Basic Linux Commands

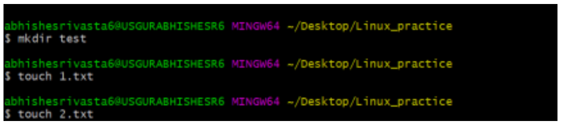
Monday, March 21, 2022 6:12 PM

Path -> Linux Basic > Git > CI/CD basic > Jenkins > Build Tool(ant, maven, gradle) , Docker basic.

Basic Commands:

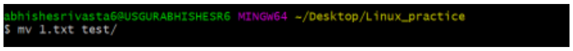
- pwd - Parent folders/file Structure.
- ls - Child folders/file structure.

- Important Directories : bin, dev, usr, tmp
- Cd- change directory
- $- regular user.
- mkdir - make directory



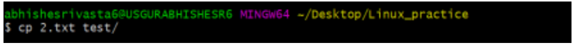
```
abhishesrivasta6@USGURABHISHESR6 MINGW64 ~/Desktop/Linux_practice
$ mkdir test

abhishesrivasta6@USGURABHISHESR6 MINGW64 ~/Desktop/Linux_practice
$ touch 1.txt

abhishesrivasta6@USGURABHISHESR6 MINGW64 ~/Desktop/Linux_practice
$ touch 2.txt
```

- touch - creates blank text-file
- mv - Move file



```
abhishesrivasta6@USGURABHISHESR6 MINGW64 ~/Desktop/Linux_practice
$ mv 1.txt test/
```

- cp - Copy file

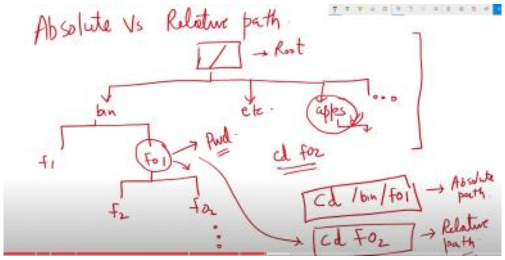


```
abhishesrivasta6@USGURABHISHESR6 MINGW64 ~/Desktop/Linux_practice
$ cp 2.txt test/
```

Users In Linux:

★ 1.Regular User - Home Directory
★ 2.Root User or Super user- Full Access [Admin], sudo
3.Service User - only working in server

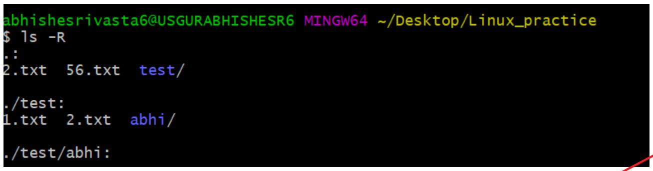
Absolute vs Relative Path



Advanced Linux Commands

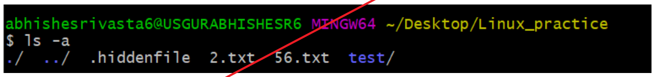
Tuesday, March 22, 2022 5:57 PM

- Apt-get Update - update list of software which is to be updated
- Apt-get Update - install all software.
- ls -R - Opens sub-directories inside directories . It doesn't show hidden files.



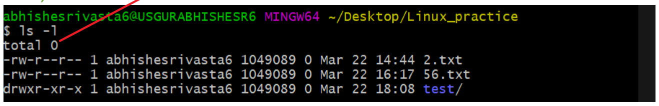
```
abhishesrivasta6@USGURABHISHESR6 MINGW64 ~/Desktop/Linux_practice
$ ls -R
.:
2.txt  56.txt  test/

./test:
1.txt  2.txt  abhi/

./test/abhi:
```

- Touch .filename - creates a hidden file.
- ls -a - shows hidden files



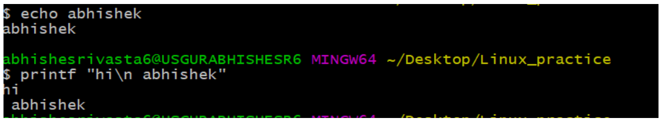
```
abhishesrivasta6@USGURABHISHESR6 MINGW64 ~/Desktop/Linux_practice
$ ls -a
./  ../  .hiddenfile  2.txt  56.txt  test/
```

- ls -l , ls-t



```
abhishesrivasta6@USGURABHISHESR6 MINGW64 ~/Desktop/Linux_practice
$ ls -l
total 0
-rw-r--r-- 1 abhishesrivasta6 1049089 0 Mar 22 14:44 2.txt
-rw-r--r-- 1 abhishesrivasta6 1049089 0 Mar 22 16:17 56.txt
drwxr-xr-x 1 abhishesrivasta6 1049089 0 Mar 22 18:08 test/
```

- clear - cleans the terminal
- History - shows commands which are being run. **Do not put PASSWORD as command ever.
- echo and printf - print statement or anything else

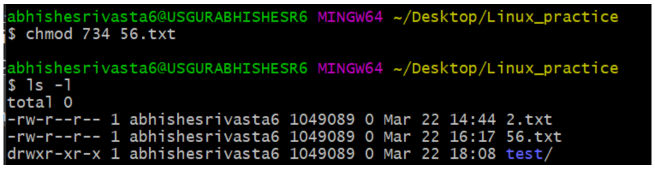


```
$ echo abhishek
abhishek

abhishesrivasta6@USGURABHISHESR6 MINGW64 ~/Desktop/Linux_practice
$ printf "hi\n abhishek"
hi
 abhishek
```

- apt install software_name - installs software
- top - list resources consumed by the running processes.
- ps - lists running processes
- kill PID (Process ID) - Close the running process or app.
- nano file_name - Edit text file > ctrl+x > enter
- vim file_name - Opens an editor

To change the permission of any directory or file :

these are read, write, execute permission for User , Group , public respectively .
Use Chmod Calculator online



```
abhishesrivasta6@USGURABHISHESR6 MINGW64 ~/Desktop/Linux_practice
$ chmod 734 56.txt

abhishesrivasta6@USGURABHISHESR6 MINGW64 ~/Desktop/Linux_practice
$ ls -l
total 0
-rw-r--r-- 1 abhishesrivasta6 1049089 0 Mar 22 14:44 2.txt
-rw-r--r-- 1 abhishesrivasta6 1049089 0 Mar 22 16:17 56.txt
drwxr-xr-x 1 abhishesrivasta6 1049089 0 Mar 22 18:08 test/
```

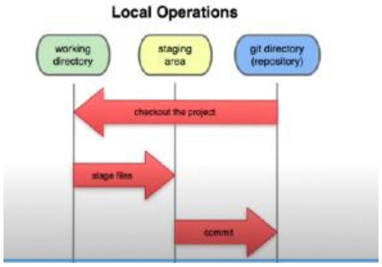
GIT 1

Wednesday, March 23, 2022 7:29 PM

It's a version control system

- Easily recovers file.
- Who introduced an issue and when.
- Roll back to previously working state.
- .git file : snapshot of whole project history
- Almost every operation is local. And later you can push to git(GitHub, GitLab, Bitbucket).
- GIT has Integrity. (having checksums[SHA] same both the sides).If Check-Sum is different , means files has been disrupted in between.
- GIT generally only adds data.

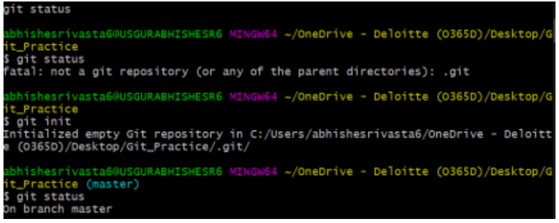
GIT - Three stage architecture



- Working Directory - where all files are present (Windows Explorer)
- Staging area- files which are to be going in commit stage
- Git Directory(repository) - (.git) where snapshot of process will be kept .i.e., files after staging.

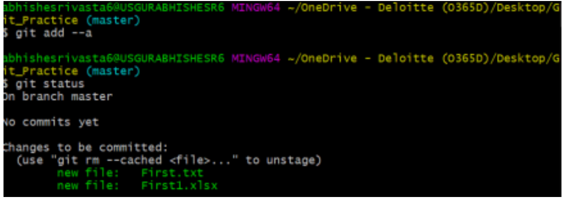
GIT Commands

- git status - to check the repository status of file/folder



```
git status

abhishesrivasta6@USGURABHISHESR6 MINGW64 ~/OneDrive - Deloitte (O365D)/Desktop/G
it_Practice
$ git status
fatal: not a git repository (or any of the parent directories): .git

abhishesrivasta6@USGURABHISHESR6 MINGW64 ~/OneDrive - Deloitte (O365D)/Desktop/G
it_Practice
$ git init
Initialized empty Git repository in C:/Users/abhishesrivasta6/OneDrive - Deloitt
e (O365D)/Desktop/Git_Practice/.git/

abhishesrivasta6@USGURABHISHESR6 MINGW64 ~/OneDrive - Deloitte (O365D)/Desktop/G
it_Practice (master)
$ git status
On branch master
```

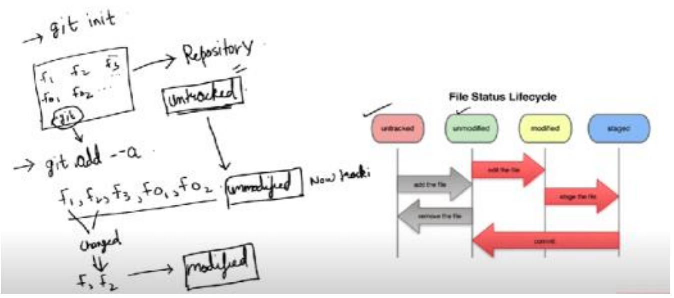
- git init - to make a folder as git repository
- git add --a/filename - promote ALL files/file to STAGING AREA.



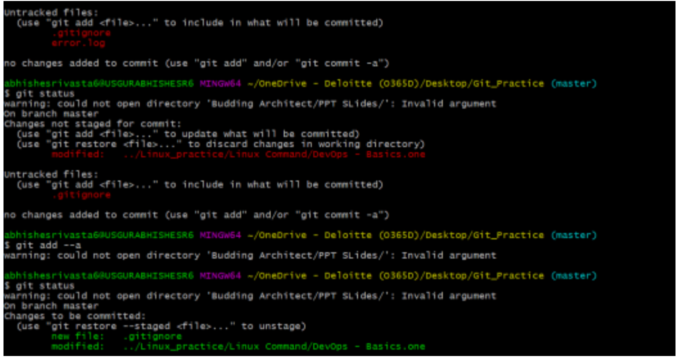
```
abhishesrivasta6@USGURABHISHESR6 MINGW64 ~/OneDrive - Deloitte (O365D)/Desktop/G
it_Practice (master)
$ git add --a

abhishesrivasta6@USGURABHISHESR6 MINGW64 ~/OneDrive - Deloitte (O365D)/Desktop/G
it_Practice (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   First.txt
        new file:   First1.xlsx
```

- git commit -m "Message" - to make the snapshot/copy of selected folder/files.
- git log - to see all commits
- rm -rf .git - deletes git repository
- git clone <URL> user_choice_folder_name - Clone the repository from GitHub.

File Status Life-Cycle:



- touch .gitignore - if any file added inside this file, it can be ignored from tracking
  Ex-Log file/*.log / directory_name we can write inside .gitignore.

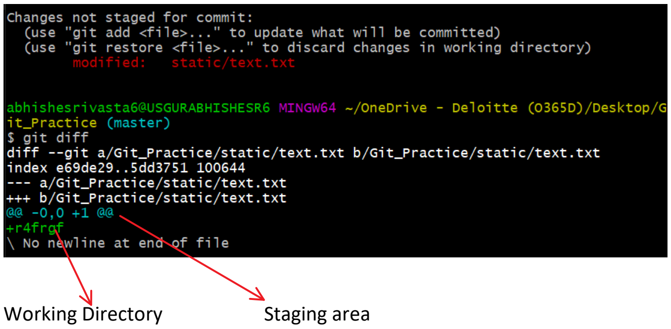


```
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .gitignore
        error.log

no changes added to commit (use "git add" and/or "git commit -a")

abhishesrivasta6@USGURABHISHESR6 MINGW64 ~/OneDrive - Deloitte (O365D)/Desktop/Git_Practice (master)
$ git status
warning: could not open directory 'Budding Architect/PPT SLides/': Invalid argument
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   ../Linux_practice/Linux Command/DevOps - Basics.one

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .gitignore

no changes added to commit (use "git add" and/or "git commit -a")

abhishesrivasta6@USGURABHISHESR6 MINGW64 ~/OneDrive - Deloitte (O365D)/Desktop/Git_Practice (master)
$ git add --a
warning: could not open directory 'Budding Architect/PPT SLides/': Invalid argument

abhishesrivasta6@USGURABHISHESR6 MINGW64 ~/OneDrive - Deloitte (O365D)/Desktop/Git_Practice (master)
$ git status
warning: could not open directory 'Budding Architect/PPT SLides/': Invalid argument
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   .gitignore
        modified:   ../Linux_practice/Linux Command/DevOps - Basics.one
```

GIT 2

Tuesday, March 29, 2022   5:20 PM

- Git diff : Showing Changes Between Staging Area & Working Director.

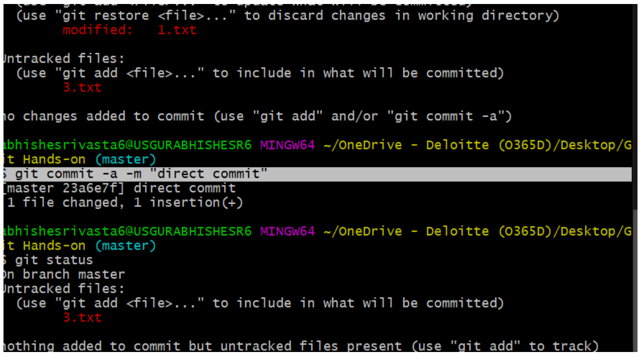


```
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   static/text.txt

abhishesrivasta6@USGURABHISHESR6 MINGW64 ~/OneDrive - Deloitte (O365D)/Desktop/G
it_Practice (master)
$ git diff
diff --git a/Git_Practice/static/text.txt b/Git_Practice/static/text.txt
index e69de29..5dd3751 100644
--- a/Git_Practice/static/text.txt
+++ b/Git_Practice/static/text.txt
@@ -0,0 +1 @@
+r4frgf
\ No newline at end of file
```

- Git diff --staged : Showing Changes Between Previous Commits & present staging area.

Skipping Staging Area

- git commit -a -m "direct commit" - Tracked file can be directly committed without going to staging area. And We can't directly commit untracked files.



```
  (use "git restore <file>..." to discard changes in working directory)
        modified:   1.txt

ntracked files:
  (use "git add <file>..." to include in what will be committed)
        3.txt

o changes added to commit (use "git add" and/or "git commit -a")

bhishesrivasta6@USGURABHISHESR6 MINGW64 ~/OneDrive - Deloitte (O365D)/Desktop/G
t Hands-on (master)
 git commit -a -m "direct commit"
master 23a6e7f] direct commit
1 file changed, 1 insertion(+)

bhishesrivasta6@USGURABHISHESR6 MINGW64 ~/OneDrive - Deloitte (O365D)/Desktop/G
t Hands-on (master)
 git status
n branch master
ntracked files:
  (use "git add <file>..." to include in what will be committed)
        3.txt

othing added to commit but untracked files present (use "git add" to track)
```

Changing Previous commit

- Git commit --amend - modify the most recent commit

Unstaging any file

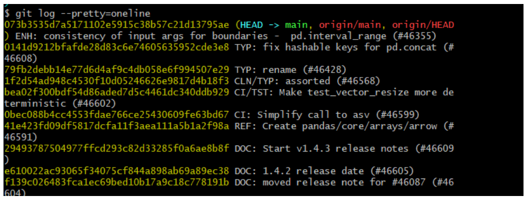
- Git restore --staged 1.txt - makes the file unstaged.
- Git checkout -- 1.txt - unmodify last staged file/undo
- Git checkout -f - undo all modified files.

