

# **DVS Technologies Aws & Devops**

Compiled and Scrutinized by

**Mr. Shaan Shaik**

**(Senior DevOps Lead)**

## **Words To The Students**

Though we have taken utmost efforts to present you this book error free, but still it may contain some errors or mistakes. Students are encouraged to bring, if there are any mistakes or errors in this document to our notice. So that it may be rectified in the next edition of this document.

**“Suppressing your doubts is Hindering your growth”.**

We urge you to work hard and make use of the facilities we are providing to you, because there is no substitute for hard work. We wish you all the best for your future.

**“The grass isn’t greener on the other side; the grass is greener where you water it.”**

You and your suggestions are valuable to us; Help us to serve you better. In case of any suggestions, grievance, or complaints, please feel free to write us your suggestions, grievance and feedback on the following

**Dvs.training@gmail.com**

# DVS Technologies Aws & Devops

## 1. Introduction to Docker:

### Learning the Basics of Docker:

Docker is an open-source Project that automates the deployment of applications inside software containers, by providing an additional layer of abstraction and automation of operating system-level virtualization on Linux.

So basically, it is a tool (or a set of tools depending on how you look at it) that packages up an application and all its dependencies in a "virtual container" so that it can be run on any linux system or distribution.

### When would we use Dockers:

There are lot of reasons to use docker. Although you will generally hear about docker used in conjunction with development and deployment of applications,

there are tons of examples for use:

- \* Configuration simplification
- \* Enhance Developer Productivity
- \* Server Consolidation and management
- \* Application isolation
- \* Rapid Deployment
- \* Build Management

Keep in mind these are only a few use cases. We are going to explore many more during our course!!

### Containers vs. Virtual Machines::

#### What is a virtual Machine:

In basic terms, a virtual machine is an emulation of a specific computer system type. They operate based on the architecture and functions of that real computer system type and its implementation can involve specialized hardware, software or both. when you think of a virtual machine, you probably think of vmware, citrix and or virtual box. Virtualization software allows you to setup one operating system with another. Although they both share the same physical hardware, the virtual machine is isolated from that hardware and has to communicate with it through something called a Hypervisor.

# DVS Technologies Aws & Devops

An aws instance is one type of virtual Machine!

What is a container?

A container is exactly what you might expect it to be based on the general definition of the word. It is an entirely isolated set of packages, libraries and/or applications that are completely independent from its surroundings. In the simplest example, you place your leftovers in a plastic container and then set it on the table. Although the table lends the platform on which the leftover are resting upon, they are independent of the table itself. What you do to one does not necessarily affect the other (although in certain instances it can)

What is the difference important?

As in most things in life, the importance is in perspective. From the perspective of getting the most performance out of hardware purchased, virtualization was invented to allow us to share but segregate server instances from each other. This way, we could protect one operating system from another without letting space CPU cycles, memory or disk space go to waste. Now, virtualization is becoming more granular. We have virtual servers, but they are based on emulating virtual hardware through a hypervisor. This means that they are heavy in terms of system requirements. Containers however, use shared operating systems and more efficient in system resource terms.

**Docker Architecture:**

Docker is a client-server application where both the daemon and client can be run on the same system or you can connect a docker client with a remote docker daemon. Docker clients and daemons communicate via sockets or through a RESTful API (representational state transfer - it is a stateless transfer over http of a web page containing an XML file that describes and includes the desired content).

**The main components of Docker are:**

- daemon
- client
- Docker.io registry

See the docker architecture picture you saved to the dockers directory.

# DVS Technologies Aws & Devops

From the picture we can conclude that docker engine manages all the resource allocations for the applications, there is no necessity to install a new OS and install application on top of it. This will not save our resources like memory, disk & hardware as well as our precious time to install and configure the OS :)

Hasn't this already been done ?

Many companies already have this concept before but Docker hit the right spot in right time. PFB companies which are already using this concept.

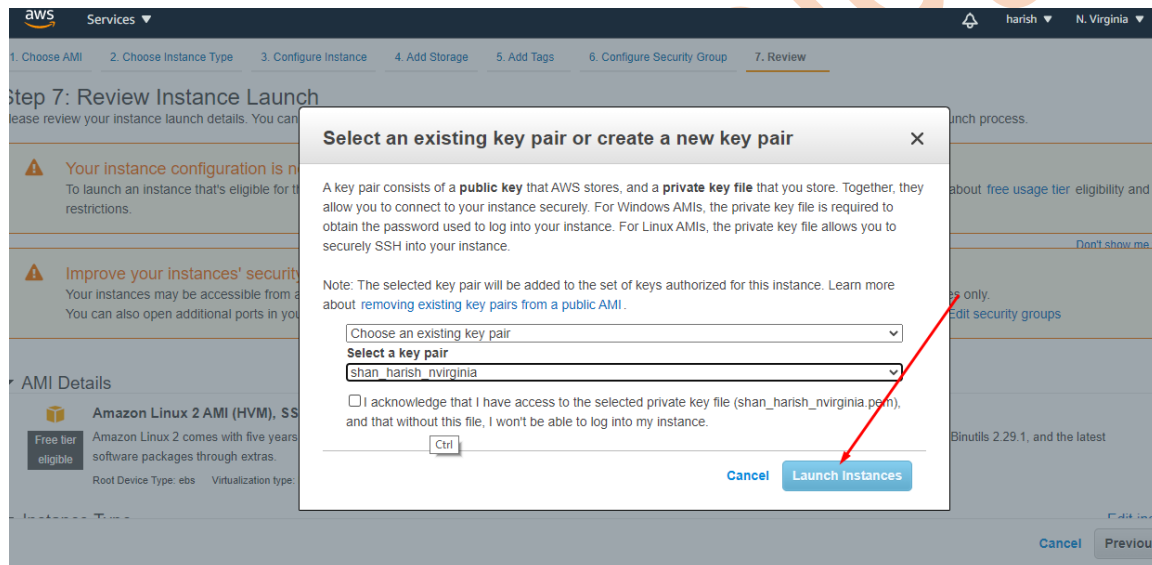
FreeBSD :-> Jails

Sun(and now oracle) Solaris :-> Zones

Google :-> IMctfy

Openvz

## 2. Installation



# DVS Technologies Aws & Devops

Instances (1/2) Info

Filter instances

Instance state: running X Instance state: pending X Clear filters

| Name           | Instance ID                | Instance state | Instance type | Status check   | Alarm Status |
|----------------|----------------------------|----------------|---------------|----------------|--------------|
| controller     | i-02f530154eea4ecb0        | Running        | t2.micro      | 2/2 checks ... | No alarms +  |
| <b>Dockers</b> | <b>i-0d6ae64abb40b108f</b> | Running        | t2.medium     | Initializing   | No alarms +  |

Instance: i-0d6ae64abb40b108f

Details Security Networking Storage Status Checks Monitoring Tags

Instances (1/2) Info

Filter instances

Instance state: running X

| Name           | Instance ID                | Instance state | Instance type | Status       |
|----------------|----------------------------|----------------|---------------|--------------|
| controller     | i-0d6ae64abb40b108f        | Running        | t2.micro      | Running      |
| <b>Dockers</b> | <b>i-0d6ae64abb40b108f</b> | Running        | t2.medium     | Initializing |

Instance ID: i-0d6ae64abb40b108f

Instance state: Running

Instance type: t2.medium

Private IPv4 address: 172.31.73.251

Private IPv4 DNS: ip-172-31-73-2

VPC ID

© 2008 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.

# DVS Technologies Aws & Devops

```
root@ip-172-31-15-231:~#
Using username "ec2-user".
Authenticating with public key "imported-openssh-key" from agent

      _ _      _ _
     _/  ( _ _  /
    _/_\  _/_/_/

Amazon Linux 2 AMI

https://aws.amazon.com/amazon-linux-2/
2 package(s) needed for security, out of 13 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-73-251 ~]$ sudo su -
[root@ip-172-31-73-251 ~]# hostnamectl set-hostname dockers
[root@ip-172-31-73-251 ~]# bash
[root@dockers ~]#
```

## Installation:

```
[root@dockers ~]#
[root@dockers ~]# yum install docker -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Resolving Dependencies
--> Running transaction check
--> Package docker.x86_64 0:19.03.6ce-4.amzn2 will be installed
--> Processing Dependency: runc >= 1.0.0 for package: docker-19.03.6ce-4.amzn2.x86_64
--> Processing Dependency: containerd >= 1.3.2 for package: docker-19.03.6ce-4.amzn2.x86_64
--> Processing Dependency: pigz for package: docker-19.03.6ce-4.amzn2.x86_64
--> Processing Dependency: libcgrouper for package: docker-19.03.6ce-4.amzn2.x86_64
--> Running transaction check
--> Package containerd.x86_64 0:1.3.2-1.amzn2 will be installed
--> Package libcgrouper.x86_64 0:0.41-21.amzn2 will be installed
--> Package pigz.x86_64 0:2.3.4-1.amzn2.0.1 will be installed
--> Package runc.x86_64 0:1.0.0-0.1.20200204.gitdc9208a.amzn2 will be installed
--> Finished Dependency Resolution
```

# DVS Technologies Aws & Devops

```
root@dockers ~]#
root@dockers ~]#
root@dockers ~]# systemctl enable docker
Created symlink from /etc/systemd/system/multi-user.target.wants/docker.service to /usr/lib/systemd/system/docker.service.
root@dockers ~]# systemctl restart docker
root@dockers ~]# systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; vendor preset: disabled)
   Active: active (running) since Wed 2020-10-07 14:43:27 UTC; 3s ago
     Docs: https://docs.docker.com
   Process: 3852 ExecStartPre=/usr/libexec/docker/docker-setup-runtimes.sh (code=exited, status=0/SUCCESS)
   Process: 3847 ExecStartPre=/bin/mkdir -p /run/docker (code=exited, status=0/SUCCESS)
  Main PID: 3860 (dockerd)
    Tasks: 10
   Memory: 37.5M
   CGroup: /system.slice/docker.service
           └─3860 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock --default-ulimit nfile=1024:4...

Oct 07 14:43:25 dockers dockerd[3860]: time="2020-10-07T14:43:25.532548785Z" level=info msg="scheme \"unix\" not ...e=grp
Oct 07 14:43:25 dockers dockerd[3860]: time="2020-10-07T14:43:25.532579435Z" level=info msg="ccResolverWrapper: s...e=grp
Oct 07 14:43:25 dockers dockerd[3860]: time="2020-10-07T14:43:25.532595650Z" level=info msg="ClientConn switching...e=grp
Oct 07 14:43:25 dockers dockerd[3860]: time="2020-10-07T14:43:25.554287149Z" level=info msg="Loading containers: start."
Oct 07 14:43:25 dockers dockerd[3860]: time="2020-10-07T14:43:25.990588820Z" level=info msg="Default bridge (dock...dress"
Oct 07 14:43:26 dockers dockerd[3860]: time="2020-10-07T14:43:26.174219959Z" level=info msg="Loading containers: done."
Oct 07 14:43:27 dockers dockerd[3860]: time="2020-10-07T14:43:27.863754439Z" level=info msg="Docker daemon" commi...3.6-ce
Oct 07 14:43:27 dockers dockerd[3860]: time="2020-10-07T14:43:27.863860993Z" level=info msg="Daemon has completed...ation"
Oct 07 14:43:27 dockers dockerd[3860]: time="2020-10-07T14:43:27.893771651Z" level=info msg="API listen on /var/r...sock"
Oct 07 14:43:27 dockers systemd[1]: Started Docker Application Container Engine.
Hint: Some lines were ellipsized, use -l to show in full.
root@dockers ~]# ls -ld /var/lib/docker/
lrwx--x--x 14 root root 182 Oct  7 14:43 /var/lib/docker/
root@dockers ~]#
```

## 3. Working with Docker

### Pulling an Image:

```
root@dockers ~]#
root@dockers ~]# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
root@dockers ~]# docker pull centos
Using default tag: latest
latest: Pulling from library/centos
3c72a8ed6814: Pull complete
Digest: sha256:76d24f3ba3317fa945743bb3746fbaf3a0b752f10b10376960de01da70685fbd
Status: Downloaded newer image for centos:latest
docker.io/library/centos:latest
root@dockers ~]# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
centos               latest             0d120b6ccaa8       8 weeks ago        215MB
root@dockers ~]# docker pull centos:7
7: Pulling from library/centos
75f829a71a1c: Pull complete
Digest: sha256:19a79828ca2e505eae0ff38c2f3fd9901f4826737295157cc5212b7a372cd2b
Status: Downloaded newer image for centos:7
docker.io/library/centos:7
root@dockers ~]# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
centos               7                  7e6257c9f8d8       8 weeks ago        203MB
centos               latest             0d120b6ccaa8       8 weeks ago        215MB
root@dockers ~]#
```

*pulling image with tag*

# DVS Technologies Aws & Devops

```
[root@dockers ~]# #docker pull centos:7
[root@dockers ~]# docker info
Client:
 Debug Mode: false

Server:
 Containers: 0
  Running: 0
  Paused: 0
  Stopped: 0
 Images: 2
 Server Version: 19.03.6-ce
 Storage Driver: overlay2
 Kernel Version: 4.14.193-149.317.amzn2.x86_64
 Operating System: Amazon Linux 2
 OSType: linux
 Architecture: x86_64
 CPUs: 2
 Total Memory: 3.851GiB
 Name: dockers
 ID: V3GS:H3CA:73G6:7GOJ:FESR:B6VR:36CY:MZU6:2ZJQ:7XM3:MN5U:ILS4
 Docker Root Dir: /var/lib/docker
 Debug Mode: false
 Registry: https://index.docker.io/v1/
 Labels:
 Experimental: false
 Insecure Registries:
  127.0.0.0/8
 Live Restore Enabled: false

[root@dockers ~]#
```

