Assignment 1

Name – Sati, Ankit Date – 2/13/2022

Section – 001

SID – as14128

Total in points (Maximum 100 points)–

Professors Comments –

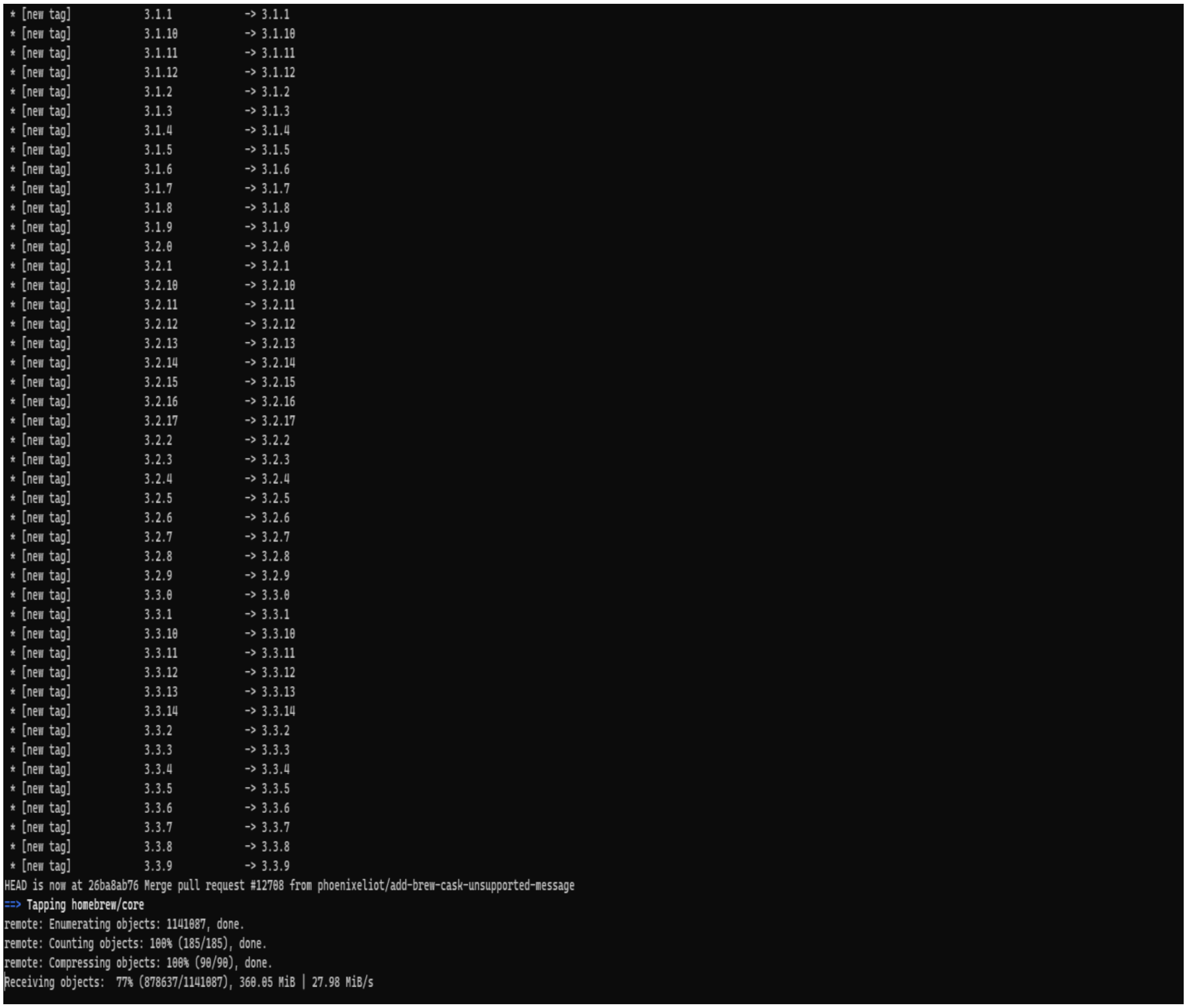
Affirmation of Independent Effort – Ankit Sati

STEP 1 – Setting up all the platforms on a windows machine.

1. Setting up Homebrew.

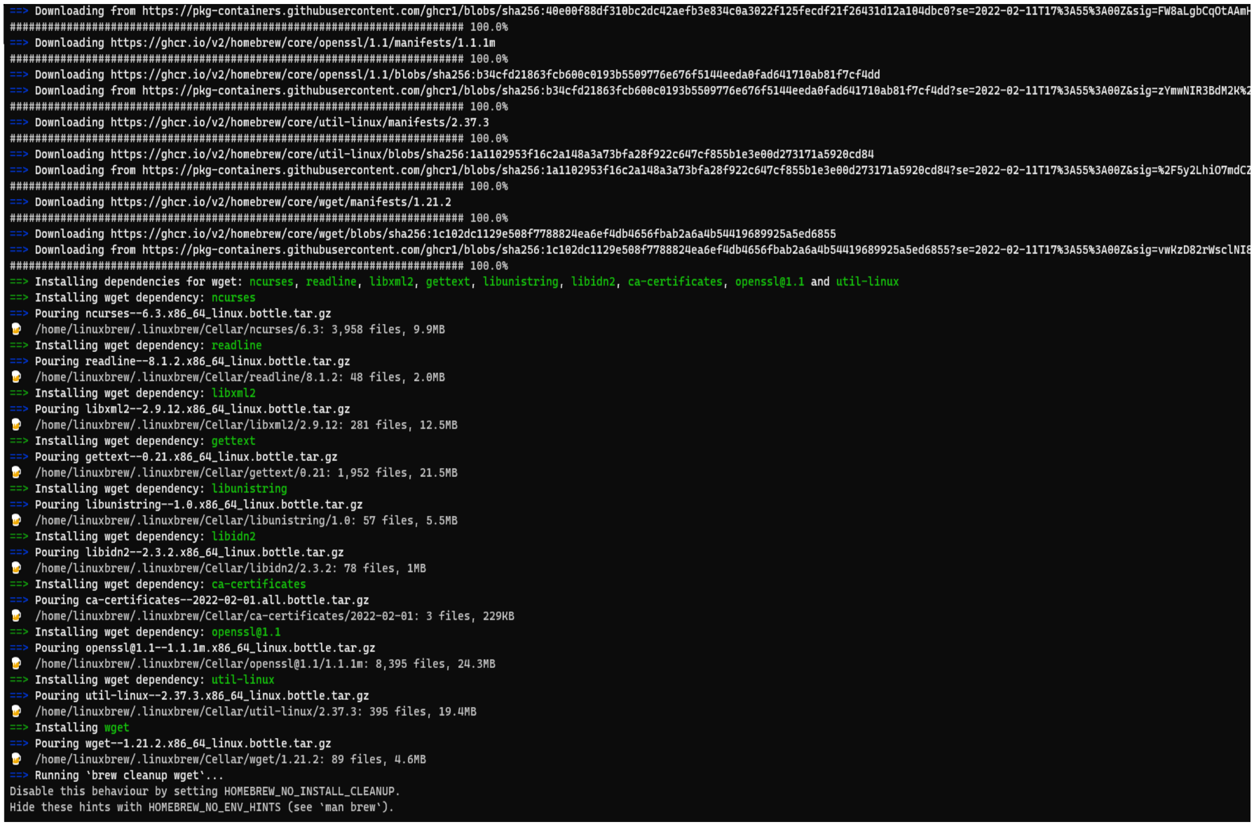
- This is a software that is needed to find and install any missing file during the entire process of setting the cloud and deploying the resources.
- Irrespective of the application being installed on the windows device, homebrew came in handy on many steps of the process for me.
  - ➔ Used while installing the Tanzu instance.
  - ➔ Later used in the cli of the AWS deployment.
  - ➔ While the SUSE(64) was deployed, this was helpful in installing the bash.
  - ➔ Finally helped me in the shell where many commands failed in the beginning and **Homebrew** was able to download the missing repositories.