CURD Operations

- git rm file_name : removes and stages file
- git mv 1.txt first.txt : renames and stages the file
- git rm --cached file_name : to untrack any file

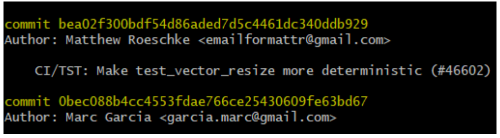
Git Log Operations

- Git log - gives log of all committed history
- Git log --p - gives summary of commits.
- Git log -n - shows the last n commits
- Git log --stat - displays the files that have been modified
- Git log --pretty=oneline - shows all commits in one line



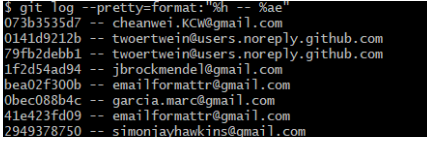
```
$ git log --pretty=oneline
073b3535d7a5171102e5915c38b57c21d13795ae (HEAD -> main, origin/main, origin/HEAD
) ENH: consistency of input args for boundaries -  pd.interval_range (#46355)
0141d9212bfafde28d83c6e74605635952cde3e8 TYP: fix hashable keys for pd.concat (#
46608)
79fb2debb14e77d6d4af9c4db058e6f994507e29 TYP: rename (#46428)
1f2d54ad948c4530f10d05246626e9817d4b18f3 CLN/TYP: assorted (#46568)
bea02f300bdf54d86aded7d5c4461dc340ddb929 CI/TST: Make test_vector_resize more de
terministic (#46602)
0bec088b4cc4553fdae766ce25430609fe63bd67 CI: Simplify call to asv (#46599)
41e423fd09df5817dcfa11f3aea111a5b1a2f98a REF: Create pandas/core/arrays/arrow (#
46591)
29493787504977ffcd293c82d33285f0a6ae8b8f DOC: Start v1.4.3 release notes (#46609
)
e610022ac93065f34075cf844a898ab69a89ec38 DOC: 1.4.2 release date (#46605)
f139c026483fca1ec69bed10b17a9c18c778191b DOC: moved release note for #46087 (#46
604)
```

- Git log --pretty=short/full - shows all Author Commit and messages.



```
commit bea02f300bdf54d86aded7d5c4461dc340ddb929
Author: Matthew Roeschke <emailformattr@gmail.com>

    CI/TST: Make test_vector_resize more deterministic (#46602)

commit 0bec088b4cc4553fdae766ce25430609fe63bd67
Author: Marc Garcia <garcia.marc@gmail.com>
```

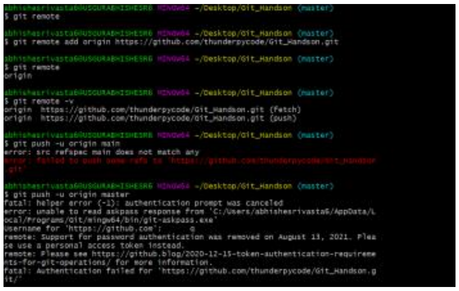
- Git log --since=n.days/weeks/months - shows n days/month/week commits.
- Git log --pretty=format:"%h -- %ae" - The format: format allows you to specify which information you want to show. It works a little bit like printf format.



```
$ git log --pretty=format:"%h -- %ae"
073b3535d7 -- cheanwei.KCW@gmail.com
0141d9212b -- twoertwein@users.noreply.github.com
79fb2debb1 -- twoertwein@users.noreply.github.com
1f2d54ad94 -- jbrockmendel@gmail.com
bea02f300b -- emailformattr@gmail.com
0bec088b4c -- garcia.marc@gmail.com
41e423fd09 -- emailformattr@gmail.com
2949378750 -- simonjayhawkins@gmail.com
```

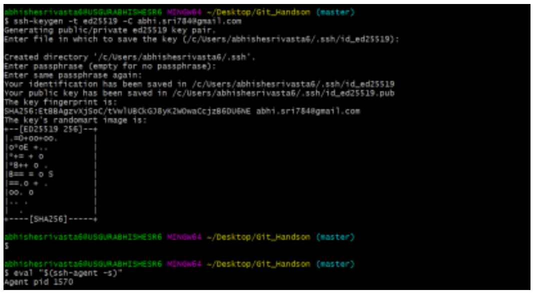
https://git-scm.com/docs/git-log

Working with Remote Repositories (GITHUB)



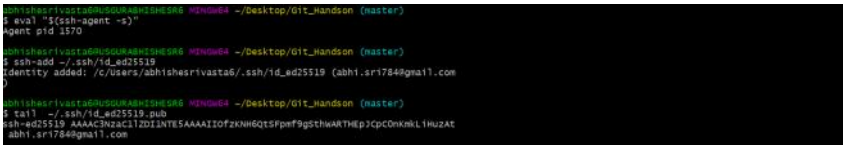
```
abhishesrivasta6@USGURABHISHESR6 MINGW64 ~/Desktop/Git_Handson (master)
$ git remote

abhishesrivasta6@USGURABHISHESR6 MINGW64 ~/Desktop/Git_Handson (master)
$ git remote add origin https://github.com/thunderpycode/Git_Handson.git

abhishesrivasta6@USGURABHISHESR6 MINGW64 ~/Desktop/Git_Handson (master)
$ git remote
origin

abhishesrivasta6@USGURABHISHESR6 MINGW64 ~/Desktop/Git_Handson (master)
$ git remote -v
origin  https://github.com/thunderpycode/Git_Handson.git (fetch)
origin  https://github.com/thunderpycode/Git_Handson.git (push)

abhishesrivasta6@USGURABHISHESR6 MINGW64 ~/Desktop/Git_Handson (master)
$ git push -u origin main
error: src refspec main does not match any
error: failed to push some refs to 'https://github.com/thunderpycode/Git_Handson
.git'

abhishesrivasta6@USGURABHISHESR6 MINGW64 ~/Desktop/Git_Handson (master)
$ git push -u origin master
fatal: helper error (-1): authentication prompt was canceled
error: unable to read askpass response from 'C:/Users/abhishesrivasta6/AppData/L
ocal/Programs/Git/mingw64/bin/git-askpass.exe'
Username for 'https://github.com':      q
remote: Support for password authentication was removed on August 13, 2021. Plea
se use a personal access token instead.
remote: Please see https://github.blog/2020-12-15-token-authentication-requireme
nts-for-git-operations/ for more information.
fatal: Authentication failed for 'https://github.com/thunderpycode/Git_Handson.g
it/'
```

(1)



```
abhishesrivasta6@USGURABHISHESR6 MINGW64 ~/Desktop/Git_Handson (master)
$ ssh-keygen -t ed25519 -C abhi.sri784@gmail.com
Generating public/private ed25519 key pair.
Enter file in which to save the key (/c/Users/abhishesrivasta6/.ssh/id_ed25519):

Created directory '/c/Users/abhishesrivasta6/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/abhishesrivasta6/.ssh/id_ed25519
Your public key has been saved in /c/Users/abhishesrivasta6/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:Et8BAgzvXjSoC/tVwlUBCkGJ8yKZWOwaCcjzB6DU6NE abhi.sri784@gmail.com
The key's randomart image is:
+--[ED25519 256]--+
|.=O+oo+oo.       |
|o*oE +..         |
|*+= + o          |
|*B++ o .         |
|B== = o S        |
|==.o + .         |
|oo. o            |
|.. .             |
|  .              |
+----[SHA256]-----+

abhishesrivasta6@USGURABHISHESR6 MINGW64 ~/Desktop/Git_Handson (master)
$

abhishesrivasta6@USGURABHISHESR6 MINGW64 ~/Desktop/Git_Handson (master)
$ eval "$(ssh-agent -s)"
Agent pid 1570
```

(2)



```
abhishesrivasta6@USGURABHISHESR6 MINGW64 ~/Desktop/Git_Handson (master)
$ eval "$(ssh-agent -s)"
Agent pid 1570

abhishesrivasta6@USGURABHISHESR6 MINGW64 ~/Desktop/Git_Handson (master)
$ ssh-add ~/.ssh/id_ed25519
Identity added: /c/Users/abhishesrivasta6/.ssh/id_ed25519 (abhi.sri784@gmail.com
)

abhishesrivasta6@USGURABHISHESR6 MINGW64 ~/Desktop/Git_Handson (master)
$ tail  ~/.ssh/id_ed25519.pub
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIIOfzKNH6QtSFpmf9gSthWARTHEpJCpCOnKmkLiHuzAt
 abhi.sri784@gmail.com
```

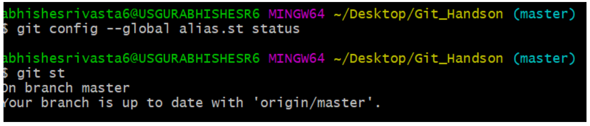
(3)

GIT 3

Tuesday, April 5, 2022 5:20 PM

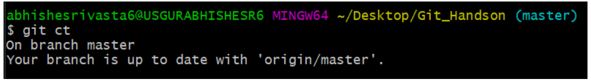
Setting Git alias (bring commands into short forms e.g. status -> st

git config --global alias.st status

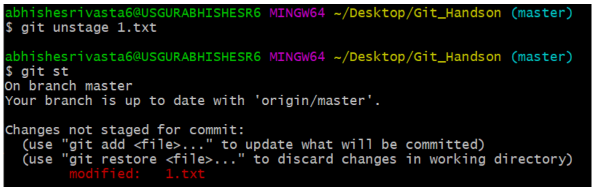


```
bhishesrivasta6@USGURABHISHESR6 MINGW64 ~/Desktop/Git_Handson (master)
$ git config --global alias.st status

bhishesrivasta6@USGURABHISHESR6 MINGW64 ~/Desktop/Git_Handson (master)
$ git st
n branch master
our branch is up to date with 'origin/master'.
```

git config --global alias.ct commit

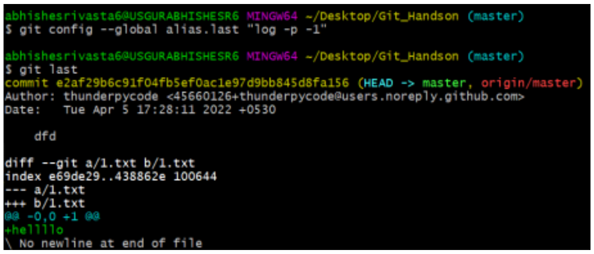


```
abhishesrivasta6@USGURABHISHESR6 MINGW64 ~/Desktop/Git_Handson (master)
$ git ct
On branch master
Your branch is up to date with 'origin/master'.
```



```
abhishesrivasta6@USGURABHISHESR6 MINGW64 ~/Desktop/Git_Handson (master)
$ git unstage 1.txt

abhishesrivasta6@USGURABHISHESR6 MINGW64 ~/Desktop/Git_Handson (master)
$ git st
On branch master
Your branch is up to date with 'origin/master'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   1.txt
```

git config --global alias.last "log -p -1"



```
abhishesrivasta6@USGURABHISHESR6 MINGW64 ~/Desktop/Git_Handson (master)
$ git config --global alias.last "log -p -1"

abhishesrivasta6@USGURABHISHESR6 MINGW64 ~/Desktop/Git_Handson (master)
$ git last
commit e2af29b6c91f04fb5ef0ac1e97d9bb845d8fa156 (HEAD -> master, origin/master)
Author: thunderpycode <45660126+thunderpycode@users.noreply.github.com>
Date:   Tue Apr 5 17:28:11 2022 +0530

    dfd

diff --git a/1.txt b/1.txt
index e69de29..438862e 100644
--- a/1.txt
+++ b/1.txt
@@ -0,0 +1 @@
+helllo
\ No newline at end of file
```

Important Script to configure SSH key in GitHub account.

1. ssh-keygen -t ed25519 -C "*your_email@example.com*"
2. eval "$(ssh-agent -s)"\
3. ssh-add ~/.ssh/id_ed25519
4. clip < ~/.ssh/id_ed25519.pub

From <https://docs.github.com/en/authentication/connecting-to-github-with-ssh/adding-a-new-ssh-key-to-your-github-account>

Pushing Branches to GitHub

Git remote - to see the remote directory name e.g., origin
Git remote -v -to see the repository folder path

1.Create new repository
2.Select HTTPS
3.Select git remote add origin https://github.com/thunderpycode/Sample-Repository.git
4. Push the git repository to GitHub git push -u origin master
5. Push the branch git push origin bugfix

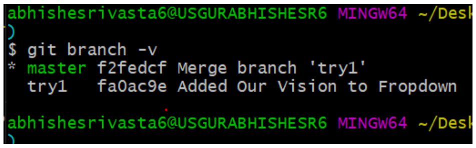
Note : Be in that branch to which branch you we want to push in GitHub Branch

Creating & Switching Branches In Git

- Git checkout -b <branch_name> - creates new branch
- Git checkout <branch_name> - to switch to any branch
- Git branch - to check all the branches in grid

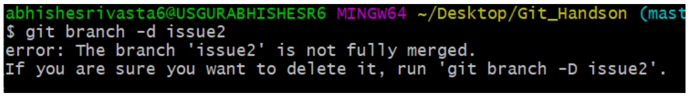
Branch Management

- Git branch -v - shows commit hash and last commit in branch



```
abhishesrivasta6@USGURABHISHESR6 MINGW64 ~/Desk
)
$ git branch -v
* master f2fedcf Merge branch 'try1'
  try1   fa0ac9e Added Our Vision to Fropdown

abhishesrivasta6@USGURABHISHESR6 MINGW64 ~/Desk
)
```

- Git branch --merged - shows already merged branches.
- Git branch --no-merged - not merged branches.
- Git branch -d try1 -gives an error if try1 branch is not merged. (does not deletes non merged branches)



```
abhishesrivasta6@USGURABHISHESR6 MINGW64 ~/Desktop/Git_Handson (mast
$ git branch -d issue2
error: The branch 'issue2' is not fully merged.
If you are sure you want to delete it, run 'git branch -D issue2'.
```

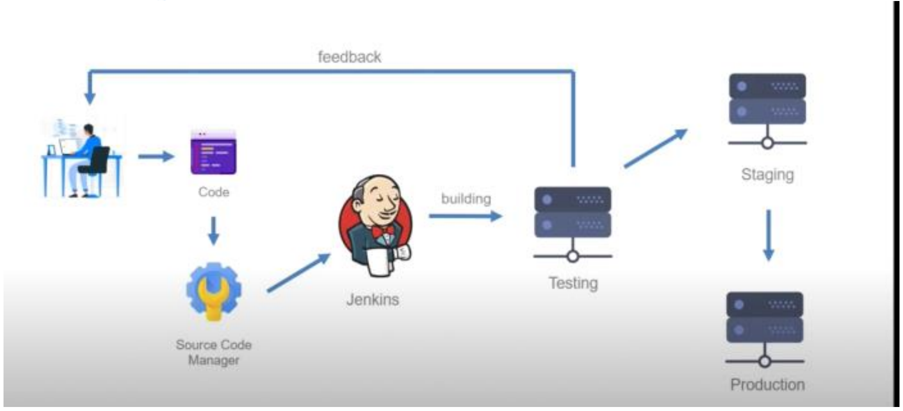
- Git branch -D try1 - deletes branch
- Git push -d origin branch_name - deletes branch from remote

Jenkins

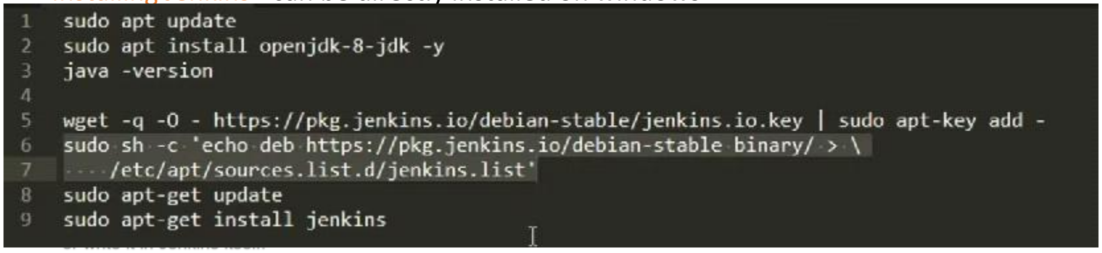
Tuesday, April 12, 2022 5:13 PM

Jenkins is a powerful application that allows continuous integration and continuous delivery of projects, regardless of the platform you are working on. It is a free source that can handle any kind of build or continuous integration. You can integrate Jenkins with a number of testing and deployment technologies. It builds and tests our software projects, which continuously making it easier for developers to integrate changes to the project, and making it easier for users to obtain a fresh build.

