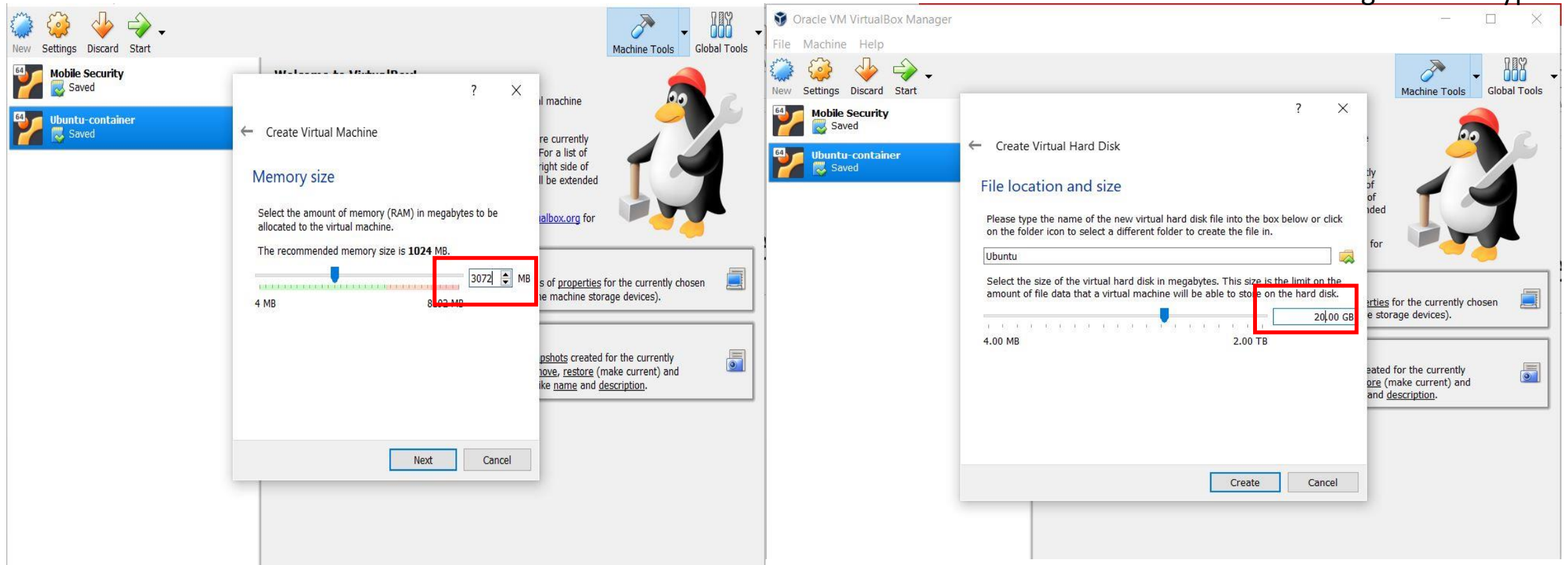
Ubuntu : <https://ubuntu.com/download/desktop>

mage: Sign up for an account in <https://hub.docker.com/>

- Also create an AWS account if you have not created one.

Illustration of key steps VM in Virtual box: Recommended size of the virtual disk

Choose fixed size disk image
after choosing VDI disk type



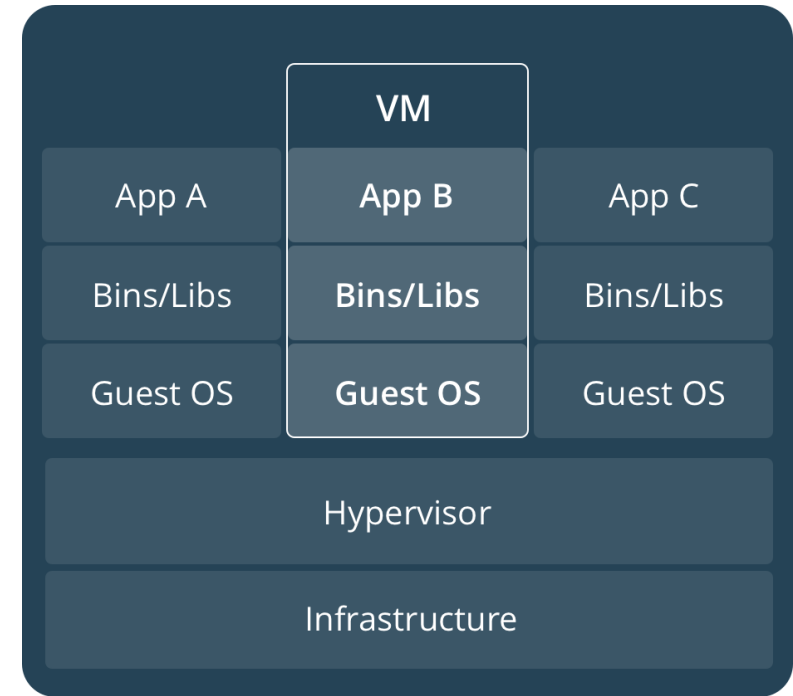
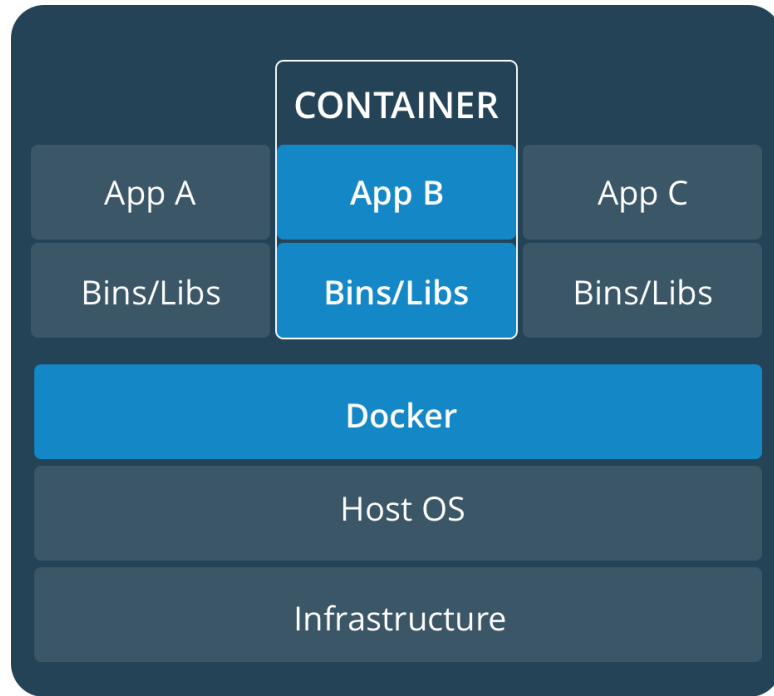
Steps

- Follow the tutorial given in:

<https://docker-curriculum.com/>

Read through all text and finish all steps 1.1-3.4 (inclusive). Please see file “P4- TutorialIndex.docx” for the step indices.

Also check the notes in the tutorial document.