Confirmation of installation.



```
 * [new tag]          3.1.1          -> 3.1.1
 * [new tag]          3.1.10         -> 3.1.10
 * [new tag]          3.1.11         -> 3.1.11
 * [new tag]          3.1.12         -> 3.1.12
 * [new tag]          3.1.2          -> 3.1.2
 * [new tag]          3.1.3          -> 3.1.3
 * [new tag]          3.1.4          -> 3.1.4
 * [new tag]          3.1.5          -> 3.1.5
 * [new tag]          3.1.6          -> 3.1.6
 * [new tag]          3.1.7          -> 3.1.7
 * [new tag]          3.1.8          -> 3.1.8
 * [new tag]          3.1.9          -> 3.1.9
 * [new tag]          3.2.0          -> 3.2.0
 * [new tag]          3.2.1          -> 3.2.1
 * [new tag]          3.2.10         -> 3.2.10
 * [new tag]          3.2.11         -> 3.2.11
 * [new tag]          3.2.12         -> 3.2.12
 * [new tag]          3.2.13         -> 3.2.13
 * [new tag]          3.2.14         -> 3.2.14
 * [new tag]          3.2.15         -> 3.2.15
 * [new tag]          3.2.16         -> 3.2.16
 * [new tag]          3.2.17         -> 3.2.17
 * [new tag]          3.2.2          -> 3.2.2
 * [new tag]          3.2.3          -> 3.2.3
 * [new tag]          3.2.4          -> 3.2.4
 * [new tag]          3.2.5          -> 3.2.5
 * [new tag]          3.2.6          -> 3.2.6
 * [new tag]          3.2.7          -> 3.2.7
 * [new tag]          3.2.8          -> 3.2.8
 * [new tag]          3.2.9          -> 3.2.9
 * [new tag]          3.3.0          -> 3.3.0
 * [new tag]          3.3.1          -> 3.3.1
 * [new tag]          3.3.10         -> 3.3.10
 * [new tag]          3.3.11         -> 3.3.11
 * [new tag]          3.3.12         -> 3.3.12
 * [new tag]          3.3.13         -> 3.3.13
 * [new tag]          3.3.14         -> 3.3.14
 * [new tag]          3.3.2          -> 3.3.2
 * [new tag]          3.3.3          -> 3.3.3
 * [new tag]          3.3.4          -> 3.3.4
 * [new tag]          3.3.5          -> 3.3.5
 * [new tag]          3.3.6          -> 3.3.6
 * [new tag]          3.3.7          -> 3.3.7
 * [new tag]          3.3.8          -> 3.3.8
 * [new tag]          3.3.9          -> 3.3.9
HEAD is now at 26ba8ab76 Merge pull request #12708 from phoenixeliot/add-brew-cask-unsupported-message
==> Tapping homebrew/core
remote: Enumerating objects: 1141087, done.
remote: Counting objects: 100% (185/185), done.
remote: Compressing objects: 100% (90/90), done.
Receiving objects:  77% (878637/1141087), 360.05 MiB | 27.98 MiB/s
```

HomeBrew Dependencies

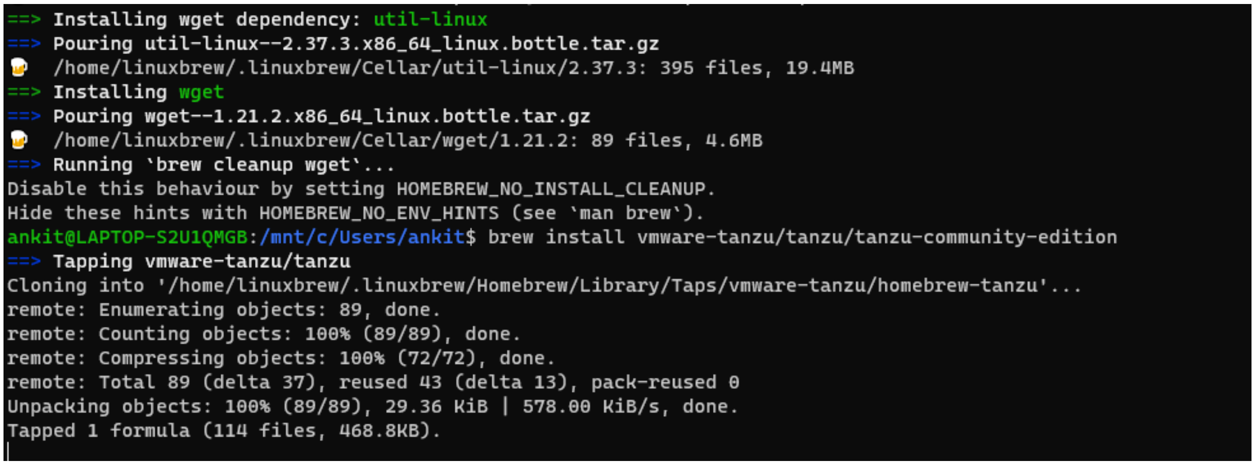


```
==> Downloading from https://pkg-containers.githubusercontent.com/ghcr1/blobs/sha256:40e00f88df310bc2dc42aefb3e834c0a3022f125fecdf21f26431d12a104dbc0?se=2022-02-11T17%3A55%3A00Z&sig=FW8aLgbCqOtAAmH
######################################################################## 100.0%
==> Downloading https://ghcr.io/v2/homebrew/core/openssl/1.1/manifests/1.1.1m
######################################################################## 100.0%
==> Downloading https://ghcr.io/v2/homebrew/core/openssl/1.1/blobs/sha256:b34cfd21863fcb600c0193b5509776e676f5144eeda0fad641710ab81f7cf4dd
==> Downloading from https://pkg-containers.githubusercontent.com/ghcr1/blobs/sha256:b34cfd21863fcb600c0193b5509776e676f5144eeda0fad641710ab81f7cf4dd?se=2022-02-11T17%3A55%3A00Z&sig=zYmwNIR3BdM2K%2
######################################################################## 100.0%
==> Downloading https://ghcr.io/v2/homebrew/core/util-linux/manifests/2.37.3
######################################################################## 100.0%
==> Downloading https://ghcr.io/v2/homebrew/core/util-linux/blobs/sha256:1a1102953f16c2a148a3a73bfa28f922c647cf855b1e3e00d273171a5920cd84
==> Downloading from https://pkg-containers.githubusercontent.com/ghcr1/blobs/sha256:1a1102953f16c2a148a3a73bfa28f922c647cf855b1e3e00d273171a5920cd84?se=2022-02-11T17%3A55%3A00Z&sig=%2F5y2LhiO7mdCZ
######################################################################## 100.0%
==> Downloading https://ghcr.io/v2/homebrew/core/wget/manifests/1.21.2
######################################################################## 100.0%
==> Downloading https://ghcr.io/v2/homebrew/core/wget/blobs/sha256:1c102dc1129e508f7788824ea6ef4db4656fbab2a6a4b54419689925a5ed6855
==> Downloading from https://pkg-containers.githubusercontent.com/ghcr1/blobs/sha256:1c102dc1129e508f7788824ea6ef4db4656fbab2a6a4b54419689925a5ed6855?se=2022-02-11T17%3A55%3A00Z&sig=vwKzD82rWsclNI8
######################################################################## 100.0%
==> Installing dependencies for wget: ncurses, readline, libxml2, gettext, libunistring, libidn2, ca-certificates, openssl@1.1 and util-linux
==> Installing wget dependency: ncurses
==> Pouring ncurses--6.3.x86_64_linux.bottle.tar.gz
🍺  /home/linuxbrew/.linuxbrew/Cellar/ncurses/6.3: 3,958 files, 9.9MB
==> Installing wget dependency: readline
==> Pouring readline--8.1.2.x86_64_linux.bottle.tar.gz
🍺  /home/linuxbrew/.linuxbrew/Cellar/readline/8.1.2: 48 files, 2.0MB
==> Installing wget dependency: libxml2
==> Pouring libxml2--2.9.12.x86_64_linux.bottle.tar.gz
🍺  /home/linuxbrew/.linuxbrew/Cellar/libxml2/2.9.12: 281 files, 12.5MB
==> Installing wget dependency: gettext
==> Pouring gettext--0.21.x86_64_linux.bottle.tar.gz
🍺  /home/linuxbrew/.linuxbrew/Cellar/gettext/0.21: 1,952 files, 21.5MB
==> Installing wget dependency: libunistring
==> Pouring libunistring--1.0.x86_64_linux.bottle.tar.gz
🍺  /home/linuxbrew/.linuxbrew/Cellar/libunistring/1.0: 57 files, 5.5MB
==> Installing wget dependency: libidn2
==> Pouring libidn2--2.3.2.x86_64_linux.bottle.tar.gz
🍺  /home/linuxbrew/.linuxbrew/Cellar/libidn2/2.3.2: 78 files, 1MB
==> Installing wget dependency: ca-certificates
==> Pouring ca-certificates--2022-02-01.all.bottle.tar.gz
🍺  /home/linuxbrew/.linuxbrew/Cellar/ca-certificates/2022-02-01: 3 files, 229KB
==> Installing wget dependency: openssl@1.1
==> Pouring openssl@1.1--1.1.1m.x86_64_linux.bottle.tar.gz
🍺  /home/linuxbrew/.linuxbrew/Cellar/openssl@1.1/1.1.1m: 8,395 files, 24.3MB
==> Installing wget dependency: util-linux
==> Pouring util-linux--2.37.3.x86_64_linux.bottle.tar.gz
🍺  /home/linuxbrew/.linuxbrew/Cellar/util-linux/2.37.3: 395 files, 19.4MB
==> Installing wget
==> Pouring wget--1.21.2.x86_64_linux.bottle.tar.gz
🍺  /home/linuxbrew/.linuxbrew/Cellar/wget/1.21.2: 89 files, 4.6MB
==> Running `brew cleanup wget`...
Disable this behaviour by setting HOMEBREW_NO_INSTALL_CLEANUP.
Hide these hints with HOMEBREW_NO_ENV_HINTS (see `man brew`).
```