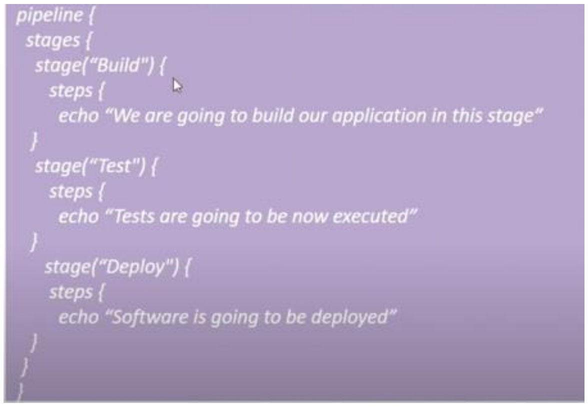
Jenkins Pipeline



Installing Jenkins - can be directly installed on windows

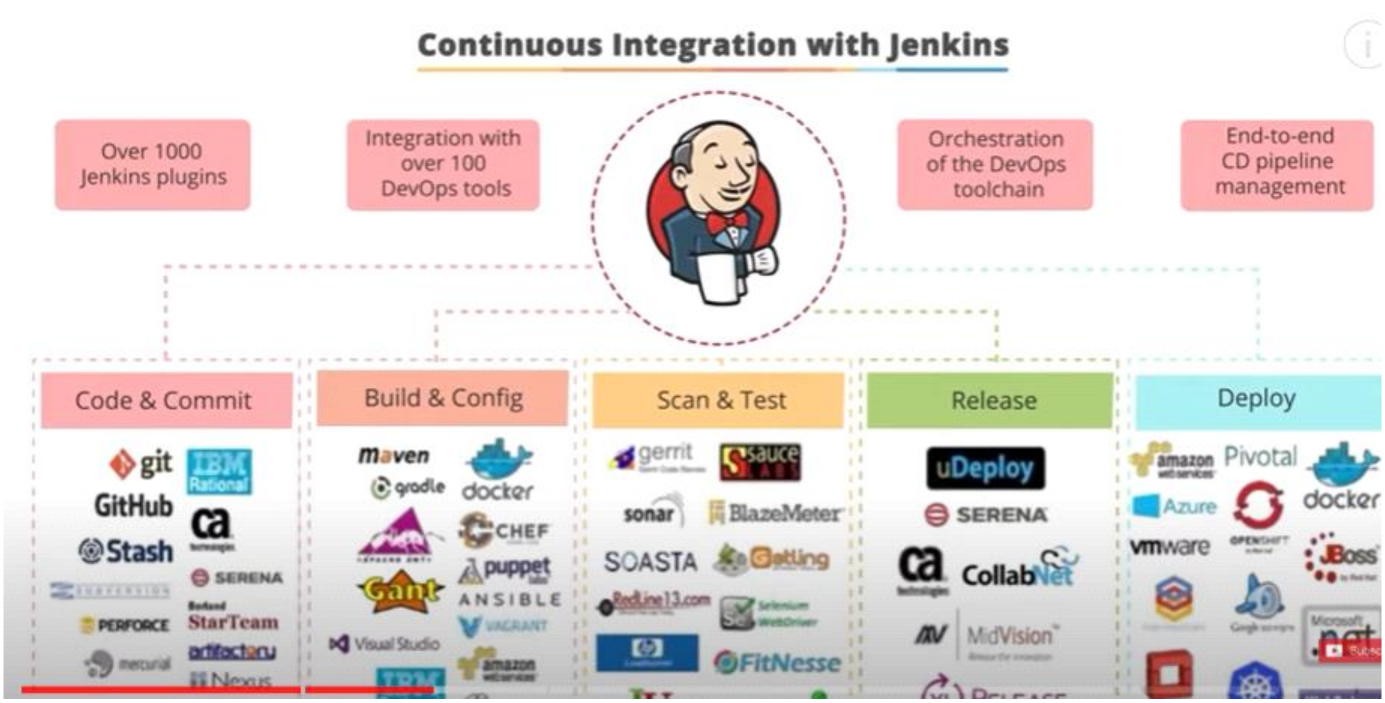


```
sudo apt update
sudo apt install openjdk-8-jdk -y
java -version

wget -q -O - https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo apt-key add -
sudo sh -c 'echo deb https://pkg.jenkins.io/debian-stable binary/ > \
    /etc/apt/sources.list.d/jenkins.list'
sudo apt-get update
sudo apt-get install jenkins
```



```
pipeline {
  stages {
    stage("Build") {
      steps {
        echo "We are going to build our application in this stage"
    }
    stage("Test") {
      steps {
        echo "Tests are going to be now executed"
    }
      stage("Deploy") {
      steps {
        echo "Software is going to be deployed"
    }
  }
}
```

Unclarity on - Maven , Jenkins Files



Continuous Integration with Jenkins

Over 1000 Jenkins plugins

Integration with over 100 DevOps tools

Orchestration of the DevOps toolchain

End-to-end CD pipeline management

Code & Commit

Build & Config

Scan & Test

Release

Deploy

Docker - 1

Wednesday, May 4, 2022 7:27 PM

AWS

Public IP : 13.233.106.30

Install Docker on Ubuntu from Default Repositories

Step 1: First let's Update exiting repositories. Run the below command to update

```
sudoapt-getupdate
```

Step 2: If you need to uninstall the older version of docker, run the below command to uninstall older version.

```
sudoapt-getremove docker docker-engine docker.io
```

Step 3: Now run the below command to install the latest version of docker.

```
sudo apt install docker.io
```

Step 4: The dockers need to be run at startup, so run the below commands one by one.

```
sudosystemctl start docker
sudosystemctl enabledocker
```

Step 5: Finally let's verify whether docker installed on not on the machine. Run the below command to verify.

```
sudodocker --version
```

Install Docker on Ubuntu from Official Repository

Step 1: As usual update the existing packages.

```
sudoapt-getupdate
```

Step 2: Run the command to allow your operating system to access the Docker repositories over HTTPS and download the dependencies.

```
 sudoapt-getinstallapt-transport-https ca-certificates curlsoftware-properties-common
```

Let's break down the above command:

- **apt-transport-https** allows the package manager to transfer files and data over https.
- **ca-certificates** allows the system to check security certificates.
- **curl** is a tool for transferring data.
- **software-properties-common** adds scripts for managing software

Step 3: Now let's add GPG keys to ensure the software we're installing is authentic.

```
 curl-fsSL https://download.docker.com/linux/ubuntu/gpg |sudoapt-key add
```

Step 4: Now let's install the docker repository

```
sudoadd-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb_release -cs)stable"
```

Here **$(lsb_release –cs)** scans and returns the codename of your Ubuntu installation – In our case, its bionic and the **stable** is the type of docker release.

The stable release is test and confirmed to work but updates are less frequent. If you need more frequent updates you can pull **edge** release.

Step 5: Now let's update the repos that we added recently.

```
sudoapt-getupdate
```

Step 6: Install the latest version of docker community edition.

```
sudoapt-getinstalldocker-ce   // ce is community edition
```

Step 7: Finally verify the installation.

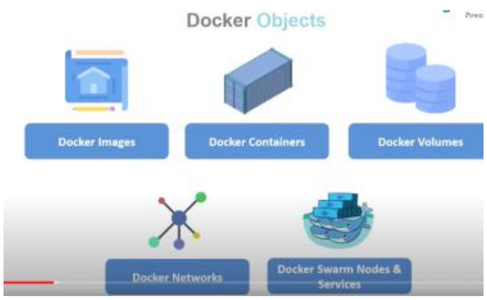
```
sudodocker --version
```

From <https://vanchiv.com/how-to-install-docker-on-ubuntu/>

Docker - 2

Wednesday, May 4, 2022 7:50 PM

Docker Engine - CLI + API + Docker Daemon

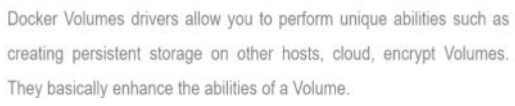


Docker Objects - Images

Docker Objects - Images - Set of instructions use to create containers and execute code inside it.

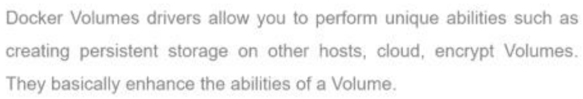
- To pull the image : Docker Pull Ubuntu
- To see Image : Docker Image
- To see the Container : docker run -it -d ubuntu
- To Create a New Container : docker run -it -d --name mycontainer ubuntu
- To See the status of Container Image : Docker ps
- To Publish a Container in any port : docker run -it -d --name mycontaine-publish -p 80:80 ubuntu
- To Remove/Kill/restart/stop the docker container : Docker rm/Kill/restart/stop <container id>
- To save the settings in Docker : Docker commit <container id> <new_image_name>

Docker Objects - Volumes



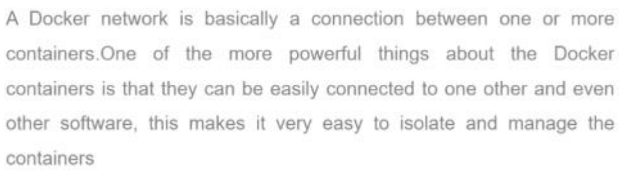
Docker Volumes drivers allow you to perform unique abilities such as creating persistent storage on other hosts, cloud, encrypt Volumes. They basically enhance the abilities of a Volume.

Docker Objects - Volume Drivers



Docker Volumes drivers allow you to perform unique abilities such as creating persistent storage on other hosts, cloud, encrypt Volumes. They basically enhance the abilities of a Volume.

Docker Objects - Networks



A Docker network is basically a connection between one or more containers.One of the more powerful things about the Docker containers is that they can be easily connected to one other and even other software, this makes it very easy to isolate and manage the containers

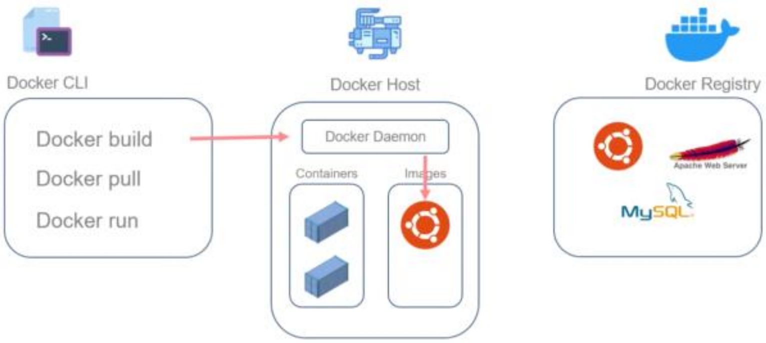
Docker - Registries

Docker - Registries (Where we can store various Images i.e., ubuntu - These images can have different versions) - DockerHub

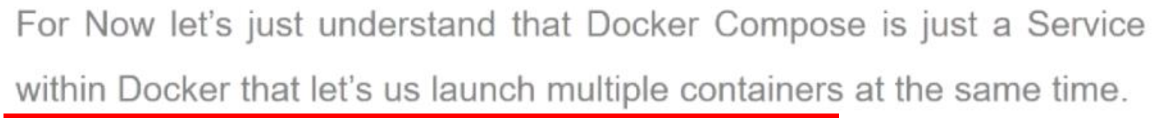
Docker Compose - It's just a service inside docker which helps to launch multiple containers at a same time.

Docker Swarm - Manages multiple containers in multiple nodes(servers).

Docker Architecture



Docker Compose


For Now let's just understand that Docker Compose is just a Service within Docker that let's us launch multiple containers at the same time.

*******yum install docker-ce docker-ce-cli containerd.io docker-compose-plugin -y

The following command will download the 1.29.2 release and save the executable file at /usr/local/bin/docker-compose, which will make this software globally accessible as docker-compose:

```
sudocurl-L "https://github.com/docker/compose/releases/download/1.29.2/docker-compose-$(uname-s)-$(uname-m)"-o /usr/local/bin/docker-compose
```

Next, set the correct permissions so that the docker-compose command is executable:

```
sudochmod+x /usr/local/bin/docker-compose
```

To verify that the installation was successful, you can run:

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

You'll see output similar to this:

Output

```
docker-compose version 1.29.2, build 5becea4c
```

From <https://www.digitalocean.com/community/tutorials/how-to-install-and-use-docker-compose-on-ubuntu-20-04>

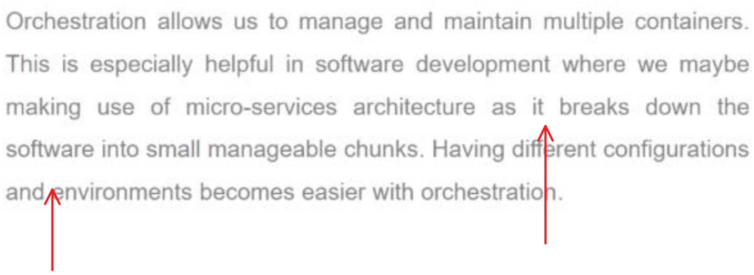
1.Nano docker-composer.yaml

2.SAMPLE YAML FILE

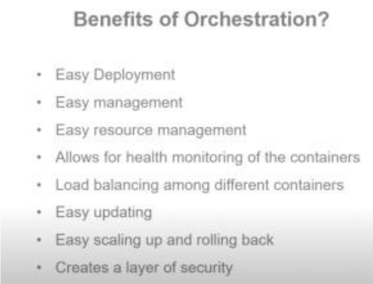
```
version: '3'
networks:
  batman:
    driver: bridge

services:
  web:
    image: "nginx:latest"
    ports:
     - "5000:5000"
    networks:
     - batman
  database:
    image: "mysql"
    networks:
      - batman
```

3. docker-composer up
4. docker -composer

Docker Orchestration



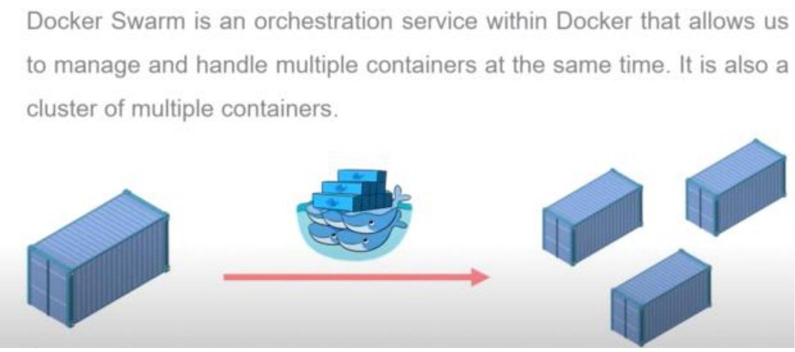
Orchestration allows us to manage and maintain multiple containers. This is especially helpful in software development where we maybe making use of micro-services architecture as it breaks down the software into small manageable chunks. Having different configurations and environments becomes easier with orchestration.



Benefits of Orchestration?

- Easy Deployment
- Easy management
- Easy resource management
- Allows for health monitoring of the containers
- Load balancing among different containers
- Easy updating
- Easy scaling up and rolling back
- Creates a layer of security

Docker Swarm



Docker Swarm is an orchestration service within Docker that allows us to manage and handle multiple containers at the same time. It is also a cluster of multiple containers.

How
When
Why
things useful

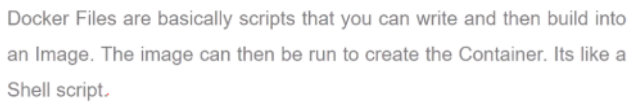
Worker Node

Manager Node

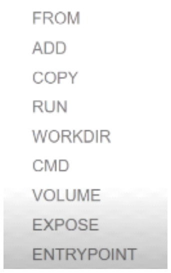
Docker - 3

Thursday, May 5, 2022 11:44 AM

Docker Files



Docker Files are basically scripts that you can write and then build into an Image. The image can then be run to create the Container. Its like a Shell script.



FROM
ADD
COPY
RUN
WORKDIR
CMD
VOLUME
EXPOSE
ENTRYPOINT

FROM <Base Image>

ADD <source eg - URL, local file location><destination>

COPY <source eg - local file location><destination> URL can't be added in the source

RUN <command> - Used at the time of creating container

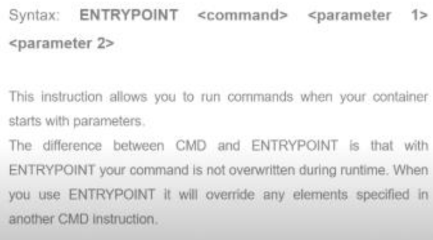
WORKDIR <Directory>

CMD <command> - Used after container is started

VOLUME <path>

EXPOSE <port> - To any port

ENTRYPOINT

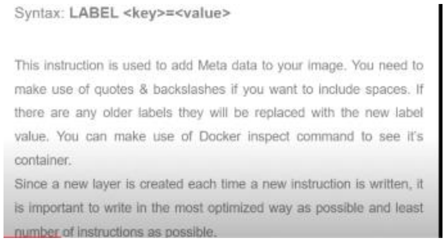


Syntax: **ENTRYPOINT <command> <parameter 1> <parameter 2>**

This instruction allows you to run commands when your container starts with parameters.

The difference between CMD and ENTRYPOINT is that with ENTRYPOINT your command is not overwritten during runtime. When you use ENTRYPOINT it will override any elements specified in another CMD instruction.

LABEL



Syntax: **LABEL <key>=<value>**

This instruction is used to add Meta data to your image. You need to make use of quotes & backslashes if you want to include spaces. If there are any older labels they will be replaced with the new label value. You can make use of Docker inspect command to see it's container.

Since a new layer is created each time a new instruction is written, it is important to write in the most optimized way as possible and least number of instructions as possible.

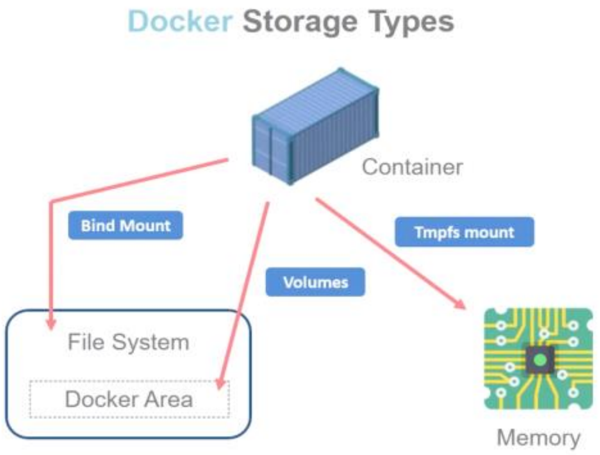
How to create Custom Docker Image ?