Steps 1.1 How to create a container

Step 1.0- Installing Docker

- Install docker using repository
- Mac: <https://docs.docker.com/docker-for-mac/install/>
- Linux: <https://docs.docker.com/install/linux/docker-ce/ubuntu/>

After installation commands to execute:

```
docker pull busybox
```

```
docker images
```

```
docker run busybox echo "hello from busybox"
```

-- Also other basic command(s) from the tutorial

Deliverable 1

- Step 1.1

Command: `docker ps -a`

Note: if you get an error like permission denied, execute all the commands using `sudo`

```
tanmoy@tanmoy-VirtualBox:~$ sudo docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED
5114da4fb605	busybox	"echo 'hello from bu..."	50 seconds ago
09d8d399fa7c	busybox	"sh"	4 minutes ago
82d446ca2ebe	busybox	"sh"	18 hours ago
73d1fddeecef	hello-world	"/hello"	19 hours ago

tanmoy@tanmoy-VirtualBox:~\$

Steps 2.0- WEBAPPS WITH DOCKER

- Images - The blueprints of applications which form the basis of containers.
- Containers - Created from Docker images and run the actual application.

Step 2.0- WEBAPPS WITH DOCKER

- Sub-steps for 2.1:

- i) `docker run --rm prakhar1989/static-site`

- ii) **Ctrl+C to stop the container.**

- iii) `docker run -d -P --name static-site
prakhar1989/static-site`

- iv) `docker port static-site`

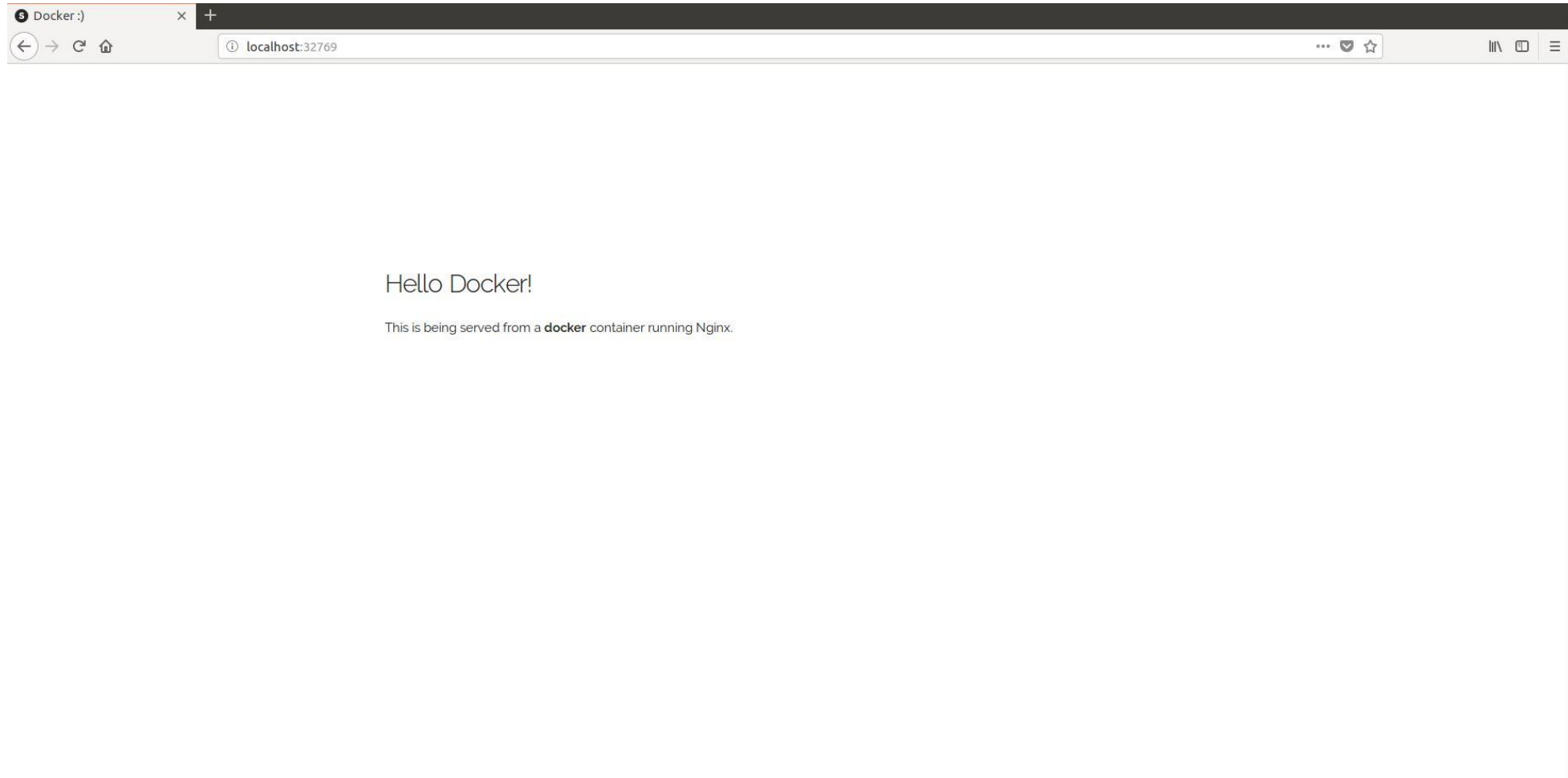
- v) open <http://localhost:32769> in your browser (Deliverable 2).

- vi) You can also run

- `docker run -p 8888:80 prakhar1989/static-site`

- For specifying a custom port (8888).

Deliverable 2



Step 2.0- WEBAPPS WITH DOCKER

- Sub-steps for 2.2:

Command: docker images (Deliverable 3)

```
tannoy@tannoy-VirtualBox:~$ sudo docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED
SIZE			
busybox	latest	83aa35aa1c79	3 weeks ago
1.22MB			
hello-world	latest	fce289e99eb9	15 months ago
1.84kB			
prakhar1989/static-site	latest	f01030e1dcf3	4 years ago
134MB			

Step 2.0- WEBAPPS WITH DOCKER

- Sub-steps for 2.3 and 2.4:

i) `git clone https://github.com/prakhar1989/docker-curriculum.git`

ii) `cd docker-curriculum/flask-app`

iii) The Dockerfile is already in the flask-app directory, you do not need to do anything