2. Setting up Tanzu

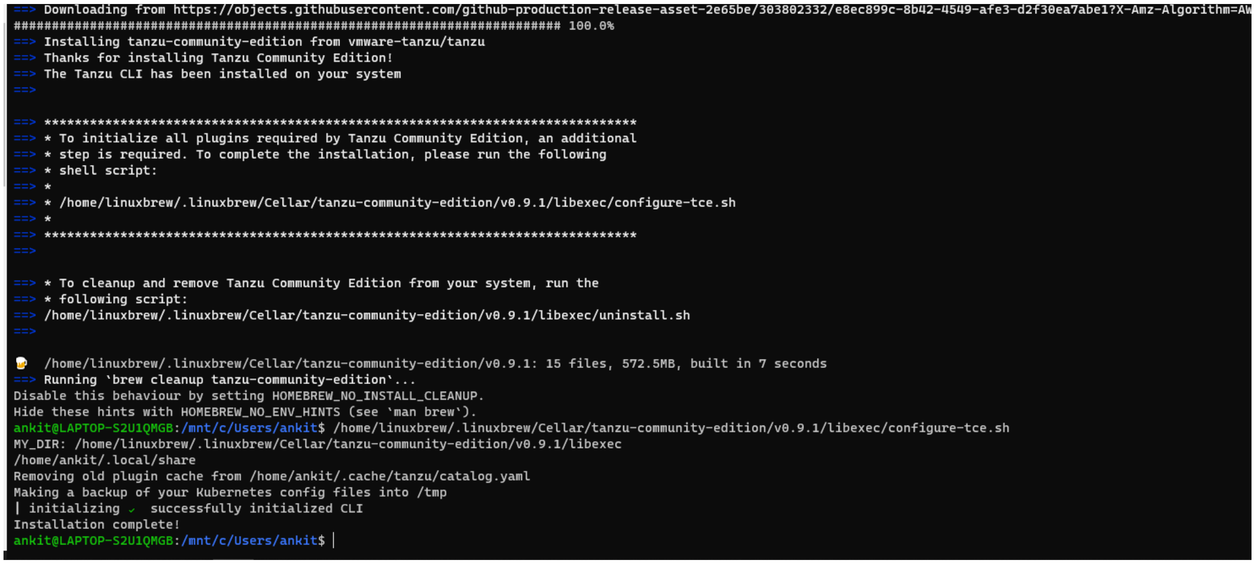
- Tanzu is a suite of products that helps users run and manage multiple Kubernetes (K8S) clusters across public and private “clouds”.
- While introducing Kubernetes as a first class VMware product, it still keeps strong ties to the VMware virtualization portfolio. It consists of: Spring Runtime – app framework.
- This came in handy in many places post the installation for managing the resources that I made available to myself on the cloud.
  - ➔ Post installation of the AWS repositories for managing instances.
  - ➔ Secondly while creating the VM instances on the cloud space, we could easily track and see the activities of the resources.
  - ➔ Aligning resources to the SUSE instance.

Confirmation of Tanzu installation



```
==> Installing wget dependency: util-linux
==> Pouring util-linux--2.37.3.x86_64_linux.bottle.tar.gz
🍺  /home/linuxbrew/.linuxbrew/Cellar/util-linux/2.37.3: 395 files, 19.4MB
==> Installing wget
==> Pouring wget--1.21.2.x86_64_linux.bottle.tar.gz
🍺  /home/linuxbrew/.linuxbrew/Cellar/wget/1.21.2: 89 files, 4.6MB
==> Running `brew cleanup wget`...
Disable this behaviour by setting HOMEBREW_NO_INSTALL_CLEANUP.
Hide these hints with HOMEBREW_NO_ENV_HINTS (see `man brew`).
ankit@LAPTOP-S2U1QMGB:/mnt/c/Users/ankit$ brew install vmware-tanzu/tanzu/tanzu-community-edition
==> Tapping vmware-tanzu/tanzu
Cloning into '/home/linuxbrew/.linuxbrew/Homebrew/Library/Taps/vmware-tanzu/homebrew-tanzu'...
remote: Enumerating objects: 89, done.
remote: Counting objects: 100% (89/89), done.
remote: Compressing objects: 100% (72/72), done.
remote: Total 89 (delta 37), reused 43 (delta 13), pack-reused 0
Unpacking objects: 100% (89/89), 29.36 KiB | 578.00 KiB/s, done.
Tapped 1 formula (114 files, 468.8KB).
```

Required resources setup and completed.



```
==> Downloading from https://objects.githubusercontent.com/github-production-release-asset-2e65be/303802332/e8ec899c-8b42-4549-afe3-d2f30ea7abe1?X-Amz-Algorithm=AW
######################################################################## 100.0%
==> Installing tanzu-community-edition from vmware-tanzu/tanzu
==> Thanks for installing Tanzu Community Edition!
==> The Tanzu CLI has been installed on your system
==>

==> ******************************************************************************
==> * To initialize all plugins required by Tanzu Community Edition, an additional
==> * step is required. To complete the installation, please run the following
==> * shell script:
==> *
==> * /home/linuxbrew/.linuxbrew/Cellar/tanzu-community-edition/v0.9.1/libexec/configure-tce.sh
==> *
==> ******************************************************************************
==>

==> * To cleanup and remove Tanzu Community Edition from your system, run the
==> * following script:
==> /home/linuxbrew/.linuxbrew/Cellar/tanzu-community-edition/v0.9.1/libexec/uninstall.sh
==>

🍺  /home/linuxbrew/.linuxbrew/Cellar/tanzu-community-edition/v0.9.1: 15 files, 572.5MB, built in 7 seconds
==> Running `brew cleanup tanzu-community-edition`...
Disable this behaviour by setting HOMEBREW_NO_INSTALL_CLEANUP.
Hide these hints with HOMEBREW_NO_ENV_HINTS (see `man brew`).
ankit@LAPTOP-S2U1QMGB:/mnt/c/Users/ankit$ /home/linuxbrew/.linuxbrew/Cellar/tanzu-community-edition/v0.9.1/libexec/configure-tce.sh
MY_DIR: /home/linuxbrew/.linuxbrew/Cellar/tanzu-community-edition/v0.9.1/libexec
/home/ankit/.local/share
Removing old plugin cache from /home/ankit/.cache/tanzu/catalog.yaml
Making a backup of your Kubernetes config files into /tmp
| initializing ✓  successfully initialized CLI
Installation complete!
ankit@LAPTOP-S2U1QMGB:/mnt/c/Users/ankit$ |
```

Finally Lets get started with the Assignment.

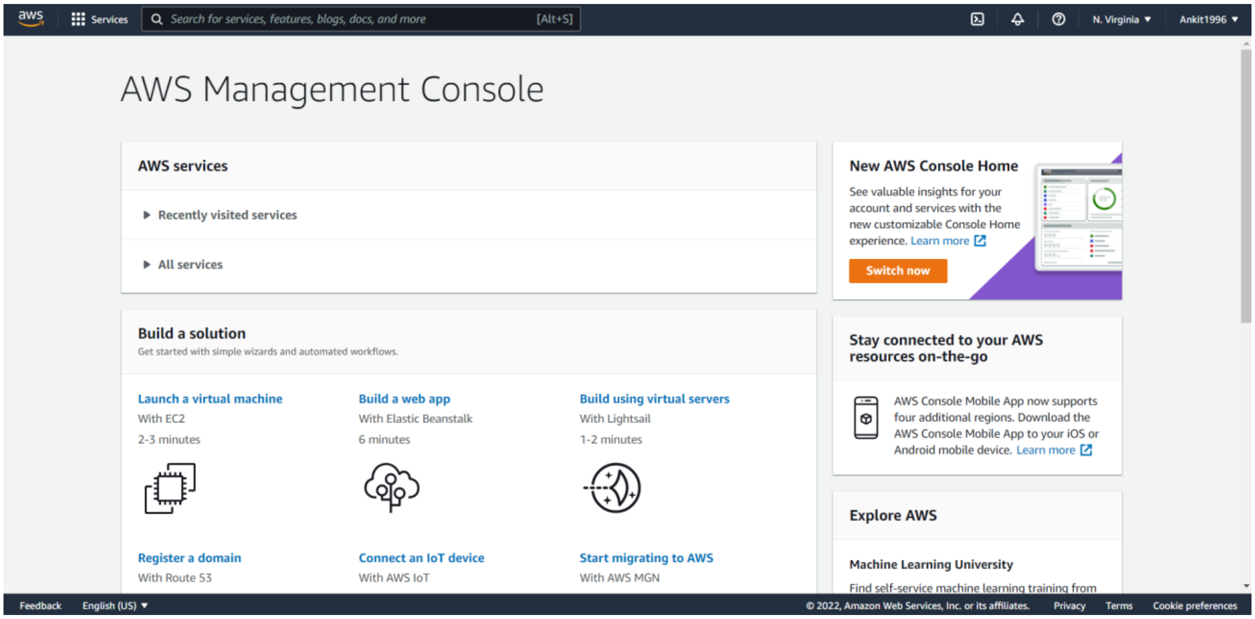
1. Setting up the AWS account with the required instances.

Required instances.

➔ EC2(**Compute server**) – This is basically a **regular server** instance that is used to deploy and the required resources over the VM as per the choices made by the users.
- This is used to deploy the VM.
- Manage resources over those VM's.
- Finally to migrate services and monitor volumes.

➔ S3(**Storage utility**) – This is a basic protocol that acts like a storage bucket.
- The prime feature of this protocol is to deal with the data as per service request.
- We need this to store the data in the **data buckets** which are later used to store and move the data across volumes created.

Initial account setup.



AWS Management Console

AWS services

▸ Recently visited services

▸ All services

New AWS Console Home

See valuable insights for your account and services with the new customizable Console Home experience. Learn more

Build a solution

Get started with simple wizards and automated workflows.