1. Open CLI in Ubuntu
2. Create folder - mkdir project and cd project
3. Create python file - nano main.py
4. Create dockerfile (text) - nano dockerfile
5. Write the following script/set of instructions

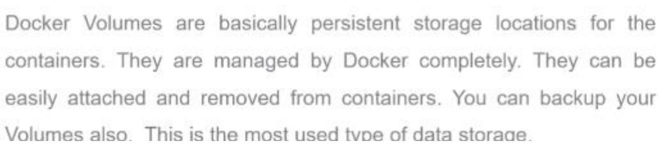
```
FROM ubuntu:latest
WORKDIR /app
ADD . /app
RUN apt-get update
RUN apt-get install pip -y
CMD python /app/main.py
LABEL color=red
```

. Signifies search file within same directory

6. Build the container - docker build -t custom-python-image .

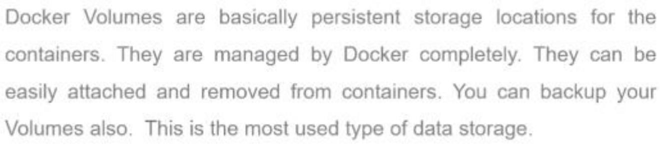


Docker Volume



Docker Volumes are basically persistent storage locations for the containers. They are managed by Docker completely. They can be easily attached and removed from containers. You can backup your Volumes also. This is the most used type of data storage.

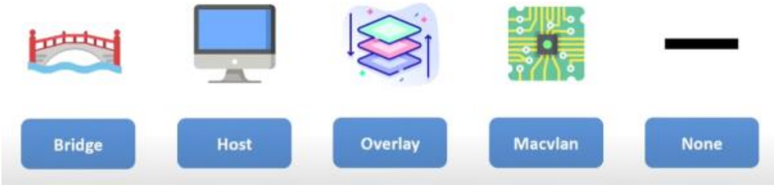
To Create Volume - docker volume create <volume name>
To Delete Volume - docker volume rm <volume name>
To inspect/detail Volume - docker inspect <volume name>
To delete more than one volume - docker volume prune



Docker Volumes are basically persistent storage locations for the containers. They are managed by Docker completely. They can be easily attached and removed from containers. You can backup your Volumes also. This is the most used type of data storage.

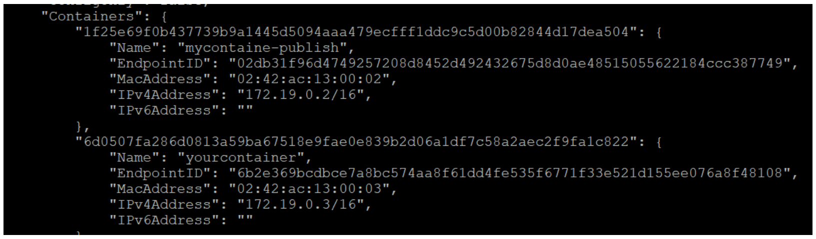
To Delete Volume - docker volume rm <volume name>
To inspect/detail Volume - docker inspect <volume name>
To delete more than one volume - docker volume prune

Docker Network Types



Bridge Network : Containers can communicate with each other with isolation without interference of third container.

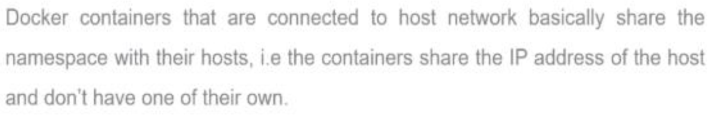
- To create bridge network -Docker network create --driver bridge <network name>
- Docker network ls
- To pre connect/ create network with container - docker run -it -d --network batman-net --name mycontainer1 -p 80:80 ubuntu
- To connect container with network : docker network connect <network name > <container name>
- To inspect network : docker network inspect batman-net



```
    "Containers": {
        "1f25e69f0b437739b9a1445d5094aaa479ecfff1ddc9c5d00b82844d17dea504": {
            "Name": "mycontaine-publish",
            "EndpointID": "02db31f96d4749257208d8452d492432675d8d0ae48515055622184ccc387749",
            "MacAddress": "02:42:ac:13:00:02",
            "IPv4Address": "172.19.0.2/16",
            "IPv6Address": ""
        },
        "6d0507fa286d0813a59ba67518e9fae0e839b2d06a1df7c58a2aec2f9fa1c822": {
            "Name": "yourcontainer",
            "EndpointID": "6b2e369bcdbce7a8bc574aa8f61dd4fe535f6771f33e521d155ee076a8f48108",
            "MacAddress": "02:42:ac:13:00:03",
            "IPv4Address": "172.19.0.3/16",
            "IPv6Address": ""
```

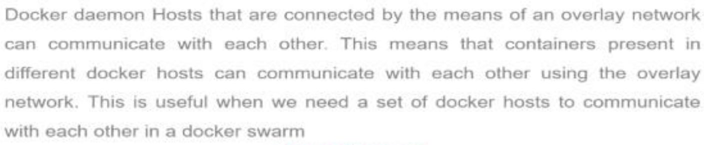
- To Disconnect from network : docker network disconnect <network name> <container name>
- To communicate two containers : docker exec -it yourcontainer bash
- apt install iputils-ping
- Ping <IP Address of other container>

Host Network :

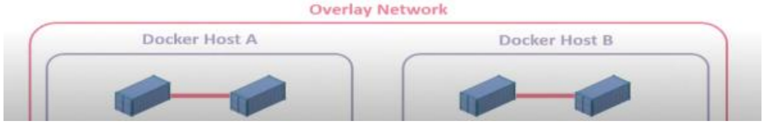


Docker containers that are connected to host network basically share the namespace with their hosts, i.e the containers share the IP address of the host and don't have one of their own.

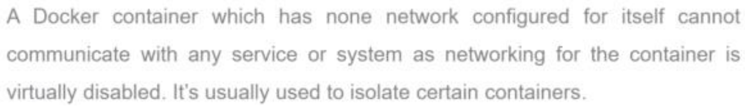
Overlay Network :



Docker daemon Hosts that are connected by the means of an overlay network can communicate with each other. This means that containers present in different docker hosts can communicate with each other using the overlay network. This is useful when we need a set of docker hosts to communicate with each other in a docker swarm



None Network :



A Docker container which has none network configured for itself cannot communicate with any service or system as networking for the container is virtually disabled. It's usually used to isolate certain containers.

```
docker run -it -d --network none --name mycontainer1  ubuntu
```

Terraform

Wednesday, May 25, 2022 12:30 PM

```
//Block Labe1 {
 //identifier = expression
//}
```

```
output hello1{
    value="HelloWorld"
}
```

In Json Format - file_name.tf.son

```
{
"output" : {
    "hello" : {
        "value" : "Hello abhishek"
}
}
}
*************************************
*
```

List-Operation

```
variable user{
    type = list
}
output printlist {
    value = "the first user is
${var.user[0]}"
    }
************************************************
```

For user Defined

```
variable username {}
output Printname {
        value = "Hello ! How are you ${var.username}"
}
```

#In CLI : 1.terraform plan
2. terraform plan -var "username=AbhishekSrivastava"

```
************************************************************************
```

File1 - age.tf

```
variable age{
type = number
 }
```

File2 - Helloworld

```
variable username {
type = string
 }

output Printname {
        value = "Hello ! How are you ${var.username} ,
your age is ${var.age}"
}
**********************************************************************
```

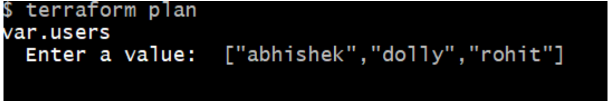
Function-operations

File1.

```
    variable users {
        type = list
    }
```

File2.

```
    output join_function {
      value="${join("--->",var.users)}"
    }
    output upper_function {
      value="${upper(var.users[1])}"
    }
    output lower_function {
      value="${lower(var.users[2])}"
    }
    output title_function {
      value="${title(var.users[0])}"
    }
```



```
$ terraform plan
var.users
  Enter a value:  ["abhishek","dolly","rohit"]
```

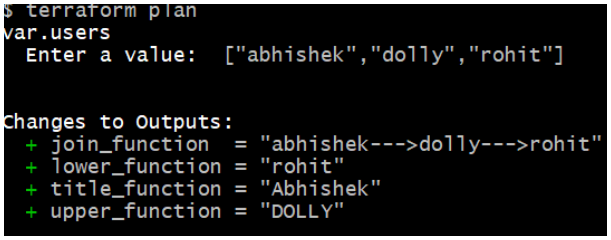
Map Function / Dictionary

```
variable userage{
    type=map
    default={
        abhishek=20
        sri=25
    }
}
variable username{
    type = string
}
output printage{
    value = "my name is ${var.username} ,my age is
${lookup(var.userage,var.username)}"
}
```

#In CLI : terraform plan -var='userage={"abhi":"20","sri":"22"}'

```
************************************************************
```

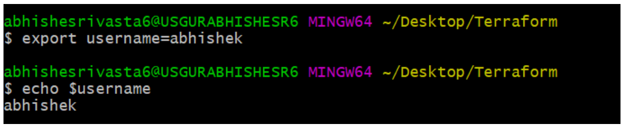
If One file have more than 1 variables (about 10-15 approx) , we use



```
$ terraform plan
var.users
  Enter a value:  ["abhishek","dolly","rohit"]


Changes to Outputs:
  + join_function  = "abhishek--->dolly--->rohit"
  + lower_function = "rohit"
  + title_function = "Abhishek"
  + upper_function = "DOLLY"
```

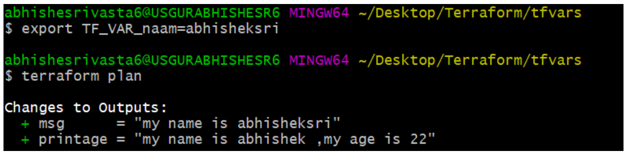
****

Export Variable name to environment variable :



```
abhishesrivasta6@USGURABHISHESR6 MINGW64 ~/Desktop/Terraform
$ export username=abhishek

abhishesrivasta6@USGURABHISHESR6 MINGW64 ~/Desktop/Terraform
$ echo $username
abhishek
```

To read from environment variables :



```
abhishesrivasta6@USGURABHISHESR6 MINGW64 ~/Desktop/Terraform/tfvars
$ export TF_VAR_naam=abhisheksri

abhishesrivasta6@USGURABHISHESR6 MINGW64 ~/Desktop/Terraform/tfvars
$ terraform plan

Changes to Outputs:
  + msg      = "my name is abhisheksri"
  + printage = "my name is abhishek ,my age is 22"
```

****

If One file have more than 1 variables (about 10-15 approx) , we use separate terraform file i.e., .tfvars

File 1.

```
variable age {
    type=number
}
variable username {
    type = string
}
output printage{
    value = "my name is ${var.username} ,my age is ${var.age}"
}
```

File2. Must be named as "terraform.tfvars"

```
age=22
username="abhishek"
```

```
For custom name file : terraform plan -var-file=filename.tfvars
```

****

Terraform - Creating Git Resource

Friday, May 27, 2022 2:07 PM

https://registry.terraform.io/providers/integrations/github/latest/docs/resources/repository

In VS-Code :

```
provider "github" {}
resource "github_repository" "terraform_local" {
  name        = "First_repo_from_terrform"
  description = "My awesome codebase"
  visibility = "public"
  auto_init = true
}
```

In CLI :

```
terraform init
terraform apply
```

Microservices - Docker 1

Monday, November 14, 2022 10:02 AM

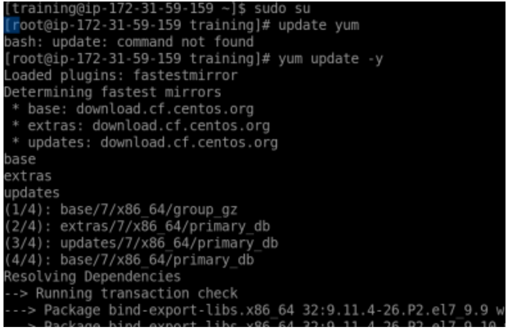
- Need for Microservices - Scaling, faster , isolation from failures
- 12 Factor App
  - Always map one process to one container

9mfoGhPWhpdErkCq

From <https://teams.microsoft.com/multi-window/?agent=electron&version=22102807200>

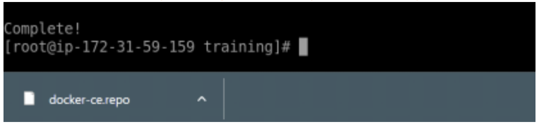
Introduction to : Docker

- sudo su
- yum update -y



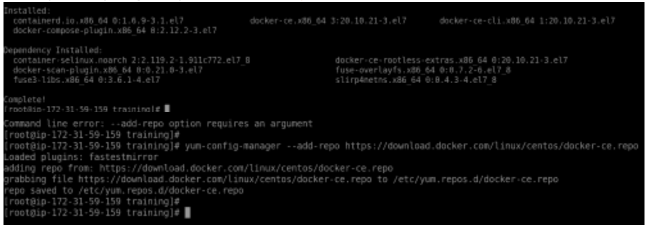
```
[training@ip-172-31-59-159 ~]$ sudo su
[root@ip-172-31-59-159 training]# update yum
bash: update: command not found
[root@ip-172-31-59-159 training]# yum update -y
Loaded plugins: fastestmirror
Determining fastest mirrors
 * base: download.cf.centos.org
 * extras: download.cf.centos.org
 * updates: download.cf.centos.org
base
extras
updates
(1/4): base/7/x86_64/group_gz
(2/4): extras/7/x86_64/primary_db
(3/4): updates/7/x86_64/primary_db
(4/4): base/7/x86_64/primary_db
Resolving Dependencies
--> Running transaction check
---> Package bind-export-libs.x86_64 32:9.11.4-26.P2.el7_9.9 w
```

- yum-config-manager --add-repo https://download.docker.com/linux/centos/docker-ce.repo (installs docker package from specified repo)



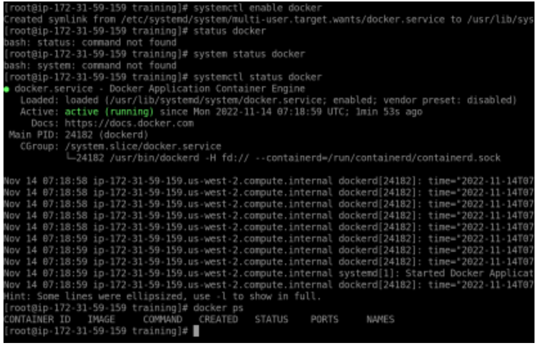
```
Complete!
[root@ip-172-31-59-159 training]# █
```

- yum install docker-ce docker-ce-cli containerd.io docker-compose-plugin -y (installs dependencies)



```
Installed:
  containerd.io.x86_64 0:1.6.9-3.1.el7          docker-ce.x86_64 3:20.10.21-3.el7       docker-ce-cli.x86_64 1:20.10.21-3.el7
  docker-compose-plugin.x86_64 0:2.12.2-3.el7

Dependency Installed:
  container-selinux.noarch 2:2.119.2-1.911c772.el7_8                docker-ce-rootless-extras.x86_64 0:20.10.21-3.el7
  docker-scan-plugin.x86_64 0:0.21.0-3.el7                          fuse-overlayfs.x86_64 0:0.7.2-6.el7_8
  fuse3-libs.x86_64 0:3.6.1-4.el7                                   slirp4netns.x86_64 0:0.4.3-4.el7_8

Complete!
[root@ip-172-31-59-159 training]# █
Command line error: --add-repo option requires an argument
[root@ip-172-31-59-159 training]#
[root@ip-172-31-59-159 training]# yum-config-manager --add-repo https://download.docker.com/linux/centos/docker-ce.repo
Loaded plugins: fastestmirror
adding repo from: https://download.docker.com/linux/centos/docker-ce.repo
grabbing file https://download.docker.com/linux/centos/docker-ce.repo to /etc/yum.repos.d/docker-ce.repo
repo saved to /etc/yum.repos.d/docker-ce.repo
[root@ip-172-31-59-159 training]#
[root@ip-172-31-59-159 training]# █
```

- systemctl start docker
- system enable docker
- system status docker
- docker ps



```
[root@ip-172-31-59-159 training]# systemctl enable docker
Created symlink from /etc/systemd/system/multi-user.target.wants/docker.service to /usr/lib/sys
[root@ip-172-31-59-159 training]# status docker
bash: status: command not found
[root@ip-172-31-59-159 training]# system status docker
bash: system: command not found
[root@ip-172-31-59-159 training]# 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 Mon 2022-11-14 07:18:59 UTC; 1min 53s ago
     Docs: https://docs.docker.com
 Main PID: 24182 (dockerd)
   CGroup: /system.slice/docker.service
           └─24182 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Nov 14 07:18:58 ip-172-31-59-159.us-west-2.compute.internal dockerd[24182]: time="2022-11-14T07
Nov 14 07:18:58 ip-172-31-59-159.us-west-2.compute.internal dockerd[24182]: time="2022-11-14T07
Nov 14 07:18:58 ip-172-31-59-159.us-west-2.compute.internal dockerd[24182]: time="2022-11-14T07
Nov 14 07:18:58 ip-172-31-59-159.us-west-2.compute.internal dockerd[24182]: time="2022-11-14T07
Nov 14 07:18:58 ip-172-31-59-159.us-west-2.compute.internal dockerd[24182]: time="2022-11-14T07
Nov 14 07:18:59 ip-172-31-59-159.us-west-2.compute.internal dockerd[24182]: time="2022-11-14T07
Nov 14 07:18:59 ip-172-31-59-159.us-west-2.compute.internal dockerd[24182]: time="2022-11-14T07
Nov 14 07:18:59 ip-172-31-59-159.us-west-2.compute.internal dockerd[24182]: time="2022-11-14T07
Nov 14 07:18:59 ip-172-31-59-159.us-west-2.compute.internal systemd[1]: Started Docker Applicat
Nov 14 07:18:59 ip-172-31-59-159.us-west-2.compute.internal dockerd[24182]: time="2022-11-14T07
Hint: Some lines were ellipsized, use -l to show in full.
[root@ip-172-31-59-159 training]# docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
[root@ip-172-31-59-159 training]# █
```