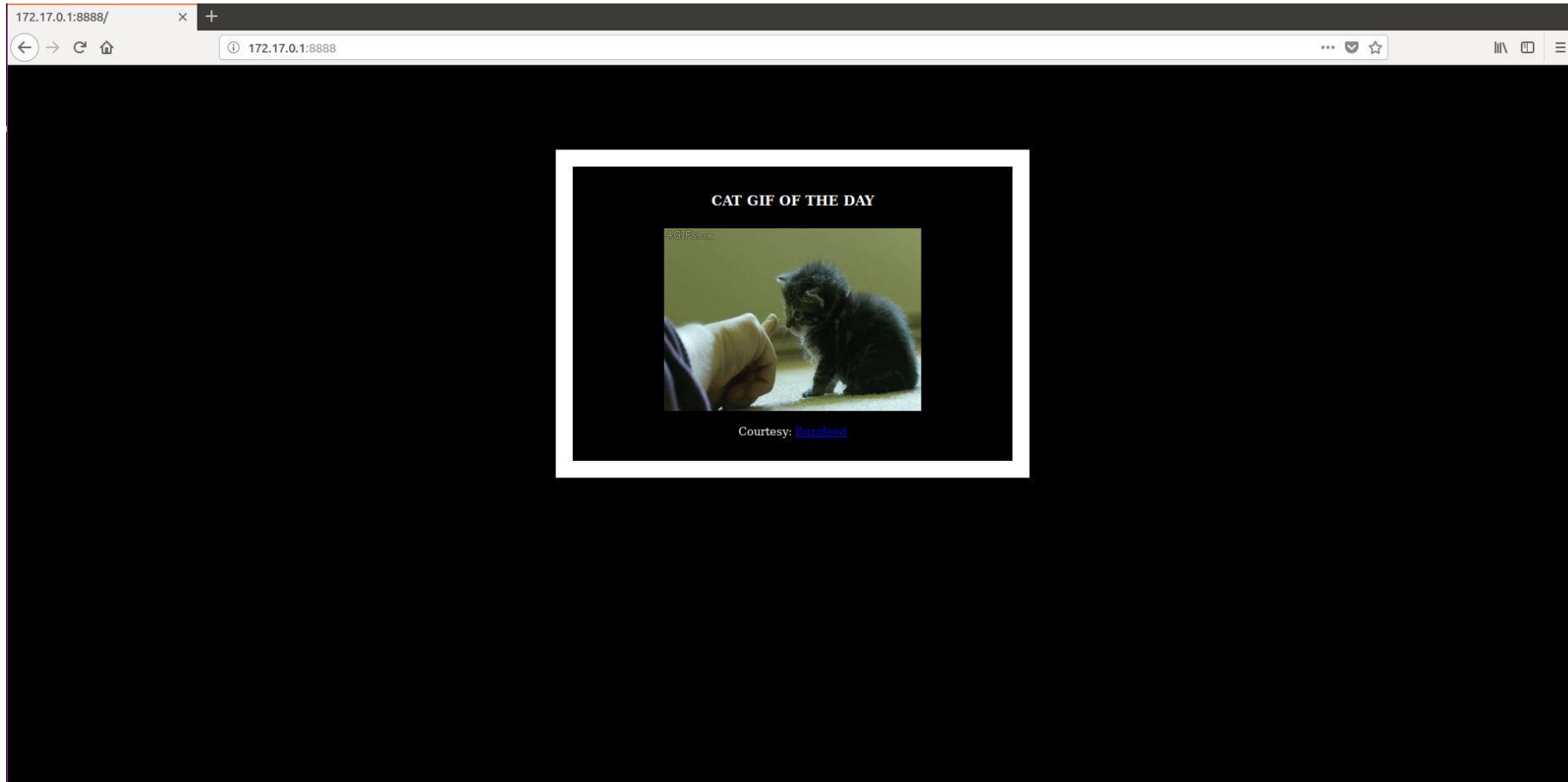
iv) `docker build -t yourusername/catnip .`

Note: Don't forget the period sign. Yourusername refers to docker hub username in this and all future references

v) `docker run -p 8888:5000 yourusername/catnip`

vi) open `http://localhost:8888` in your browser (Deliverable 4).

Deliverable 4



Step 2.0- WEBAPPS WITH DOCKER

- Sub-steps for 2.5:

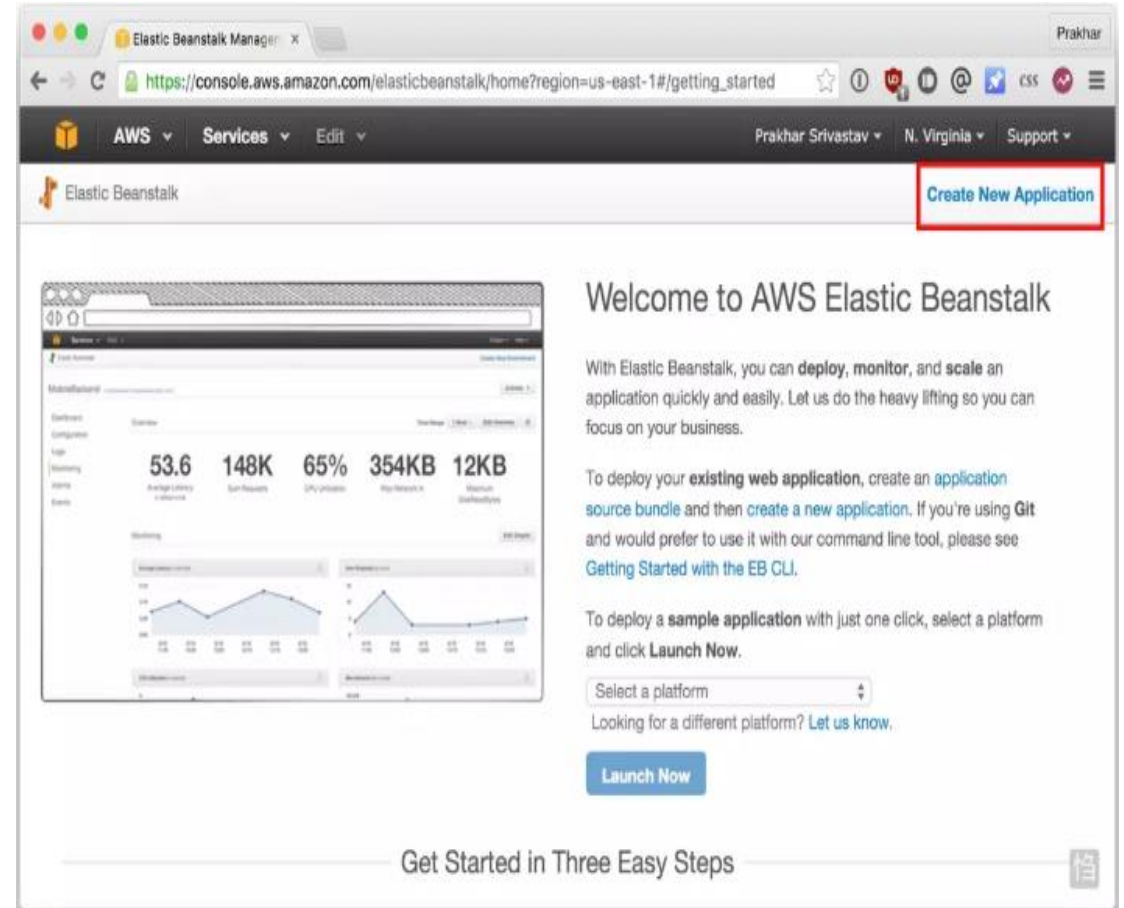
- i) `docker login`

- ii) Username and password from your docker hub account.

- iii) `docker push yourusername/catnip`

Step 2.0- WEBAPPS WITH DOCKER

- Sub-steps for 2.5 (AWS Beanstalk)
- i) Login to your AWS console.
 - ii) Click to Elastic Beanstalk console [link](#)
 - iii) Click on "Create New Application" in the top right
 - iv) Give your app a memorable (but unique) name and provide an (optional) description



Step 2.0- WEBAPPS WITH DOCKER

- Sub-steps for 2.5

v) In the **New Environment** screen, create a new environment and choose the **Web Server Environment**.

vi) Fill in the environment information by choosing a domain. This URL is what you'll share with your friends so make sure it's easy to remember.

vii) Under base configuration section. Choose *Docker* from the *predefined platform*.

Base configuration

Platform ☒ Preconfigured platform
Platforms published and maintained by AWS Elastic Beanstalk.

Docker

☐ Custom platform
Platforms created and owned by you. [Learn more](#)

-- Choose a custom platform --

Application code ☐ Sample application
Get started right away with sample code.

☐ Existing version
Application versions that you have uploaded for trial.