## Creating Container:

### Base machine Image:

```
[root@dockers ~]# uname -a
Linux dockers 4.14.193-149.317.amzn2.x86_64 #1 SMP Thu Sep 3 19:04:44 UTC 2020 x86_64 x86_64 x86_64 GNU/Linux

[root@dockers ~]# docker run -i -t --name mycont1 centos /bin/bash
[root@7f7446b7bff6 /]# uname -a
Linux 7f7446b7bff6 4.14.193-149.317.amzn2.x86_64 #1 SMP Thu Sep 3 19:04:44 UTC 2020 x86_64 x86_64 x86_64 GNU/Linux
[root@7f7446b7bff6 /]# cat /etc/redhat-release
CentOS Linux release 8.2.2004 (Core)
```

*container creation*

*os inside container*

## Now lets exit from the container & lets see what will happen:

```
[root@7f7446b7bff6 /]# cat /etc/redhat-release
CentOS Linux release 8.2.2004 (Core)
[root@7f7446b7bff6 /]# exit
exit
[root@dockers ~]# docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS
MES
[root@dockers ~]# docker ps -a
CONTAINER ID        NAMES              IMAGE               COMMAND             CREATED             STATUS              PORTS
7f7446b7bff6        mycont1            centos              "/bin/bash"         2 minutes ago       Exited (0) About a minute ago
```

*Gives active containers*

*Gives both active & passive containers*



# DVS Technologies Aws & Devops

## Restart a container:

```
[root@ip-172-31-73-251 ~]# bash
[root@dockers ~]# docker restart mycont1
mycont1
[root@dockers ~]# docker ps
```

| CONTAINER ID | IMAGE  | COMMAND     | CREATED       | STATUS       | PORTS | NAMES   |
|--------------|--------|-------------|---------------|--------------|-------|---------|
| 7f7446b7bff6 | centos | "/bin/bash" | 7 minutes ago | Up 3 seconds |       | mycont1 |

[root@dockers ~]#

```
[root@dockers ~]# docker run -it --name mycont2 centos:7 /bin/bash
[root@75e51f4eb8c8 /]# exit
[root@dockers ~]# docker ps
```

| CONTAINER ID | IMAGE    | COMMAND     | CREATED        | STATUS                    | PORTS | NAMES   |
|--------------|----------|-------------|----------------|---------------------------|-------|---------|
| 7f7446b7bff6 | centos   | "/bin/bash" | 15 minutes ago | Up 8 minutes              |       | mycont1 |
| 75e51f4eb8c8 | centos:7 | "/bin/bash" | 28 seconds ago | Exited (0) 11 seconds ago |       | mycont2 |
| 7f7446b7bff6 | centos   | "/bin/bash" | 16 minutes ago | Up 8 minutes              |       | mycont1 |

[root@dockers ~]# docker run -it --name mycont3 centos:7 /bin/bash  
[root@99abc69fd2a9 /]# cat /etc/redhat-release  
CentOS Linux release 7.8.2003 (Core)  
[root@99abc69fd2a9 /]# #ctrl+p+q

```
[root@99abc69fd2a9 /]# docker ps
```

| CONTAINER ID | IMAGE    | COMMAND     | CREATED            | STATUS            | PORTS | NAMES   |
|--------------|----------|-------------|--------------------|-------------------|-------|---------|
| 99abc69fd2a9 | centos:7 | "/bin/bash" | About a minute ago | Up About a minute |       | mycont3 |
| 7f7446b7bff6 | centos   | "/bin/bash" | 17 minutes ago     | Up 10 minutes     |       | mycont1 |

[root@dockers ~]#

*In this case container will get killed*

*In this case container will be there as we are using ctrl+p+q to come out of container*

## Running container in detached mode:

```
[root@dockers ~]# docker ps
```

| CONTAINER ID | IMAGE    | COMMAND     | CREATED        | STATUS        | PORTS | NAMES       |
|--------------|----------|-------------|----------------|---------------|-------|-------------|
| 72880c892472 | centos   | "/bin/bash" | 2 minutes ago  | Up 2 minutes  |       | vigilant_tu |
| 99abc69fd2a9 | centos:7 | "/bin/bash" | 10 minutes ago | Up 10 minutes |       | mycont3     |
| 7f7446b7bff6 | centos   | "/bin/bash" | 27 minutes ago | Up 19 minutes |       | mycont1     |

```
[root@dockers ~]# docker run -itd --name mycont4 centos:7 /bin/bash
039cd9dcf6e9708526b33a0aa54a6e1d0e6d769cd5769053454627a65306b0
```

```
[root@dockers ~]# docker ps
```

| CONTAINER ID | IMAGE    | COMMAND     | CREATED        | STATUS        | PORTS | NAMES       |
|--------------|----------|-------------|----------------|---------------|-------|-------------|
| 039cd9dcf6e9 | centos:7 | "/bin/bash" | 12 seconds ago | Up 11 seconds |       | mycont4     |
| 72880c892472 | centos   | "/bin/bash" | 3 minutes ago  | Up 3 minutes  |       | vigilant_tu |
| 99abc69fd2a9 | centos:7 | "/bin/bash" | 11 minutes ago | Up 11 minutes |       | mycont3     |
| 7f7446b7bff6 | centos   | "/bin/bash" | 28 minutes ago | Up 21 minutes |       | mycont1     |

## Renaming a container:

```
[root@dockers ~]# docker ps
```

| CONTAINER ID | IMAGE    | COMMAND     | CREATED            | STATUS            | PORTS | NAMES       |
|--------------|----------|-------------|--------------------|-------------------|-------|-------------|
| 039cd9dcf6e9 | centos   | "/bin/bash" | About a minute ago | Up About a minute |       | mycont4     |
| 72880c892472 | centos   | "/bin/bash" | 5 minutes ago      | Up 5 minutes      |       | vigilant_tu |
| 99abc69fd2a9 | centos:7 | "/bin/bash" | 12 minutes ago     | Up 12 minutes     |       | mycont3     |
| 7f7446b7bff6 | centos   | "/bin/bash" | 29 minutes ago     | Up 22 minutes     |       | mycont1     |

```
[root@dockers ~]# docker rename vigilant_tu mycont5
```

```
[root@dockers ~]# docker ps
```

| CONTAINER ID | IMAGE    | COMMAND     | CREATED        | STATUS        | PORTS | NAMES   |
|--------------|----------|-------------|----------------|---------------|-------|---------|
| 039cd9dcf6e9 | centos:7 | "/bin/bash" | 2 minutes ago  | Up 2 minutes  |       | mycont4 |
| 72880c892472 | centos   | "/bin/bash" | 5 minutes ago  | Up 5 minutes  |       | mycont5 |
| 99abc69fd2a9 | centos:7 | "/bin/bash" | 13 minutes ago | Up 13 minutes |       | mycont3 |
| 7f7446b7bff6 | centos   | "/bin/bash" | 30 minutes ago | Up 23 minutes |       | mycont1 |

[root@dockers ~]#

# DVS Technologies Aws & Devops

Login in to the container:

```
7f7446b7bfff        centos                "/bin/bash"          28 minutes ago      Up 21 minutes      mycont1
[root@dockers ~]# docker exec -it mycont4 /bin/bash
[root@039cd9dcf6e9 /]# cat /etc/redhat-release
CentOS Linux release 7.8.2003 (Core)
[root@039cd9dcf6e9 /]# read escape sequence
```

OR

```
[root@dockers ~]# docker ps
CONTAINER ID        IMAGE               COMMAND              CREATED            STATUS              PORTS              NAMES
039cd9dcf6e9       centos:7           "/bin/bash"         6 minutes ago     Up 6 minutes              mycont4
72880c892472       centos:7           "/bin/bash"         9 minutes ago     Up 9 minutes              mycont5
99abc69fd2a9       centos:7           "/bin/bash"         17 minutes ago    Up 17 minutes             mycont3
7f7446b7bfff       centos              "/bin/bash"         34 minutes ago    Up 26 minutes             mycont1

[root@dockers ~]# docker attach mycont5
[root@72880c892472 /]# cat /etc/redhat-release
CentOS Linux release 8.2.2004 (Core)
[root@72880c892472 /]# read escape sequence
[root@dockers ~]#
[root@dockers ~]#
[root@dockers ~]#
```

Testing the test.txt file existence in two different containers:

```
[root@dockers ~]# docker run -itd --name mytestcont1 centos /bin/bash
2118ebc1f6e91d9442cf43a73dc62ede9ee887ca0f0224c1ff525b859a6805c
[root@dockers ~]# docker run -itd --name mytestcont2 centos /bin/bash
ce266335ebdc66c6986c755e4493f3beaf611516b4efaf75607c972deab90df2
[root@dockers ~]# docker ps | grep -i mytestcon
ce266335ebdc        centos              "/bin/bash"          9 seconds ago       Up 8 seconds         mytestcont2
2118ebc1f6e9       centos              "/bin/bash"          17 seconds ago      Up 16 seconds        mytestcont1

[root@dockers ~]# docker exec -it mytestcont1 /bin/bash
[root@2118ebc1f6e9 /]# cd /tmp/
[root@2118ebc1f6e9 tmp]# echo "Hi I am inside mytestcont1" > /tmp/test.txt
[root@2118ebc1f6e9 tmp]# read escape sequence
[root@dockers ~]#
[root@dockers ~]#
[root@dockers ~]# #Testing if my test.txt file exists in mycont2 or not
[root@dockers ~]#
[root@dockers ~]# docker exec -it mytestcont2 /bin/bash
[root@ce266335ebdc /]# ls -l /tmp/test.txt
ls: cannot access '/tmp/test.txt': No such file or directory
[root@ce266335ebdc /]#
```

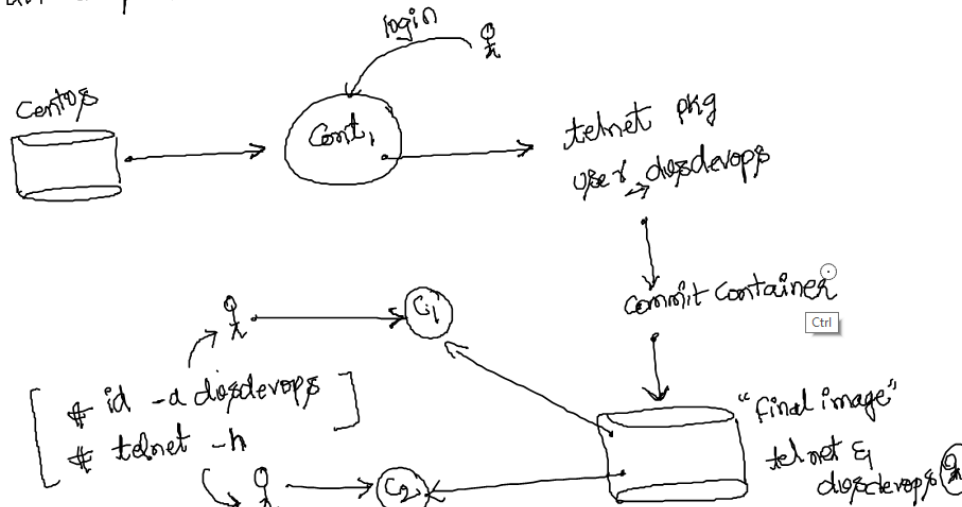
*Testing the Existence  
in two diff containers*

# DVS Technologies Aws & Devops

## 4. Image Customization

Creating an Image from Container :

5/5



### 1. Create an image from container:

#### 1 Creating a container from centos7 image:

```
[root@dockers ~]# docker run -itd --name myimagecont1 centos /bin/bash
91d81c30bcc8216ba56cc7cc781324a6b1805b6e510a90975e82a4631d4d600
[root@dockers ~]# docker ps|grep -i myimage
91d81c30bcc8      centos          "/bin/bash"        6 seconds ago    Up 5 seconds
[root@dockers ~]#
```

#### 2 Login in to the container & testing the user & package

```
[root@dockers ~]# docker exec -it myimagecont1 /bin/bash
[root@91d81c30bcc8 /]#
[root@91d81c30bcc8 /]# #Testing user & package
[root@91d81c30bcc8 /]#
[root@91d81c30bcc8 /]#
[root@91d81c30bcc8 /]# id -a dysdevops
id: 'dysdevops': no such user
[root@91d81c30bcc8 /]# telnet -h
bash: telnet: command not found
[root@91d81c30bcc8 /]#
```

# DVS Technologies Aws & Devops

## 3 Creating user & installing telnet package inside the above container

```
[root@91d81c30bcc8 /]# useradd dvsdevops
[root@91d81c30bcc8 /]# id -a dvsdevops
uid=1000(dvsdevops) gid=1000(dvsdevops) groups=1000(dvsdevops)
```

```
[root@91d81c30bcc8 /]# yum install telnet -y
Failed to set locale, defaulting to C.UTF-8
CentOS-8 - AppStream                23 MB/s | 5.8 MB    00:00
CentOS-8 - Base                     10 MB/s | 2.2 MB    00:00
CentOS-8 - Extras                   156 kB/s | 8.1 kB   00:00
Dependencies resolved.
```

| Package | Architecture | Version | Repository | Size |
|---------|--------------|---------|------------|------|
|---------|--------------|---------|------------|------|

```
[root@91d81c30bcc8 /]#
[root@91d81c30bcc8 /]# telnet
telnet> q
[root@91d81c30bcc8 /]#
```

## 4 creating image from container:

```
[root@91d81c30bcc8 /]#
[root@91d81c30bcc8 /]# read escape sequence ← ctrl+p+q Come out of Container
[root@dockers ~]# docker ps|grep -i image
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
91d81c30bcc8   centos   "/bin/bash"   5 minutes ago   Up 5 minutes   ports alive   myimagecont1
[root@dockers ~]# docker images
REPOSITORY    TAG       IMAGE ID       CREATED       SIZE
centos        7         7e6257c9f8d8   8 weeks ago   203MB
centos        latest    0d120b6ccaa8   8 weeks ago   215MB
[root@dockers ~]# docker commit myimagecont1 mycustomimage:v1
sha256:0e2342aecdd5acf9c2a9e9985e23f7216f67a2c57818668f71196f3c8514a0e7
[root@dockers ~]# docker images
REPOSITORY    TAG       IMAGE ID       CREATED       SIZE
mycustomimage v1        0e2342aecdd5   3 seconds ago 246MB
centos        7         7e6257c9f8d8   8 weeks ago   203MB
centos        latest    0d120b6ccaa8   8 weeks ago   215MB
[root@dockers ~]#
```