Launch a virtual machine
With EC2
2-3 minutes

Build a web app
With Elastic Beanstalk
6 minutes

Build using virtual servers
With Lightsail
1-2 minutes

Register a domain
With Route 53

Connect an IoT device
With AWS IoT

Start migrating to AWS
With AWS MGN

Stay connected to your AWS resources on-the-go

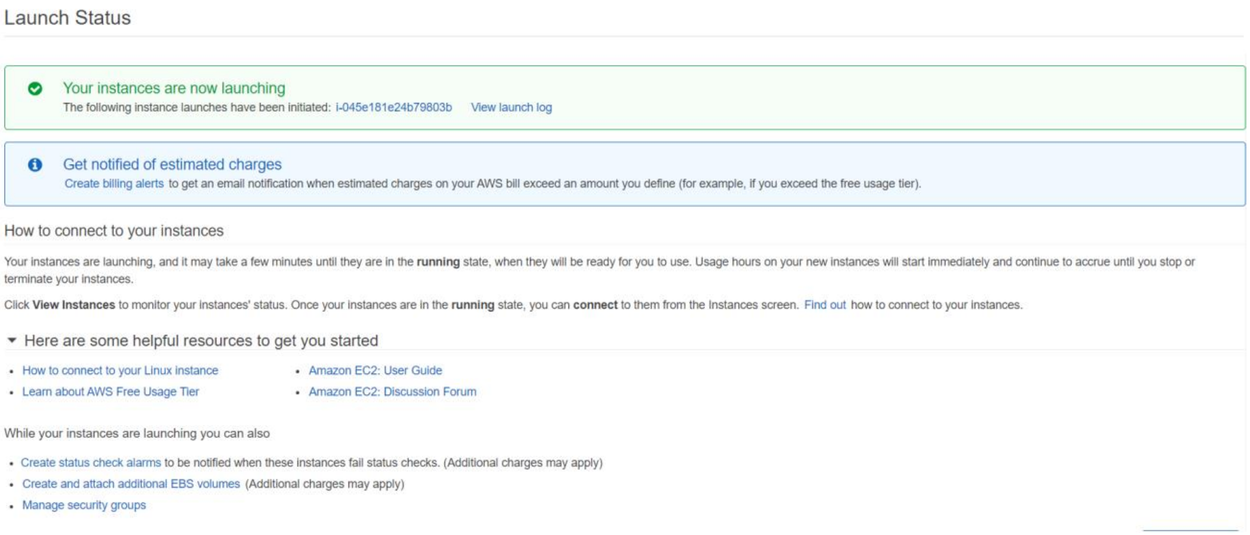
AWS Console Mobile App now supports four additional regions. Download the AWS Console Mobile App to your iOS or Android mobile device. Learn more

Explore AWS

Machine Learning University

Find self-service machine learning training from

© 2022, Amazon Web Services, Inc. or its affiliates.



Launch Status

Your instances are now launching
The following instance launches have been initiated: i-045e181e24b79803b View launch log

Get notified of estimated charges
Create billing alerts to get an email notification when estimated charges on your AWS bill exceed an amount you define (for example, if you exceed the free usage tier).

How to connect to your instances

Your instances are launching, and it may take a few minutes until they are in the **running** state, when they will be ready for you to use. Usage hours on your new instances will start immediately and continue to accrue until you stop or terminate your instances.

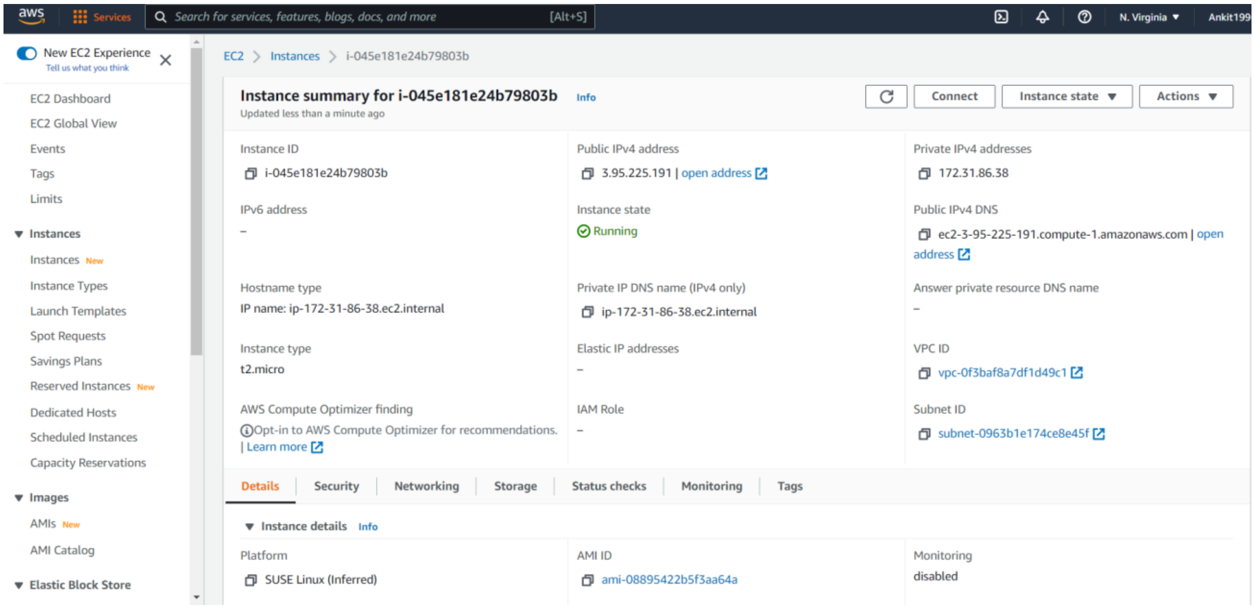
Click **View Instances** to monitor your instances' status. Once your instances are in the **running** state, you can **connect** to them from the Instances screen. Find out how to connect to your instances.

▾ Here are some helpful resources to get you started

- How to connect to your Linux instance
- Learn about AWS Free Usage Tier
- Amazon EC2: User Guide
- Amazon EC2: Discussion Forum

While your instances are launching you can also

- Create status check alarms to be notified when these instances fail status checks. (Additional charges may apply)
- Create and attach additional EBS volumes (Additional charges may apply)
- Manage security groups



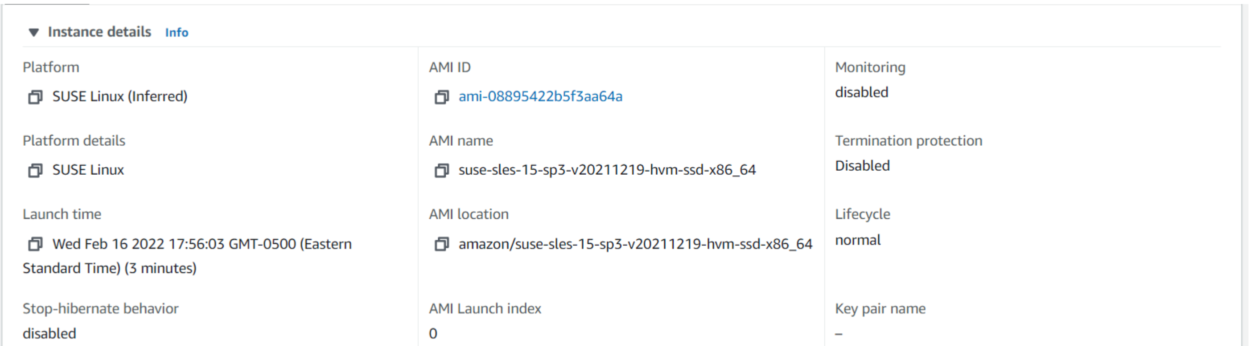
Instance summary for i-045e181e24b79803b Info
Updated less than a minute ago

Instance ID i-045e181e24b79803b	Public IPv4 address 3.95.225.191 | open address	Private IPv4 addresses 172.31.86.38
IPv6 address –	Instance state ✓ Running	Public IPv4 DNS ec2-3-95-225-191.compute-1.amazonaws.com | open address
Hostname type IP name: ip-172-31-86-38.ec2.internal	Private IP DNS name (IPv4 only) ip-172-31-86-38.ec2.internal	Answer private resource DNS name –
Instance type t2.micro	Elastic IP addresses –	VPC ID vpc-0f3baf8a7df1d49c1
AWS Compute Optimizer finding Opt-in to AWS Compute Optimizer for recommendations. | Learn more	IAM Role –	Subnet ID subnet-0963b1e174ce8e45f