-- Choose a version --

☒ Upload your code
Upload a source bundle from your computer or copy one from Amazon S3.

trial-source

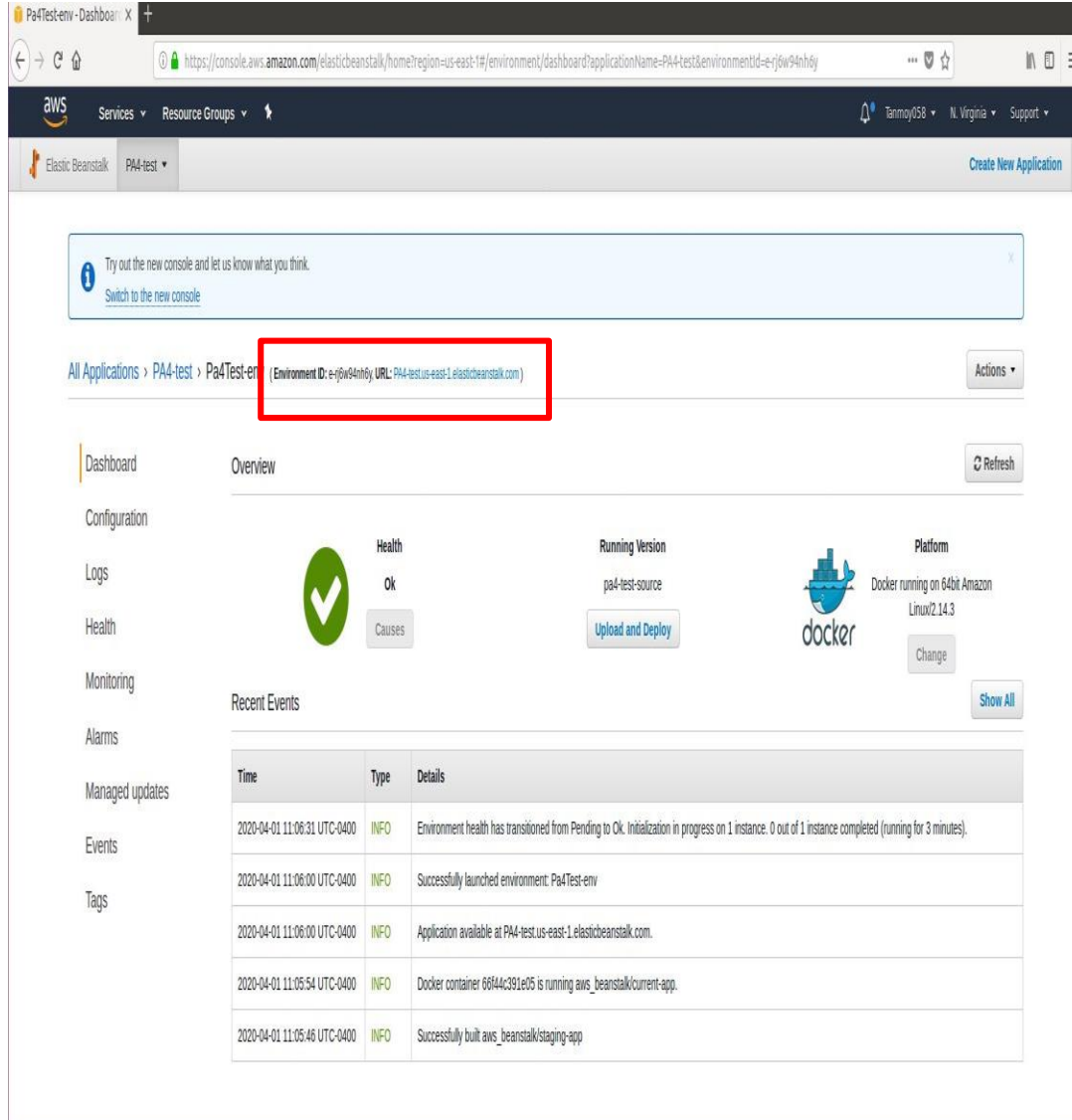
Step 2.0- WEBAPPS WITH DOCKER

- Sub-steps for 2.5 (AWS Beanstalk)

viii) Now we need to upload our application code. But since our application is packaged in a Docker container, we just need to tell EB about our container. Open the `Dockerrun.aws.json` file located in the flask-app folder and edit the Name of the image to your image's name (change the username with docker hub username).

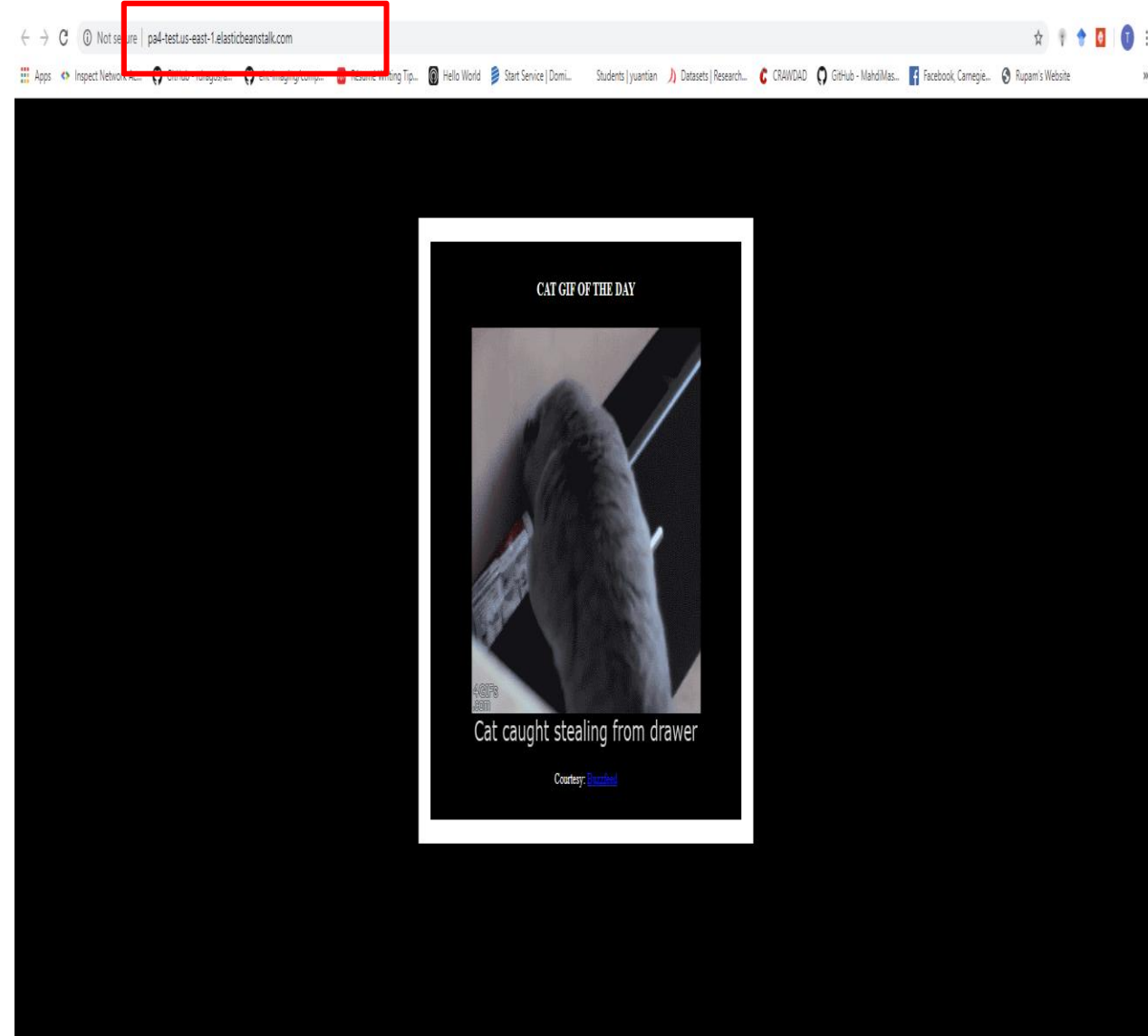
ix) Now click on "Create environment". The final screen that you see will have a few spinners indicating that your environment is being set up. It typically takes around 5 minutes for the first-time setup.

