

# Running Docker Apps

**Sang Shin**  
**JPassion.com**  
**“Code with Passion!”**



# Topics

- Run Linux (Alpine, Ubuntu, Fedora, etc.)
- Run Helloworld apps
- Docker containers
- Run Web application with Port binding
- Run Nginx, Tomcat with Port binding
- Volume

**Run Alpine/Ubuntu**

## Lab: Running Alpine or Ubuntu Linux

- `docker pull alpine`
- `docker images`
- `docker run alpine ls -l` (`docker run ubuntu ls l`)
- `docker run alpine echo "hello from alpine"`
- `docker run alpine /bin/sh`
- `docker ps`
- `docker ps -a`
- `docker run -it alpine /bin/sh`
  - > Run it in interactive mode

## Lab: Pulling/Installing a image from the registry

```
C:\Users\sangs>docker pull alpine
```

```
Using default tag: latest
```

```
latest: Pulling from library/alpine
```

```
627beaf3eaaf: Pull complete
```

```
Digest: sha256:58e1a1bb75db1b5a24a462dd5e2915277ea06438c3f105138f97eb53149673c4
```

```
Status: Downloaded newer image for alpine:latest
```

```
C:\Users\sangs>docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
alpine	latest	4a415e366388	3 weeks ago	3.99 MB

## Lab: Pulling/Installing an image from registry

```
C:\Users\sangs>docker pull ubuntu
```

```
Using default tag: latest
```

```
latest: Pulling from library/ubuntu
```

```
d54efb8db41d: Pull complete
```

```
f8b845f45a87: Pull complete
```

```
e8db7bf7c39f: Pull complete
```

```
9654c40e9079: Pull complete
```

```
6d9ef359eaaa: Pull complete
```

```
Digest: sha256:dd7808d8792c9841d0b460122f1acf0a2dd1f56404f8d1e56298048885e45535
```

```
Status: Downloaded newer image for ubuntu:latest
```

```
C:\Users\sangs>docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
alpine	latest	4a415e366388	3 weeks ago	3.99 MB
ubuntu	latest	0ef2e08ed3fa	4 weeks ago	130 MB

## Lab: Run Alpine or Ubuntu image – Creates container instance

```
C:\Users\sangs>docker run alpine ls -l
```

```
total 52
drwxr-xr-x  2 root  root    4096 Mar  3 11:20 bin
drwxr-xr-x  5 root  root    340 Mar 27 22:24 dev
drwxr-xr-x 14 root  root   4096 Mar 27 22:24 etc
...
drwxrwxrwt  2 root  root    4096 Mar  3 11:20 tmp
drwxr-xr-x  7 root  root    4096 Mar  3 11:20 usr
drwxr-xr-x 12 root  root    4096 Mar  3 11:20 var
```

```
C:\Users\sangs>docker run ubuntu echo "Code with Passion!"
```

```
Code with Passion!
```

docker run <image> <command> <arguments>

```
C:\Users\sangs>docker run alpine echo "Code with Passion!" "Love"
```

```
Code with Passion! Love
```

```
C:\Users\sangs>docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
--------------	-------	---------	---------	--------	-------	-------

# Lab: Run an image in interactive mode

```
C:\Users\sangs>docker run -i -t alpine /bin/sh
```

```
/ # ls
```

```
bin  etc  lib  mnt  root  sbin  sys  usr  
dev  home media proc run  srv  tmp  var
```

```
/ # pwd
```

```
/
```

```
/ # cd tmp
```

```
/tmp # ls
```

```
/tmp # touch hello
```

```
/tmp # ls
```

```
hello
```

```
/tmp #
```

-i for interactive mode  
-t for creating a new terminal to interact

In other terminal