▾ Instance details Info

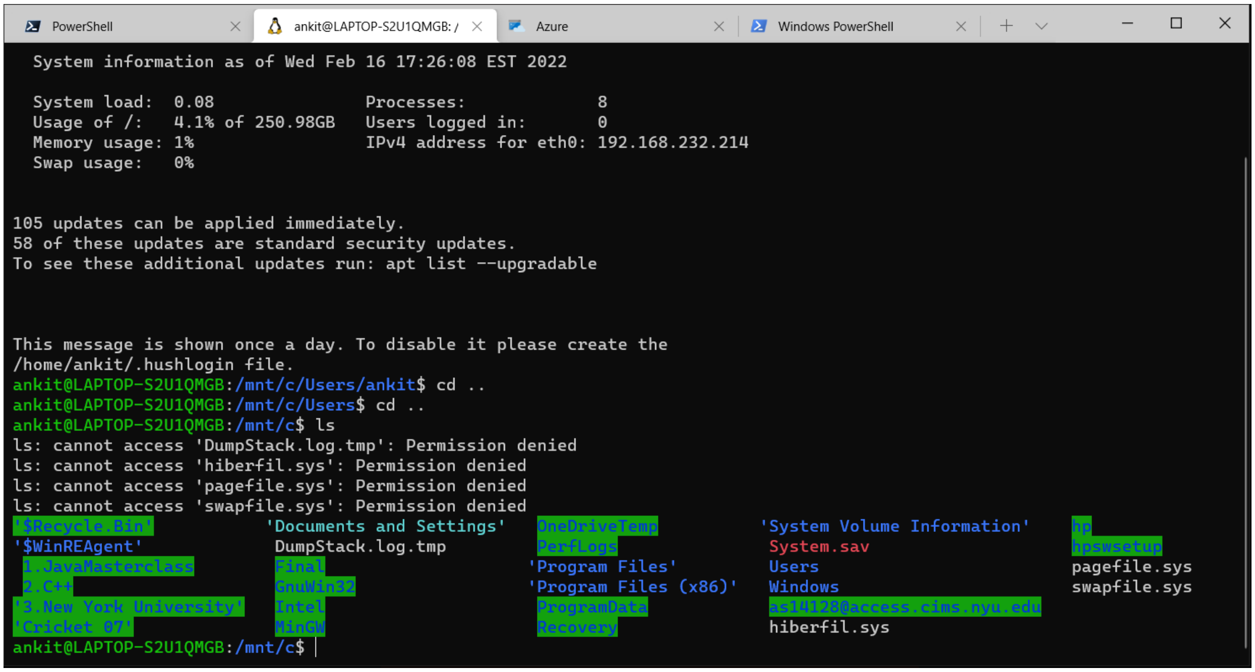
Platform SUSE Linux (Inferred)	AMI ID ami-08895422b5f3aa64a	Monitoring disabled

2. Setting up the SUSE 64 bit and Kubectls Instance to manage services.



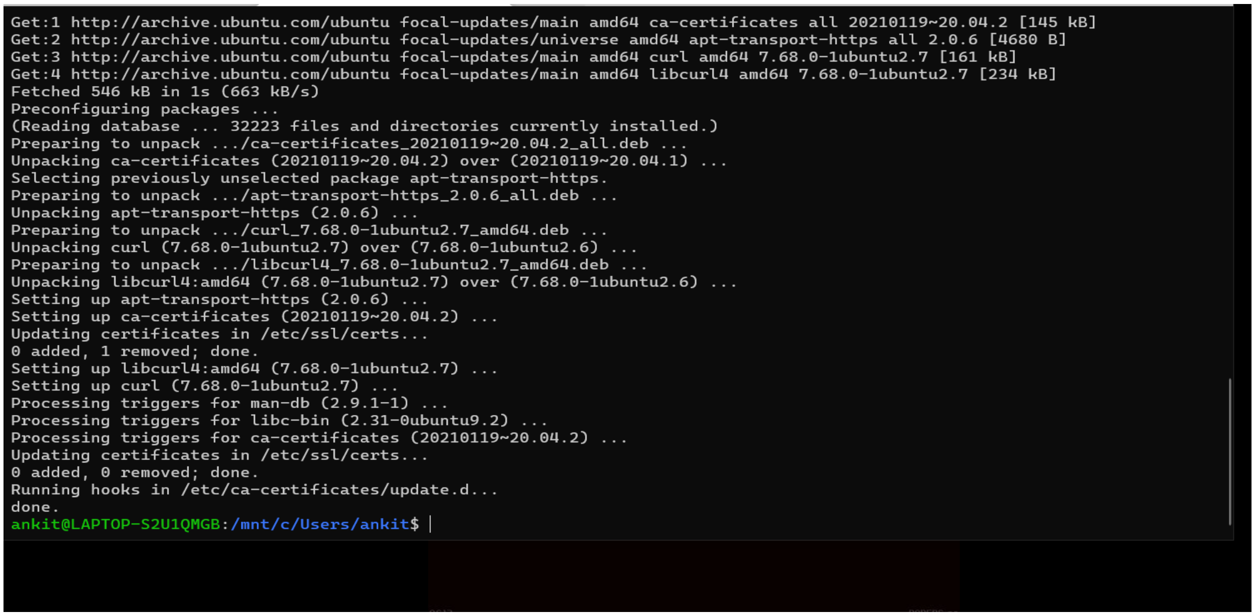
Platform SUSE Linux (Inferred)	AMI ID ami-08895422b5f3aa64a	Monitoring disabled
Platform details SUSE Linux	AMI name suse-sles-15-sp3-v20211219-hvm-ssd-x86_64	Termination protection Disabled
Launch time Wed Feb 16 2022 17:56:03 GMT-0500 (Eastern Standard Time) (3 minutes)	AMI location amazon/suse-sles-15-sp3-v20211219-hvm-ssd-x86_64	Lifecycle normal
Stop-hibernate behavior disabled	AMI Launch index 0	Key pair name –

- This is just another instance to manage the services of all the cloud applications.
    - ➔ Firstly We need to download a CLI client to manage all the different shell services like Azure, AWS SUSE etc. I will use the powerShell application as attached below.



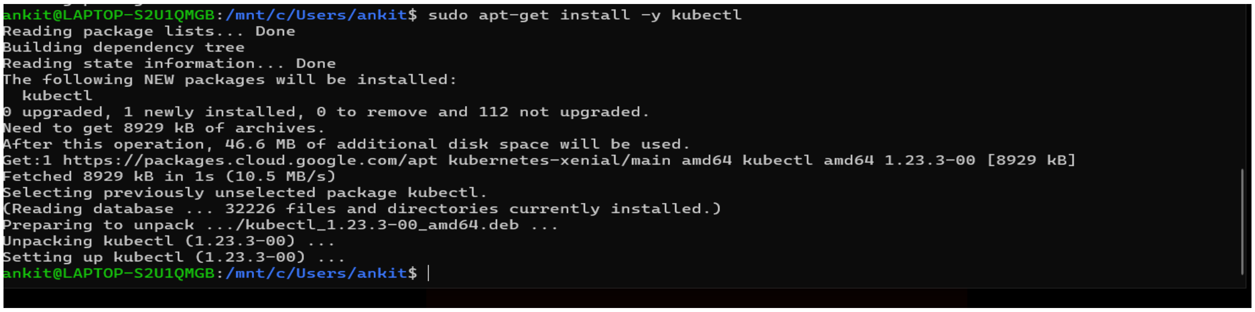
```
  System information as of Wed Feb 16 17:26:08 EST 2022

  System load:  0.08              Processes:            8
  Usage of /:   4.1% of 250.98GB  Users logged in:      0
  Memory usage: 1%                IPv4 address for eth0: 192.168.232.214
  Swap usage:   0%


105 updates can be applied immediately.
58 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable



This message is shown once a day. To disable it please create the
/home/ankit/.hushlogin file.
ankit@LAPTOP-S2U1QMGB:/mnt/c/Users/ankit$ cd ..
ankit@LAPTOP-S2U1QMGB:/mnt/c/Users$ cd ..
ankit@LAPTOP-S2U1QMGB:/mnt/c$ ls
ls: cannot access 'DumpStack.log.tmp': Permission denied
ls: cannot access 'hiberfil.sys': Permission denied
ls: cannot access 'pagefile.sys': Permission denied
ls: cannot access 'swapfile.sys': Permission denied
'$Recycle.Bin'             'Documents and Settings'   OneDriveTemp          'System Volume Information'   hp
'$WinREAgent'               DumpStack.log.tmp         PerfLogs               System.sav                   hpswsetup
 1.JavaMasterclass          Final                    'Program Files'         Users                        pagefile.sys
 2.C++                      GnuWin32                 'Program Files (x86)'   Windows                      swapfile.sys
'3.New York University'     Intel                     ProgramData            as14128@access.cims.nyu.edu
'Cricket 07'                MinGW                     Recovery               hiberfil.sys
ankit@LAPTOP-S2U1QMGB:/mnt/c$
```

- ➔ Secondly we need to set up the SUSE instance as required in this HW assignment.(Download SUSE rep)