Final Image

## 5 Testing customized image:

```
[root@dockers ~]# docker images
REPOSITORY    TAG       IMAGE ID       CREATED       SIZE
mycustomimage v1        0e2342aecdd5   3 minutes ago 246MB
centos        7         7e6257c9f8d8   8 weeks ago   203MB
centos        latest    0d120b6ccaa8   8 weeks ago   215MB
[root@dockers ~]# docker run -it --name mycustomimagetest1 mycustomimage:v1 /bin/bash
[root@e5085ca1154f /]#
[root@e5085ca1154f /]# #I am inside container of my customimage
[root@e5085ca1154f /]#
[root@e5085ca1154f /]# id -a dvsdevops
uid=1000(dvsdevops) gid=1000(dvsdevops) groups=1000(dvsdevops)
[root@e5085ca1154f /]# telnet
telnet> 1
?Invalid command
telnet> q
[root@e5085ca1154f /]#
```

# DVS Technologies Aws & Devops

Drawback with this approach is we don't have history of changes what we are performing for creating the image. Hence we always opt for Dockerfile

## Image creation using Dockerfile:

### Dockerfile references:

#### FROM:

```
alpine latest a24bb4015296
[root@dockers httpd]# cat Dockerfile
FROM centos:7
RUN useradd dvsdevops && \
```

#### RUN:

```
[root@dockers httpd]#
[root@dockers httpd]# cat Dockerfile
FROM centos:7
RUN useradd dvsdevops && \
    yum install telnet -y
[root@dockers httpd]# docker build -t "dvsdevops:v1" .
Sending build context to Docker daemon 2.048kB
Step 1/2 : FROM centos:7
--> 7e6257c9f8d8
Step 2/2 : RUN useradd dvsdevops && yum install telnet -y
--> Running in 8ad679d8cbe5
Loaded plugins: fastestmirror, ovl
Determining fastest mirrors
 * base: d36uatko69830t.cloudfront.net
 * extras: d36uatko69830t.cloudfront.net
 * updates: d36uatko69830t.cloudfront.net
Resolving Dependencies
--> Running transaction check
--> Package telnet.x86_64 1:0.17-65.el7_8 will be installed
--> Finished Dependency Resolution
[root@dockers httpd]# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
dvsdevops            v1                 01337634ebe6       5 minutes ago      296MB
<none>              <none>             786e1e169665       9 minutes ago      296MB
mycustomimage        v1                 0e2342aecdd5       23 hours ago       246MB
centos               7                  7e6257c9f8d8       8 weeks ago        292MB
```

# DVS Technologies Aws & Devops

## COPY:

```
[root@dockers httpd]# ls -l
total 4
-rw-r--r-- 1 root root 61 Oct  8 14:51 Dockerfile
[root@dockers httpd]# tar -cvf tmp.tar /tmp
tar: Removing leading `/' from member names
/tmp/
/tmp/.XIM-unix/
/tmp/.X11-unix/
/tmp/.Test-unix/
/tmp/.ICE-unix/
/tmp/.font-unix/
/tmp/systemd-private-5eba5fb79ca64f599631dbfdc5ea46be-chrond.service-ydULOu/
/tmp/systemd-private-5eba5fb79ca64f599631dbfdc5ea46be-chrond.service-ydULOu/tmp/
[root@dockers httpd]#
[root@dockers httpd]# ls -l
total 16
-rw-r--r-- 1 root root 61 Oct  8 14:51 Dockerfile
-rw-r--r-- 1 root root 10240 Oct  8 14:55 tmp.tar
```

```
[root@dockers httpd]# cat Dockerfile
FROM centos:7
RUN useradd dvsdevops && yum install telnet -y
COPY tmp.tar /root/
[root@dockers httpd]# docker build -t "dvsdevops:COPY" .
Sending build context to Docker daemon 12.8kB
Step 1/3 : FROM centos:7
--> 7e6257c9f8d8
Step 2/3 : RUN useradd dvsdevops && yum install telnet -y
--> Running in a58e31a66e2d
Loaded plugins: fastestmirror, ovl
Determining fastest mirrors
 * base: d36uatko69830t.cloudfront.net
 * extras: d36uatko69830t.cloudfront.net
```

```
[root@dockers httpd]# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
dvsdevops            COPY               df4072186bc5       9 seconds ago      296MB
dvsdevops            v1                 01337634ebe6       9 minutes ago      296MB
```

## Final testing:

```
copy
[root@dockers httpd]# docker run --name copy -it dvsdevops:COPY /bin/bash
[root@f089bf2c9c56 /]# cd /root/
[root@f089bf2c9c56 ~]# ls -l
total 16
-rw----- 1 root root 3416 Aug  9 21:39 anaconda-ks.cfg
-rw-r--r-- 1 root root 10240 Oct  8 14:55 tmp.tar
[root@f089bf2c9c56 ~]#
```

# DVS Technologies Aws & Devops

## ADD:

```
[root@dockers httpd]# ls -l
total 16
-rw-r--r-- 1 root root    81 Oct  8 14:57 Dockerfile
-rw-r--r-- 1 root root 10240 Oct  8 14:55 tmp.tar
[root@dockers httpd]# vi Dockerfile
[root@dockers httpd]# cat Dockerfile
FROM centos:7
RUN useradd dvsdevops && yum install telnet -y
ADD tmp.tar /root/
[root@dockers httpd]# docker build -t "dvsdevops:ADD" .
Sending build context to Docker daemon  12.8KB
Step 1/3 : FROM centos:7
----> 7e6257c9f8d8
Step 2/3 : RUN useradd dvsdevops && yum install telnet -y
----> Using cache
----> 7a9c70445332
Step 3/3 : ADD tmp.tar /root/
----> 9dc96ff5d735
Successfully built 9dc96ff5d735
Successfully tagged dvsdevops:ADD
[root@dockers httpd]# docker run --name add -it dvsdevops:ADD /bin/bash
[root@4474e19ce597 /]# cd /root/
[root@4474e19ce597 ~]# ls -l
total 4
-rw----- 1 root root 3416 Aug  9 21:39 anaconda-ks.cfg
drwxrwxrwt 8 root root 172 Oct  8 14:51 tmp
[root@4474e19ce597 ~]#
```

ADD:-  
Will untar & then copy the data  
In the below I can see my data  
as "tmp"  
where in COPY it will  
come as "tmp.tar"

## ENV:

Before in centos:7 image:

```
[root@dockers httpd]# docker run -it --name env-test centos:7 /bin/bash
[root@c41559056227 /]# env|grep -i MYBATCH
[root@c41559056227 /]#
```

After in our custom image:

# DVS Technologies Aws & Devops

```
[root@dockers httpd]#  
[root@dockers httpd]# cat Dockerfile  
FROM centos:7  
RUN useradd dvsdevops && yum install telnet -y  
ADD tmp.tar /root/  
ENV MYBATCH="dvsdevops4"  
[root@dockers httpd]# docker build -t "dvsdevops:ENV" .  
Sending build context to Docker daemon 12.8kB  
Step 1/4 : FROM centos:7  
--> 7e6257c9f8d8  
Step 2/4 : RUN useradd dvsdevops && yum install telnet -y  
--> Using cache  
--> 7a9c70445332  
Step 3/4 : ADD tmp.tar /root/  
--> Using cache  
--> 9dc96ff5d735  
Step 4/4 : ENV MYBATCH="dvsdevops4"  
--> Using cache  
--> d691ef6fcd53  
Successfully built d691ef6fcd53  
Successfully tagged dvsdevops:ENV  
[root@dockers httpd]#
```

```
[root@dockers httpd]# docker run -it --name env dvsdevops:ENV /bin/bash  
[root@6aa41908fe8b /]# env|grep -i MY  
MYBATCH=dvsdevops4  
[root@6aa41908fe8b /]#
```

## USER:

```
[root@dockers httpd]# cat Dockerfile  
FROM centos:7  
RUN useradd dvsdevops && yum install telnet -y  
ADD tmp.tar /root/  
ENV MYBATCH="dvsdevops4"  
USER dvsdevops  
[root@dockers httpd]# docker build -t "dvsdevops:USER" .  
Sending build context to Docker daemon 12.8kB  
Step 1/5 : FROM centos:7  
--> 7e6257c9f8d8  
Step 2/5 : RUN useradd dvsdevops && yum install telnet -y  
--> Using cache  
--> 7a9c70445332  
Step 3/5 : ADD tmp.tar /root/  
--> Using cache  
--> 9dc96ff5d735  
Step 4/5 : ENV MYBATCH="dvsdevops4"  
--> Using cache  
--> d691ef6fcd53  
Step 5/5 : USER dvsdevops  
--> Using cache  
--> ccc492f4d52b  
Successfully built ccc492f4d52b  
Successfully tagged dvsdevops:USER  
[root@dockers httpd]# docker run -it --name user dvsdevops:USER /bin/bash  
[dvsdevops@6efbde3e74b8 /]$ whoami  
dvsdevops  
[dvsdevops@6efbde3e74b8 /]$
```



# DVS Technologies Aws & Devops

## EXPOSE:

General container from nginx image:

```
[root@dockers httpd]# docker run --name dockers-nginx -d nginx
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
d121f8d1c412: Pull complete
66a200539fd6: Pull complete
e9738820db15: Pull complete
d74ea5811e8a: Pull complete
ffdacbb6928: Pull complete
Digest: sha256:fc66cdef5ca33809823182c9c5d72ea86fd2cef7713cf3363e1a0b12a5d77500
Status: Downloaded newer image for nginx:latest
5a1b1f47aa2993bcf86fde042d8f44fca5128d9fc9e2172d4b15602945b9122a
[root@dockers httpd]# docker ps |grep dockers-nginx
5a1b1f47aa29    nginx          "/docker-entrypoint..." 14 seconds ago    Up 11 seconds    80/tcp
dockers-nginx
[root@dockers httpd]#
```

*Docker hub image*

Let's build our own Nginx:

```
[root@dockers httpd]# cat Dockerfile
FROM centos
RUN yum install nginx -y
CMD ["nginx", "-g", "daemon off;"]
[root@dockers httpd]# docker build -t "dvsdevops:EXPOSE" .
Sending build context to Docker daemon 12.8kB
Step 1/3 : FROM centos
--> 0d120b6ccaa8
Step 2/3 : RUN yum install nginx -y
--> Running in 29c56ca1189b
CentOS-8 - AppStream          52 MB/s | 5.8 MB      00:00
CentOS-8 - Base               11 MB/s | 2.2 MB      00:00
CentOS-8 - Extras            54 kB/s | 8.1 kB      00:00
Dependencies resolved.
=====
Package                        Arch      Version                               Repo      Size
=====
Installing:
nginx                          x86_64    1:1.14.1-9.module_el8.0.0+184+e34fea82 AppStream 570 k
Installing dependencies:
dejavu-fonts-common           noarch    2.35-6.el8                           BaseOS    74 k
dejavu-sans-fonts             noarch    2.35-6.el8                           BaseOS    1.5 M
fontconfig                    x86_64    2.13.1-3.el8                         BaseOS    275 k
fontpackages-filesystem       noarch    1.44-22.el8                          BaseOS    16 k
=====
```

```
[root@dockers httpd]# docker images|grep -i EXPOSE
dvsdevops    EXPOSE      a383ffb152d4    41 seconds ago    289MB
[root@dockers httpd]#
```

```
[root@dockers httpd]# docker images|grep -i EXPOSE
dvsdevops    EXPOSE      a383ffb152d4    41 seconds ago    289MB
[root@dockers httpd]# docker run --name myown-nginx -d dvsdevops:EXPOSE
b5ecc7bd80da7b869ce4f11d9519cbbd49a3b244a169672f06a9524f4e3cdc8
[root@dockers httpd]# docker ps|grep -i nginx
b5ecc7bd80da    dvsdevops:EXPOSE    "nginx -g 'daemon of..." 8 seconds ago    Up 6 seconds
myown-nginx
5a1b1f47aa29    nginx          "/docker-entrypoint..." 6 minutes ago    Up 6 minutes    80/tcp
dockers-nginx
[root@dockers httpd]#
```

*Not*

# DVS Technologies Aws & Devops

## Final Code:

```
[root@dockers httpd]# cat Dockerfile
FROM centos
RUN yum install nginx -y
EXPOSE 80
CMD ["nginx", "-g", "daemon off;"]
[root@dockers httpd]# docker build -t "dvsdevops:EXPOSE" .
Sending build context to Docker daemon 12.8kB
Step 1/4 : FROM centos
--> 0d120b6ccaa8
Step 2/4 : RUN yum install nginx -y
--> Using cache
--> 6da3907a60df
Step 3/4 : EXPOSE 80
--> Running in aala25241c2f
Removing intermediate container aala25241c2f
--> b32d539384d1
Step 4/4 : CMD ["nginx", "-g", "daemon off;"]
--> Running in 2a791bf108ac
Removing intermediate container 2a791bf108ac
--> 3641d32d9d95
Successfully built 3641d32d9d95
Successfully tagged dvsdevops:EXPOSE
[root@dockers httpd]# docker run --name myown-nginx-final -d dvsdevops:EXPOSE
48adace71fdce90959fd27969a416551ded989bf8a6dafae5bddc70c89aac546
[root@dockers httpd]# docker ps | grep -i nginx
48adace71fdce90959fd27969a416551ded989bf8a6dafae5bddc70c89aac546   dvsdevops:EXPOSE   "nginx -g 'daemon of..."   4 seconds ago       Up 3 seconds        80/tcp
b5ecc7bd80daa383ffb152d4   "nginx -g 'daemon of..."   3 minutes ago       Up 3 minutes        80/tcp
5a1b1f47aa29   nginx   "/docker-entrypoint..."   9 minutes ago       Up 9 minutes        80/tcp
[root@dockers httpd]#
```