```
C:\>docker-machine env
```

```
...
```

```
C:\>@FOR /f "tokens=*" %i IN ('docker-machine env') DO @%i
```

```
C:\>docker ps
```

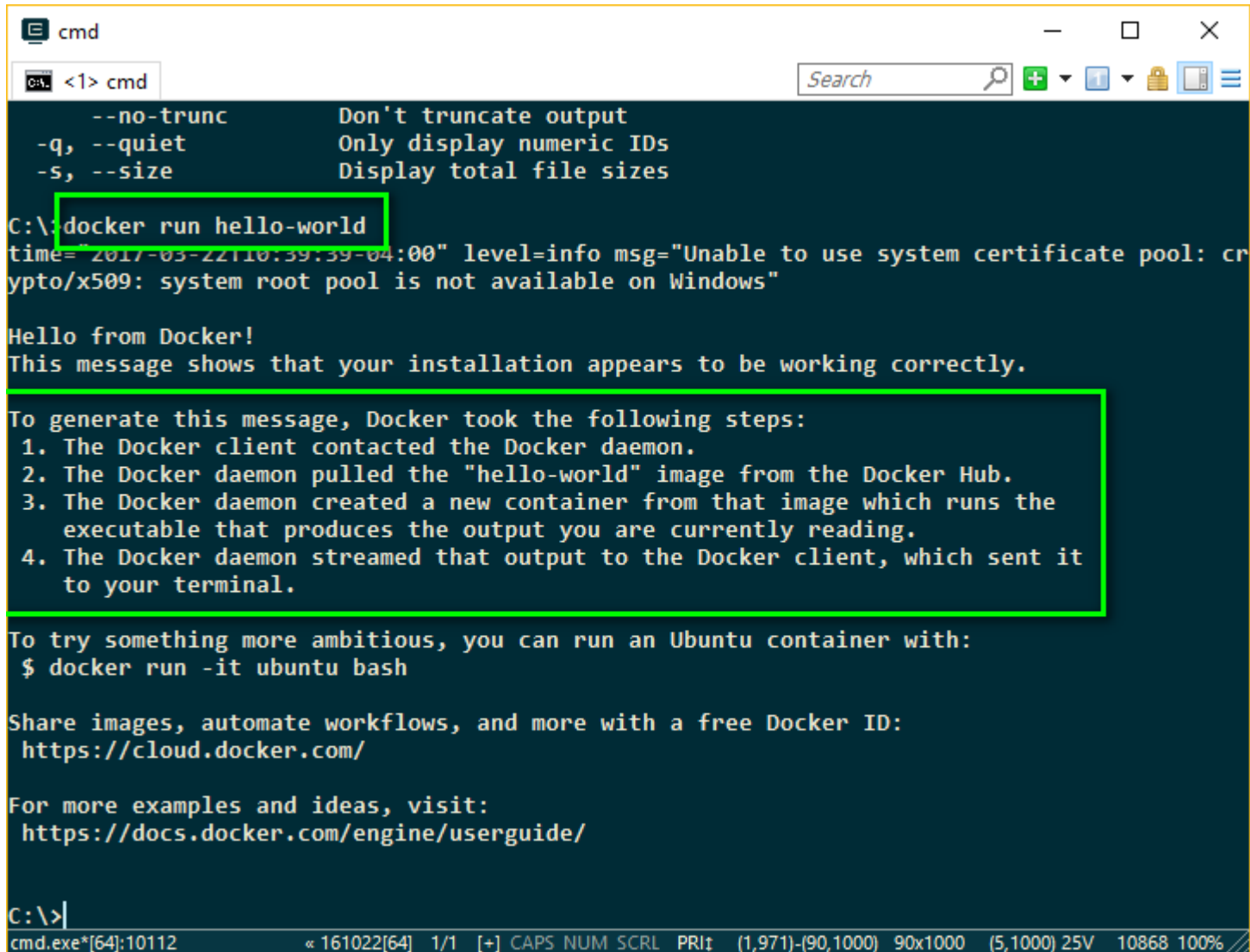
CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
702304e36cfc	alpine	"/bin/sh"	3 minutes ago	Up 3 minutes		pedantic_woznial



The background is a solid orange color with a repeating pattern of vertical, wavy lines. On the right side, there is a large, white, curved shape that resembles a stylized letter 'C' or a partial circle.

# Run Helloworld Apps

# Lab: Run “hello-world” image



The screenshot shows a Windows command prompt window titled 'cmd'. The command prompt has a search bar and several icons on the right. The command `docker run hello-world` has been entered and executed. The output shows a message from Docker indicating that the installation is working correctly. A green box highlights the command and the first part of the output. Another green box highlights a list of steps that Docker took to generate the message. The status bar at the bottom shows the command prompt's internal state.