Deliverable 5



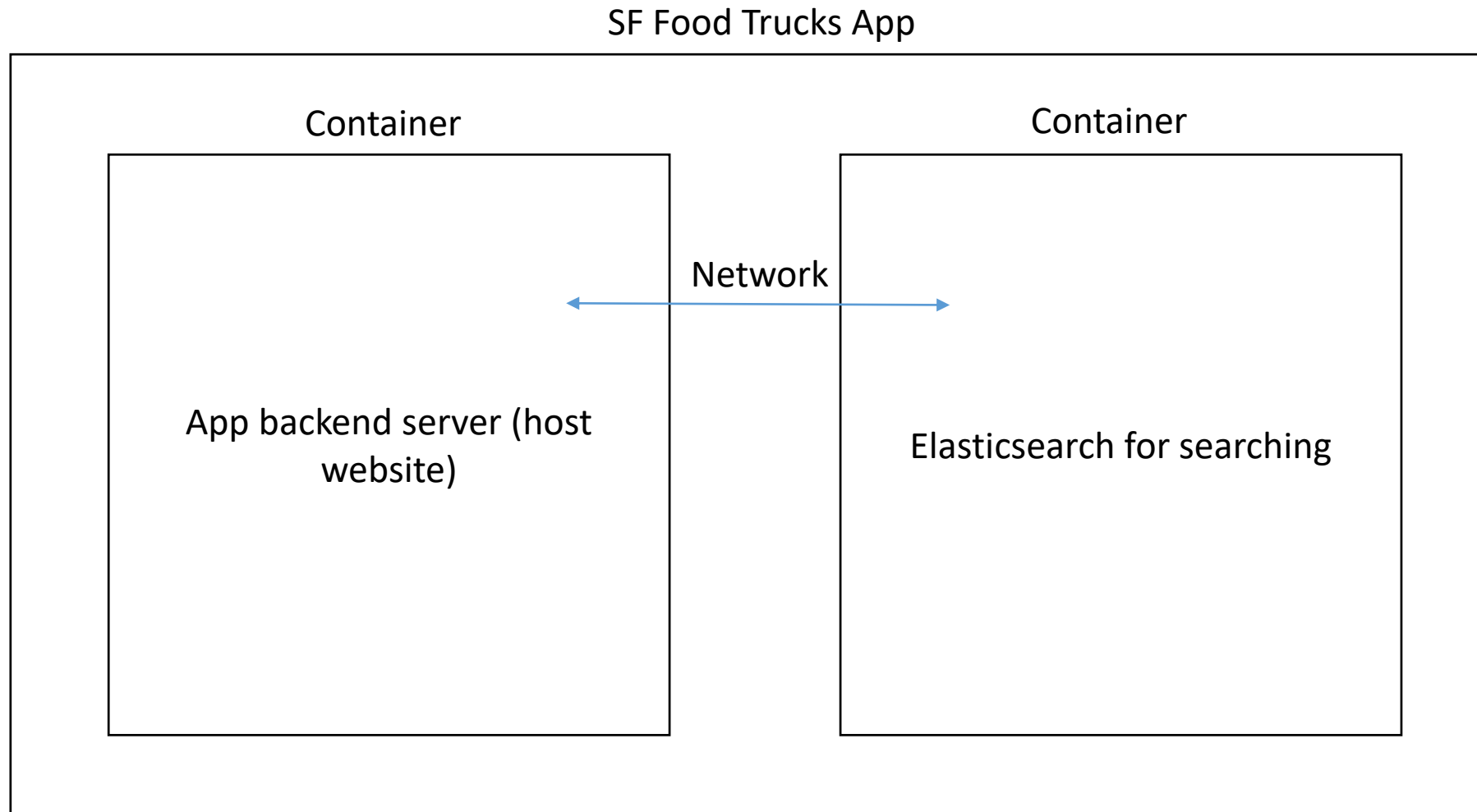
The screenshot shows the AWS Elastic Beanstalk console for the environment 'Pa4Test-env'. The breadcrumb navigation path is 'All Applications > PA4-test > Pa4Test-env'. A red box highlights the environment details: 'Environment ID: e-jfw94nhdy' and 'URL: PA4-test-us-east-1.elasticbeanstalk.com'. The dashboard includes a 'Health' section with a green checkmark and 'Ok' status, a 'Running Version' section with 'pa4-test-source' and an 'Upload and Deploy' button, and a 'Platform' section with 'Docker running on 64bit Amazon Linux/2.14.3' and a 'Change' button. The 'Recent Events' table shows the following data:

Time	Type	Details
2020-04-01 11:06:31 UTC-0400	INFO	Environment health has transitioned from Pending to Ok. Initialization in progress on 1 instance. 0 out of 1 instance completed (running for 3 minutes).
2020-04-01 11:06:00 UTC-0400	INFO	Successfully launched environment: Pa4Test-env
2020-04-01 11:06:00 UTC-0400	INFO	Application available at PA4-test-us-east-1.elasticbeanstalk.com.
2020-04-01 11:05:54 UTC-0400	INFO	Docker container 66f44c391e05 is running aws_beanstalk/current-app.
2020-04-01 11:05:46 UTC-0400	INFO	Successfully built aws_beanstalk/staging-app



The screenshot shows a web browser with the address bar containing 'pa4-test-us-east-1.elasticbeanstalk.com'. Below the browser is a meme titled 'CAT GIF OF THE DAY'. The meme features a video of a cat stealing a small object from a drawer. The text 'Cat caught stealing from drawer' is overlaid on the video. The source is credited to 'Courtesy: Tazbeel'.

Step 3.0- MULTI-CONTAINER ENVIRONMENTS



Step 3.0- MULTI-CONTAINER ENVIRONMENTS

- Sub-steps 3.1

- i) `git clone https://github.com/prakhar1989/FoodTrucks`
- ii) `cd FoodTrucks` (All the commands after this are run inside this directory)
- iii) `docker search elasticsearch`
- iv) `docker pull
docker.elastic.co/elasticsearch/elasticsearch:6.3.2`
- v) `docker run -d --name es -p 9200:9200 -p 9300:9300 -e
"discovery.type=single-node"
docker.elastic.co/elasticsearch/elasticsearch:6.3.2`
- vi) `docker build -t yourusername/foodtrucks-web .`

Note: Step vi fails because of not being able to connect to elastic search.