## CMD & ENTRYPOINT:

### CMD:

```
[root@dockers httpd]# cat script1.sh
#!/bin/bash
echo "Hi I am Script1"
[root@dockers httpd]# cat script2.sh
#!/bin/bash
echo "Hi I am Script2"
[root@dockers httpd]# vi Dockerfile
[root@dockers httpd]# cat Dockerfile
FROM centos
COPY script1.sh script2.sh /tmp/
CMD ["/bin/bash", "/tmp/script1.sh"]
[root@dockers httpd]# docker build -t "dvsdevops:CMD" .
Sending build context to Docker daemon 14.85kB
Step 1/3 : FROM centos
--> 0d120b6ccaa8
Step 2/3 : COPY script1.sh script2.sh /tmp/
--> e181a0a98da6
Step 3/3 : CMD ["/bin/bash", "/tmp/script1.sh"]
--> Running in 82bc66624961
Removing intermediate container 82bc66624961
--> b5e5b9758840
Successfully built b5e5b9758840
Successfully tagged dvsdevops:CMD
[root@dockers httpd]# docker run --name cmd dvsdevops:CMD
Hi I am Script1
[root@dockers httpd]# docker run --name cmd2 dvsdevops:CMD /bin/bash /tmp/script2.sh
Hi I am Script2
[root@dockers httpd]#
```

# DVS Technologies Aws & Devops

## ENTRYPOINT:

```
[root@dockers httpd]# cat script1.sh
#!/bin/bash
echo "Hi I am Script1"
[root@dockers httpd]# cat script2.sh
#!/bin/bash
echo "Hi I am Script2"
[root@dockers httpd]# cat Dockerfile
FROM centos
COPY script1.sh script2.sh /tmp/
ENTRYPOINT ["/bin/bash", "/tmp/script1.sh"]
[root@dockers httpd]# docker build -t "dvsdevops:ENTRYPOINT" .
Sending build context to Docker daemon 14.85KB
Step 1/3 : FROM centos
--> 0d120b6ccaa8
Step 2/3 : COPY script1.sh script2.sh /tmp/
--> Using cache
--> e181a0a98da6
Step 3/3 : ENTRYPOINT ["/bin/bash", "/tmp/script1.sh"]
--> Running in 176c86b9e119
Removing intermediate container 176c86b9e119
--> a633815fd6db
Successfully built a633815fd6db
Successfully tagged dvsdevops:ENTRYPOINT
```

```
[root@dockers httpd]# docker run --name entrypoint dvsdevops:ENTRYPOINT
Hi I am Script1
[root@dockers httpd]# docker run --name entrypoint2 dvsdevops:ENTRYPOINT /bin/bash /tmp/script2.sh
Hi I am Script2
[root@dockers httpd]#
```

## FINAL CODE:

**FROM** centos  
**RUN** yum install telnet nginx -y && useradd test1  
**ENV** myenv=100  
**ADD** tmp.tar /tmp/  
**COPY** tmp.tar /tmp/  
**EXPOSE** 80  
**VOLUME** ["/usr/share/nginx/html"]  
**CMD** ["nginx", "-g", "daemon off;"]

# DVS Technologies Aws & Devops

## 5. Storage Management

```
[root@dockers httpd]# docker run -it --name mycont1 centos:7 /bin/bash
[root@53ec7077bc92 /]# cd /tmp/
[root@53ec7077bc92 tmp]# echo "Hi All I am inside container mycont1" > mytestfile.txt
[root@53ec7077bc92 tmp]# cat mytestfile.txt
Hi All I am inside container mycont1
[root@53ec7077bc92 tmp]# [root@dockers httpd]#
[root@dockers httpd]#
[root@dockers httpd]# docker rm -f mycont1
mycont1
[root@dockers httpd]# docker exec -it mycont1 /bin/bash
Error: No such container: mycont1
[root@dockers httpd]# docker restart mycont1
Error response from daemon: No such container: mycont1
[root@dockers httpd]# docker ps|grep -i mycont1
[root@dockers httpd]# docker ps -algrep -i mycont1
[root@dockers httpd]#
```

*once container is deleted we cannot retrieve data. In simple data will be lost.*

Let's create our container with our own volume so that data will be persistent & we can retrieve the data even if the container crashes or gets delete using below.

### 1. Add a disk

The screenshot shows the AWS Management Console 'Volumes' page. A table lists several volumes, with 'mydockervol1' (ID: vol-0080e3a2a3f7f34bd, Size: 5 GiB) highlighted. Below the table, the 'Description' tab for 'mydockervol1' is open, showing details like Volume ID, Alarm status, Snapshot, Availability Zone, Encryption, and KMS Key ID. The 'Attachment information' section shows the volume is attached to the instance 'dockers' at '/dev/sdf'.

| Name         | Volume ID             | Size   | Volume Type | IOPS | Snapshot        | Created         |
|--------------|-----------------------|--------|-------------|------|-----------------|-----------------|
| mydockervol1 | vol-0080e3a2a3f7f34bd | 5 GiB  | gp2         | 100  |                 | October 8, 2020 |
|              | vol-0f778e8e5361e8e87 | 20 GiB | gp2         | 100  | snap-0299d08... | October 7, 2020 |
|              | vol-0c11995dc348d5e0f | 8 GiB  | gp2         | 100  | snap-0299d08... | October 2, 2020 |
|              | vol-0ccda7742b7d47cef | 8 GiB  | gp2         | 100  | snap-0299d08... | October 2, 2020 |
|              | vol-01fce8e5c5d769dd6 | 8 GiB  | gp2         | 100  | snap-0299d08... | October 2, 2020 |
|              | vol-03b8bb0411d9b8e3a | 8 GiB  | gp2         | 100  | snap-010d360... | October 1, 2020 |

Volumes: vol-0080e3a2a3f7f34bd (mydockervol1)

**Description** | Status Checks | Monitoring | Tags

Volume ID: vol-0080e3a2a3f7f34bd  
Alarm status: None  
Snapshot: -  
Availability Zone: us-east-1f  
Encryption: Not Encrypted  
KMS Key ID: -

Outposts ARN: -  
Size: 5 GiB  
Created: October 8, 2020 at 8:10:25 PM UTC+4  
State: in-use

Attachment information: i-0d6ae64abb40b108f (dockers) /dev/sdf (attached)  
Volume type: gp2  
Product codes: -

# DVS Technologies Aws & Devops

```
└─xvda1 202:1    0 20G 0 part /
[root@dockers httpd]# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
xvda        202:0    0 20G  0 disk
└─xvda1 202:1    0 20G  0 part /
xvdf        202:80   0   5G  0 disk
[root@dockers httpd]#
```

## 2. Create a file system:

```
[root@dockers httpd]# vgcreate myvg /dev/xvdf
Physical volume "/dev/xvdf" successfully created.
Volume group "myvg" successfully created.
[root@dockers httpd]# lvcreate -L +4G -n mylv myvg
Logical volume "mylv" created.
[root@dockers httpd]# mkfs.ext4 /dev/myvg/mylv
mke2fs 1.42.9 (28-Dec-2013)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
262144 inodes, 1048576 blocks
52428 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=1073741824
32 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done

[root@dockers httpd]# mkdir /app
[root@dockers httpd]# mount -t ext4 /dev/myvg/mylv /app
[root@dockers httpd]# df -hT /app
Filesystem            Type      Size  Used Avail Use% Mounted on
/dev/mapper/myvg-mylv ext4       3.9G  16M  3.6G   1% /app
[root@dockers httpd]#
```

## 3. Create a container mapping the volume to the above filesystem

```
[root@dockers httpd]# df -hT /app
Filesystem            Type      Size  Used Avail Use% Mounted on
/dev/mapper/myvg-mylv ext4       3.9G  16M  3.6G   1% /app
[root@dockers httpd]# docker run -it --name myvoll -v /app:/mydata centos:7 /bin/bash
[root@45a7911866ae /]# df -hT /mydata/
Filesystem            Type      Size  Used Avail Use% Mounted on
/dev/mapper/myvg-mylv ext4       3.9G  16M  3.6G   1% /mydata
[root@45a7911866ae /]#
```

*Container*

# DVS Technologies Aws & Devops

## 4. Testing the data existence

Let's create some file from the container:

Note: "-v"

```
[root@dockers httpd]# docker run -it --name myvoll -v /app:/mydata centos:7 /bin/bash
[root@45a7911866ae /]# df -hT /mydata/
Filesystem      Type      Size  Used Avail Use% Mounted on
/dev/mapper/myvg-mylv ext4      3.9G   16M   3.6G   1% /mydata
[root@45a7911866ae /]# pwd
/
[root@45a7911866ae /]# cd /mydata/
[root@45a7911866ae mydata]# echo "Hi I am from container" > test.txt
[root@45a7911866ae mydata]# cat test.txt
Hi I am from container
[root@45a7911866ae mydata]#
```

*note: I am inside container*

```
[root@dockers httpd]#
[root@dockers httpd]# ls -l /app/
total 20
drwx----- 2 root root 16384 Oct  8 16:13 lost+found
-rw-r--r-- 1 root root    23 Oct  8 16:17 test.txt
[root@dockers httpd]# cat /app/test.txt
Hi I am from container
[root@dockers httpd]# echo "Hi I am from the base machine" >> /app/test.txt
[root@dockers httpd]# cat /app/test.txt
Hi I am from container
Hi I am from the base machine
[root@dockers httpd]#
```

*note: I am in Base Machine*

```
[root@dockers httpd]# docker exec -it myvoll /bin/bash
[root@45a7911866ae /]# cat /mydata/test.txt
Hi I am from container
Hi I am from the base machine
[root@45a7911866ae /]#
```

*note: I am inside container*

```
[root@dockers httpd]#
[root@dockers httpd]# docker rm -f myvoll
myvoll
[root@dockers httpd]# docker ps -a|grep -i myvoll
[root@dockers httpd]# cat /app/test.txt
Hi I am from container
Hi I am from the base machine
[root@dockers httpd]#
```

# DVS Technologies Aws & Devops

## 6. Network Management

Gathering container details:

```
root@dockers ~]# docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS              NAMES
MES

root@dockers ~]#
root@dockers ~]# docker run -itd --name test1 centos:7 /bin/bash
4dea621413ea8f5cb3ece9f52ee61d15d7b3d32fd5cfd1d2709a5f6f2802d7180
root@dockers ~]# docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS              NAMES
4dea621413ea      centos:7            "/bin/bash"         2 seconds ago       Up 1 second         te
st1
root@dockers ~]# docker inspect test1|more
{
  "Id": "4dea621413ea8f5cb3ece9f52ee61d15d7b3d32fd5cfd1d2709a5f6f2802d7180",
  "Created": "2020-10-09T14:40:48.118118527Z",
  "Path": "/bin/bash",
  "Args": [],
  "State": {
    "Status": "running",
    "Running": true,
    "Paused": false,
    "Restarting": false,
    "OOMKilled": false,
    "Dead": false,
    "Pid": 5434,
    "ExitCode": 0,
    "Error": "",
    "StartedAt": "2020-10-09T14:40:48.810006458Z",
```

Let's check the ipaddress & gateway for the containers:

```
[root@dockers ~]# docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS              NAMES
b3b75033a2b5      centos:7            "/bin/bash"         2 minutes ago       Up About a minute   t
st2
4dea621413ea      centos:7            "/bin/bash"         11 minutes ago      Up 10 minutes       t
st1
[root@dockers ~]#
```

```
see docker --help
[root@dockers ~]# docker inspect test1|grep -w -e "IPAddress\|Gateway"
  "Gateway": "172.17.0.1",
  "IPAddress": "172.17.0.2",
  "Gateway": "172.17.0.1",
  "IPAddress": "172.17.0.2",
[root@dockers ~]# docker inspect test2|grep -w -e "IPAddress\|Gateway"
  "Gateway": "172.17.0.1",
  "IPAddress": "172.17.0.3",
  "Gateway": "172.17.0.1",
  "IPAddress": "172.17.0.3",
[root@dockers ~]# docker ps
```

*Gateway (docker0)*  
*Container IPs*

# DVS Technologies Aws & Devops

```
[root@dockers ~]# ifconfig -a
docker0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 172.17.0.1 netmask 255.255.0.0 broadcast 172.17.255.255
    inet6 fe80::42:90ff:fee5:f2b6 prefixlen 64 scopeid 0x20<link>
    ether 02:42:90:e5:f2:b6 txqueuelen 0 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 5 bytes 446 (446.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 9001
    inet 172.31.73.251 netmask 255.255.240.0 broadcast 172.31.79.255
    inet6 fe80::14de:28ff:feb2:5bfb prefixlen 64 scopeid 0x20<link>
    ether 16:de:28:b2:5b:fb txqueuelen 1000 (Ethernet)
    RX packets 1643 bytes 170457 (166.4 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1603 bytes 189504 (185.0 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 0 bytes 0 (0.0 B)
```

## Man Page for Docker:

```
[root@dockers ~]# man docker-network
[root@dockers ~]# man docker-network-create
[root@dockers ~]#
```

## Creating custom Adatper:

```
[root@dockers ~]# docker network ls
NETWORK ID          NAME                DRIVER              SCOPE
ef175ef7d845        bridge             bridge             local
1b1cc7b2895f        host              host              local
21242c38bfd3        none              null              local
[root@dockers ~]# docker network create \
--driver=bridge \
--subnet=10.1.0.0/16 \
--ip-range=10.1.4.0/24 \
--gateway=10.1.0.1 \
dvsbatch4
d18a942c62d5c13824549b7dedf599647182b06974e9e1b45a423396ee185232
[root@dockers ~]# docker network ls
NETWORK ID          NAME                DRIVER              SCOPE
ef175ef7d845        bridge             bridge             local
d18a942c62d5        dvsbatch4          bridge             local
1b1cc7b2895f        host              host              local
21242c38bfd3        none              null              local
```