```
cmd
C:\> docker run hello-world
time="2017-03-22T10:39:39-04:00" level=info msg="Unable to use system certificate pool: crypto/x509: system root pool is not available on Windows"

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
3. The Docker daemon created a new container from that image which runs the
   executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
   to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://cloud.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/engine/userguide/

C:\>
```

cmd.exe\*[64]:10112 « 161022[64] 1/1 [+] CAPS NUM SCRL PRI: (1,971)-(90,1000) 90x1000 (5,1000) 25V 10868 100%

## Lab: Run another Helloworld image

```
C:\>docker run docker/whalesay cowsay "Hello world"
```

```
Unable to find image 'docker/whalesay:latest' locally
```

```
latest: Pulling from docker/whalesay
```

```
e190868d63f8: Pull complete
```

```
909cd34c6fd7: Pull complete
```

```
0b9bfabab7c1: Pull complete
```

```
a3ed95caeb02: Pull complete
```

```
00bf65475aba: Pull complete
```

```
c57b6bcc83e3: Pull complete
```

```
8978f6879e2f: Pull complete
```

```
8eed3712d2cf: Pull complete
```

```
Digest: sha256:178598e51a26abbc958b8a2e48825c90bc22e641de3d31e18aaf55f3258ba93b
```

```
Status: Downloaded newer image for docker/whalesay:latest
```

```
< "Hello world" >
```

```
-----
 \
  \
   \
    ##
   ## ## ## ==
  ## ## ## ## ===
 /##### \ ===
 ~~~ {~~ ~~~~ ~~~ ~~~~ ~~~ / ==~- ~~~
   \_____o_____ /
   \ \   \
    \ \   \
```

# Lab: Run another Helloworld image

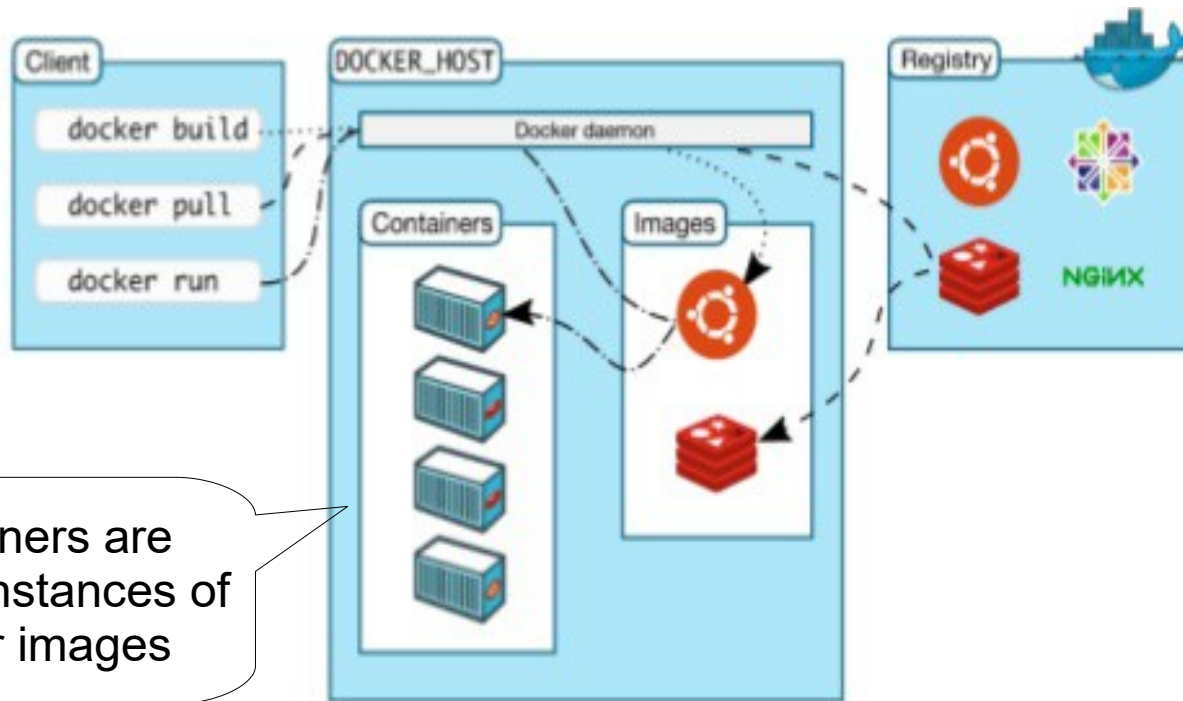
C:\>docker run docker/whalesay cowsay "Code with Passion!"