Step 3.0- MULTI-CONTAINER ENVIRONMENTS

- Sub-steps 3.2

vii) `docker container stop es`

viii) `docker container rm es`

ix) `docker build -t yourusername/foodtrucks-web .`

x) `docker network create foodtrucks-net`

xi) `docker run -d --name es --net foodtrucks-net -p 9200:9200 -p 9300:9300 -e "discovery.type=single-node" docker.elastic.co/elasticsearch/elasticsearch:6.3.2`

xii) `docker run -d --net foodtrucks-net -p 5000:5000 --name foodtrucks-web yourusername/foodtrucks-web`

Note: you can use the `docker network inspect foodtrucks-net` to check whether elastic search container is connected to the desired bridge

Step 3.0- MULTI-CONTAINER ENVIRONMENTS

- Sub-steps 3.3:

If you're running Mac, Docker Compose is already installed as it comes in the Docker Toolbox. Linux users can easily get their hands on Docker Compose by following the instructions on the docs. Since Compose is written in Python, you can also simply do `pip install docker-compose`

Then check the version with the following command

```
docker-compose --version
```


Step 3.0- MULTI-CONTAINER ENVIRONMENTS

- Sub-steps 3.3:

- i) `docker stop es foodtrucks-web`

- ii) `docker rm es foodtrucks-web`

- iii) `docker-compose up` (**Deliverable 6– get the screenshot**)

- iv) `docker-compose down -v`

Deliverable 6

```
Creating es
Creating foodtrucks_web_1
Attaching to es, foodtrucks_web_1
es      | OpenJDK 64-Bit Server VM warning: Option UseConcMarkSweepGC was deprecated in version 9.0 and will likely be removed in a future rele
ase.
es      | [2018-11-20T06:21:41,899][INFO ][o.e.n.Node               ] [] initializing ...
es      | [2018-11-20T06:21:42,145][INFO ][o.e.e.NodeEnvironment ] [QMyhyPPp] using [1] data paths, mounts [[/usr/share/elasticsearch/data (/
dev/mapper/ubuntu--vg-root)]], net usable space [10.6gb], net total space [20gb], types [ext4]
es      | [2018-11-20T06:21:42,146][INFO ][o.e.e.NodeEnvironment ] [QMyhyPPp] heap size [1015.6mb], compressed ordinary object pointers [true
]
es      | [2018-11-20T06:21:42,155][INFO ][o.e.n.Node               ] [QMyhyPPp] node name derived from node ID [QMyhyPPp15cmEVXJSn79KLA]; set [n
ode.name] to override
es      | [2018-11-20T06:21:42,155][INFO ][o.e.n.Node               ] [QMyhyPPp] version[6.3.2], pid[1], build[default/tar/053779d/2018-07-20T05
:20:23.451332Z], OS[Linux/4.4.0-31-generic/amd64], JVM["Oracle Corporation"/OpenJDK 64-Bit Server VM/10.0.2/10.0.2+13]
es      | [2018-11-20T06:21:42,157][INFO ][o.e.n.Node               ] [QMyhyPPp] JVM arguments [-Xmsig, -Xmxig, -XX:+UseConcMarkSweepGC, -XX:CMS
InitiatingOccupancyFraction=75, -XX:+UseCMSInitiatingOccupancyOnly, -XX:+AlwaysPreTouch, -Xssin, -Djava.awt.headless=true, -Dfile.encoding=UTF
-8, -Djna.nosys=true, -XX:-OmitStackTraceInFastThrow, -Dio.netty.noUnsafe=true, -Dio.netty.noKeySetOptimization=true, -Dio.netty.recycler.maxC
apacityPerThread=0, -Dlog4j.shutdownHookEnabled=false, -Dlog4j2.disable.jmx=true, -Djava.io.tmpdir=/tmp/elasticsearch.GJPA3Rz, -XX:+HeapDumpO
nOutOfMemoryError, -XX:HeapDumpPath=data, -XX:ErrorFile=logs/hs_err_pid%p.log, -Xlog:gc*,gc+age=trace,safepoint:file=logs/gc.log:utctime,pid,t
ags:filecount=32,filesize=64M, -Djava.locale.providers=COMPAT, -XX:UseAVX=2, -Des.cgroups.hierarchy.override=/, -Des.path.home=/usr/share/elas
ticsearch, -Des.path.conf=/usr/share/elasticsearch/config, -Des.distribution.flavor=default, -Des.distribution.type=tar]
es      | [2018-11-20T06:21:48,559][INFO ][o.e.p.PluginsService       ] [QMyhyPPp] loaded module [aggs-matrix-stats]
es      | [2018-11-20T06:21:48,560][INFO ][o.e.p.PluginsService       ] [QMyhyPPp] loaded module [analysis-common]
es      | [2018-11-20T06:21:48,560][INFO ][o.e.p.PluginsService       ] [QMyhyPPp] loaded module [ingest-common]
es      | [2018-11-20T06:21:48,561][INFO ][o.e.p.PluginsService       ] [QMyhyPPp] loaded module [lang-expression]
es      | [2018-11-20T06:21:48,561][INFO ][o.e.p.PluginsService       ] [QMyhyPPp] loaded module [lang-mustache]
es      | [2018-11-20T06:21:48,562][INFO ][o.e.p.PluginsService       ] [QMyhyPPp] loaded module [lang-painless]
es      | [2018-11-20T06:21:48,562][INFO ][o.e.p.PluginsService       ] [QMyhyPPp] loaded module [mapper-extras]
es      | [2018-11-20T06:21:48,562][INFO ][o.e.p.PluginsService       ] [QMyhyPPp] loaded module [parent-join]
es      | [2018-11-20T06:21:48,563][INFO ][o.e.p.PluginsService       ] [QMyhyPPp] loaded module [percolator]
es      | [2018-11-20T06:21:48,563][INFO ][o.e.p.PluginsService       ] [QMyhyPPp] loaded module [rank-eval]
```

Step 3.0- MULTI-CONTAINER ENVIRONMENTS

Sub-step 3.4:

i) Install Amazon ECS CLI [Link](#)

Mac: `sudo curl -o /usr/local/bin/ecs-cli https://amazon-ecs-cli.s3.amazonaws.com/ecs-cli-darwin-amd64-latest`

Linux: `sudo curl -o /usr/local/bin/ecs-cli https://amazon-ecs-cli.s3.amazonaws.com/ecs-cli-linux-amd64-latest`

ii) Verify

Mac: `curl -s https://amazon-ecs-cli.s3.amazonaws.com/ecs-cli-darwin-amd64-latest.md5 && md5 -q /usr/local/bin/ecs-cli`