## Command:

```
docker network create \
--driver=bridge \
--subnet=10.1.0.0/16 \
--ip-range=10.1.4.0/24 \
--gateway=10.1.0.1 \
dvsbatch4
```



# DVS Technologies Aws & Devops

```
[root@dockers ~]# docker network ls
NETWORK ID          NAME                DRIVER
ef175ef7d845        bridge             bridge
d18a942c62d5        dvsbatch4          bridge
1b1cc7b2895f        host               host
21242c38bfd3        none               null

[root@dockers ~]# ifconfig -a|more
br-118a942c62d5: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    inet 10.1.0.1 netmask 255.255.0.0 broadcast 10.1.0.255
    inet6 fe80::42:7aff:fe14:3a3d prefixlen 64 scopeid 0x20:::
    ether 02:42:7a:14:3a:3d txqueuelen 0 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

docker0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 172.17.0.1 netmask 255.255.0.0 broadcast 172.17.0.255
    inet6 fe80::42:90ff:fee5:f2b6 prefixlen 64 scopeid 0x20:::
    ether 02:42:90:e5:f2:b6 txqueuelen 0 (Ethernet)
```

## Inspecting adapter details:

```
[root@dockers ~]# docker network ls
NETWORK ID          NAME                DRIVER          SCOPE
ef175ef7d845        bridge             bridge          local
d18a942c62d5        dvsbatch4          bridge          local
1b1cc7b2895f        host               host            local
21242c38bfd3        none               null            local

[root@dockers ~]# docker inspect dvsbatch4
[
  {
    "Name": "dvsbatch4",
    "Id": "d18a942c62d5c13824549b7dedf599647182b06974e9e1b45a423396ee185232",
    "Created": "2020-10-09T14:59:31.216064363Z",
    "Scope": "local",
    "Driver": "bridge",
    "EnableIPv6": false,
    "IPAM": {
      "Driver": "default",
      "Options": {},
      "Config": [
        {
          "Subnet": "10.1.0.0/16",
          "IPRange": "10.1.4.0/24",
          "Gateway": "10.1.0.1"
        }
      ]
    },
    "Internal": false,
    "Attachable": false,
    "Containers": {}
  }
]
```

## Creating a container with custom network:

# DVS Technologies Aws & Devops

```
[root@dockers ~]# docker run -itd --name mynet1 --net dvsbatch4 centos:7 /bin/bash
30707655feded25398feca2e2e1d73eee3da2237db2597de0121a8f44ae4ab9a
[root@dockers ~]# docker inspect mynet1|grep -w -e "IPAddress\|Gateway"
"Gateway": "",
"IPAddress": "",
"Gateway": "10.1.0.1",
"IPAddress": "10.1.4.0",

[root@dockers ~]#
```

Untitled - Notepad

```
File Edit Format View Help
Ctrl
docker network create \
--driver=bridge \
--subnet=10.1.0.0/16 \
--ip-range=10.1.4.0/24 \
--gateway=10.1.0.1 \
dvsbatch4
```

Creating a container from custom ipaddress from the custom adapter:

```
[root@dockers ~]# docker run -itd --name mynet2 --net dvsbatch4 --ip 10.1.4.3 centos:7 /bin/bash
014b190a497ff6dfeab2d43fe0c88a7912b4b008d506f7ce4f3245939fcf5a6
[root@dockers ~]# docker inspect mynet2|grep -w -e "IPAddress\|Gateway"
"Gateway": "",
"IPAddress": "",
"Gateway": "10.1.0.1",
"IPAddress": "10.1.4.3",

[root@dockers ~]#
```

Untitled - Notepad

```
File Edit Format View Help
Ctrl
docker network create \
--driver=bridge \
--subnet=10.1.0.0/16 \
--ip-range=10.1.4.0/24 \
--gateway=10.1.0.1 \
dvsbatch4
```

Delete our containers & network:

Removing containers:

```
[root@dockers ~]# docker ps|grep -i net
014b190a497f centos:7 "/bin/bash" 2 minutes ago Up 2 minutes m
net2
30707655fede centos:7 "/bin/bash" 5 minutes ago Up 5 minutes m
net1
[root@dockers ~]# docker rm -f mynet1 mynet2
mynet1
mynet2
[root@dockers ~]# docker ps -a|grep -i net
[root@dockers ~]#
```

Removing the network:

# DVS Technologies Aws & Devops

```
[root@dockers ~]# docker network ls
NETWORK ID          NAME                DRIVER              SCOPE
ef175ef7d845        bridge              bridge              local
d18a942c62d5        dvsbatch4           bridge              local
1b1cc7b2895f        host                host                local
21242c38bfd3        none                null                local
[root@dockers ~]# docker network remove dvsbatch4
dvsbatch4
[root@dockers ~]# docker network ls
NETWORK ID          NAME                DRIVER              SCOPE
ef175ef7d845        bridge              bridge              local
1b1cc7b2895f        host                host                local
21242c38bfd3        none                null                local
[root@dockers ~]#
```

# DVS Technologies Aws & Devops

## 7. Monitoring Docker containers & housekeeping

Verifying CPU & Memory usage:

```
[root@dockers ~]# docker top test2
UID                PID                PPID                C                    STIME               TTY               TI
root               5568              5529               0                   14:19              pts/0              00
:00:00             /bin/bash

[root@dockers ~]# docker stats
CONTAINER ID        NAME                CPU %               MEM USAGE / LIMIT   MEM %               NET I/O
b3b75033a2b5       test2              0.01%              1.332MiB / 3.851GiB 0.03%              1.08kB / 0B
4dea621413ea       test1              0.01%              1.496MiB / 3.851GiB 0.04%              1.59kB / 0B
CONTAINER ID        NAME                CPU %               MEM USAGE / LIMIT   MEM %               NET I/O
BLOCK I/O          PIDS
```

Removing Active Containers:

```
[root@dockers ~]# docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              musing_benz
981b573429fb       centos:7           "/bin/bash"        4 seconds ago      Up 2 seconds       priceless_eng
9ffd1d0729d3       centos:7           "/bin/bash"        5 seconds ago      Up 3 seconds
lbart
abef689a51e5       centos:7           "/bin/bash"        5 seconds ago      Up 4 seconds       relaxed_chaply
gin
fcd8b5ec1360       centos:7           "/bin/bash"        6 seconds ago      Up 5 seconds       intelligent_sh
tern
6ee8ca97648d       centos:7           "/bin/bash"        7 seconds ago      Up 6 seconds       optimistic_sam
met
a0c6b4347a6f       centos:7           "/bin/bash"        17 seconds ago     Up 15 seconds      test2

[root@dockers ~]# docker ps -q
981b573429fb
9ffd1d0729d3
abef689a51e5
fcd8b5ec1360
6ee8ca97648d
a0c6b4347a6f

[root@dockers ~]# docker ps -q|wc -l
6

[root@dockers ~]# docker rm -f 981b573429fb 9ffd1d0729d3 abef689a51e5 fcd8b5ec1360 6ee8ca97648d a0c6b4347a6f
[root@dockers ~]# docker ps -q
981b573429fb
9ffd1d0729d3
abef689a51e5
fcd8b5ec1360
6ee8ca97648d
a0c6b4347a6f

[root@dockers ~]# echo $(docker ps -q)
981b573429fb 9ffd1d0729d3 abef689a51e5 fcd8b5ec1360 6ee8ca97648d a0c6b4347a6f
[root@dockers ~]# docker rm -f $(docker ps -q)
981b573429fb
9ffd1d0729d3
abef689a51e5
```

*Removing all my containers*

# DVS Technologies Aws & Devops

## Removing Passive Containers:

```
[root@dockers ~]# docker ps -a -q
9f8014f236f3
a34f6c23dd70
b85b058fd1bb
9cd6d3896da1
48adace71fdc
b5ecc7bd80da
33233298b216
5a1b1f47aa29
6efbde3e74b8
2fdd6373af2b
6aa41908fe8b
c41559056227
4474e19ce597
f089bf2c9c56
5c3a1a8dd646
53878541749d
bfad17d2231c
e5085ca1154f
91d81c30bcc8
ce266335ebdc
2118ebc1f6e9
039cd9dcf6e9
72880c892472
99abc69fd2a9
75e51f4eb8c8
[root@dockers ~]# docker rm -f $(docker ps -a -q)
```

**Note:** If you want to print only passive/Exited container id's then use the below  
`docker ps -a|grep -i exited|awk '{print $1}'`

## Removing Images:

**Command:** `docker rmi -f <imageid>`

```
[root@dockers ~]# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
dvsdevops            ENTRYPOINT         a633815fd6db       24 hours ago       215MB
dvsdevops            CMD                b5e5b9758840       24 hours ago       215MB
dvsdevops            EXPOSE            3641d32d9d95       24 hours ago       289MB
<none>               <none>             a383fffb152d4       24 hours ago       289MB
dvsdevops            USER             ccc492f4d52b       24 hours ago       296MB
dvsdevops            ENV               d691ef6fcd53       24 hours ago       296MB
dvsdevops            ADD               9dc96ff5d735       25 hours ago       296MB
dvsdevops            COPY              df4072186bc5       25 hours ago       296MB
dvsdevops            v1                01337634e6e6       25 hours ago       296MB
<none>               <none>             786e1e169665       25 hours ago       296MB
mycustomimage        v1                0e2342aecdd5       2 days ago         246MB
nginx                latest            992e3b7be046       3 days ago         133MB
centos                7                7e6257c9f8d8       8 weeks ago        203MB
centos                latest           0d120b6ccaa8       8 weeks ago        215MB
alpine                latest           a24bb4013296       4 months ago       5.57MB
[root@dockers ~]# docker rmi -f 0e2342aecdd5
Untagged: mycustomimage:v1
Deleted: sha256:0e2342aecdd5ac9c2a9e9985e23f7216f67a2c57818668f71196f3c8514a0e7
Deleted: sha256:08c2805893b3d7fe990dff29ed897ec55add002e545500ccae72ab3cf6c6e620
[root@dockers ~]# docker images|grep -i mycustomimage
[root@dockers ~]#
```

# DVS Technologies Aws & Devops

Want to delete all the images except centos:

```
[root@dockers ~]# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
dvsdevops            ENTRYPOINT          a633815fd6db       24 hours ago       215MB
dvsdevops            CMD                 b5e5b9758840       24 hours ago       215MB
dvsdevops            EXPOSE             3641d32d9d95       24 hours ago       289MB
<none>               <none>             a383ffb152d4       24 hours ago       289MB
dvsdevops            USER              ccc492f4d52b       25 hours ago       296MB
dvsdevops            ENV               d691ef6fcd53       25 hours ago       296MB
dvsdevops            ADD               9dc96ff5d735       25 hours ago       296MB
dvsdevops            COPY              df4072186bc5       25 hours ago       296MB
dvsdevops            v1                01337634ebe6       25 hours ago       296MB
<none>               <none>             786e1e169665       25 hours ago       296MB
nginx                latest            992e3b7be046       3 days ago         133MB
centos                7                7e6257c9f8d8       8 weeks ago        203MB
centos                latest           0d120b6ccaa8       8 weeks ago        215MB
alpine               latest           a24bb4013296       4 months ago       5.57MB

[root@dockers ~]# docker images|grep -v -e "centos\|REPOSITORY"|awk '{print $3}'
a633815fd6db
b5e5b9758840
3641d32d9d95
a383ffb152d4
ccc492f4d52b
d691ef6fcd53
9dc96ff5d735
df4072186bc5
01337634ebe6
786e1e169665
992e3b7be046
a24bb4013296

[root@dockers ~]# echo $(docker images|grep -v -e "centos\|REPOSITORY"|awk '{print $3}')
```

```
[root@dockers ~]# docker rmi -f $(docker images|grep -v -e "centos\|REPOSITORY"|awk '{print $3}')
Untagged: dvsdevops:ENTRYPOINT
Deleted: sha256:a633815fd6db0f1d472729f5352301bf9ed3f1be5004d20741e6cb9324351c39
Untagged: dvsdevops:CMD
Deleted: sha256:b5e5b975884021c4ae6a64982bf1df96f6e786bb7b1a97a19b969371516cf
Deleted: sha256:e181a0a98da64927e7b6031f0053f0b43e7d1899f09096799f2a1bce15526a44
Deleted: sha256:ba1c69cd69a0566a8722d55f000089461392029447557ec649a42a802858ebe1
Untagged: dvsdevops:EXPOSE
Deleted: sha256:3641d32d9d95c27c88815db05f162077de79fa0eae940b00b1ea594db3742e83
Deleted: sha256:b32d539384d1a7147266f1d3b8c5ce6a964f056bb40868b74741cc65394e032c
Deleted: sha256:a383ffb152d4795c8880478669700a7f38d4d34cecaaedabd6a805eda1946635
Deleted: sha256:6da3907a60df5b3e36757351989f83ac8f8418963522803e00fd4daa7dbbe13f
Deleted: sha256:72ba07fec68973c8b6daef227fd73fadba4a7f7d5f9ee43c28a3bbd9940ce9f94
Untagged: dvsdevops:USER
Deleted: sha256:ccc492f4d52b9a97fe00a64d9486daac49201bd6acc0788349c7af916df196e1
Untagged: dvsdevops:ENV
Deleted: sha256:d691ef6fcd538d8f952e4c882e75c186f457ea2ce45d37a8e36600a4cbfa9c8b
Untagged: dvsdevops:ADD
Deleted: sha256:9dc96ff5d735cf3aed6fbf625b008b36de710f6a54030fce79c9b176381bb7f0
Deleted: sha256:b29c2dbd10e5a91b1a26f687b2b368bca12c068a9fa6719d308e0ce76b8e3bbb
Untagged: dvsdevops:COPY
```