Creating Docker file

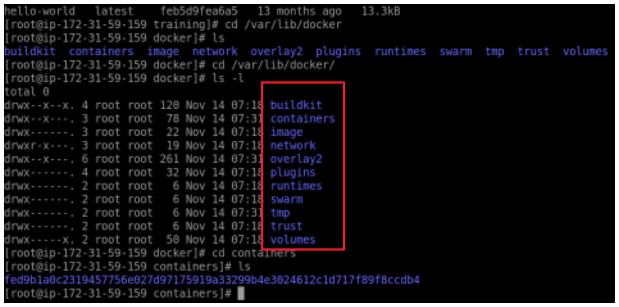
- mkdir Dockerfolder
- vi DockerFile

```
FROM ubuntu:18.04
RUN apt-get update
RUN apt-get install apache2 -y
# Create web page
RUN echo "docker training" > /var/www/html/index.html
# Create script to start Apache in fore ground
RUN echo ". /etc/apache2/envvars" > /run.sh
RUN echo "mkdir -p /var/run/apache2" >> /run.sh
RUN echo "mkdir -p /var/lock/apache2" >> /run.sh
RUN echo "/usr/sbin/apache2 -D FOREGROUND" >> /run.sh

RUN chmod 755 /run.sh

EXPOSE 80

CMD /run.sh
```

esc > :wq >enter

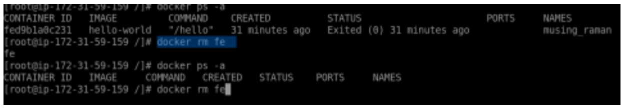
https://hub.docker.com/ -> Search hello-world

- docker run "image name"
- docker ps --help
- docker images (this command list the images which we have run on local host)
- cd /var/lib/docker



```
hello-world   latest    feb5d9fea6a5   13 months ago   13.3kB
[root@ip-172-31-59-159 training]# cd /var/lib/docker
[root@ip-172-31-59-159 docker]# ls
buildkit  containers  image  network  overlay2  plugins  runtimes  swarm  tmp  trust  volumes
[root@ip-172-31-59-159 docker]# cd /var/lib/docker/
[root@ip-172-31-59-159 docker]# ls -l
total 0
drwx--x--x. 4 root root 120 Nov 14 07:18 buildkit
drwx--x---. 3 root root  78 Nov 14 07:31 containers
drwx------. 3 root root  22 Nov 14 07:18 image
drwxr-x---. 3 root root  19 Nov 14 07:18 network
drwx--x---. 6 root root 261 Nov 14 07:31 overlay2
drwx------. 4 root root  32 Nov 14 07:18 plugins
drwx------. 2 root root   6 Nov 14 07:18 runtimes
drwx------. 2 root root   6 Nov 14 07:18 swarm
drwx------. 2 root root   6 Nov 14 07:31 tmp
drwx------. 2 root root   6 Nov 14 07:18 trust
drwx-----x. 2 root root  50 Nov 14 07:18 volumes
[root@ip-172-31-59-159 docker]# cd containers
[root@ip-172-31-59-159 containers]# ls
fed9b1a0c2319457756e027d97175919a33299b4e3024612c1d717f89f8ccdb4
[root@ip-172-31-59-159 containers]# 
```

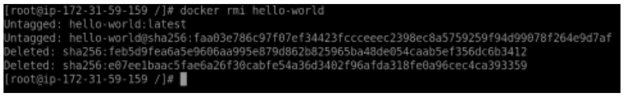
- docker rm fe (remove exited/stopped container)



```
[root@ip-172-31-59-159 /]# docker ps -a
CONTAINER ID   IMAGE         COMMAND    CREATED          STATUS                      PORTS     NAMES
fed9b1a0c231   hello-world   "/hello"   31 minutes ago   Exited (0) 31 minutes ago             musing_raman
[root@ip-172-31-59-159 /]# docker rm fe
fe
[root@ip-172-31-59-159 /]# docker ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
[root@ip-172-31-59-159 /]# docker rm fe
```

- $docker ps -a -f status=exited -f status=created -q - remove all exited containers , it’s a Linux command not docker specific.

- docker rmi hello-world - removes image



```
[root@ip-172-31-59-159 /]# docker rmi hello-world
Untagged: hello-world:latest
Untagged: hello-world@sha256:faa03e786c97f07ef34423fccceeec2398ec8a5759259f94d99078f264e9d7af
Deleted: sha256:feb5d9fea6a5e9606aa995e879d862b825965ba48de054caab5ef356dc6b3412
Deleted: sha256:e07ee1baac5fae6a26f30cabfe54a36d3402f96afda318fe0a96cec4ca393359
[root@ip-172-31-59-159 /]# 
```

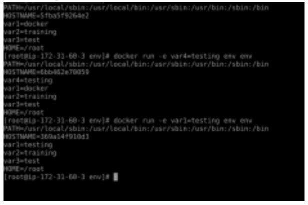
- Docker pull "image name" - pull image form registry.

Microservices - Docker 2

Wednesday, November 16, 2022 10:13 AM

Concept of Env variables :

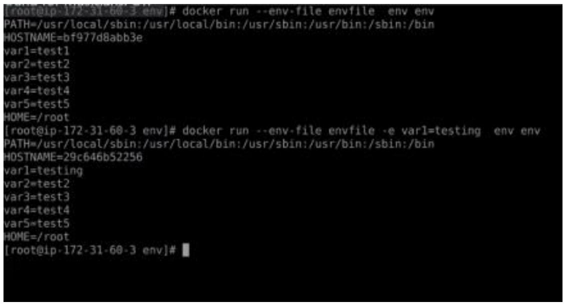
- Running Envs in docker



```
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
HOSTNAME=5fba5f9264e2
var1=docker
var2=training
var3=test
HOME=/root
[root@ip-172-31-60-3 env]# docker run -e var4=testing env env
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
HOSTNAME=6bb462e70059
var4=testing
var1=docker
var2=training
var3=test
HOME=/root
[root@ip-172-31-60-3 env]# docker run -e var1=testing env env
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
HOSTNAME=369a14f910d3
var1=testing
var2=training
var3=test
HOME=/root
[root@ip-172-31-60-3 env]#
```

Precedence for runtime variable will be high

- Pass env as a file



```
[root@ip-172-31-60-3 env]# docker run --env-file envfile  env env
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
HOSTNAME=bf977d8abb3e
var1=test1
var2=test2
var3=test3
var4=test4
var5=test5
HOME=/root
[root@ip-172-31-60-3 env]# docker run --env-file envfile -e var1=testing  env env
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
HOSTNAME=29c646b52256
var1=testing
var2=test2
var3=test3
var4=test4
var5=test5
HOME=/root
[root@ip-172-31-60-3 env]#
```

Precedence for runtime variable will be high

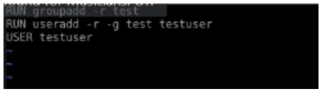
- Copying directive as binary based file in docker containers

Vi dockerfile
FROM Ubuntu
COPY code.jar /tmp

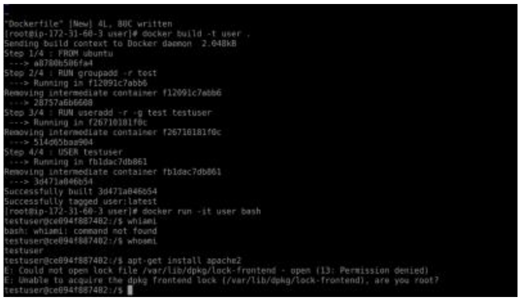
Docker build -t copy .
Docker run -it copy bash
cat dockerfile

- Adding groups and userss

FROM ubuntu



```
RUN groupadd -r test
RUN useradd -r -g test testuser
USER testuser
~
~
~
```

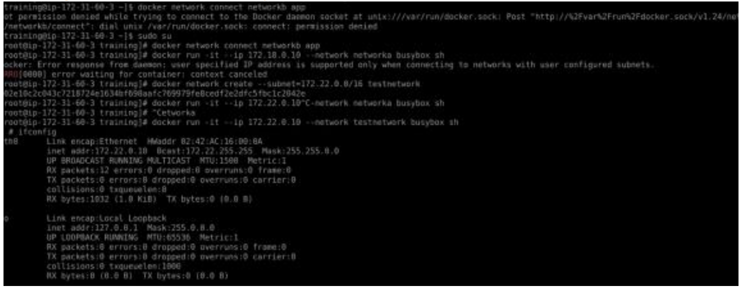


```
~
"Dockerfile" [New] 4L, 80C written
[root@ip-172-31-60-3 user]# docker build -t user .
Sending build context to Docker daemon  2.048kB
Step 1/4 : FROM ubuntu
 ---> a8780b506fa4
Step 2/4 : RUN groupadd -r test
 ---> Running in f12091c7abb6
Removing intermediate container f12091c7abb6
 ---> 28757a6b6608
Step 3/4 : RUN useradd -r -g test testuser
 ---> Running in f26710181f0c
Removing intermediate container f26710181f0c
 ---> 514d65baa904
Step 4/4 : USER testuser
 ---> Running in fb1dac7db861
Removing intermediate container fb1dac7db861
 ---> 3d471a046b54
Successfully built 3d471a046b54
Successfully tagged user:latest
[root@ip-172-31-60-3 user]# docker run -it user bash
testuser@ce094f887402:/$ whiami
bash: whiami: command not found
testuser@ce094f887402:/$ whoami
testuser
testuser@ce094f887402:/$ apt-get install apache2
E: Could not open lock file /var/lib/dpkg/lock-frontend - open (13: Permission denied)
E: Unable to acquire the dpkg frontend lock (/var/lib/dpkg/lock-frontend), are you root?
testuser@ce094f887402:/$
```

Networking concept:

- Ifconfig
- Docker network ls : list all the network when we installed docker - bridge (default), host, none(when we want to run without internet connectivity).
- docker images>docker run -d --network host deloitteabhi>docker ps>docker inspect "container_id"
- Running three containers
  - First terminal : docker run -it ---network network1 --name app busybox sh
  - Second Terminal : docker run -it ---network network1 --name app busybox sh
  - Third Terminal : docker run -it ---network network1 --name db busybox sh
  - Fourth Terminal : docker network connect networkb
  - ping -c 2 172.18.0.3 from terminal first.

Attaching custom subnet with static ip :

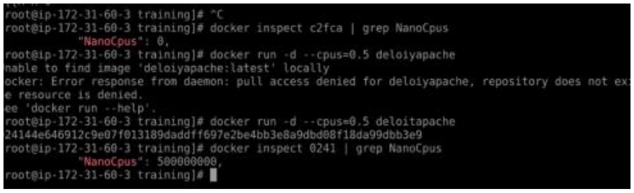


```
training@ip-172-31-60-3 ~]$ docker network connect networkb app
ot permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Post "http://%2Fvar%2Frun%2Fdocker.sock/v1.24/ne
/networkb/connect": dial unix /var/run/docker.sock: connect: permission denied
training@ip-172-31-60-3 ~]$ sudo su
root@ip-172-31-60-3 training]# docker network connect networkb app
root@ip-172-31-60-3 training]# docker run -it --ip 172.18.0.10 --network networka busybox sh
ocker: Error response from daemon: user specified IP address is supported only when connecting to networks with user configured subnets.
RRO[0000] error waiting for container: context canceled
root@ip-172-31-60-3 training]# docker network create --subnet=172.22.0.0/16 testnetwork
02e10c2c043c7218724e1634bf698aafc769979fe8cedf2e2dfc5fbc1c2042e
root@ip-172-31-60-3 training]# docker run -it --ip 172.22.0.10^C-network networka busybox sh
root@ip-172-31-60-3 training]# ^Cetworka
root@ip-172-31-60-3 training]# docker run -it --ip 172.22.0.10 --network testnetwork busybox sh
 # ifconfig
th0       Link encap:Ethernet  HWaddr 02:42:AC:16:00:0A
          inet addr:172.22.0.10  Bcast:172.22.255.255  Mask:255.255.0.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:12 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:1032 (1.0 KiB)  TX bytes:0 (0.0 B)

o         Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

Restricting container from using CPU/Memory:

Memory :docker run -d --memory='256m' deloitteapache

CPU:



```
root@ip-172-31-60-3 training]# ^C
root@ip-172-31-60-3 training]# docker inspect c2fca | grep NanoCpus
            "NanoCpus": 0,
root@ip-172-31-60-3 training]# docker run -d --cpus=0.5 deloiyapache
nable to find image 'deloiyapache:latest' locally
ocker: Error response from daemon: pull access denied for deloiyapache, repository does not ex
e resource is denied.
ee 'docker run --help'.
root@ip-172-31-60-3 training]# docker run -d --cpus=0.5 deloitapache
24144e646912c9e07f013189daddff697e2be4bb3e8a9dbd08f18da99dbb3e9
root@ip-172-31-60-3 training]# docker inspect 0241 | grep NanoCpus
            "NanoCpus": 500000000,
root@ip-172-31-60-3 training]#
```

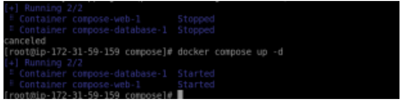
Docker Compose

docker compose version

Sample yaml file

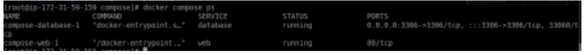
```
version: '4.2'
services:
 database:
  image: mysql
  ports:
  - "3306:3306"
  environment:
  - MYSQL_ROOT_PASSWORD=password
  - MYSQL_USER=user
  - MYSQL_PASSWORD=password
  - MYSQL_DATABASE=demo
 web:
  image: nginx
```

- mkdir compose >cd compose >vi compose-file.yml
- docker compose up
- docker compose up -d -->run in Daemon mode



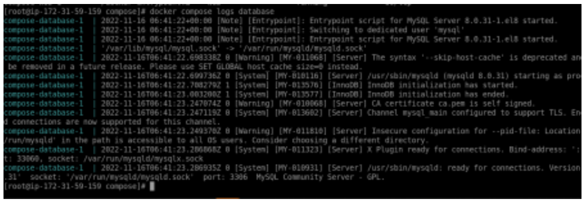
```
[+] Running 2/2
 ⠿ Container compose-web-1       Stopped
 ⠿ Container compose-database-1  Stopped
canceled
[root@ip-172-31-59-159 compose]# docker compose up -d
[+] Running 2/2
 ⠿ Container compose-database-1  Started
 ⠿ Container compose-web-1       Started
[root@ip-172-31-59-159 compose]#
```

- docker compose ps



```
[root@ip-172-31-59-159 compose]# docker compose ps
NAME                 COMMAND                  SERVICE             STATUS              PORTS
compose-database-1   "docker-entrypoint.s…"   database            running             0.0.0.0:3306->3306/tcp, :::3306->3306/tcp, 33060/t
cp
compose-web-1        "/docker-entrypoint.…"   web                 running             80/tcp
```

- docker compose logs database

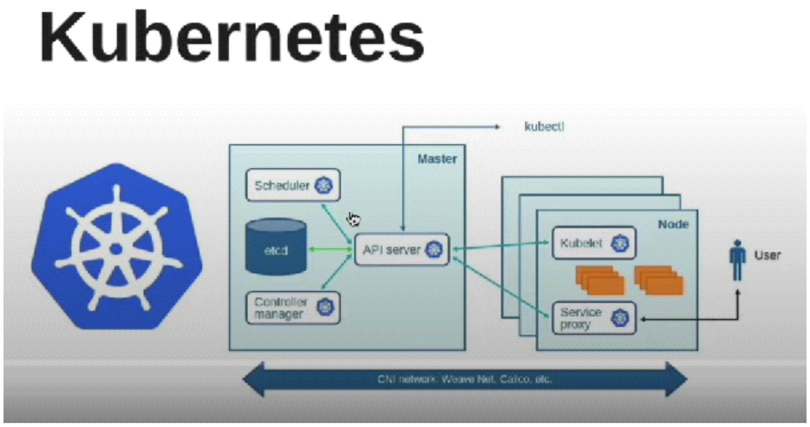