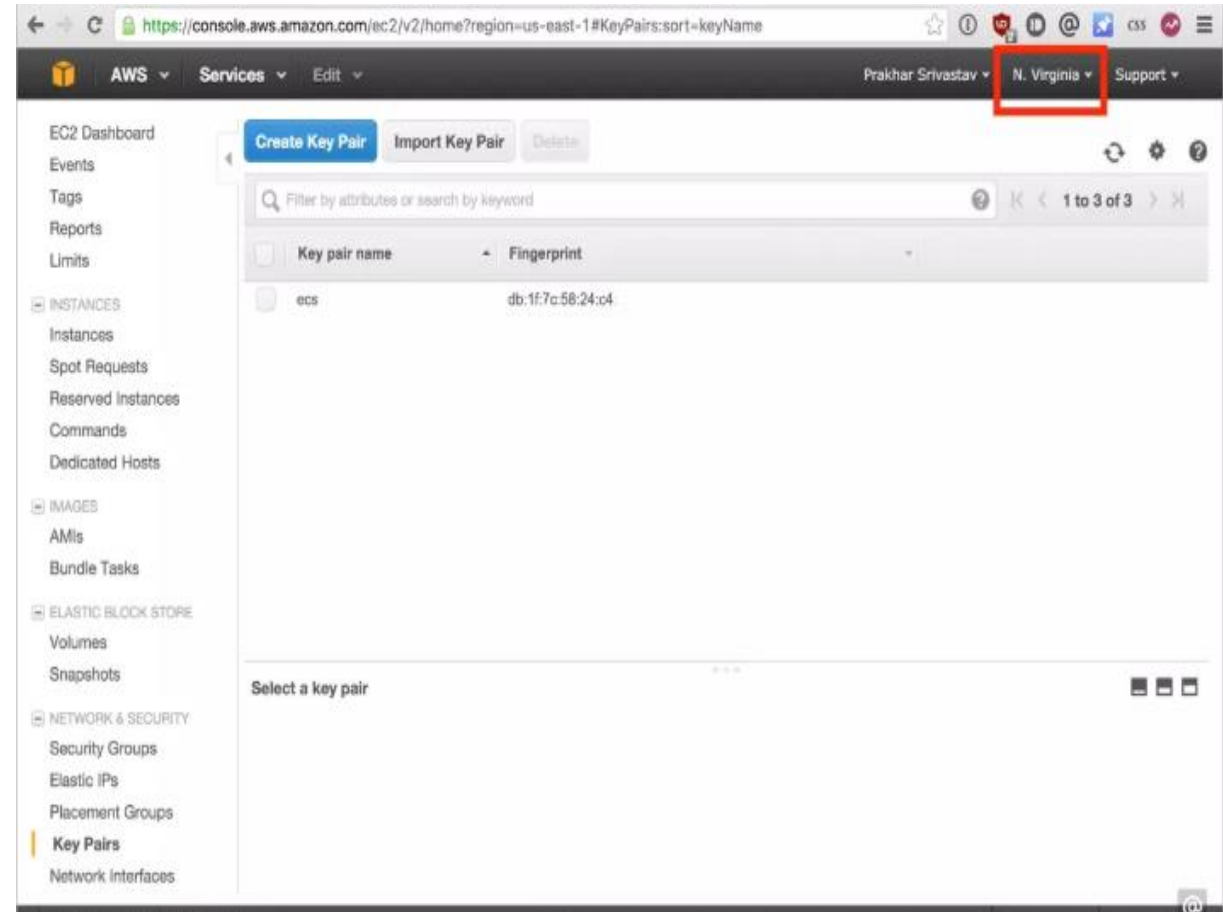
Linux: `echo "$(curl -s https://amazon-ecs-cli.s3.amazonaws.com/ecs-cli-linux-amd64-latest.md5)
/usr/local/bin/ecs-cli" | md5sum -c -`

Step 3.0- MULTI-CONTAINER ENVIRONMENTS

Sub-step 3.4:

iii) get a keypair which we'll be using to log into the instances. Head over to your [EC2 Console](#) and create a new keypair. Download the keypair and store it in a safe location.

Another thing to note before you move away from this screen is the region name. In my case, I have named my key - ecs and set my region as us-east-1. This is what I'll assume for the rest of this walkthrough.



Step 3.0- MULTI-CONTAINER ENVIRONMENTS

Step 3.4:

iv) Configure [Link](#):

```
ecs-cli configure profile --profile-name  
profile_name --access-key $AWS_ACCESS_KEY_ID --  
secret-key $AWS_SECRET_ACCESS_KEY
```

Profile_name : your desired profile name

\$AWS_ACCESS_KEY_ID and **\$AWS_SECRET_ACCESS_KEY**: access them from your AWS security credentials (as in PA3)

Step 3.0- MULTI-CONTAINER ENVIRONMENTS

v) `ecs-cli configure --region us-east-1 --cluster foodtrucks`

vi) `ecs-cli up --keypair ecs --capability-iam --size 2 --instance-type t2.micro`

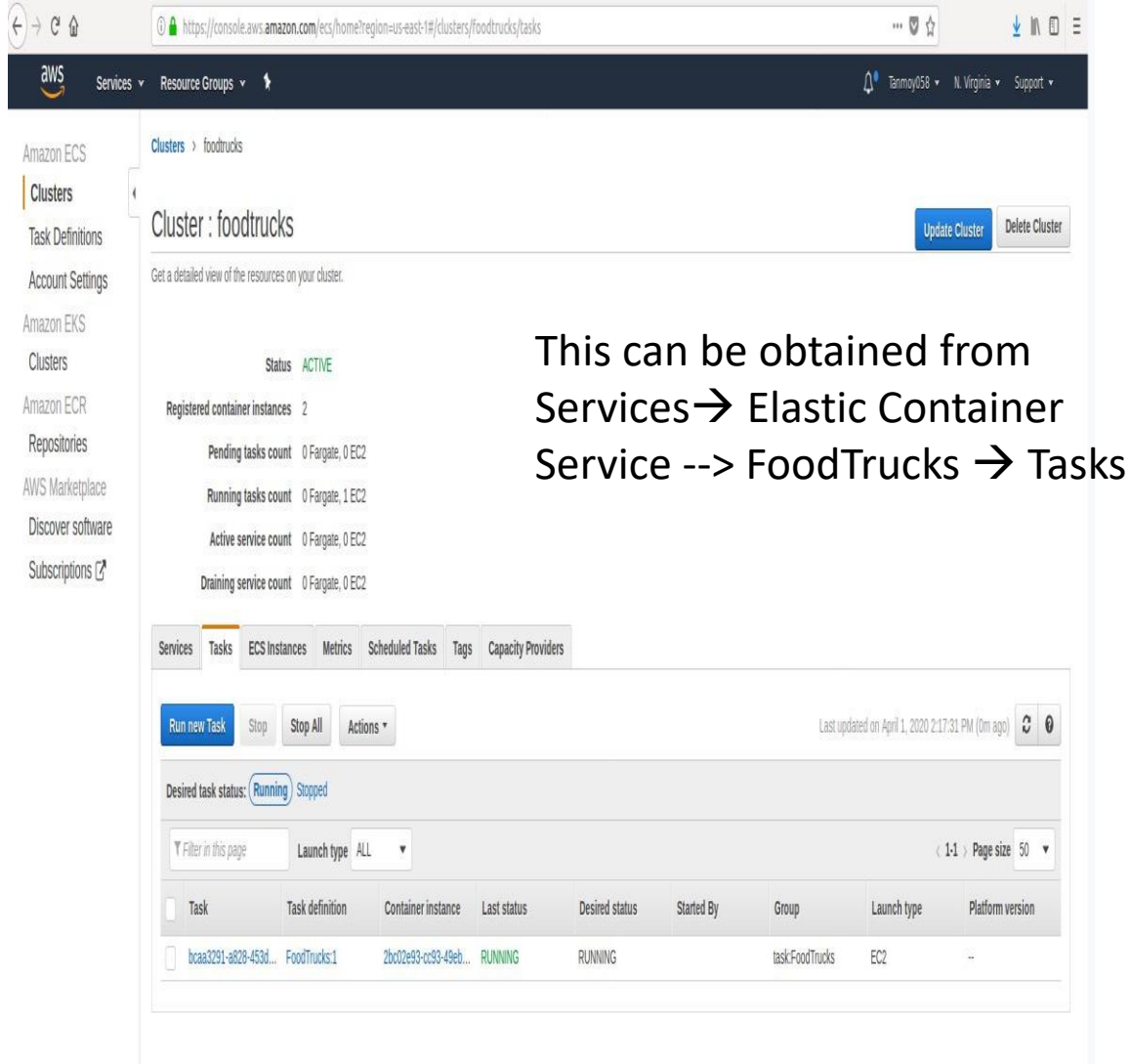
vii) `docker push yourusername/foodtrucks-web`

viii) `ecs-cli compose --file aws-compose.yml up`

ix) `ecs-cli ps`

Type the ip in the browser and observe the output (screenshot in the next slide)

Deliverable 7 (Either one of them is ok)



The screenshot shows the AWS Management Console for the 'foodtrucks' cluster. The cluster is in an 'ACTIVE' state. The 'Registered container instances' section shows 2 instances. The 'Services' tab is selected, and the 'Tasks' section shows a table of tasks.