# DVS Technologies Aws & Devops

## 8. Working with docker registry

Let's build our own image & perform the below.

```
[root@dockers myimage]# vi Dockerfile
[root@dockers myimage]# cat Dockerfile
FROM centos
RUN useradd dvsbatch4 && yum install telnet -y
USER dvsbatch4

[root@dockers myimage]# docker build -t "mycustomimage:v1" .
Sending build context to Docker daemon 2.048kB
Step 1/3 : FROM centos
--> 0d120b6ccaa8
Step 2/3 : RUN useradd dvsbatch4 && yum install telnet -y
--> Running in 1d4ddal26cbc
CentOS-8 - AppStream          22 MB/s | 5.8 MB   00:00
CentOS-8 - Base              10 MB/s | 2.2 MB   00:00
CentOS-8 - Extras            175 kB/s | 8.0 kB  00:00
Dependencies resolved.
=====
Package      Architecture Version                      Repository      Size
=====
Installing:
telnet       x86_64      1:0.17-73.el8_1.1           AppStream       72 k
Transaction Summary
=====
Install 1 Package
Total download size: 72 k
Installed size: 153 k
Downloading Packages:
telnet-0.17-73.el8_1.1.x86_64.rpm          12 MB/s | 72 kB   00:00
--> Running in dfalbb28ba39
Removing intermediate container dfalbb28ba39
--> 6e628cb66029
Successfully built 6e628cb66029
Successfully tagged mycustomimage:v1
[root@dockers myimage]# docker images|grep -i custom
mycustomimage    v1                6e628cb66029      56 seconds ago    246MB
[root@dockers myimage]#
```

Testing my image configuration via a container:

```
[root@dockers myimage]#
[root@dockers myimage]# docker images|grep -i custom
mycustomimage    v1                6e628cb66029      About a minute ago    246MB
[root@dockers myimage]# docker run -it --name mycustomtest1 mycustomimage:v1 /bin/bash
[dvsbatch4@87968aeef02f /]$ id -a
uid=1000(dvsbatch4) gid=1000(dvsbatch4) groups=1000(dvsbatch4)
[dvsbatch4@87968aeef02f /]$ telnet
telnet> q
[dvsbatch4@87968aeef02f /]$
```

Let's save out image as below:

### 1st way via tar:

1. docker save -o myimage.tar mycustom
2. docker load -i myimage.tar

# DVS Technologies Aws & Devops

Creating a backup from the existing image:

```
[root@dockers myimage]# docker images
REPOSITORY    TAG       IMAGE ID      CREATED        SIZE
mycustomimage  v1        6e628cb66029 3 minutes ago  246MB
centos         7         7e6257c9f8d8 8 weeks ago    203MB
centos         latest    0d120b6ccaa8 8 weeks ago    215MB

[root@dockers myimage]# docker save -o mycustombackup.tar 6e628cb66029
[root@dockers myimage]# ls -l mycustombackup.tar
-rw-r--r-- 1 root root 253436416 Oct  9 16:01 mycustombackup.tar
[root@dockers myimage]# gzip mycustombackup.tar
[root@dockers myimage]# ls -l mycustombackup.tar.gz
-rw-r--r-- 1 root root 90930068 Oct  9 16:01 mycustombackup.tar.gz
[root@dockers myimage]# du -sh mycustombackup.tar.gz
87M    mycustombackup.tar.gz
[root@dockers myimage]#
```

Verification:

```
[root@dockers myimage]# docker images
REPOSITORY    TAG       IMAGE ID      CREATED        SIZE
mycustomimage  v1        6e628cb66029 3 minutes ago  246MB
centos         7         7e6257c9f8d8 8 weeks ago    203MB
centos         latest    0d120b6ccaa8 8 weeks ago    215MB

[root@dockers myimage]# docker rm -f 87968aee02f
87968aee02f
[root@dockers myimage]# docker rmi -f 6e628cb66029
Deleted: sha256:6e628cb66029cf8ba35ee62f200c980242cbb82ad077750a416a7ae5793c5a5a
Deleted: sha256:1a10200708689aa6018977bf57a0449aed1c79896e7de44f2cccd2be2a164023
Deleted: sha256:2132d5f33048b0c9497a81e4dad1e510e29a4eb9d113614fdced5814fd6f8de4
[root@dockers myimage]# docker images
REPOSITORY    TAG       IMAGE ID      CREATED        SIZE
centos         7         7e6257c9f8d8 8 weeks ago    203MB
centos         latest    0d120b6ccaa8 8 weeks ago    215MB
[root@dockers myimage]#
```

*Note: Before you delete image make sure you are deleting depend containers first*

Now load the image to the system using below.

```
@6ca5008c0673/
[... Options Remote Control Drawing 1c ...]

[root@dockers myimage]# docker images
REPOSITORY    TAG       IMAGE ID      CREATED        SIZE
centos         7         7e6257c9f8d8 8 weeks ago    203MB
centos         latest    0d120b6ccaa8 8 weeks ago    215MB

[root@dockers myimage]# ls -l mycustombackup.tar.gz
-rw-r--r-- 1 root root 90930068 Oct  9 16:01 mycustombackup.tar.gz
[root@dockers myimage]# docker load -i mycustombackup.tar.gz
0f463ff85269: Loading layer [=====>] 31.07MB/31.07MB
Loaded image ID: sha256:6e628cb66029cf8ba35ee62f200c980242cbb82ad077750a416a7ae5793c5a5a
[root@dockers myimage]# docker images
REPOSITORY    TAG       IMAGE ID      CREATED        SIZE
<none>        <none>    6e628cb66029 9 minutes ago  246MB
centos         7         7e6257c9f8d8 8 weeks ago    203MB
centos         latest    0d120b6ccaa8 8 weeks ago    215MB

[root@dockers myimage]# docker run -it --name mycustomtest2 6e628cb66029 /bin/bash
[dvsbatch4@6ca5008c0673 /]$ id -a
uid=1000(dvsbatch4) gid=1000(dvsbatch4) groups=1000(dvsbatch4)
[dvsbatch4@6ca5008c0673 /]$ telnet
telnet> q
[dvsbatch4@6ca5008c0673 /]$
```

## 2nd Way via Dockerhub:

1. Customize one docker image and try to tag it & push it to docker hub
2. Save & load the images locally



# DVS Technologies Aws & Devops

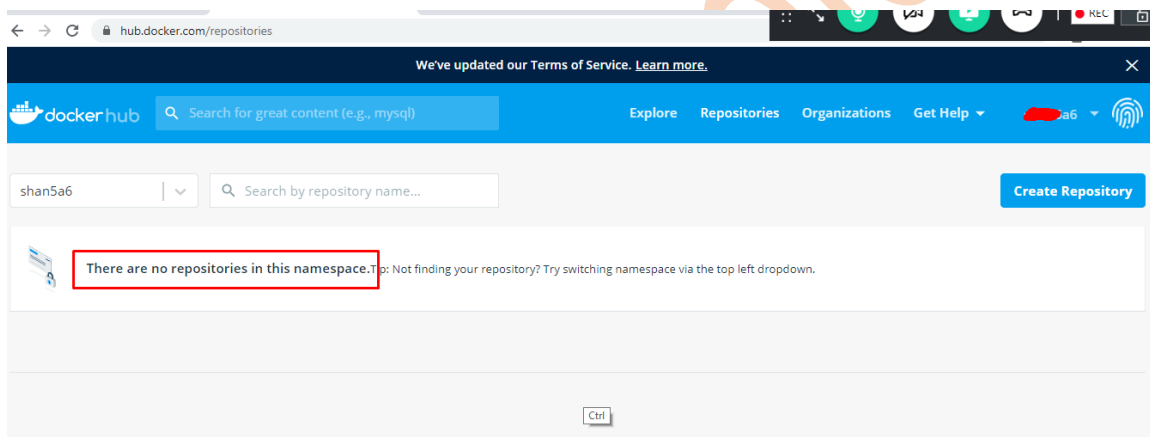
3. Remove the local image & pull the dockerhub image pushed from step 1 & show the output.

Note: User docker login & logout to login to the dockerhub

Login to the dockerhub:

```
[root@dockers myimage]# docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
mycustomimage  v1        16bd7eeb2396   10 seconds ago 246MB
centos         7         7e6257c9f8d8   8 weeks ago   203MB
centos         latest    0d120b6ccaa8   8 weeks ago   215MB
[root@dockers myimage]# docker login <registryurl>
[root@dockers myimage]# docker login
Login with your Docker ID to push and pull images from Docker Hub. If you don't have a Docker ID, head over to https://hub.docker.com to create one.
Username: shan5a6
Password:
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store
Login Succeeded
```

Before pushing image:

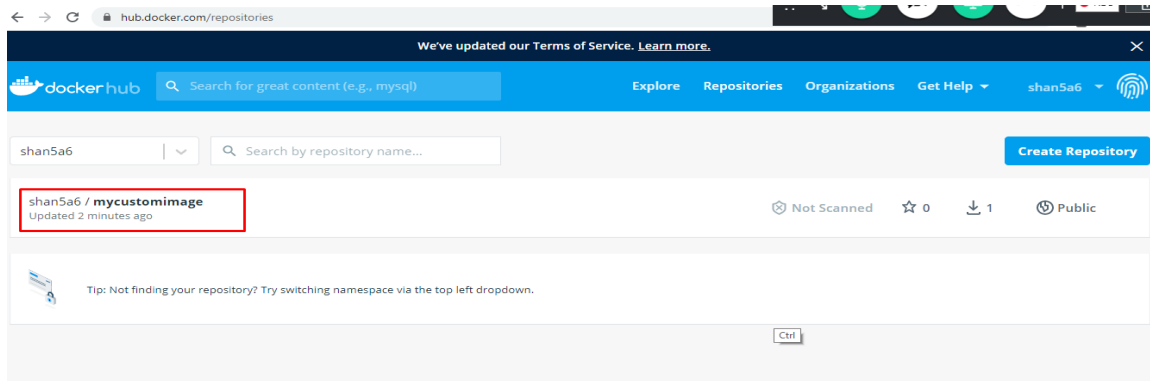


Pushing image to dockerhub:

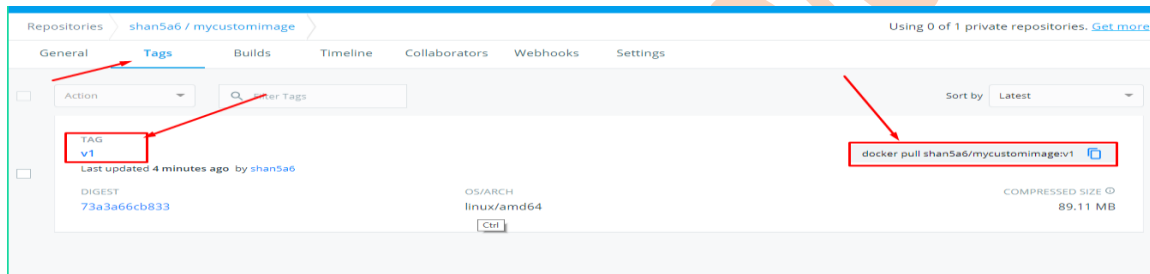
```
[root@dockers myimage]# docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
mycustomimage  v1        16bd7eeb2396   5 minutes ago 246MB
centos         7         7e6257c9f8d8   8 weeks ago   203MB
centos         latest    0d120b6ccaa8   8 weeks ago   215MB
[root@dockers myimage]# docker tag mycustomimage:v1 shan5a6/mycustomimage:v1
[root@dockers myimage]# docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
mycustomimage  v1        16bd7eeb2396   5 minutes ago 246MB
shan5a6/mycustomimage  v1        16bd7eeb2396   5 minutes ago 246MB
centos         7         7e6257c9f8d8   8 weeks ago   203MB
centos         latest    0d120b6ccaa8   8 weeks ago   215MB
[root@dockers myimage]# docker push shan5a6/mycustomimage:v1
The push refers to repository [docker.io/shan5a6/mycustomimage]
ec491f94c6ef: Pushed
291f6e44771a: Mounted from library/centos
v1: digest: sha256:73a3a66cb83330d0dd79899f19bde99cb5a6a29e78180769cb47baf9a337ea7741
[root@dockers myimage]# docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
mycustomimage  v1        16bd7eeb2396   5 minutes ago 246MB
shan5a6/mycustomimage  v1        16bd7eeb2396   5 minutes ago 246MB
centos         7         7e6257c9f8d8   8 weeks ago   203MB
centos         latest    0d120b6ccaa8   8 weeks ago   215MB
[root@dockers myimage]#
```

# DVS Technologies Aws & Devops

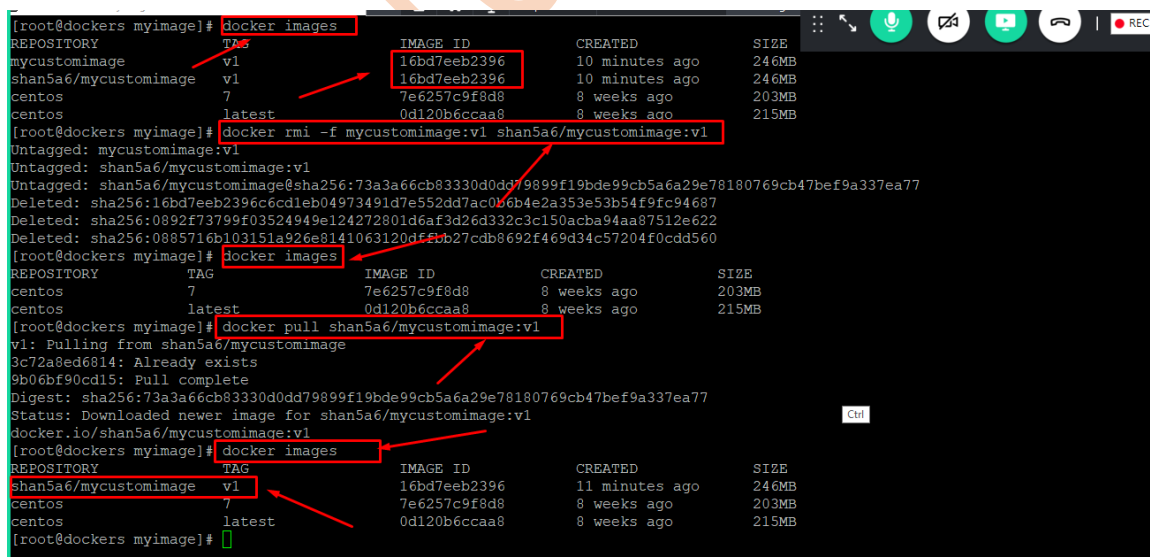
Now if you verify in the dockerhub console:



Verify the tags for the pushed image:

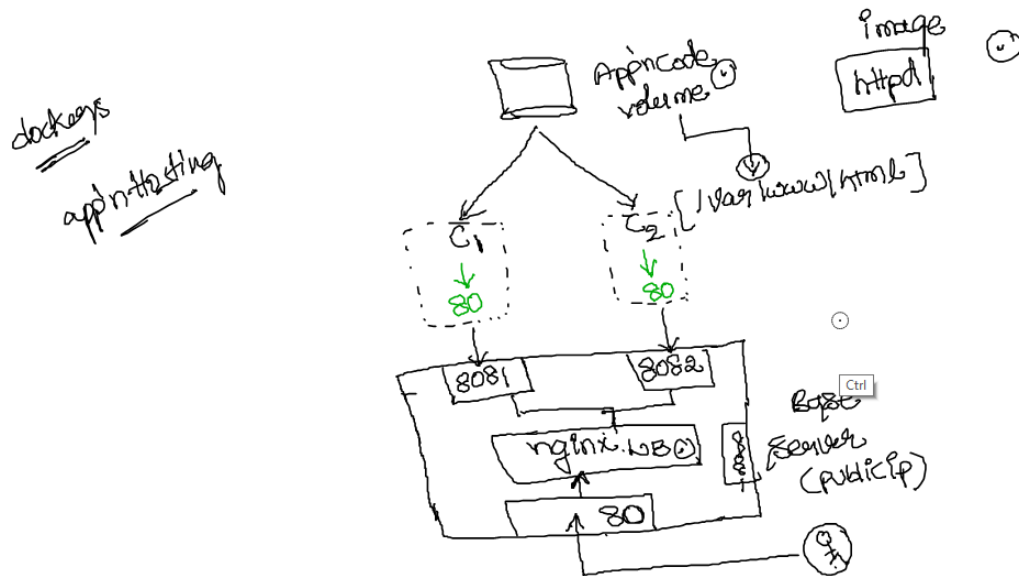


Final testing:



# DVS Technologies Aws & Devops

## 9. Application Hosting



Let's build our own apache image:

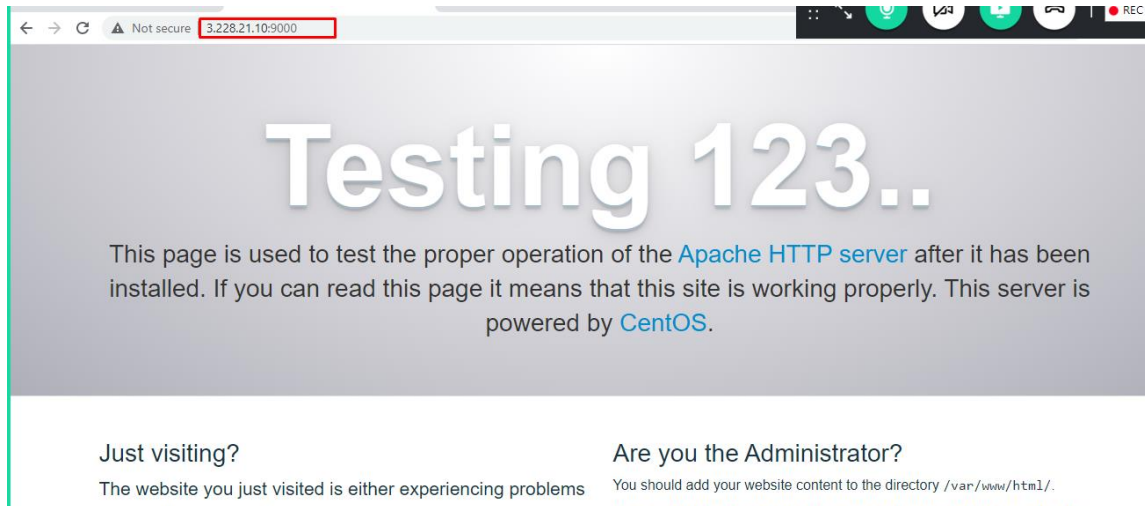
```
[root@dockers ~]#  
[root@dockers ~]# mkdir myapache  
[root@dockers ~]# cd myapache/  
[root@dockers myapache]# ls -l  
total 0  
[root@dockers myapache]# vi Dockerfile  
[root@dockers myapache]#  
[root@dockers myapache]# cat Dockerfile  
FROM centos:7  
VOLUME ["/var/www/html"]  
EXPOSE 80  
RUN yum install httpd -y  
CMD ["/usr/sbin/httpd","-D","FOREGROUND"]  
  
[root@dockers myapache]# docker build -t "myapache:v1" .  
Sending build context to Docker daemon 2.048KB  
Step 1/5 : FROM centos:7  
----> 7e6257c9f8d8  
Step 2/5 : VOLUME ["/var/www/html"]  
----> Running in fff52308e95f  
Removing intermediate container fff52308e95f  
----> 394aaf9e7010  
Step 3/5 : EXPOSE 80  
----> Running in fa58784fee67
```

Testing our docker image:

```
testapache  
[root@dockers myapache]# docker run -d --name testapache -p 9000:80 myapache:v1  
f01be2fe4aa62de00853ad5b062292af4567a78cdd8999135b8102ff583da0ac  
[root@dockers myapache]# docker ps  
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS  
f01be2fe4aa6        myapache:v1        "/usr/sbin/httpd -D ..." 2 seconds ago       Up 1 second        0.0.0.0:9000->  
/tcp testapache  
[root@dockers myapache]#
```

note: 9000 is base machine port

# DVS Technologies Aws & Devops



Let's configure out application as defined above:

Downloading application code:

```
[root@dockers myapache]#  
[root@dockers myapache]# mkdir /myapp  
[root@dockers myapache]# cd /myapp/  
[root@dockers myapp]# wget -O /tmp/bluefreedom3.zip http://static.oswd.org/designs/3682/bluefreedom3.zip  
--2020-10-10 10:07:10-- http://static.oswd.org/designs/3682/bluefreedom3.zip  
Resolving static.oswd.org (static.oswd.org)... 99.84.185.170, 99.84.185.64, 99.84.185.77, ...  
Connecting to static.oswd.org (static.oswd.org)|99.84.185.170:80... connected.  
HTTP request sent, awaiting response... 200 OK  
Length: 13412 (13K) [application/zip]  
Saving to: '/tmp/bluefreedom3.zip'  
  
100%[=====>] 13,412 --.-K/s in 0s  
  
2020-10-10 10:07:10 (200 MB/s) - '/tmp/bluefreedom3.zip' saved [13412/13412]  
  
[root@dockers myapp]# unzip -d /myapp /tmp/bluefreedom3.zip  
Archive: /tmp/bluefreedom3.zip  
  creating: /myapp/bluefreedom3/images/  
    inflating: /myapp/bluefreedom3/images/bott.jpg  
    inflating: /myapp/bluefreedom3/images/hidr.jpg  
    inflating: /myapp/bluefreedom3/images/midd.jpg  
    inflating: /myapp/bluefreedom3/images/minimal.jpg  
    inflating: /myapp/bluefreedom3/images/topp.jpg  
    inflating: /myapp/bluefreedom3/index.html  
    inflating: /myapp/bluefreedom3/style.css  
[root@dockers myapp]# ls -l  
total 0  
drwxr-xr-x 3 root root 55 Oct 10 10:07 bluefreedom3  
[root@dockers myapp]# mv bluefreedom3/* .  
[root@dockers myapp]# ls -l  
total 8  
drwxr-xr-x 2 root root 6 Oct 10 10:07 bluefreedom3  
drwxr-xr-x 2 root root 89 Jan 18 2007 images
```

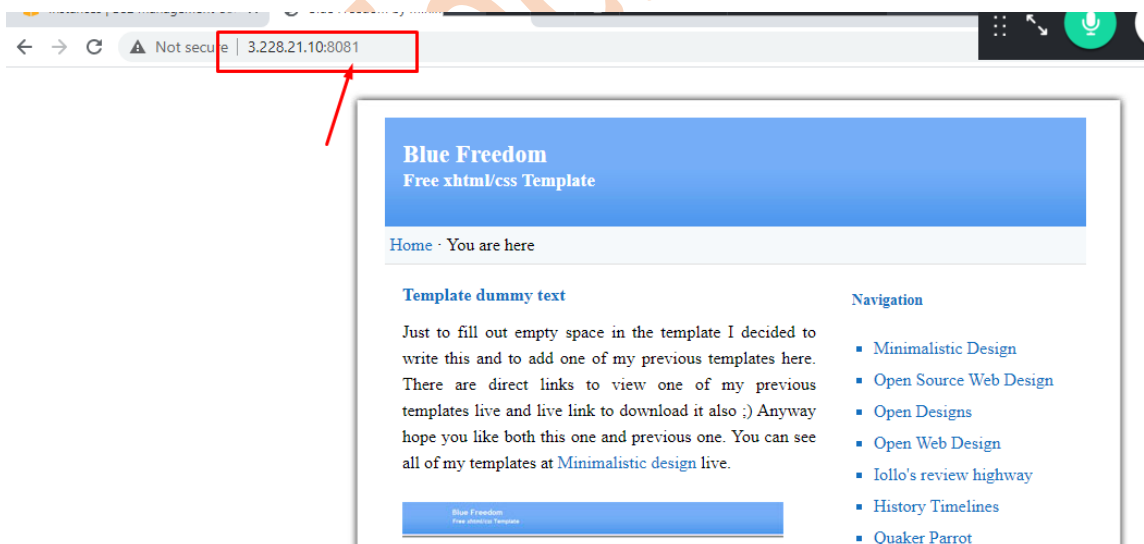
# DVS Technologies Aws & Devops

```
[root@dockers myapp]# ls -l
total 8
drwxr-xr-x 2 root root 6 Oct 10 10:07 bluefreedom3
drwxr-xr-x 2 root root 89 Jan 18 2007 images
-rw-r--r-- 1 root root 2078 Jan 18 2007 index.html
-rw-r--r-- 1 root root 1628 Jan 8 2007 style.css
[root@dockers myapp]# rm -rf bluefreedom3
[root@dockers myapp]# ls -l
total 8
drwxr-xr-x 2 root root 89 Jan 18 2007 images
-rw-r--r-- 1 root root 2078 Jan 18 2007 index.html
-rw-r--r-- 1 root root 1628 Jan 8 2007 style.css
[root@dockers myapp]# pwd
/myapp
[root@dockers myapp]#
```

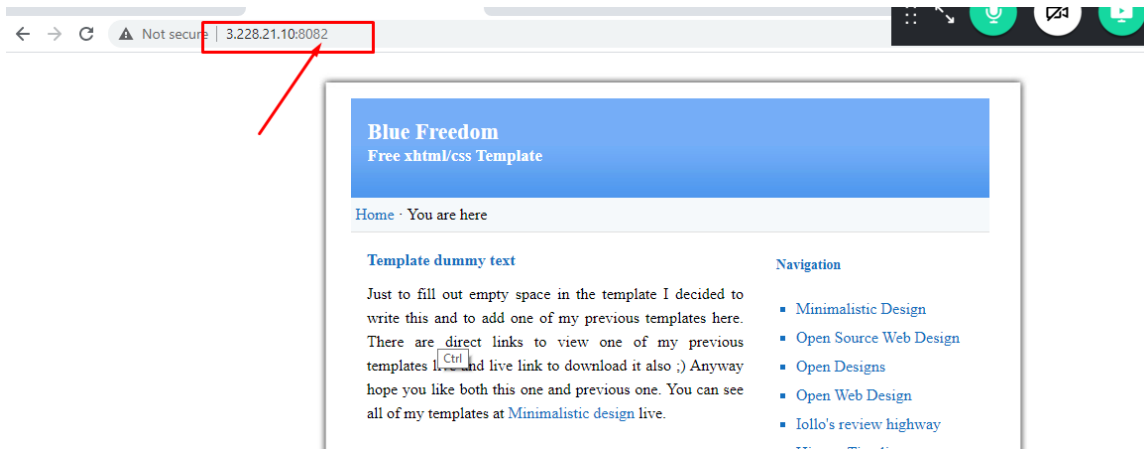
Let's spin up the containers:

```
[root@dockers myapp]# ls -l /myapp/
total 8
drwxr-xr-x 2 root root 89 Jan 18 2007 images
-rw-r--r-- 1 root root 2078 Jan 18 2007 index.html
-rw-r--r-- 1 root root 1628 Jan 8 2007 style.css
[root@dockers myapp]# docker images|grep -i myap
myapache          v1                29f4acc63068      10 minutes ago      328MB
[root@dockers myapp]# docker run -d --name myweb1 -p 8081:80 -v /myapp:/var/www/html/ myapache:v1
b002b06b3c4699ec08195fba775304415de3ed3982bb53d4f1123e5bbc21b408
[root@dockers myapp]# docker run -d --name myweb2 -p 8082:80 -v /myapp:/var/www/html/ myapache:v1
0443eda08c14fc0fb964bdd23b328aee07565e959802862c318d6d025b8799a4
[root@dockers myapp]# docker ps|grep -i myweb
0443eda08c14      myapache:v1      "/usr/sbin/httpd -D ..." 7 seconds ago      Up 6 seconds      0.0.0.0:8082->80/
tcp      myweb2
b002b06b3c46      myapache:v1      "/usr/sbin/httpd -D ..." 24 seconds ago      Up 24 seconds      0.0.0.0:8081->80/
tcp      myweb1
[root@dockers myapp]#
```