```
[root@ip-172-31-59-159 compose]# docker compose logs database
compose-database-1  | 2022-11-16 06:41:22+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 8.0.31-1.el8 started.
compose-database-1  | 2022-11-16 06:41:22+00:00 [Note] [Entrypoint]: Switching to dedicated user 'mysql'
compose-database-1  | 2022-11-16 06:41:22+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 8.0.31-1.el8 started.
compose-database-1  | '/var/lib/mysql/mysql.sock' -> '/var/run/mysqld/mysqld.sock'
compose-database-1  | 2022-11-16T06:41:22.698338Z 0 [Warning] [MY-011068] [Server] The syntax '--skip-host-cache' is deprecated an
 be removed in a future release. Please use SET GLOBAL host_cache_size=0 instead.
compose-database-1  | 2022-11-16T06:41:22.699736Z 0 [System] [MY-010116] [Server] /usr/sbin/mysqld (mysqld 8.0.31) starting as pro
compose-database-1  | 2022-11-16T06:41:22.708279Z 1 [System] [MY-013576] [InnoDB] InnoDB initialization has started.
compose-database-1  | 2022-11-16T06:41:23.003200Z 1 [System] [MY-013577] [InnoDB] InnoDB initialization has ended.
compose-database-1  | 2022-11-16T06:41:23.247074Z 0 [Warning] [MY-010068] [Server] CA certificate ca.pem is self signed.
compose-database-1  | 2022-11-16T06:41:23.247119Z 0 [System] [MY-013602] [Server] Channel mysql_main configured to support TLS. En
d connections are now supported for this channel.
compose-database-1  | 2022-11-16T06:41:23.249370Z 0 [Warning] [MY-011810] [Server] Insecure configuration for --pid-file: Location
/run/mysqld' in the path is accessible to all OS users. Consider choosing a different directory.
compose-database-1  | 2022-11-16T06:41:23.286868Z 0 [System] [MY-011323] [Server] X Plugin ready for connections. Bind-address: ':
t: 33060, socket: /var/run/mysqld/mysqlx.sock
compose-database-1  | 2022-11-16T06:41:23.286935Z 0 [System] [MY-010931] [Server] /usr/sbin/mysqld: ready for connections. Version
.31'  socket: '/var/run/mysqld/mysqld.sock'  port: 3306  MySQL Community Server - GPL.
[root@ip-172-31-59-159 compose]# 
```

- docker compose logs database -> view database specific logs
- docker compose logs web -> view web container specific log

Kubernetes

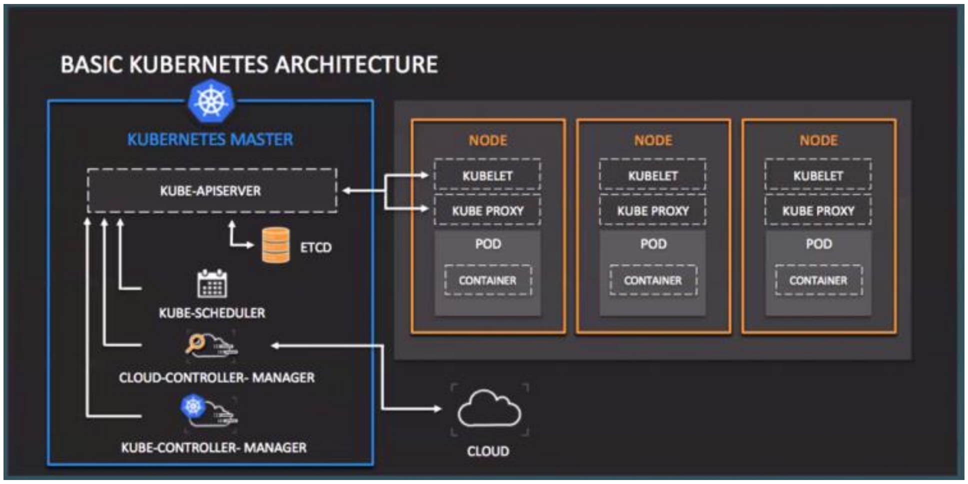
Monday, November 21, 2022 1:54 PM



Kubernetes

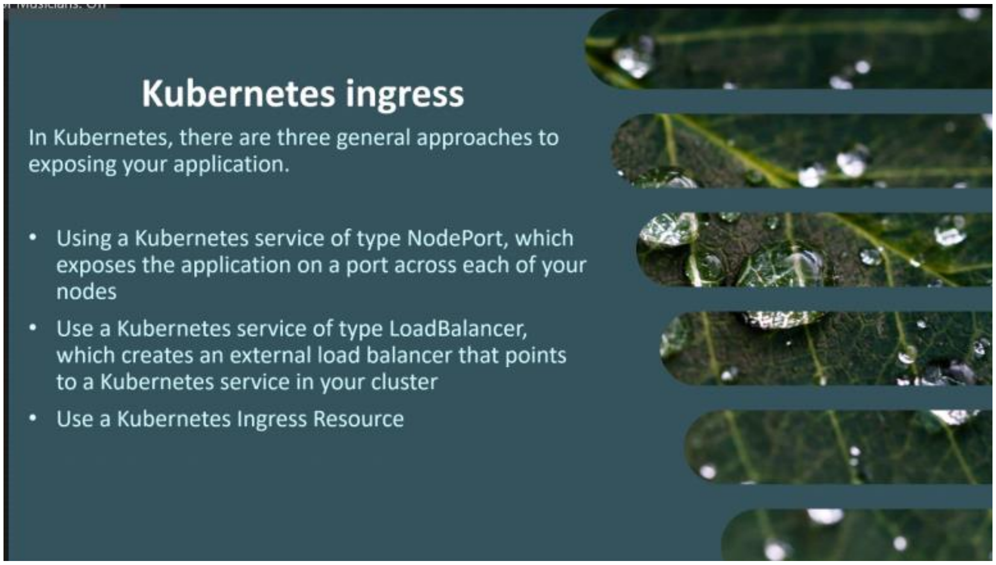
Microservices - Kubernetes Concepts

Wednesday, November 16, 2022 12:41 PM



Details : https://www.vmware.com/topics/glossary/content/kubernetes-architecture.html#:~:text=Kubernetes%20is%20an%20architecture%20that,one%20or%20more%20compute%20nodes.

combination of clusters called pods



Kubernetes ingress

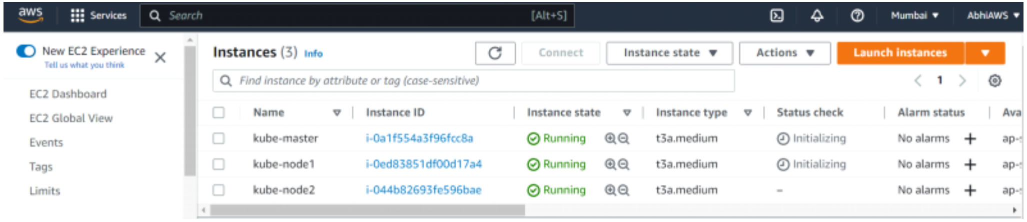
In Kubernetes, there are three general approaches to exposing your application.

- Using a Kubernetes service of type NodePort, which exposes the application on a port across each of your nodes
- Use a Kubernetes service of type LoadBalancer, which creates an external load balancer that points to a Kubernetes service in your cluster
- Use a Kubernetes Ingress Resource

these 3 can expose service to external world

This is manual process of doing the the file configuration

Setup 3 Amazon instances -with Debian



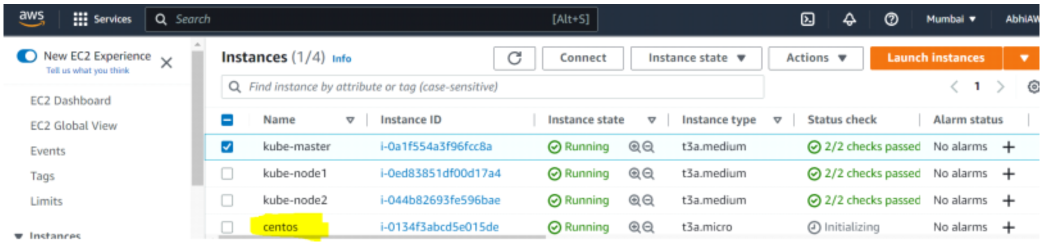
Instances (3) Info

□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Ava
□	kube-master	i-0a1f554a3f96fcc8a	Running	t3a.medium	Initializing	No alarms	ap-
□	kube-node1	i-0ed83851df00d17a4	Running	t3a.medium	Initializing	No alarms	ap-
□	kube-node2	i-044b82693fe596bae	Running	t3a.medium	-	No alarms	ap-

IPV4 Private Address
Masteriaddress-172.31.54.2
Node1-172.31.58.33
Node2-172.31.59.248

In security group, edit inbound rules > allow all traffic

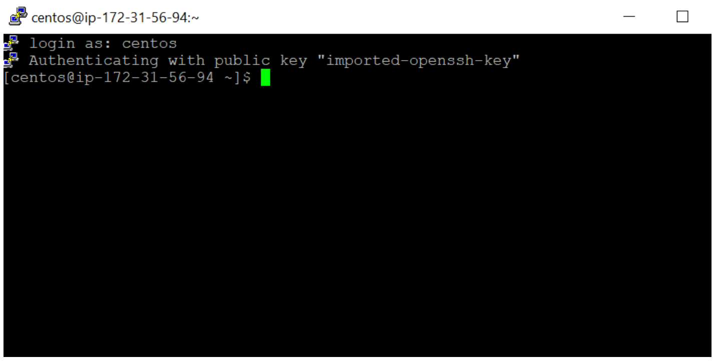
launch centOS as another instance : this instance will have all configuration files



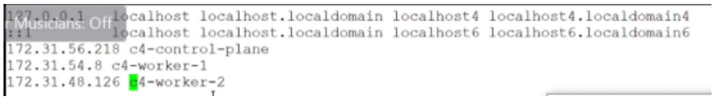
Instances (1/4) Info

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status
☑	kube-master	i-0a1f554a3f96fcc8a	Running	t3a.medium	2/2 checks passed	No alarms
□	kube-node1	i-0ed83851df00d17a4	Running	t3a.medium	2/2 checks passed	No alarms
□	kube-node2	i-044b82693fe596bae	Running	t3a.medium	2/2 checks passed	No alarms
□	centos	i-0134f3abcd5e015de	Running	t3a.micro	Initializing	No alarms

To connect with centos:
Open puttygen > load > save private key
Open putty >add centos public IPV4>add ppk file in Auth



```
login as: centos
Authenticating with public key "imported-openssh-key"
[centos@ip-172-31-56-94 ~]$
```



```
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
172.31.56.218 c4-control-plane
172.31.54.8 c4-worker-1
172.31.48.126 c4-worker-2
```

https://www.oueta.com/linux/create-kubernetes-cluster-the-hard-way/

Master actions :
hostnamectl set-hostname c4-worker-1
root@ip-172-31-58-33:/home/admin# vi /etc/hosts > 127.0.0.1 c4-worker-1
root@ip-172-31-58-33:/home/admin# exec bash

Microservices - Enabling cluster by Terraform

Friday, November 18, 2022 10:10 AM

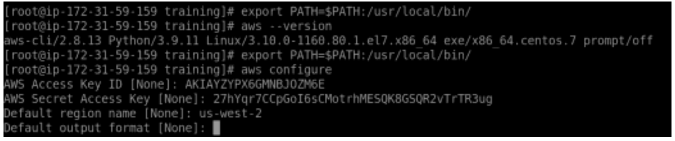
AKIAYZYPX6GMNBJOZM6E
27hYqr7CCpGoI6sCMotrhMES
QK8GSQR2vTrTR3ug

Trainee08	605086544280	clouduser	9mfoGhPWhpdErkCq	https://605086544280.signin.aws.amazon.com/console	us-west-2

Provisioning Kubernetes to Terraform

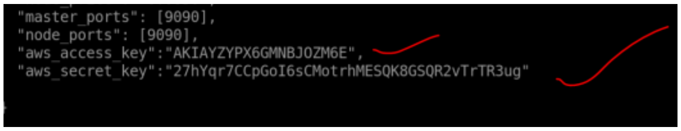
Configuring AWS

Go to AWS > IAM > users >select > security credentials > add access key



```
[root@ip-172-31-59-159 training]# export PATH=$PATH:/usr/local/bin/
[root@ip-172-31-59-159 training]# aws --version
aws-cli/2.8.13 Python/3.9.11 Linux/3.10.0-1160.80.1.el7.x86_64 exe/x86_64.centos.7 prompt/off
[root@ip-172-31-59-159 training]# export PATH=$PATH:/usr/local/bin/
[root@ip-172-31-59-159 training]# aws configure
AWS Access Key ID [None]: AKIAYZYPX6GMNBJOZM6E
AWS Secret Access Key [None]: 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
Default region name [None]: us-west-2
Default output format [None]:
```

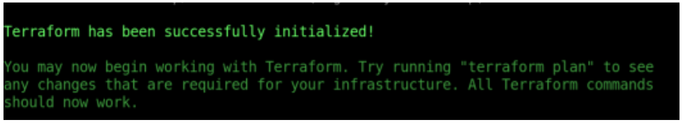
Edit terraform.tvars.json



```
  "master_ports": [9090],
  "node_ports": [9090],
  "aws_access_key":"AKIAYZYPX6GMNBJOZM6E",
  "aws_secret_key":"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"
```

Terraform.tfvars.json >

$terraform init

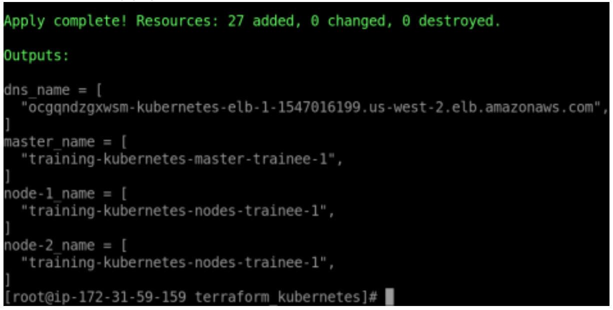


```
Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.
```

$terraform plan -

The terraform plan command evaluates a Terraform configuration to determine the desired state of all the resources it declares, then compares that desired state to the real infrastructure objects being managed with the current working directory and workspace.

$terraform apply



```
Apply complete! Resources: 27 added, 0 changed, 0 destroyed.

Outputs:

dns_name = [
  "ocgqndzgxwsm-kubernetes-elb-1-1547016199.us-west-2.elb.amazonaws.com",
]
master_name = [
  "training-kubernetes-master-trainee-1",
]
node-1_name = [
  "training-kubernetes-nodes-trainee-1",
]
node-2_name = [
  "training-kubernetes-nodes-trainee-1",
]
[root@ip-172-31-59-159 terraform_kubernetes]#
```

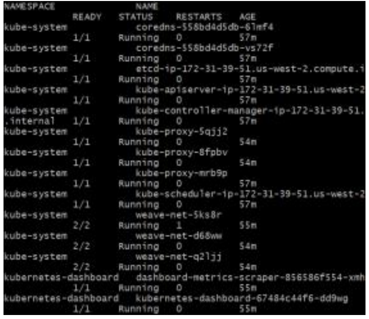
See for load-balancer, attached to master instance

Master public ip - 52.12.100.76
Node1-35.93.7.164
Node2-

Login to master using putty :
sudo su > kubectl get nodes

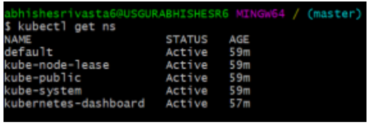
Some Basic Kubernetes Commands (on 4th slave node -local/slave)

$ kubectl get pod --all-namespaces --> to see all containers and clusters



```
NAMESPACE                  NAME
               READY   STATUS    RESTARTS   AGE
kube-system          coredns-558bd4d5db-6lmf4
               1/1     Running   0          57m
kube-system          coredns-558bd4d5db-vs72f
               1/1     Running   0          57m
kube-system          etcd-ip-172-31-39-51.us-west-2.compute.i
               1/1     Running   0          57m
kube-system          kube-apiserver-ip-172-31-39-51.us-west-2
               1/1     Running   0          57m
kube-system          kube-controller-manager-ip-172-31-39-51.
.internal      1/1     Running   0          57m
kube-system          kube-proxy-5qjj2
               1/1     Running   0          54m
kube-system          kube-proxy-8fpbv
               1/1     Running   0          54m
kube-system          kube-proxy-mrb9p
               1/1     Running   0          57m
kube-system          kube-scheduler-ip-172-31-39-51.us-west-2
               1/1     Running   0          57m
kube-system          weave-net-5ks8r
               2/2     Running   1          55m
kube-system          weave-net-d68ww
               2/2     Running   0          54m
kube-system          weave-net-q2ljj
               2/2     Running   0          54m
kubernetes-dashboard   dashboard-metrics-scraper-856586f554-xmh
               1/1     Running   0          55m
kubernetes-dashboard   kubernetes-dashboard-67484c44f6-dd9wg
               1/1     Running   0          55m
```

$ kubectl get ns