Cluster: foodtrucks

Status: ACTIVE

Registered container instances: 2

Pending tasks count: 0 Fargate, 0 EC2

Running tasks count: 0 Fargate, 1 EC2

Active service count: 0 Fargate, 0 EC2

Draining service count: 0 Fargate, 0 EC2

Services: Tasks ECS Instances Metrics Scheduled Tasks Tags Capacity Providers

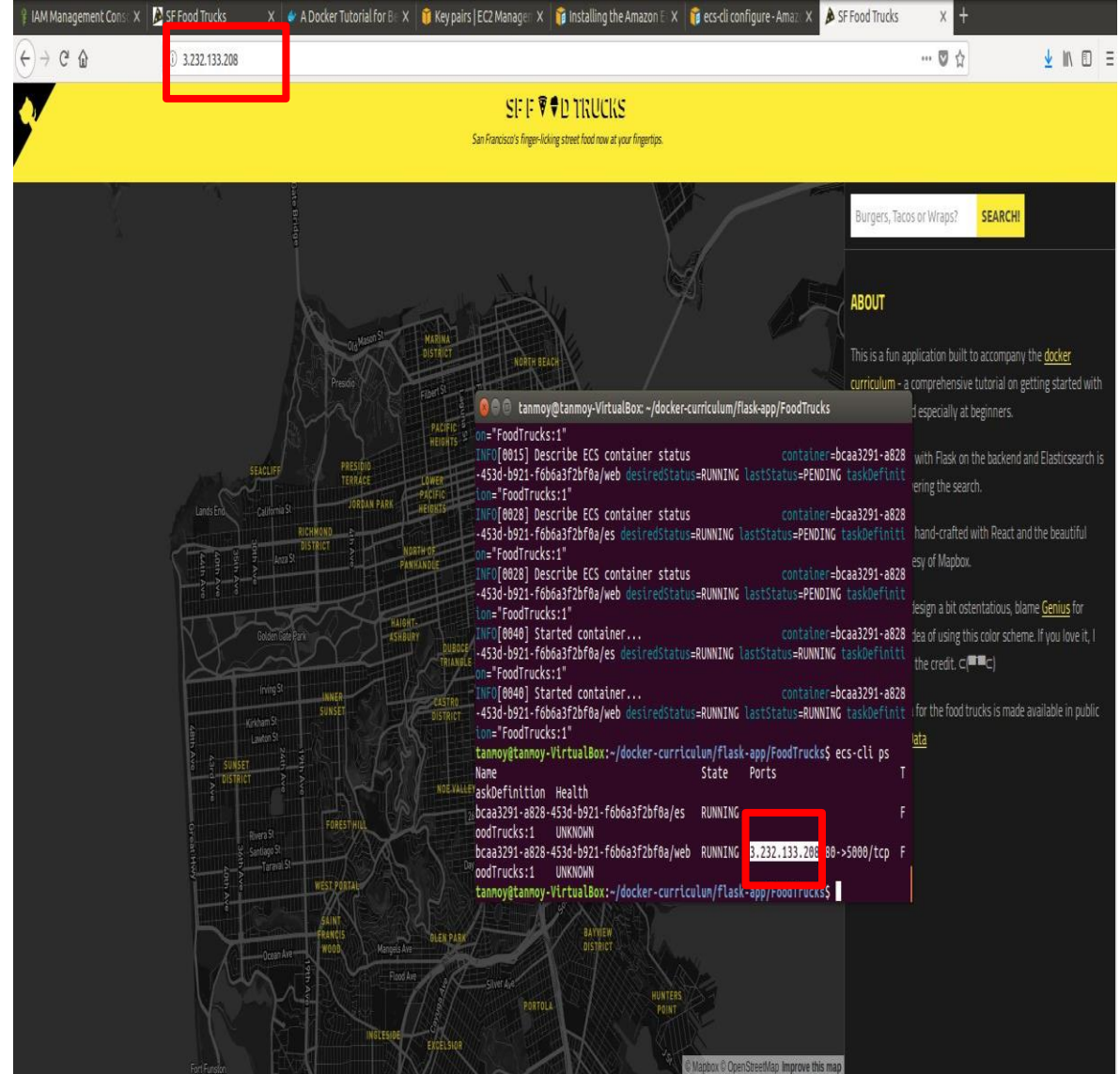
Run new Task Stop Stop All Actions

Desired task status: Running Stopped

Filter in this page Launch type ALL

Task	Task definition	Container instance	Last status	Desired status	Started By	Group	Launch type	Platform version
<input type="checkbox"/>	bcaa3291-a828-453d-...	FoodTrucks:1	2b02e93-c939-49eb...	RUNNING	RUNNING	task-FoodTrucks	EC2	--

This can be obtained from
Services → Elastic Container
Service → FoodTrucks → Tasks



The screenshot shows the SF Food Trucks application running in a browser. The application displays a map of San Francisco with food truck locations. The browser's address bar shows the IP address 3.232.133.208, which is highlighted with a red box. The application's terminal output shows the command to run the application and the resulting output.

3.232.133.208

SF FOOD TRUCKS

San Francisco's finger-licking street food now at your fingertips.

Burgers, Tacos or Wraps? SEARCH!

ABOUT

This is a fun application built to accompany the [docker curriculum](#) - a comprehensive tutorial on getting started with Docker, especially at beginners.

with Flask on the backend and Elasticsearch in the front-end.

hand-crafted with React and the beautiful Mapbox.

design a bit ostentatious, blame [Genius](#) for this.

idea of using this color scheme. If you love it, I'd be happy to hear from you.

for the food trucks is made available in public.

```
tannoy@tannoy-VirtualBox: ~/docker-curriculum/flask-app/FoodTrucks
$ docker-compose up
INFO[0015] Describe ECS container status container=bcaa3291-a828-453d-b921-f0b6a3f2bf0a/web desiredStatus=RUNNING lastStatus=PENDING taskDefinition=FoodTrucks:1
INFO[0028] Describe ECS container status container=bcaa3291-a828-453d-b921-f0b6a3f2bf0a/es desiredStatus=RUNNING lastStatus=PENDING taskDefinition=FoodTrucks:1
INFO[0028] Describe ECS container status container=bcaa3291-a828-453d-b921-f0b6a3f2bf0a/web desiredStatus=RUNNING lastStatus=PENDING taskDefinition=FoodTrucks:1
INFO[0040] Started container... container=bcaa3291-a828-453d-b921-f0b6a3f2bf0a/es desiredStatus=RUNNING lastStatus=RUNNING taskDefinition=FoodTrucks:1
INFO[0040] Started container... container=bcaa3291-a828-453d-b921-f0b6a3f2bf0a/web desiredStatus=RUNNING lastStatus=RUNNING taskDefinition=FoodTrucks:1
tannoy@tannoy-VirtualBox:~/docker-curriculum/flask-app/FoodTrucks$ docker-compose ps
Name                        State      Ports
bcaa3291-a828-453d-b921-f0b6a3f2bf0a/es    RUNNING   80->5080/tcp
bcaa3291-a828-453d-b921-f0b6a3f2bf0a/web    RUNNING   3.232.133.208->5080/tcp
tannoy@tannoy-VirtualBox:~/docker-curriculum/flask-app/FoodTrucks$
```

- Remember to terminate all the services.
 - Delete the cluster after getting the screenshot in the previous slide.
 - Go to elastic beanstalk and delete the application.
 - Also terminate your running EC2 instances.

Questions?