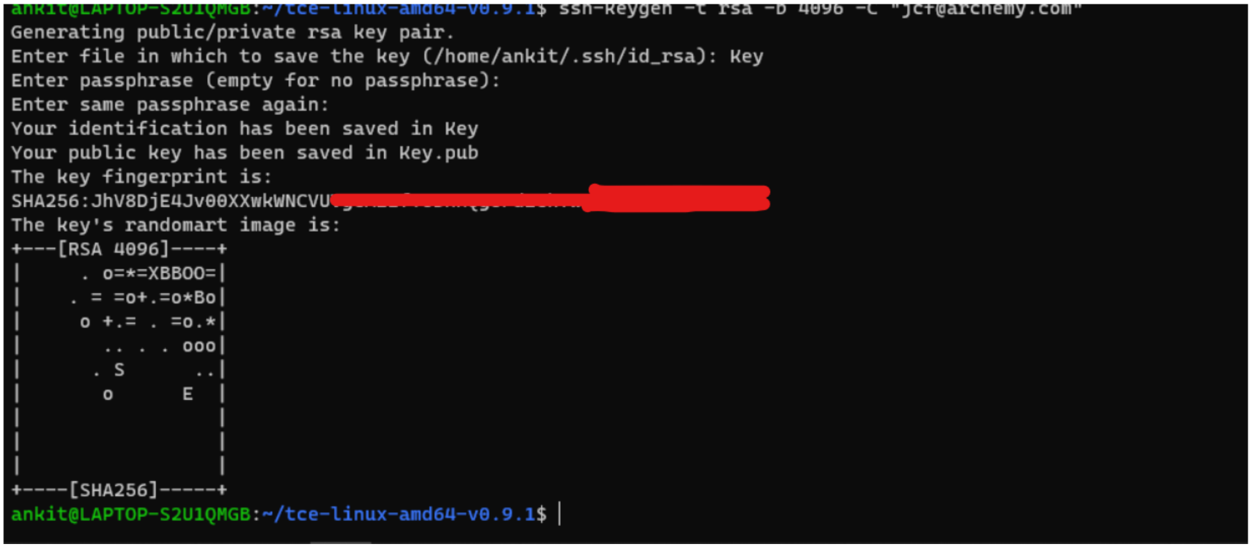
```
Get:1 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 ca-certificates all 20210119~20.04.2 [145 kB]
Get:2 http://archive.ubuntu.com/ubuntu focal-updates/universe amd64 apt-transport-https all 2.0.6 [4680 B]
Get:3 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 curl amd64 7.68.0-1ubuntu2.7 [161 kB]
Get:4 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 libcurl4 amd64 7.68.0-1ubuntu2.7 [234 kB]
Fetched 546 kB in 1s (663 kB/s)
Preconfiguring packages ...
(Reading database ... 32223 files and directories currently installed.)
Preparing to unpack .../ca-certificates_20210119~20.04.2_all.deb ...
Unpacking ca-certificates (20210119~20.04.2) over (20210119~20.04.1) ...
Selecting previously unselected package apt-transport-https.
Preparing to unpack .../apt-transport-https_2.0.6_all.deb ...
Unpacking apt-transport-https (2.0.6) ...
Preparing to unpack .../curl_7.68.0-1ubuntu2.7_amd64.deb ...
Unpacking curl (7.68.0-1ubuntu2.7) over (7.68.0-1ubuntu2.6) ...
Preparing to unpack .../libcurl4_7.68.0-1ubuntu2.7_amd64.deb ...
Unpacking libcurl4:amd64 (7.68.0-1ubuntu2.7) over (7.68.0-1ubuntu2.6) ...
Setting up apt-transport-https (2.0.6) ...
Setting up ca-certificates (20210119~20.04.2) ...
Updating certificates in /etc/ssl/certs...
0 added, 1 removed; done.
Setting up libcurl4:amd64 (7.68.0-1ubuntu2.7) ...
Setting up curl (7.68.0-1ubuntu2.7) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for libc-bin (2.31-0ubuntu9.2) ...
Processing triggers for ca-certificates (20210119~20.04.2) ...
Updating certificates in /etc/ssl/certs...
0 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d...
done.
ankit@LAPTOP-S2U1QMGB:/mnt/c/Users/ankit$ |
```

- ➔ the final Build.



```
ankit@LAPTOP-S2U1QMGB:/mnt/c/Users/ankit$ sudo apt-get install -y kubectl
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  kubectl
0 upgraded, 1 newly installed, 0 to remove and 112 not upgraded.
Need to get 8929 kB of archives.
After this operation, 46.6 MB of additional disk space will be used.
Get:1 https://packages.cloud.google.com/apt kubernetes-xenial/main amd64 kubectl amd64 1.23.3-00 [8929 kB]
Fetched 8929 kB in 1s (10.5 MB/s)
Selecting previously unselected package kubectl.
(Reading database ... 32226 files and directories currently installed.)
Preparing to unpack .../kubectl_1.23.3-00_amd64.deb ...
Unpacking kubectl (1.23.3-00) ...
Setting up kubectl (1.23.3-00) ...
ankit@LAPTOP-S2U1QMGB:/mnt/c/Users/ankit$ |
```

- ➔ Saving the required keys as mentioned in the assignment.
  Key.pub identification
  Type – RSA 4096
  Key – Screenshot attached (Blurred the actual key for privacy)

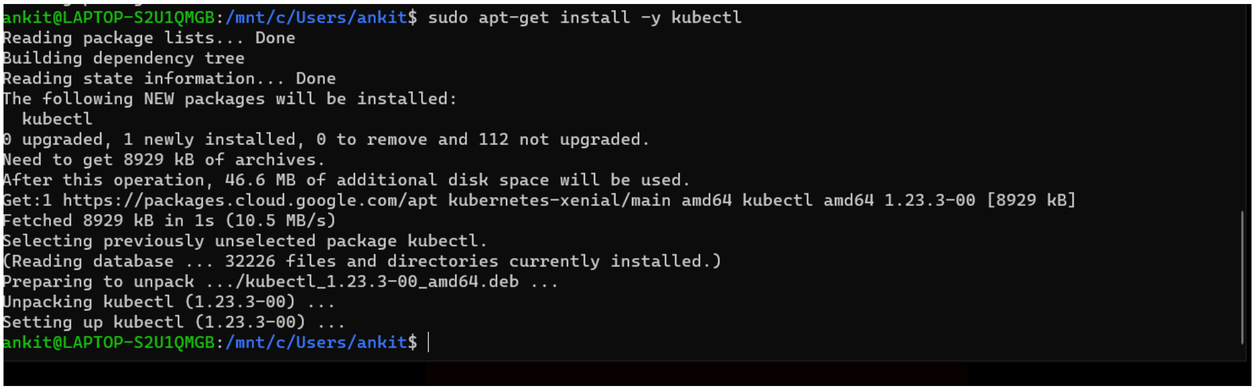


```
ankit@LAPTOP-S2U1QMGB:~/tce-linux-amd64-v0.9.1$ ssh-keygen -t rsa -b 4096 -C "jcf@archemy.com"
Generating public/private rsa key pair.
Enter file in which to save the key (/home/ankit/.ssh/id_rsa): Key
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in Key
Your public key has been saved in Key.pub
The key fingerprint is:
SHA256:JhV8DjE4Jv00XXwkWNCVU[redacted]
The key's randomart image is:
+---[RSA 4096]----+
|      . o=*=XBBOO=|
|     . = =o+.=o*Bo|
|      o +.= . =o.*|
|        .. . . ooo|
|       . S      ..|
|        o      E  |
|                  |
|                  |
|                  |
+----[SHA256]-----+
ankit@LAPTOP-S2U1QMGB:~/tce-linux-amd64-v0.9.1$ |
```

3. SSH into the instance that we have created.

- ➔ The final build needs to be created on the GUI first.
- ➔ Post that we need to setup the EC2 instance and the S3 storage bucket
- ➔ Finally, we can **ssh** into the created instance.

- **Screenshot of the final build**
- **Moreover I have installed the KUBEctl application to manage the containers created across the cloud platform.**



```
ankit@LAPTOP-S2U1QMGB:/mnt/c/Users/ankit$ sudo apt-get install -y kubectl
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  kubectl
0 upgraded, 1 newly installed, 0 to remove and 112 not upgraded.
Need to get 8929 kB of archives.
After this operation, 46.6 MB of additional disk space will be used.
Get:1 https://packages.cloud.google.com/apt kubernetes-xenial/main amd64 kubectl amd64 1.23.3-00 [8929 kB]
Fetched 8929 kB in 1s (10.5 MB/s)
Selecting previously unselected package kubectl.
(Reading database ... 32226 files and directories currently installed.)
Preparing to unpack .../kubectl_1.23.3-00_amd64.deb ...
Unpacking kubectl (1.23.3-00) ...
Setting up kubectl (1.23.3-00) ...
ankit@LAPTOP-S2U1QMGB:/mnt/c/Users/ankit$ |
```

4. Setting up the Tanzu application.

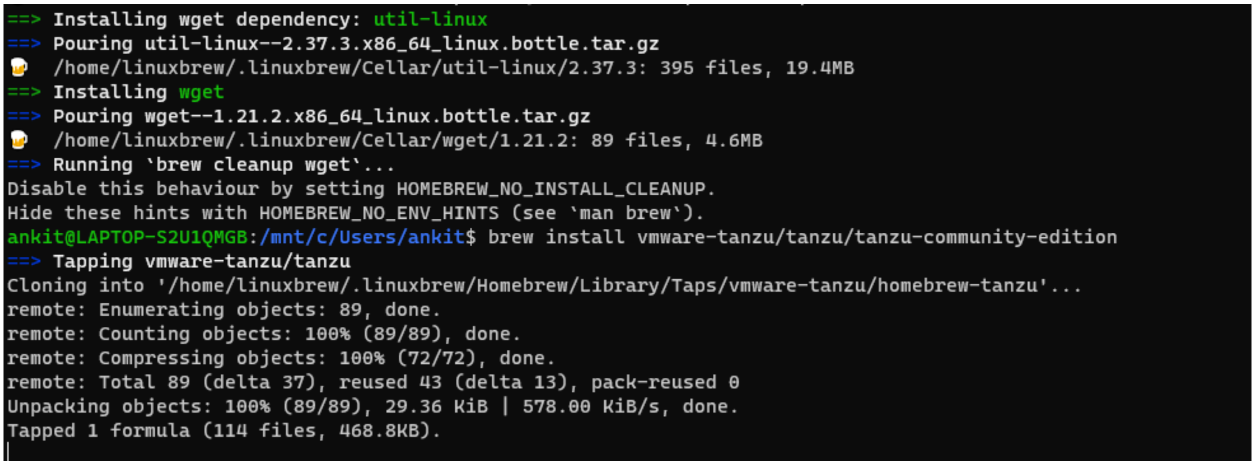
- ➔ I had discussed with the professor regarding cloud foundry. He wanted us to setup Tanzu instead of this service.