< "Code with Passion!" >

```
-----
\
 \
  \
   ##
  ## ## ## ==
 ## ## ## ## ===
/##### \ ===
~~~{~~ ~~~~ ~~~ ~~~~ ~~~ / ===- ~~~
 \_____ 0 _____/
  \   \       /
   \   \_____/
```

# Docker Containers

# Docker Containers



Containers are running instances of docker images

## Lab: Run Containers

- `docker ps`
  - > List all running containers
- `docker ps -a`
  - > List all containers (including the ones that exited)
- `docker run ubuntu /bin/echo "Hello, Sang!"`
  - > `docker run <image> <command> <arguments>`
  - > Run a container from the <image> and then run command using arguments
- `docker run -i -t ubuntu /bin/echo "Hello, Sang!"`
  - > Run container in interface mode (-i) and create a terminal to interact with (-t)

## Lab: Run Containers in the Detached Mode (Background)

- `docker run -d ubuntu /bin/sh <some background operation>`
  - > Run container in detached mode (background)
- `docker attach <container-id>`
  - > Attach to the container that is running in the detached mode



## Lab: Run Container in Detached Mode

```
C:\>docker run -d ubuntu /bin/sh -c "while true; do echo $(date); sleep 10; done"  
413dc7c0e672fc4410bea1b8dca6d256e414c3a86ed9886bbcd72a6cb41e823f
```

```
C:\>docker ps  
413dc7c0e672      ubuntu  "/bin/sh -c 'while...'  25 seconds ago Up 24 seconds
```

```
C:\>docker logs 413dc7c0e672fc4410bea1b8dca6d256e414c3a86ed9886bbcd72a6cb41e823f  
Wed Mar 22 18:55:01 UTC 2017  
Wed Mar 22 18:55:11 UTC 2017  
Wed Mar 22 18:55:21 UTC 2017  
Wed Mar 22 18:55:31 UTC 2017
```

## Lab: Container Status “docker ps -a”

```
C:\>docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
7c3336dfd309	ubuntu	"/bin/sh -c 'while..."	5 minutes ago	Exited (137)	About a minute ago
loving_ride4425edd202eb	ubuntu	"/bin/sh"	7 minutes ago	Exited (2)	6 minutes ago
kind_murdock					
dd69d7f85fec	ubuntu	"/bin/bash"	9 minutes ago	Exited (0)	9 minutes ago
stupefied_perlman					
be9d06b60a92	ubuntu	"/bin/bash"	16 minutes ago	Exited (0)	16 minutes ago
loving_pasteur					
784021474b5c	busybox	"ping google.com"	18 hours ago	Up	18 hours
optimistic_aryabhata					
d09ae6ad4f49	jiwhiz/spring-boot-docker-mysql	"java -Djava.secur..."	18 hours ago	Up	18 hours
0.0.0.0:8080->8080/tcp	demo-app				

## Lab: Remove exited containers

```
C:\>docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
7c3336dfd309	ubuntu	"/bin/sh -c 'while..."	5 minutes ago	Exited (137)	About a minute ago
loving_ride4425edd202eb	ubuntu	"/bin/sh"	7 minutes ago	Exited (2)	6 minutes ago
kind_murdock					
dd69d7f85fec	ubuntu	"/bin/bash"	9 minutes ago	Exited (0)	9 minutes ago
stupefied_perlman					
be9d06b60a92	ubuntu	"/bin/bash"	16 minutes ago	Exited (0)	16 minutes ago
loving_pasteur					
784021474b5c	busybox	"ping google.com"	18 hours ago	Up	18 hours
optimistic_aryabhata					
d09ae6ad4f49	jiwhiz/spring-boot-docker-mysql	"java -Djava.secur..."	18 hours ago	Up	18 hours
0.0.0.0:8080->8080/tcp	demo-app				

```
C:\>docker stop <container-id>
```

```
C:\>docker rm <container-id>
```

## Lab: Remove all exited containers

```
C:\>docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
7c3336dfd309	ubuntu	"/bin/sh -c 'while..."	5 minutes ago	Exited (137)	About a minute ago
loving_ride4425edd202eb	ubuntu	"/bin/sh"	7 minutes ago	Exited (2)	6 minutes ago
kind_murdock					
dd69d7f85fec	ubuntu	"/bin/bash"	9 minutes ago	Exited (0)	9 minutes ago
stupefied_perlman					
be9d06b60a92	ubuntu	"/bin/bash"	16 minutes ago	Exited (0)	16 minutes ago
loving_pasteur					
784021474b5c	busybox	"ping google.com"	18 hours ago	Up	18 hours
optimistic_aryabhata					
d09ae6ad4f49	jiwhiz/spring-boot-docker-mysql	"java -Djava.secur..."	18 hours ago	Up	18 hours
0.0.0.0:8080->8080/tcp	demo-app				

```
C:\>docker stop $(docker ps -a -q)
```

```
C:\>docker rm $(docker ps -a -q)
```

```
C:\>docker container prune
```