Testing the application on containers:



# DVS Technologies Aws & Devops



Now let's bring the Nginx LB on top of these two containers:

**amazon-linux-extras install nginx1.12 -y**  
**systemctl enable nginx**  
**systemctl restart nginx**

```
[root@dockers myapp]# vim /etc/nginx/conf.d/default.conf
[root@dockers myapp]# cat /etc/nginx/conf.d/default.conf
upstream containerapp {
    server 3.228.21.10:8081;
    server 3.228.21.10:8082;
}

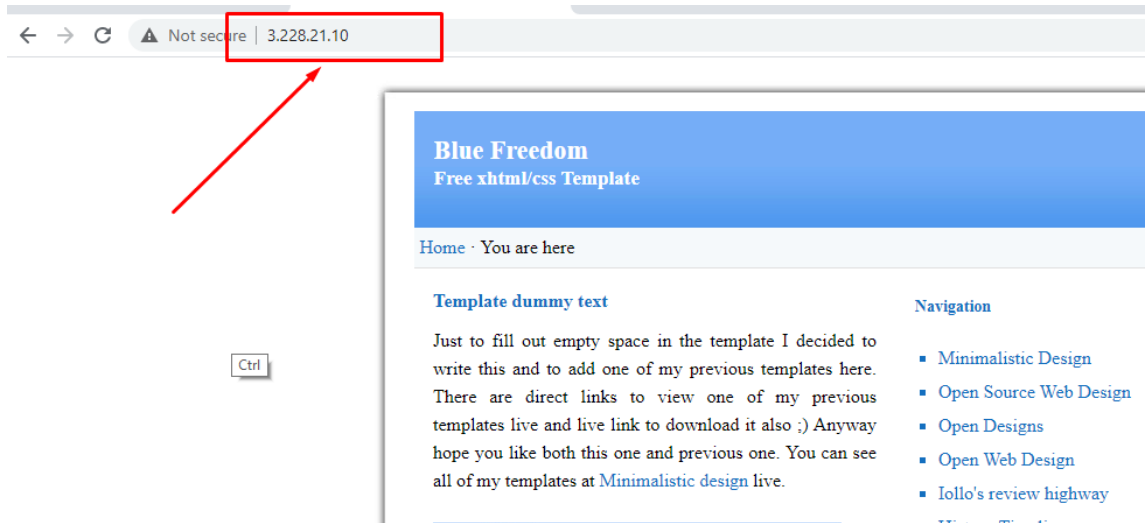
server {
    listen *:80;

    server_name 3.228.21.10;
    index index.html index.htm index.php;

    access_log /var/log/nginx/localweb.log;
    error_log /var/log/nginx/localerr.log;

    location / {
        proxy_pass http://containerapp;
    }
}
[root@dockers myapp]# systemctl restart nginx
[root@dockers myapp]#
```

# DVS Technologies Aws & Devops



## Testing our Application:

```
[root@dockers myapp]# docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS
a4e95dd5f3bd   myapache:v1  "/usr/sbin/httpd -D ..."  28 seconds ago  Up 1 second   0.0.0.0:8081->80
tcp myweb1
b87254d686df   myapache:v1  "/usr/sbin/httpd -D ..."  36 seconds ago  Up 35 seconds  0.0.0.0:8082->80
tcp myweb2
[root@dockers myapp]# docker stop myweb1
myweb1
[root@dockers myapp]# docker stop myweb2
myweb2
[root@dockers myapp]# docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS        NAMES
```



# DVS Technologies Aws & Devops

```
[root@dockers myapp]# docker start myweb1
myweb1
[root@dockers myapp]# docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS
a4e95dd5f3bd       myapache:v1        "/usr/sbin/httpd -D ..."  54 seconds ago      Up 7 seconds        0.0.0.0:8081->80/
tcp_myweb1
[root@dockers myapp]# docker start myweb2
myweb2
[root@dockers myapp]# docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS
a4e95dd5f3bd       myapache:v1        "/usr/sbin/httpd -D ..."  59 seconds ago      Up 12 seconds        0.0.0.0:8081->80/
tcp_myweb1
b87254d686df       myapache:v1        "/usr/sbin/httpd -D ..."  About a minute ago  Up 1 second         0.0.0.0:8082->80/
tcp_myweb2
[root@dockers myapp]#
```





# DVS Technologies Aws & Devops

## 10. Migrating on prem Java Based applications to Dockers

Source code you can download from

["https://github.com/shan5a6/javaDockerDeployment.git"](https://github.com/shan5a6/javaDockerDeployment.git)

### Base Image for the infrastructure:

Base Image for the infrastructure ::

Dockerfile ::

```
FROM centos:7
# Installing Java
ENV JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk
ENV PATH=$PATH:$JAVA_HOME
RUN yum install java-1.8.0-openjdk-devel wget -y
EXPOSE 8080

# Installing Maven
ENV Mvn_Version=3.6.3
ENV M2_HOME=/usr/local/apache-maven/apache-maven-${Mvn_Version}
ENV M2="`${M2_HOME}/bin"
ENV PATH=$PATH:$M2

RUN wget https://downloads.apache.org/maven/maven-3/${Mvn_Version}/binaries/apache-maven-${Mvn_Version}-bin.tar.gz && \
    tar xvfz apache-maven-${Mvn_Version}-bin.tar.gz && \
    mkdir /usr/local/apache-maven/apache-maven-${Mvn_Version} -p && \
    mv apache-maven-${Mvn_Version}/* /usr/local/apache-maven/apache-maven-${Mvn_Version}/

# Installing and configuring Tomcat
ENV Tomcat_Version=8.5.54
RUN wget http://www-eu.apache.org/dist/tomcat/tomcat-8/v${Tomcat_Version}/bin/apache-tomcat-${Tomcat_Version}.tar.gz && \
    tar xvfz apache-tomcat-${Tomcat_Version}.tar.gz && \
    mkdir -p /opt/tomcat/ /opt/myapplication/ -p && \
    mv apache-tomcat-${Tomcat_Version}.tar.gz /tmp/ && \
    mv apache-tomcat-${Tomcat_Version}/* /opt/tomcat/
COPY context.xml /opt/tomcat/webapps/manager/META-INF/
COPY tomcat-users.xml /opt/tomcat/conf/
CMD ["/opt/tomcat/bin/catalina.sh", "run"]
```

### Dockerfile:

FROM centos:7

# Installing Java

ENV JAVA\_HOME=/usr/lib/jvm/java-1.8.0-openjdk

ENV PATH=\$PATH:\$JAVA\_HOME

RUN yum install java-1.8.0-openjdk-devel wget -y

EXPOSE 8080

# Installing Maven

ENV Mvn\_Version=3.6.3

ENV M2\_HOME=/usr/local/apache-maven/apache-maven-\${Mvn\_Version}

ENV M2="`\${M2\_HOME}/bin"

ENV PATH=\$PATH:\$M2

# DVS Technologies Aws & Devops

```
RUN wget https://downloads.apache.org/maven/maven-3/\${Mvn\_Version}/binaries/apache-maven-\${Mvn\_Version}-bin.tar.gz && \
tar xvfz apache-maven-${Mvn_Version}-bin.tar.gz && \
mkdir /usr/local/apache-maven/apache-maven-${Mvn_Version} -p && \
mv apache-maven-${Mvn_Version}/* /usr/local/apache-maven/apache-maven-${Mvn_Version}/
```

## # Installing and configuring Tomcat

ENV Tomcat\_Version=8.5.54

```
RUN wget http://www-eu.apache.org/dist/tomcat/tomcat-8/v\${Tomcat\_Version}/bin/apache-tomcat-\${Tomcat\_Version}.tar.gz && \
tar xvfz apache-tomcat-${Tomcat_Version}.tar.gz && \
mkdir -p /opt/tomcat/ /opt/myapplication/ -p && \
mv apache-tomcat-${Tomcat_Version}.tar.gz /tmp/ && \
mv apache-tomcat-${Tomcat_Version}/* /opt/tomcat/.
```

COPY context.xml /opt/tomcat/webapps/manager/META-INF/

COPY tomcat-users.xml /opt/tomcat/conf/

CMD ["/opt/tomcat/bin/catalina.sh", "run"]

## Application Image for developers:

Application Image for developers ::

Dockerfile ::

```
FROM myapppbaseimage
WORKDIR /opt/myapplication
RUN yum install git -y \
    && git clone https://github.com/shan5a6/myweb.git /opt/myapplication \
    && mvn clean install \
    && mv ./target/myweb*.war /opt/tomcat/webapps/app.war
CMD ["/opt/tomcat/bin/catalina.sh", "run"]
```

## Dockerfile:

FROM myapppbaseimage

WORKDIR /opt/myapplication

RUN yum install git -y \

&& git clone <https://github.com/shan5a6/myweb.git> /opt/myapplication \

&& mvn clean install \

&& mv ./target/myweb\*.war /opt/tomcat/webapps/app.war

CMD ["/opt/tomcat/bin/catalina.sh", "run"]

# DVS Technologies Aws & Devops

## 11. Multistage Builds

Multi stage build helps to reduce the size of the images i.e., we are gonna divide our docker images into build & run images.

```
FROM centos:7 as build
# Installing Java
ENV JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk
ENV PATH=$PATH:$JAVA_HOME
RUN yum install java-1.8.0-openjdk-devel wget git -y
# Installing Maven
ENV Mvn_Version=3.6.3
ENV M2_HOME=/usr/local/apache-maven/apache-maven-${Mvn_Version}
ENV M2="${M2_HOME}/bin"
ENV PATH=$PATH:$M2
RUN wget https://downloads.apache.org/maven/maven-3/${Mvn_Version}/binaries/apache-maven-${Mvn_Version}-bin.tar.gz && \
    tar xvfz apache-maven-${Mvn_Version}-bin.tar.gz && \
    mkdir /usr/local/apache-maven/apache-maven-${Mvn_Version} /opt/myapplication/ -p && \
    mv apache-maven-${Mvn_Version}/* /usr/local/apache-maven/apache-maven-${Mvn_Version}/ && \
    git clone https://github.com/shan5a6/myweb.git /opt/myapplication/
WORKDIR /opt/myapplication/
RUN mvn clean install

# Installing and configuring Tomcat
FROM centos:7
EXPOSE 8080
ENV Tomcat_Version=8.5.59

RUN yum install java-1.8.0-openjdk-devel wget -y

RUN wget http://www-eu.apache.org/dist/tomcat/tomcat-8/v${Tomcat_Version}/bin/apache-tomcat-${Tomcat_Version}.tar.gz && \
    tar xvfz apache-tomcat-${Tomcat_Version}.tar.gz && \
    mkdir -p /opt/tomcat/ /opt/myapplication/ -p && \
    mv apache-tomcat-${Tomcat_Version}.tar.gz /tmp/ && \
    mv apache-tomcat-${Tomcat_Version}/* /opt/tomcat/
COPY context.xml /opt/tomcat/webapps/manager/META-INF/
COPY tomcat-users.xml /opt/tomcat/conf/
COPY --from=build /opt/myapplication/target/myweb-0.12.0.war /opt/tomcat/webapps/myapp.war
CMD ["/opt/tomcat/bin/catalina.sh", "run"]
```

Code:

**FROM centos:7 as build**

**# Installing Java**

**ENV JAVA\_HOME=/usr/lib/jvm/java-1.8.0-openjdk**

**ENV PATH=\$PATH:\$JAVA\_HOME**

**RUN yum install java-1.8.0-openjdk-devel wget git -y**

**# Installing Maven**

**ENV Mvn\_Version=3.6.3**

**ENV M2\_HOME=/usr/local/apache-maven/apache-maven-\${Mvn\_Version}**

**ENV M2="\${M2\_HOME}/bin"**

**ENV PATH=\$PATH:\$M2**

# DVS Technologies Aws & Devops

```
RUN wget https://downloads.apache.org/maven/maven-3/\${Mvn\_Version}/binaries/apache-maven-\${Mvn\_Version}-bin.tar.gz && \
tar xvfz apache-maven-${Mvn_Version}-bin.tar.gz && \
mkdir /usr/local/apache-maven/apache-maven-${Mvn_Version} /opt/myapplication/ -p
&& \
mv apache-maven-${Mvn_Version}/* /usr/local/apache-maven/apache-maven-${Mvn_Version}/ && \
git clone https://github.com/shan5a6/myweb.git /opt/myapplication/
WORKDIR /opt/myapplication/
RUN mvn clean install
```

```
# Installing and configuring Tomcat
FROM centos:7
EXPOSE 8080
ENV Tomcat_Version=8.5.59
```

```
RUN yum install java-1.8.0-openjdk-devel wget -y
```

```
RUN wget http://www-eu.apache.org/dist/tomcat/tomcat-8/v\${Tomcat\_Version}/bin/apache-tomcat-\${Tomcat\_Version}.tar.gz && \
tar xvfz apache-tomcat-${Tomcat_Version}.tar.gz && \
mkdir -p /opt/tomcat/ /opt/myapplication/ -p && \
mv apache-tomcat-${Tomcat_Version}.tar.gz /tmp/ && \
mv apache-tomcat-${Tomcat_Version}/* /opt/tomcat/
COPY context.xml /opt/tomcat/webapps/manager/META-INF/
COPY tomcat-users.xml /opt/tomcat/conf/
COPY --from=build /opt/myapplication/target/myweb-0.12.0.war
/opt/tomcat/webapps/myapp.war
CMD ["/opt/tomcat/bin/catalina.sh", "run"]
```