- ➔ Secondly Tanzu needs to be installed prior to the above so that it can access all the cores that the CPU has to offer.
- ➔ Above I have discussed the setting up of Tanzu, but I will paste the same procedure below.

Tanzu

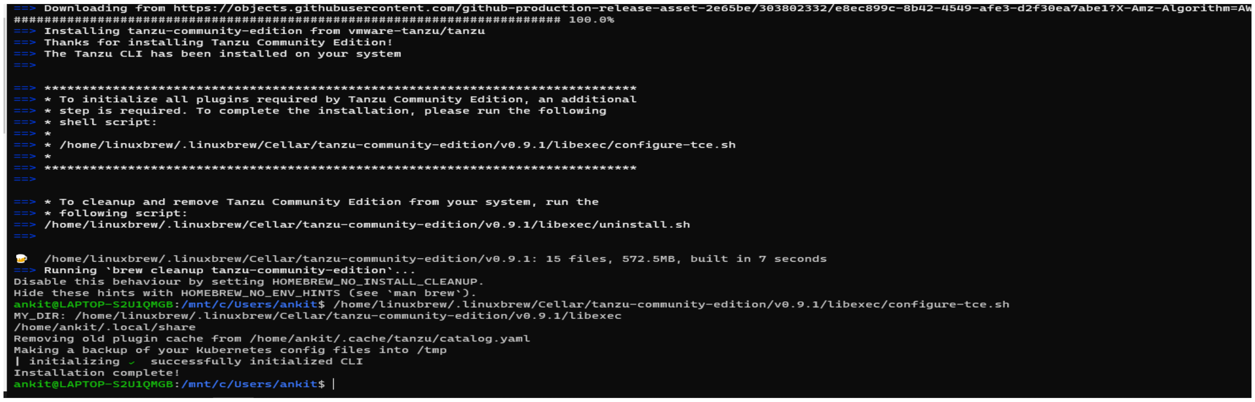
- Tanzu is a suite of products that helps users run and manage multiple Kubernetes (K8S) clusters across public and private "clouds".
- While introducing Kubernetes as a first class VMware product, it still keeps strong ties to the VMware virtualization portfolio. It consists of: Spring Runtime – app framework.
- This came in handy in many places post the installation for managing the resources that I made available to myself on the cloud.
  - ➔ Post installation of the AWS repositories for managing instances.
  - ➔ Secondly while creating the VM instances on the cloud space, we could easily track and see the activities of the resources.
  - ➔ Aligning resources to the SUSE instance.

Confirmation of Tanzu installation



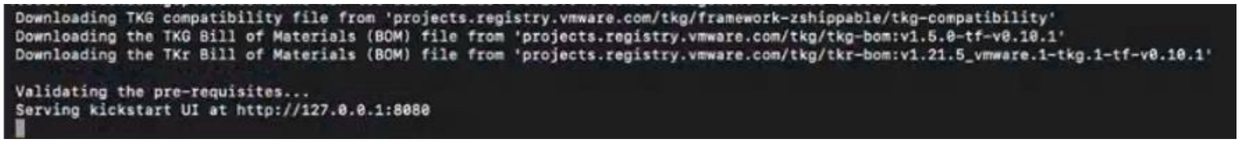
```
==> Installing wget dependency: util-linux
==> Pouring util-linux--2.37.3.x86_64_linux.bottle.tar.gz
🍺  /home/linuxbrew/.linuxbrew/Cellar/util-linux/2.37.3: 395 files, 19.4MB
==> Installing wget
==> Pouring wget--1.21.2.x86_64_linux.bottle.tar.gz
🍺  /home/linuxbrew/.linuxbrew/Cellar/wget/1.21.2: 89 files, 4.6MB
==> Running `brew cleanup wget`...
Disable this behaviour by setting HOMEBREW_NO_INSTALL_CLEANUP.
Hide these hints with HOMEBREW_NO_ENV_HINTS (see `man brew`).
ankit@LAPTOP-S2U1QMGB:/mnt/c/Users/ankit$ brew install vmware-tanzu/tanzu/tanzu-community-edition
==> Tapping vmware-tanzu/tanzu
Cloning into '/home/linuxbrew/.linuxbrew/Homebrew/Library/Taps/vmware-tanzu/homebrew-tanzu'...
remote: Enumerating objects: 89, done.
remote: Counting objects: 100% (89/89), done.
remote: Compressing objects: 100% (72/72), done.
remote: Total 89 (delta 37), reused 43 (delta 13), pack-reused 0
Unpacking objects: 100% (89/89), 29.36 KiB | 578.00 KiB/s, done.
Tapped 1 formula (114 files, 468.8KB).
```

Required resources setup and completed.

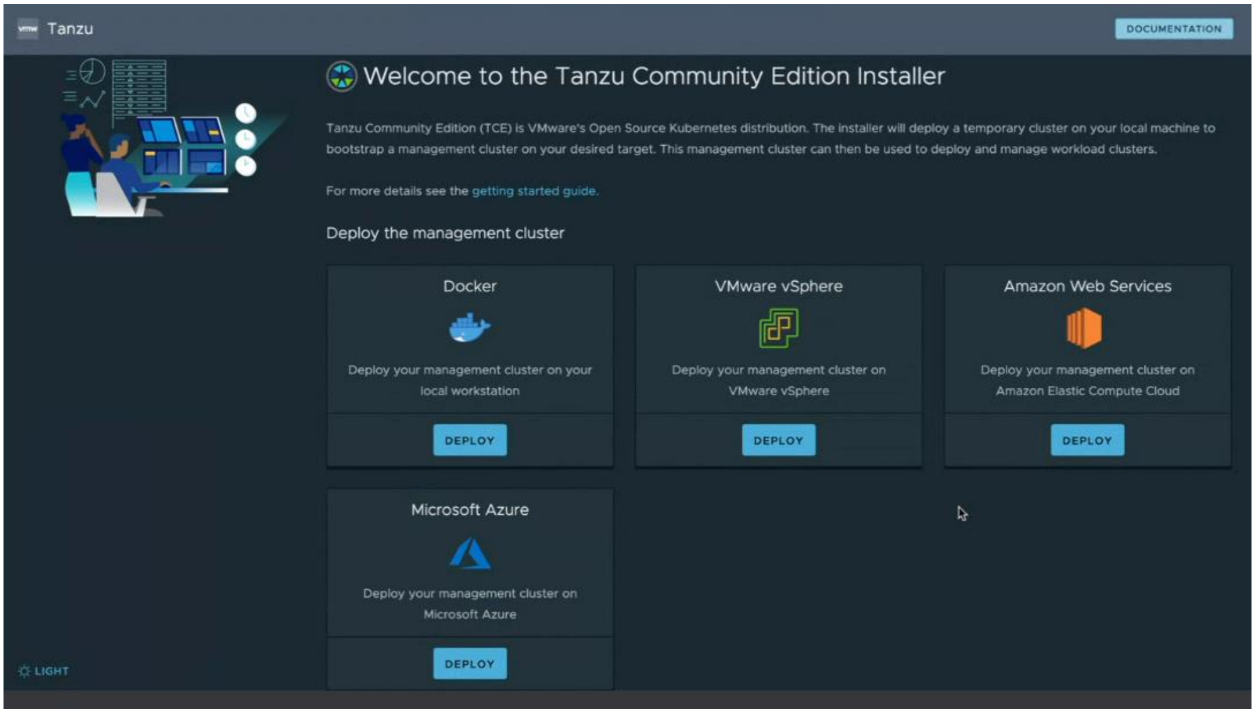


```
==> Downloading from https://objects.githubusercontent.com/github-production-release-asset-2e65be/303802332/e8ec899c-8b42-4549-afe3-d2f30ea7abe1?X-Amz-Algorithm=AW
######################################################################## 100.0%
==> Installing tanzu-community-edition from vmware-tanzu/tanzu
==> Thanks for installing Tanzu Community Edition!
==> The Tanzu CLI has been installed on your system
==>

==> ******************************************************************************
==> * To initialize all plugins required by Tanzu Community Edition, an additional
==> * step is required. To complete the installation, please run the following
==> * shell script:
==> *
==> * /home/linuxbrew/.linuxbrew/Cellar/tanzu-community-edition/v0.9.1/libexec/configure-tce.sh
==> *
==> ******************************************************************************
==>

==> * To cleanup and remove Tanzu Community Edition from your system, run the
==> * following script:
==> /home/linuxbrew/.linuxbrew/Cellar/tanzu-community-edition/v0.9.1/libexec/uninstall.sh
==>

🍺  /home/linuxbrew/.linuxbrew/Cellar/tanzu-community-edition/v0.9.1: 15 files, 572.5MB, built in 7 seconds
==> Running `brew cleanup tanzu-community-edition`...
Disable this behaviour by setting HOMEBREW_NO_INSTALL_CLEANUP.
Hide these hints with HOMEBREW_NO_ENV_HINTS (see `man brew`).
ankit@LAPTOP-S2U1QMGB:/mnt/c/Users/ankit$ /home/linuxbrew/.linuxbrew/Cellar/tanzu-community-edition/v0.9.1/libexec/configure-tce.sh
MY_DIR: /home/linuxbrew/.linuxbrew/Cellar/tanzu-community-edition/v0.9.1/libexec
/home/ankit/.local/share
Removing old plugin cache from /home/ankit/.cache/tanzu/catalog.yaml
Making a backup of your Kubernetes config files into /tmp
| initializing ✓  successfully initialized CLI
Installation complete!
ankit@LAPTOP-S2U1QMGB:/mnt/c/Users/ankit$ |
```

Final Deployment.



```
Downloading TKG compatibility file from 'projects.registry.vmware.com/tkg/framework-zshippable/tkg-compatibility'
Downloading the TKG Bill of Materials (BOM) file from 'projects.registry.vmware.com/tkg/tkg-bom:v1.5.0-tf-v0.10.1'
Downloading the TKr Bill of Materials (BOM) file from 'projects.registry.vmware.com/tkg/tkr-bom:v1.21.5_vmware.1-tkg.1-tf-v0.10.1'

Validating the pre-requisites...
Serving kickstart UI at http://127.0.0.1:8080
```



Welcome to the Tanzu Community Edition Installer

Tanzu Community Edition (TCE) is VMware's Open Source Kubernetes distribution. The installer will deploy a temporary cluster on your local machine to bootstrap a management cluster on your desired target. This management cluster can then be used to deploy and manage workload clusters.