# **Run Web Application with Port Binding**

## Lab: Running a static web application

- `docker run -d sequence/static-site`
- `docker ps`
- `docker stop <container-id>` and `docker rm <container-id>`
- `docker run --name static-site -e AUTHOR="Your Name" -d -P sequence/static-site`
- `docker port static-site`
  - > 443/tcp -> 0.0.0.0.32772
  - > 80/tcp → 0.0.0.0.32773
- Then using your browser
  - > `http://localhost:32773` or
  - > `docker-machine ip default` and then `http://192.168.99.100/32773`

# Lab: Run static web application without porting binding

```
C:\>docker run -d sequence/static-site
30b7a66141581b77936bc382eba693149f53f4306f0afd4f95c613247928530b
```

```
C:\>docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
30b7a6614158	sequence/static-site	"/bin/sh -c 'cd /u..."	30 seconds ago	Up 34 seconds	80/tcp, 443/tcp	angry_mirzak
372cdf8e7a8c	mysql	"docker-entryptoint..."	About an hour ago	Up About an hour	3306/tcp	db
816a53993cfb	ubuntu:latest	"/bin/bash"	13 hours ago	Up 13 hours		ubuntu
702304e36cfc	alpine	"/bin/sh"	14 hours ago	Up 14 hours		pedantic_woz

```
C:\>docker port 30b7a6614158
```

No port mapping between container's port to the host

```
C:\>docker port angry_mirzakhani
```

```
C:\>
```

## Lab: Run static web application with port binding -P

C:\>docker ps

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
30b7a6614158	sequence/static-site	"/bin/sh -c 'cd /u..."	30 seconds ago	Up 34 seconds	80/tcp, 443/tcp	angry_mirzak
372cdf8e7a8c	mysql	"docker-entrypt..."	About an hour ago	Up About an hour	3306/tcp	db
..						

C:\>docker stop angry\_mirzakhani

C:\>docker run --name static-site -e AUTHOR="Sang" -d -P sequence/static-site  
63072ba56255b41f2e399f075d042230dbaaf28fa27adc6bda4c4f6b706be03e

C:\>docker ps

CONTAINER ID	IMAGE	COMMAND	...	PORTS	NAMES
63072ba56255	sequence/static-site	"/bin/sh -c 'cd /u..."	...	0.0.0.0:32773->80/tcp, 0.0.0.0:32772->443/tcp	static-site
372cdf8e7a8c	mysql	"docker-entrypt..."		3306/tcp	db
..					

C:\>docker port static-site  
443/tcp -> 0.0.0.0:32772  
80/tcp -> 0.0.0.0:32773

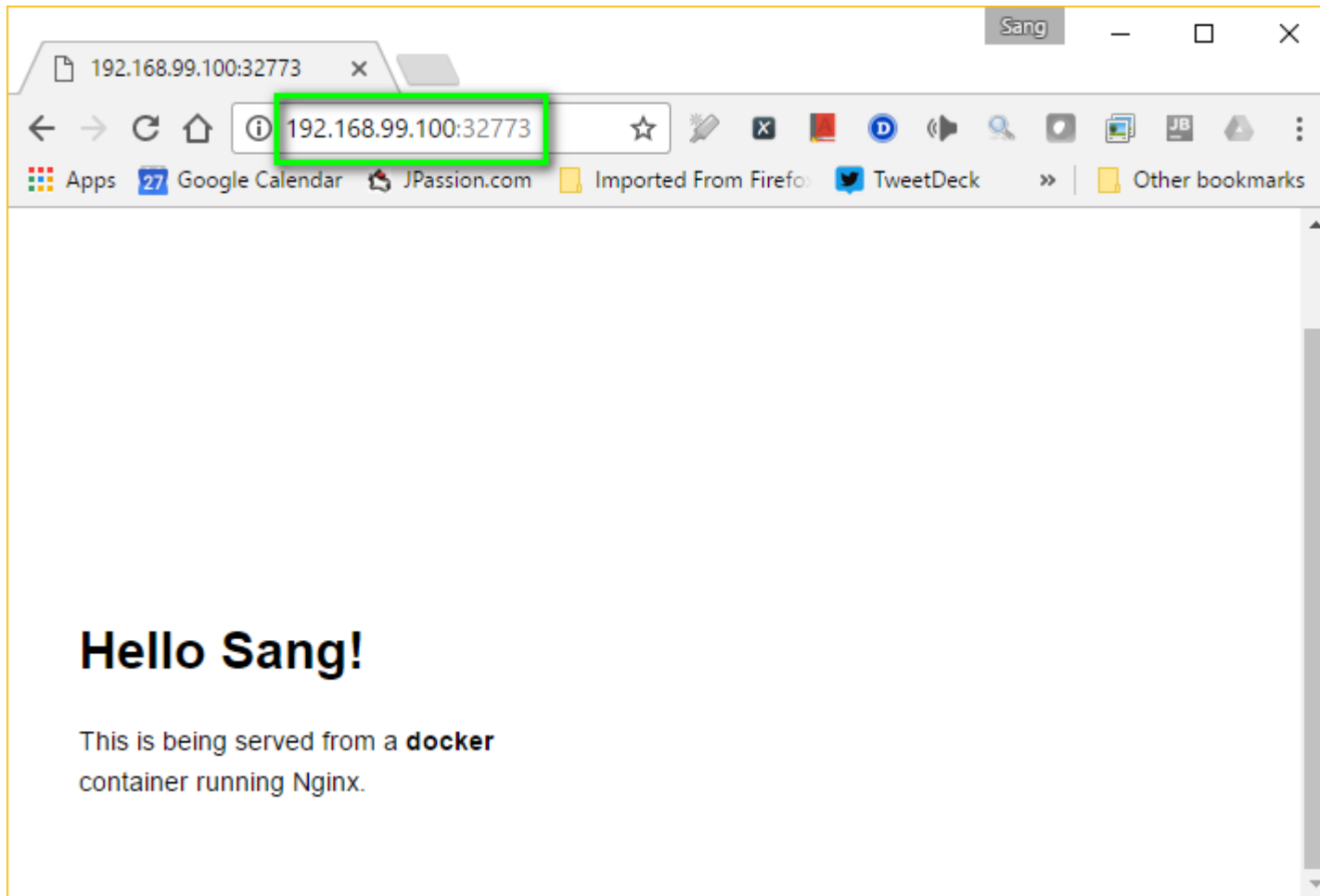
container ports

host ports



# Lab: Access the static web application