```
abhishesrivasta6@USGURABHISHESR6 MINGW64 / (master)
$ kubectl get ns
NAME                   STATUS   AGE
default                Active   59m
kube-node-lease        Active   59m
kube-public            Active   59m
kube-system            Active   59m
kubernetes-dashboard   Active   57m
```

Namespaces : Namespaces are a way to organize clusters into virtual sub-clusters — they can be helpful when different teams or projects share a Kubernetes cluster. Any number of namespaces are supported within a cluster, each logically separated from others but with the ability to communicate with each other.

Initial namespaces

Kubernetes starts with four initial namespaces:

- Default : Kubernetes includes this namespace so that you can start using your new cluster without first creating a namespace.
- kube-node-lease : This namespace holds Lease objects associated with each node. Node leases allow the kubelet to send heartbeats so that the control plane can detect node failure.
- kube-public : This namespace is readable by all clients (including those not authenticated). This namespace is mostly reserved for cluster usage, in case that some resources should be visible and readable publicly throughout the whole cluster. The public aspect of this namespace is only a convention, not a requirement.
- kube-system :The namespace for objects created by the Kubernetes system.

$ kubectl get po -n kube-system --> get all pods
$ kubectl get svc -n kube-system
$ kubectl get deploy -n kube-system
$ kubectl delete pode_name -n kube-system -->deletes pod but adm will automaticall recreate the pod

Dashboard monitoring:

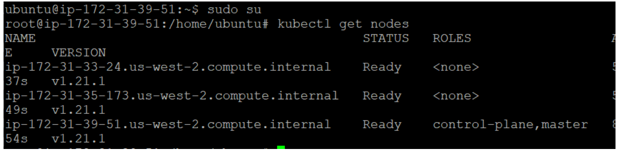
$ kubectl apply -f https://raw.githubusercontent.com/kubernetes/dashboard/v2.6.1/aio/deploy/recommended.yaml

$ kubectl proxy
Kubectl will make Dashboard available

Node2-

Login to master using putty :

sudo su > kubectl get nodes



```
ubuntu@ip-172-31-39-51:~$ sudo su
root@ip-172-31-39-51:/home/ubuntu# kubectl get nodes
NAME                                         STATUS   ROLES                  A
E      VERSION
ip-172-31-33-24.us-west-2.compute.internal   Ready    <none>                 5
37s    v1.21.1
ip-172-31-35-173.us-west-2.compute.internal  Ready    <none>                 5
49s    v1.21.1
ip-172-31-39-51.us-west-2.compute.internal   Ready    control-plane,master   8
54s    v1.21.1
```

Custer is created !

From Master :

cat ~/.kube/config

copy and paste the content to local > users>abhishek>.kube>   config file

From Local :

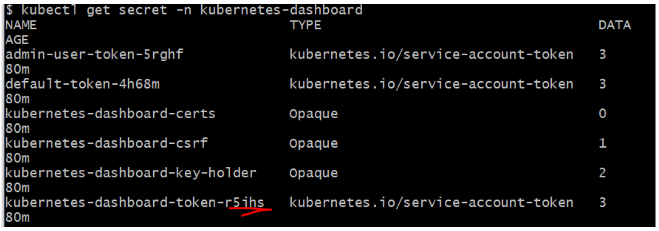
kubectl get nodes in local

eploy/recommended.yaml

$ kubectl proxy

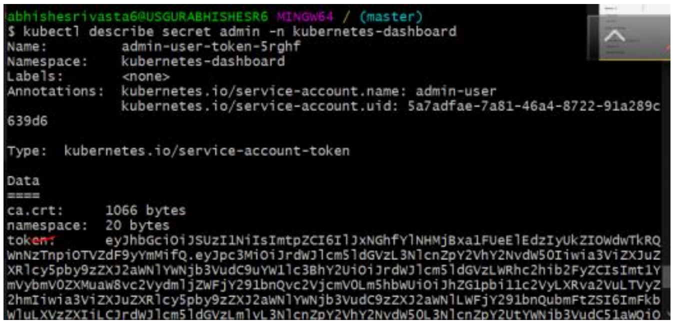
Kubectl will make Dashboard available at http://localhost:8001/api/v1/namespaces/kubernetes-dashboard/services/https:kubernetes-dashboard:/proxy/.

$ kubectl get secret -n kubernetes-dashboard -->provides token for login



```
$ kubectl get secret -n kubernetes-dashboard
NAME                               TYPE                                  DATA
AGE
admin-user-token-5rghf             kubernetes.io/service-account-token   3
80m
default-token-4h68m                kubernetes.io/service-account-token   3
80m
kubernetes-dashboard-certs         Opaque                                0
80m
kubernetes-dashboard-csrf          Opaque                                1
80m
kubernetes-dashboard-key-holder    Opaque                                2
80m
kubernetes-dashboard-token-r5ihs   kubernetes.io/service-account-token   3
80m
```

$ kubectl describe secret admin -n kubernetes-dashboard-->generates token



```
abhishesrivasta6@USGURABHISHESR6 MINGW64 / (master)
$ kubectl describe secret admin -n kubernetes-dashboard
Name:         admin-user-token-5rghf
Namespace:    kubernetes-dashboard
Labels:       <none>
Annotations:  kubernetes.io/service-account.name: admin-user
              kubernetes.io/service-account.uid: 5a7adfae-7a81-46a4-8722-91a289c
639d6

Type:  kubernetes.io/service-account-token

Data
====
ca.crt:     1066 bytes
namespace:  20 bytes
token:      eyJhbGciOiJSUzI1NiIsImtpZCI6IlJxNGhfYlNHMjBxa1FUeElEdzIyUkZIOWdwTkRQ
WnNzTnpiOTVZdF9yYmMifQ.eyJpc3MiOiJrdWJlcm5ldGVzL3NlcnZpY2VhY2NvdW50Iiwia3ViZXJuZ
XRlcy5pby9zZXJ2aWNlYWNjb3VudC9uYW1lc3BhY2UiOiJrdWJlcm5ldGVzLWRhc2hib2FyZCIsImt1Y
mVybmV0ZXMuaW8vc2VydmljZWFjY291bnQvc2VjcmV0Lm5hbWUiOiJhZG1pbi11c2VyLXRva2VuLTVyZ
2hmIiwia3ViZXJuZXRlcy5pby9zZXJ2aWNlYWNjb3VudC9zZXJ2aWNlLWFjY291bnQubmFtZSI6ImFkb
WluLXVzZXIiLCJrdWJlcm5ldGVzLmlvL3NlcnZpY2VhY2NvdW50L3NlcnZpY2UtYWNjb3VudC51aWQiO
```

Microservices - Deployments

Monday, November 21, 2022 10:30 AM

https://kubernetes.io/docs/reference/kubernetes-api/workload-resources/deployment-v1/

From <https://teams.microsoft.com/multi-window/?agent=electron&version=22102807200>

Creating deployment :

Yaml :

```
piVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  labels:
    app: nginx
spec:
  replicas: 4
  selector:
    matchLabels:
      app: nginx1
  template:
    metadata:
      labels:
        app: nginx1
    spec:
      containers:
      - name: nginxpod
        image: nginx
        ports:
        - containerPort: 80
---
apiVersion: v1
kind: Service
metadata:
  name: webservice
  labels:
    app: nginx
spec:
  ports:
  - port: 80
    protocol: TCP
  selector:
    app: nginx
  type: LoadBalancer
```

```
kubectl apply -f firstapp.yml
```

Microservices - Blue/Green Deployment Strategy

Tuesday, November 22, 2022 10:23 AM

Blue-Deployment

Vi blue.yaml
Kubectl apply -f blue.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
 name: nginx-blue
 labels:
  app: nginx
spec:
 replicas: 2
 selector:
  matchLabels:
   app: nginx
 template:
  metadata:
   labels:
    app: nginx
    version: "blue"
  spec:
   containers:
   - name: nginxpod
    image: nginx
    ports:
    - containerPort: 80
```

Create Service

Vi service.yaml

Kubectl apply -f service.yaml
Kubectl get svc

```
apiVersion: v1
kind: Service
metadata:
 name: webappbluegreen
 labels:
  app: nginx
spec:
 ports:
 - port: 80
  protocol: TCP
 selector:
  app: nginx
  version: "blue"
 type: LoadBalancer
```

Service.yaml with changes

```
apiVersion: v1
kind: Service
metadata:
 name: webappbluegreen
 labels:
  app: nginx
spec:
 ports:
 - port: 80
  protocol: TCP
 selector:
  app: nginx
  version: "green"
 type: LoadBalancer
```

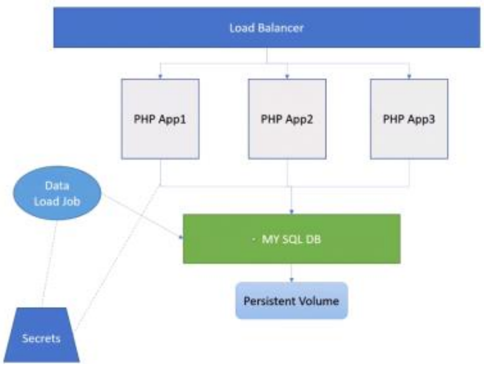
Green -Deployment

Vi green.yaml
Kubectl apply -f green.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
 name: nginx-green
 labels:
  app: nginx
spec:
 replicas: 2
 selector:
  matchLabels:
   app: nginx
 template:
  metadata:
   labels:
    app: nginx
    version: "green"
  spec:
   containers:
   - name: nginxpod
    image: nginx:1.22.1
    ports:
    - containerPort: 80
```

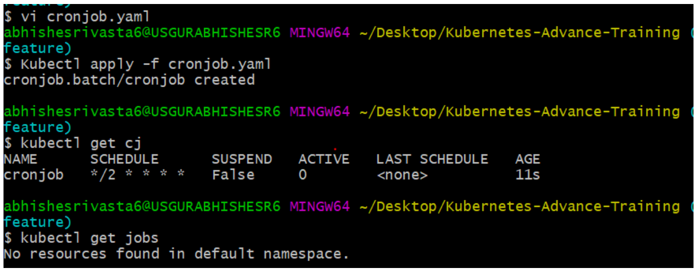
Cron-Job - Running container on pre-defined schedule

```
apiVersion: batch/v1
kind: CronJob
metadata:
 name: cronjob
spec:
 schedule: "*/2 * * * *"
 jobTemplate:
  spec:
   template:
    spec:
     containers:
     - name: testjob
      image: bash
      command: ["echo", "k8s training"] --> this will print k8s
training and go away
     restartPolicy: OnFailure
```



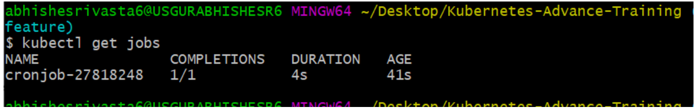
creating volume and attaching it to machine
2creating DB to attach volume
1creating secret-credentials to pass to DB
creating internal service: to expose service to clusterIP

```
        command: ["echo", "k8s training"] --> this will print k8s training and go away
      restartPolicy: OnFailure
```



```
$ vi cronjob.yaml
abhishesrivasta6@USGURABHISHESR6 MINGW64 ~/Desktop/Kubernetes-Advance-Training (
feature)
$ Kubectl apply -f cronjob.yaml
cronjob.batch/cronjob created

abhishesrivasta6@USGURABHISHESR6 MINGW64 ~/Desktop/Kubernetes-Advance-Training (
feature)
$ kubectl get cj
NAME      SCHEDULE      SUSPEND   ACTIVE   LAST SCHEDULE   AGE
cronjob   */2 * * * *   False     0        <none>          11s

abhishesrivasta6@USGURABHISHESR6 MINGW64 ~/Desktop/Kubernetes-Advance-Training (
feature)
$ kubectl get jobs
No resources found in default namespace.
```

This will greate one job with pod



```
abhishesrivasta6@USGURABHISHESR6 MINGW64 ~/Desktop/Kubernetes-Advance-Training (
feature)
$ kubectl get jobs
NAME               COMPLETIONS   DURATION   AGE
cronjob-27818248   1/1           4s         41s

abhishesrivasta6@USGURABHISHESR6 MINGW64 ~/Desktop/Kubernetes-Advance-Training
```

creating volume and attaching it to machine
2creating DB to attach volume
1creating secret-credentials to pass to DB
creating internal service: to expose service to clusterIP

livenessprobe: healthiness of container
volumemounts : which directly volume will be mounted

1.Creating secret

```
apiVersion: v1
kind: Secret
metadata:
  name: mysql-credentials
  labels:
    "ostraining": lamp
type: Opaque
data:
  rootpw: dmFyTXlSb290UGFzcw==
  user: dmFyTXlEQlVzZXI=
  password: dmFyTXlEQlBhc3M=
```

2.Creating Database with volume

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: database
  labels:
    "ostraining": lamp
  annotations:
    volume.alpha.kubernetes.io/storage-class: default
spec:
  accessModes:
  - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: mysql
  labels:
    "ostraining": lamp
spec:
  replicas: 1
  revisionHistoryLimit: 5
  selector:
```

3.Creating dataloading job

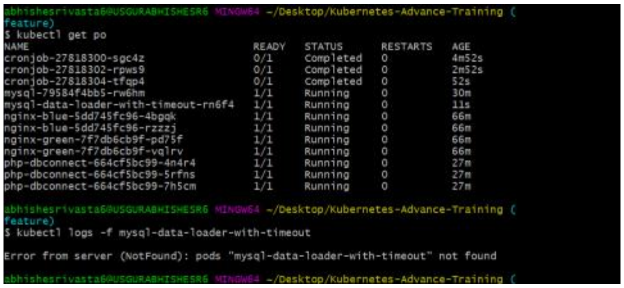
```
apiVersion: batch/v1
kind: Job
metadata:
  name: mysql-data-loader-with-timeout
  labels:
    "ostraining": lamp
spec:
  activeDeadlineSeconds: 100
  template:
    metadata:
      name: mysql-data-loader
    spec:
      containers:
      - name: mysql-data-loader
        image: sathishbob/example-php-dbconnect
        env:
        - name: MYSQL_USER
          valueFrom:
            secretKeyRef:
              name: mysql-credentials
              key: user
        - name: MYSQL_PASSWORD
          valueFrom:
            secretKeyRef:
              name: mysql-credentials
              key: password
        - name: MYSQL_HOST
          value: mysql.default.svc.cluster.local
        command: ["/tmp/mysql-sakila-data-loader.sh"]
      restartPolicy: OnFailure
```

4.Creating webapp

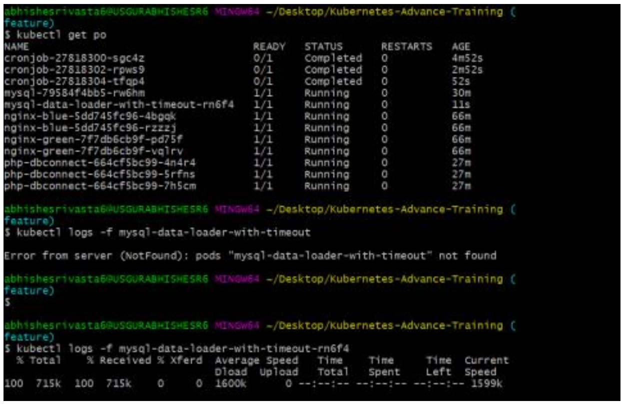
```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: php-dbconnect
  labels:
    "ostraining": lamp
```

```
spec:
  replicas: 1
  revisionHistoryLimit: 5
  selector:
    matchLabels:
      app: mysql
  template:
    metadata:
      labels:
        app: mysql
    spec:
      containers:
      - image: mysql:5.7
        name: mysql
        env:
        - name: MYSQL_ROOT_PASSWORD
          valueFrom:
            secretKeyRef:
              name: mysql-credentials
              key: rootpw
        - name: MYSQL_USER
          valueFrom:
            secretKeyRef:
              name: mysql-credentials
              key: user
        - name: MYSQL_PASSWORD
          valueFrom:
            secretKeyRef:
              name: mysql-credentials
              key: password
        - name: MYSQL_DATABASE
          value: sakila
        livenessProbe:
          tcpSocket:
            port: 3306
        ports:
        - containerPort: 3306
        volumeMounts:
        - mountPath: /var/lib/mysql
          subPath: data
          name: database
      volumes:
      - name: database
        persistentVolumeClaim:
          claimName: database
---
apiVersion: v1
kind: Service
metadata:
  name: mysql
  labels:
    "ostraining": lamp
spec:
  ports:
  - port: 3306
    protocol: TCP
  selector:
    app: mysql
  type: ClusterIP
```

```
metadata:
  name: php-dbconnect
  labels:
    "ostraining": lamp
spec:
  replicas: 3
  revisionHistoryLimit: 5
  selector:
    matchLabels:
      app: php-dbconnect
  template:
    metadata:
      labels:
        app: php-dbconnect
    spec:
      containers:
      - image: sathishbob/example-php-dbconnect
        imagePullPolicy: Always
        name: php-dbconnect
        env:
        - name: MYSQL_USER
          valueFrom:
            secretKeyRef:
              name: mysql-credentials
              key: user
        - name: MYSQL_PASSWORD
          valueFrom:
            secretKeyRef:
              name: mysql-credentials
              key: password
        - name: MYSQL_HOST
          value: mysql.default.svc.cluster.local
        livenessProbe:
          tcpSocket:
            port: 80
        ports:
        - containerPort: 80
---
apiVersion: v1
kind: Service
metadata:
  name: web
  labels:
    "ostraining": lamp
spec:
  ports:
  - port: 80
    protocol: TCP
  selector:
    app: php-dbconnect
  type: LoadBalancer
```



```
abhishesrivasta6@USGURABHISHESR6 MINGW64 ~/Desktop/Kubernetes-Advance-Training (
feature)
$ kubectl get po
NAME                                  READY   STATUS      RESTARTS   AGE
cronjob-27818300-sgc4z                0/1     Completed   0          4m52s
cronjob-27818302-rpws9                0/1     Completed   0          2m52s
cronjob-27818304-tfqp4                0/1     Completed   0          52s
mysql-79584f4bb5-rw6hm                1/1     Running     0          30m
mysql-data-loader-with-timeout-rn6f4  1/1     Running     0          11s
nginx-blue-5dd745fc96-4bgqk           1/1     Running     0          66m
nginx-blue-5dd745fc96-rzzzj           1/1     Running     0          66m
nginx-green-7f7db6cb9f-pd75f          1/1     Running     0          66m
nginx-green-7f7db6cb9f-vqlrv          1/1     Running     0          66m
php-dbconnect-664cf5bc99-4n4r4        1/1     Running     0          27m
php-dbconnect-664cf5bc99-5rfns        1/1     Running     0          27m
php-dbconnect-664cf5bc99-7h5cm        1/1     Running     0          27m

abhishesrivasta6@USGURABHISHESR6 MINGW64 ~/Desktop/Kubernetes-Advance-Training (
feature)
$ kubectl logs -f mysql-data-loader-with-timeout

Error from server (NotFound): pods "mysql-data-loader-with-timeout" not found

abhishesrivasta6@USGURABHISHESR6 MINGW64 ~/Desktop/Kubernetes-Advance-Training (
 - port: 3306
   protocol: TCP
 selector:
   app: mysql
 type: ClusterIP
```