```
C:\>docker-machine ip  
192.168.99.100
```



# Port binding options on “docker run” command

```
C:\>docker run --help
```

```
Usage: docker run [OPTIONS] IMAGE [COMMAND] [ARG...]
```

Run a command in a new container

Options:

--add-host list	Add a custom host-to-IP mapping (host:ip) (default [])
-a, --attach list	Attach to STDIN, STDOUT or STDERR (default [])
--name string	Assign a name to the container
--privileged	Give extended privileges to this container
-p, --publish list	Publish a container's port(s) to the host (default [])
-P, --publish-all	Publish all exposed ports to random ports

-p Create a port mapping rule like -p ip:hostPort:containerPort. containerPort is required. If no hostPort is specified, Docker will automatically allocate one.

-P Map a dynamically allocated host port to all container ports that have been exposed by the Dockerfile

# Lab: Run static web application with port binding -p

```
C:\>docker stop static-site
```

```
C:\>docker ps -a
```

```
C:\>docker container prune
```

```
WARNING! This will remove all stopped containers.
```

```
Are you sure you want to continue? [y/N]
```

```
y
```

```
Deleted Containers:
```

```
63072ba56255b41f2e399f075d042230dbaaf28fa27adc6bda4c4f6b706be03e
```

```
30b7a66141581b77936bc382eba693149f53f4306f0afd4f95c613247928530b
```

```
C:\>docker run --name static-site -e AUTHOR="JPassion" -d -p 8081:80 sequence/static-site
```

```
41598ad663dd512fcd96fc35ab2a01500384e9078727f86cf576ea79e3e8a656
```

```
C:\>docker ps
```

CONTAINER ID	IMAGE	COMMAND	... PORTS	NAMES
41598ad663dd	sequence/static-site	"/bin/sh -c 'cd /u..."	443/tcp, 0.0.0.0:8081->80/tcp	static-site
372cdf8e7a8c	mysql	"docker-entrypoint..."	3306/tcp	db

```
..
```

```
C:\>docker port static-site
```

```
80/tcp -> 0.0.0.0:8081
```

container port

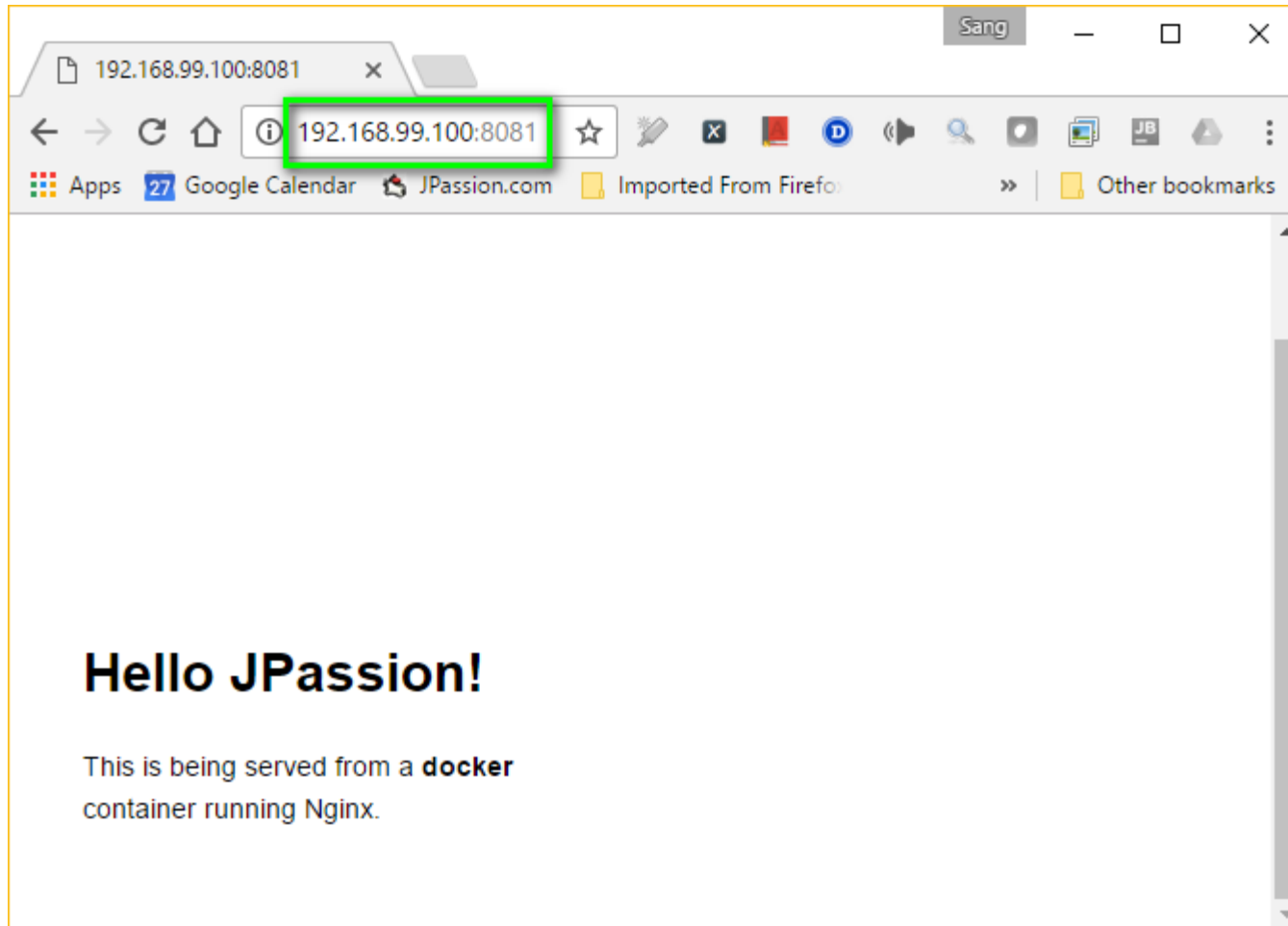
host port

host port

container port

## Lab: Access the static web application

```
C:\>docker-machine ip  
192.168.99.100
```



# **Run Nginx, Tomcat with Port Binding**

# Lab: Run nginx

```
C:\>docker run --name mynginx -P -d nginx
```

```
Unable to find image 'nginx:latest' locally
```

```
latest: Pulling from library/nginx
```

```
...
```

```
9abee7e1ef9d: Pull complete
```

```
Digest: sha256:52f84ace6ea43f2f58937e5f9fc562e99ad6876e82b99d171916c1ece587c188
```

```
Status: Downloaded newer image for nginx:latest
```

```
fc09723aea492b0451b93e012a835ede53c74ed3c05e3da5d06b16688f124009
```

```
C:\>docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
fc09723aea49	nginx	"nginx -g 'daemon ...'"	22 seconds ago	Up 21	0.0.0.0:32769->80/tcp, 0.0.0.0:32768->443/tcp

```
C:\>docker-machine ip
```

```
192.168.99.100
```

```
C:\>curl 192.168.99.100:32769
```