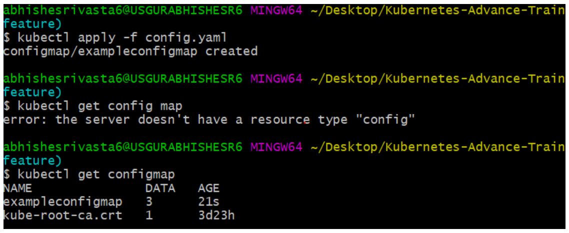
```
abhishesrivasta6@USGURABHISHESR6 MINGW64 ~/Desktop/Kubernetes-Advance-Training (
feature)
$ kubectl get po
NAME                                  READY   STATUS      RESTARTS   AGE
cronjob-27818300-sgc4z                0/1     Completed   0          4m52s
cronjob-27818302-rpws9                0/1     Completed   0          2m52s
cronjob-27818304-tfqp4                0/1     Completed   0          52s
mysql-79584f4bb5-rw6hm                1/1     Running     0          30m
mysql-data-loader-with-timeout-rn6f4  1/1     Running     0          11s
nginx-blue-5dd745fc96-4bgqk           1/1     Running     0          66m
nginx-blue-5dd745fc96-rzzzj           1/1     Running     0          66m
nginx-green-7f7db6cb9f-pd75f          1/1     Running     0          66m
nginx-green-7f7db6cb9f-vqlrv          1/1     Running     0          66m
php-dbconnect-664cf5bc99-4n4r4        1/1     Running     0          27m
php-dbconnect-664cf5bc99-5rfns        1/1     Running     0          27m
php-dbconnect-664cf5bc99-7h5cm        1/1     Running     0          27m

abhishesrivasta6@USGURABHISHESR6 MINGW64 ~/Desktop/Kubernetes-Advance-Training (
feature)
$ kubectl logs -f mysql-data-loader-with-timeout

Error from server (NotFound): pods "mysql-data-loader-with-timeout" not found

abhishesrivasta6@USGURABHISHESR6 MINGW64 ~/Desktop/Kubernetes-Advance-Training (
feature)
$

abhishesrivasta6@USGURABHISHESR6 MINGW64 ~/Desktop/Kubernetes-Advance-Training (
feature)
$ kubectl logs -f mysql-data-loader-with-timeout-rn6f4
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  715k  100  715k    0     0  1600k      0 --:--:-- --:--:-- --:--:-- 1599k
```

Config-Map

Adding env variables with pod

Configmap.yaml

```
apiVersion: v1
kind: ConfigMap
metadata:
 name: exampleconfigmap
data:
 database: mysql
 databaseurl: mysql://db.training.com:3306
 keys: |
   name.training=kubernetes
   program.training=configmap
   trainer.training=sathish
```

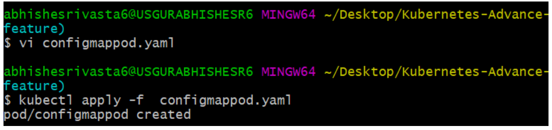


```
abhishesrivasta6@USGURABHISHESR6 MINGW64 ~/Desktop/Kubernetes-Advance-Train
feature)
$ kubectl apply -f config.yaml
configmap/exampleconfigmap created

abhishesrivasta6@USGURABHISHESR6 MINGW64 ~/Desktop/Kubernetes-Advance-Train
feature)
$ kubectl get config map
error: the server doesn't have a resource type "config"

abhishesrivasta6@USGURABHISHESR6 MINGW64 ~/Desktop/Kubernetes-Advance-Train
feature)
$ kubectl get configmap
NAME               DATA   AGE
exampleconfigmap   3      21s
kube-root-ca.crt   1      3d23h
```

Bring pod and attach configmap to pod

configmappod.yaml

```
apiVersion: v1
kind: Pod
metadata:
 name: configmappod
spec:
 containers:
   - name: configmappod
     image: nginx
     envFrom:
       - configMapRef:
           name: exampleconfigmap
```



```
abhishesrivasta6@USGURABHISHESR6 MINGW64 ~/Desktop/Kubernetes-Advance-
feature)
$ vi configmappod.yaml

abhishesrivasta6@USGURABHISHESR6 MINGW64 ~/Desktop/Kubernetes-Advance-
feature)
$ kubectl apply -f  configmappod.yaml
pod/configmappod created
```

```
abhishesrivasta6@USGURABHISHESR6 MINGW64 ~/Desktop/Kubernetes-Advance-Training (
```

Attaching env var files in running container

First file mounted to machine then mount to container

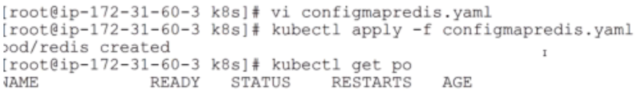
redis-config.yaml

```
maxmemory 2mb
maxmemory-policy allkeys-lru
```

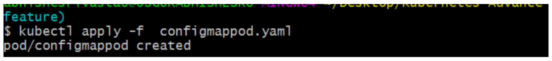
```
$ kubectl create configmap redis --from-file=redis-config.yaml
```

confimapredis.yaml

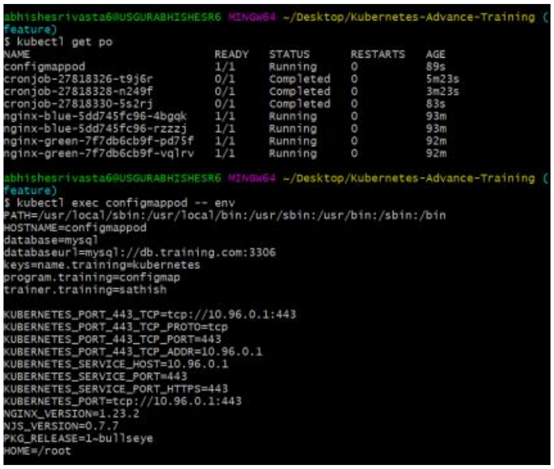
```
apiVersion: v1
kind: Pod
metadata:
 name: redis
spec:
 containers:
 - name: redis
   image: redis:5.0.4
   command:
     - redis-server
     - "/redis-master/redis.conf"
   env:
   - name: MASTER
     value: "true"
   ports:
   - containerPort: 6379
   volumeMounts:
   - mountPath: /redis-master-data
     name: data
   - mountPath: /redis-master
     name: config
 volumes:
   - name: data
     emptyDir: {}
   - name: config
     configMap:
       name: redis
       items:
       - key: redis-config
         path: redis.conf
```



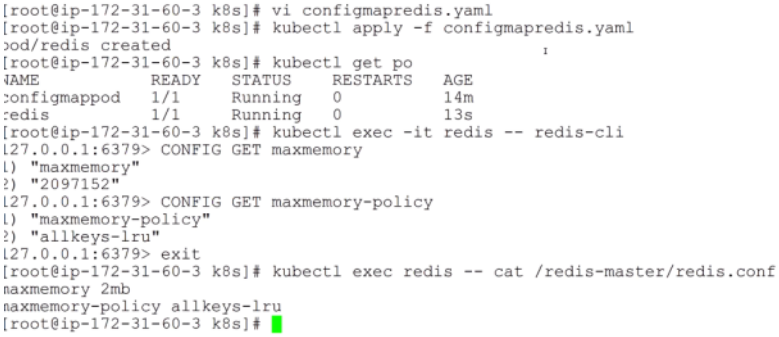
```
[root@ip-172-31-60-3 k8s]# vi configmapredis.yaml
[root@ip-172-31-60-3 k8s]# kubectl apply -f configmapredis.yaml
pod/redis created
[root@ip-172-31-60-3 k8s]# kubectl get po
NAME             READY    STATUS     RESTARTS    AGE
```



```
abhishesrivasta6@USGURABHISHESR6 MINGW64 ~/Desktop/Kubernetes-Advance
feature)
$ kubectl apply -f  configmappod.yaml
pod/configmappod created
```



```
abhishesrivasta6@USGURABHISHESR6 MINGW64 ~/Desktop/Kubernetes-Advance-Training (
feature)
$ kubectl get po
NAME                             READY   STATUS      RESTARTS   AGE
configmappod                     1/1     Running     0          89s
cronjob-27818326-t9j6r           0/1     Completed   0          5m23s
cronjob-27818328-n249f           0/1     Completed   0          3m23s
cronjob-27818330-5s2rj           0/1     Completed   0          83s
nginx-blue-5dd745fc96-4bgqk      1/1     Running     0          93m
nginx-blue-5dd745fc96-rzzzj      1/1     Running     0          93m
nginx-green-7f7db6cb9f-pd75f     1/1     Running     0          92m
nginx-green-7f7db6cb9f-vqlrv     1/1     Running     0          92m

abhishesrivasta6@USGURABHISHESR6 MINGW64 ~/Desktop/Kubernetes-Advance-Training (
feature)
$ kubectl exec configmappod -- env
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
HOSTNAME=configmappod
database=mysql
databaseurl=mysql://db.training.com:3306
keys=name.training=kubernetes
program.training=configmap
trainer.training=sathish

KUBERNETES_PORT_443_TCP=tcp://10.96.0.1:443
KUBERNETES_PORT_443_TCP_PROTO=tcp
KUBERNETES_PORT_443_TCP_PORT=443
KUBERNETES_PORT_443_TCP_ADDR=10.96.0.1
KUBERNETES_SERVICE_HOST=10.96.0.1
KUBERNETES_SERVICE_PORT=443
KUBERNETES_SERVICE_PORT_HTTPS=443
KUBERNETES_PORT=tcp://10.96.0.1:443
NGINX_VERSION=1.23.2
NJS_VERSION=0.7.7
PKG_RELEASE=1~bullseye
HOME=/root
```



```
[root@ip-172-31-60-3 k8s]# vi configmapredis.yaml
[root@ip-172-31-60-3 k8s]# kubectl apply -f configmapredis.yaml
pod/redis created
[root@ip-172-31-60-3 k8s]# kubectl get po
NAME           READY   STATUS    RESTARTS   AGE
configmappod   1/1     Running   0          14m
redis          1/1     Running   0          13s
[root@ip-172-31-60-3 k8s]# kubectl exec -it redis -- redis-cli
127.0.0.1:6379> CONFIG GET maxmemory
1) "maxmemory"
2) "2097152"
127.0.0.1:6379> CONFIG GET maxmemory-policy
1) "maxmemory-policy"
2) "allkeys-lru"
127.0.0.1:6379> exit
[root@ip-172-31-60-3 k8s]# kubectl exec redis -- cat /redis-master/redis.conf
maxmemory 2mb
maxmemory-policy allkeys-lru
[root@ip-172-31-60-3 k8s]#
```

Microservices - Multi Container Deployment in a POD

Tuesday, November 22, 2022 12:40 PM

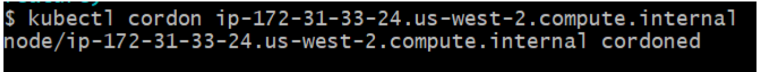
One pod generating data
One pod dispatching data in xml format

Multicontainer.yaml

```
apiVersion: v1
kind: Pod
metadata:
 name: multicontainer
spec:
 containers:
 - name: datadispatcher
   image: nginx
   volumeMounts:
   - name: html
     mountPath: /usr/share/nginx/html
 - name: datagenerator
   image: debian
   volumeMounts:
   - name: html
     mountPath: /html
   command: ["/bin/sh","-c"]
   args:
     - while true; do
         date >> /html/index.html;
         sleep 30;
       done
 volumes:
 - name: html
   emptyDir: {}
```

To see data on particular pod : $ kubectl exec multicontainer -c datagenerator --cat /html/index.html

If you don't want to deploy any new pod to particular node - cordon

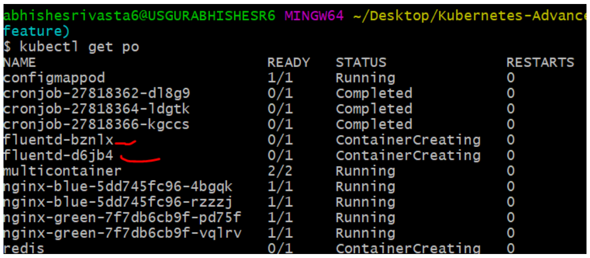


```
$ kubectl cordon ip-172-31-33-24.us-west-2.compute.internal
node/ip-172-31-33-24.us-west-2.compute.internal cordoned
```

Adding pod again to node 0r adding node again to cluster - uncordon

Creating Daemon Set
deamonset.yaml

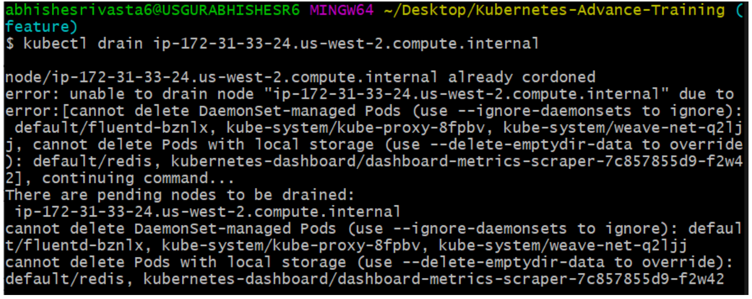
```
apiVersion: apps/v1
kind: DaemonSet
metadata:
 name: fluentd
 labels:
   app: fluentd
spec:
 selector:
   matchLabels:
     name: fluentd
 template:
   metadata:
     labels:
       name: fluentd
   spec:
     containers:
     - name: fluentd
       image: registry.hub.docker.com/monotek/fluentd-elasticsearch:27
       volumeMounts:
       - name: varlog
         mountPath: /var/log
       - name: varlibdockercontainers
         mountPath: /var/lib/docker/containers
         readOnly: true
     volumes:
     - name: varlog
       hostPath:
         path: /var/log
     - name: varlibdockercontainers
       hostPath:
         path: /var/lib/docker/containers
```



```
abhishesrivasta6@USGURABHISHESR6 MINGW64 ~/Desktop/Kubernetes-Advance
feature)
$ kubectl get po
NAME                              READY   STATUS              RESTARTS
configmappod                      1/1     Running             0
cronjob-27818362-dl8g9            0/1     Completed           0
cronjob-27818364-ldgtk            0/1     Completed           0
cronjob-27818366-kgccs            0/1     Completed           0
fluentd-bznlx                     0/1     ContainerCreating   0
fluentd-d6jb4                     0/1     ContainerCreating   0
multicontainer                    2/2     Running             0
nginx-blue-5dd745fc96-4bgqk       1/1     Running             0
nginx-blue-5dd745fc96-rzzzj       1/1     Running             0
nginx-green-7f7db6cb9f-pd75f      1/1     Running             0
nginx-green-7f7db6cb9f-vqlrv      1/1     Running             0
redis                             0/1     ContainerCreating   0
```

Stateful Set

migrate pod to other available node - drain



```
abhishesrivasta6@USGURABHISHESR6 MINGW64 ~/Desktop/Kubernetes-Advance-Training (
feature)
$ kubectl drain ip-172-31-33-24.us-west-2.compute.internal

node/ip-172-31-33-24.us-west-2.compute.internal already cordoned
error: unable to drain node "ip-172-31-33-24.us-west-2.compute.internal" due to
error:[cannot delete DaemonSet-managed Pods (use --ignore-daemonsets to ignore):
 default/fluentd-bznlx, kube-system/kube-proxy-8fpbv, kube-system/weave-net-q2lj
j, cannot delete Pods with local storage (use --delete-emptydir-data to override
): default/redis, kubernetes-dashboard/dashboard-metrics-scraper-7c857855d9-f2w4
2], continuing command...
There are pending nodes to be drained:
 ip-172-31-33-24.us-west-2.compute.internal
cannot delete DaemonSet-managed Pods (use --ignore-daemonsets to ignore): defaul
t/fluentd-bznlx, kube-system/kube-proxy-8fpbv, kube-system/weave-net-q2ljj
cannot delete Pods with local storage (use --delete-emptydir-data to override):
default/redis, kubernetes-dashboard/dashboard-metrics-scraper-7c857855d9-f2w42
```