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

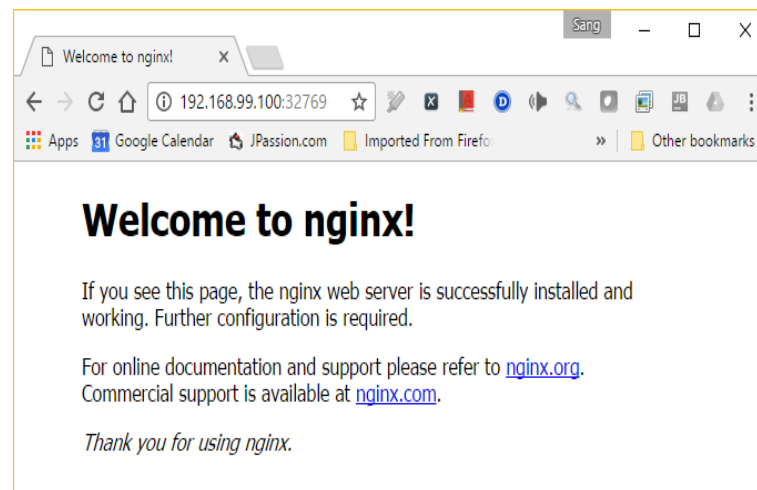
```
<title>Welcome to nginx!</title>
```

```
<style>
```

```
body {
```

```
width: 35em;
```

```
margin: 0 auto;
```



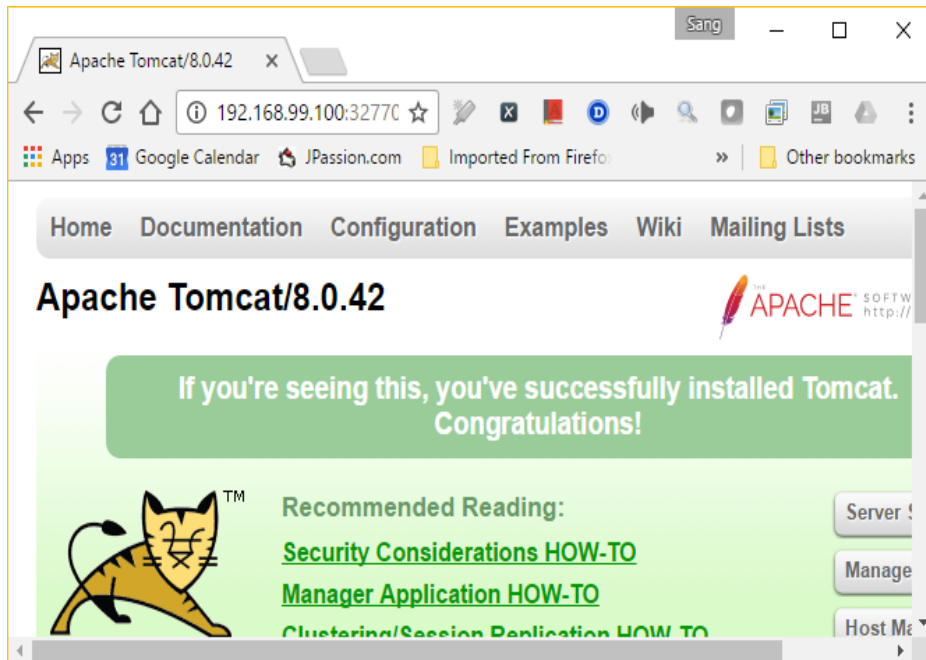
# Lab: Run Tomcat

```
C:\>docker run --name mytomcat -P -d tomcat
```

...

```
C:\>docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
5ffb9aa3b373	tomcat	"catalina.sh run"	17 seconds ago	Up 16 seconds	0.0.0.0:32770->8080/tcp	mytomcat
fc09723aea49	nginx	"nginx -g 'daemon ...'"	17 minutes ago	Up 17 minutes	0.0.0.0:32769->80/tcp, 0.0.0.0:32768->443/tcp	



The background is a solid orange color with a repeating pattern of small, wavy, vertical lines. On the right side, there is a large, white, curved shape that resembles a stylized letter 'C' or a partial circle, extending from the top to the bottom of the frame.

# Volume



**Code with Passion!**  
**JPassion.com**