For more details see the getting started guide.

Deploy the management cluster

Docker — Deploy your management cluster on your local workstation — DEPLOY

VMware vSphere — Deploy your management cluster on VMware vSphere — DEPLOY

Amazon Web Services — Deploy your management cluster on Amazon Elastic Compute Cloud — DEPLOY

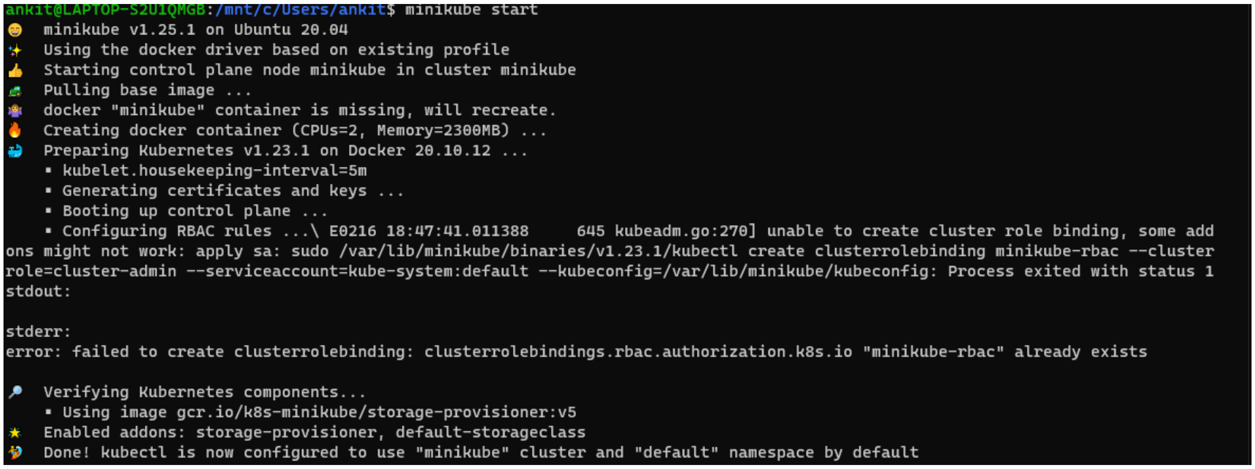
Microsoft Azure — Deploy your management cluster on Microsoft Azure — DEPLOY

5. Kuard and MiniKube Setup.

- ➔ Step 1. Get the kuard application.

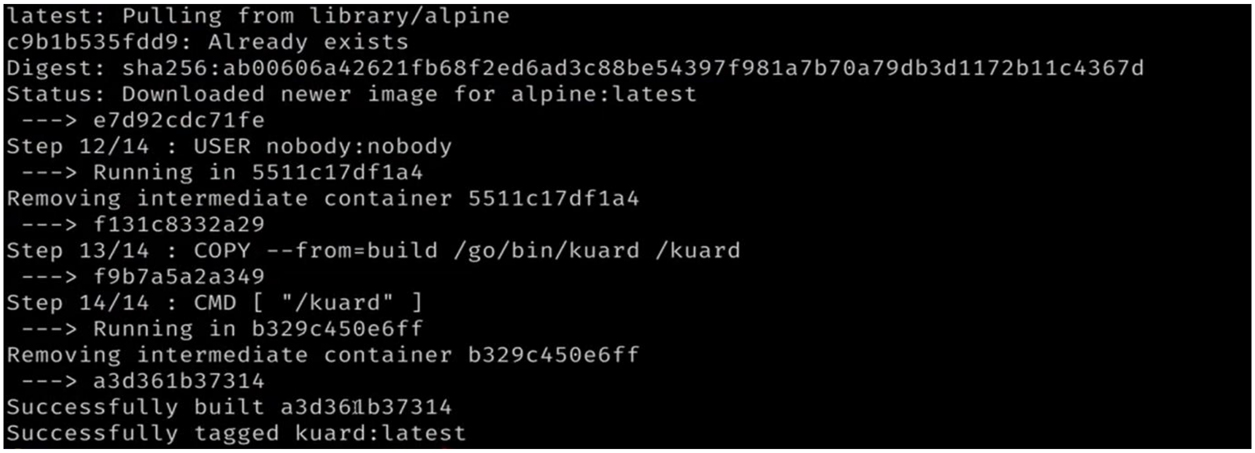
```
remote: Enumerating objects: 1418, done.
remote: Counting objects: 100% (8/8), done.
remote: Compressing objects: 100% (8/8), done.
Receremote: Total 1418 (delta 2), reused 2 (delta 0), pack-reused 1410
Receiving objects: 100% (1418/1418), 2.18 MiB | 2.29 MiB/s, done.
Resolving deltas: 100% (498/498), done.
```

- ➔ Step 2 – Install the GoPl
- ➔ Step 3 – Install the kubectl.
- ➔ Step 4 – Run minikube and make sure that all the above steps (1-4) are setup.
- ➔ Step 5. – Enable and check If the docker is able to deploy all.

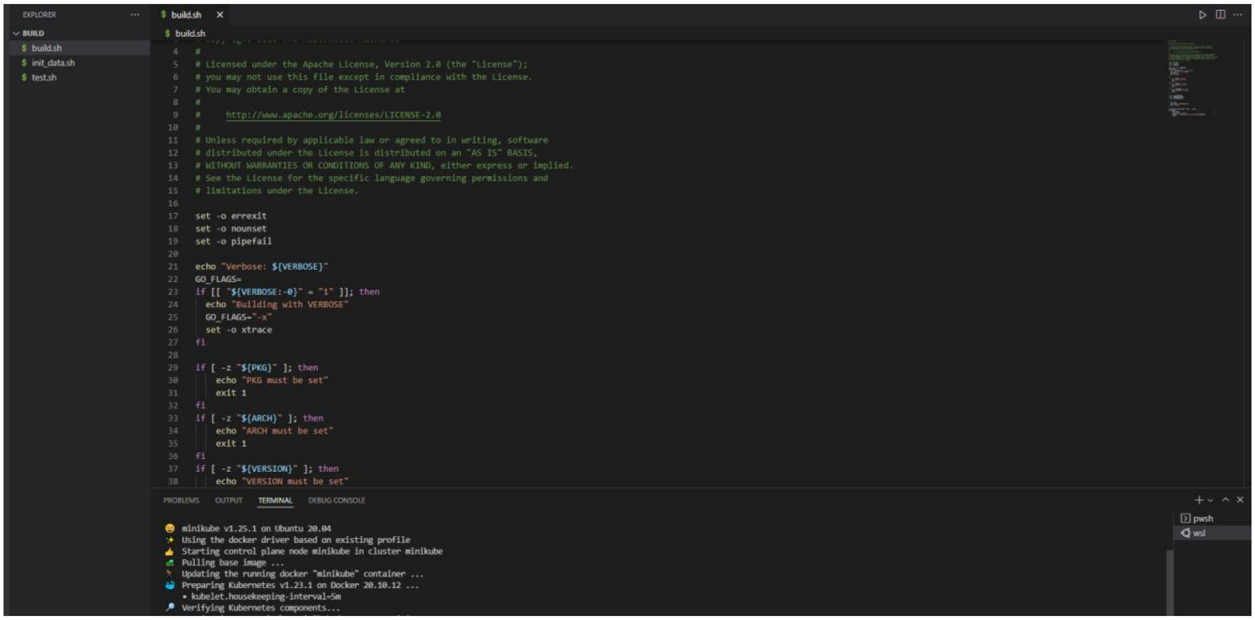


```
ankit@LAPTOP-S2U1QMGB:/mnt/c/Users/ankit$ minikube start
   minikube v1.25.1 on Ubuntu 20.04
   Using the docker driver based on existing profile
   Starting control plane node minikube in cluster minikube
   Pulling base image ...
   docker "minikube" container is missing, will recreate.
   Creating docker container (CPUs=2, Memory=2300MB) ...
   Preparing Kubernetes v1.23.1 on Docker 20.10.12 ...
    ▪ kubelet.housekeeping-interval=5m
    ▪ Generating certificates and keys ...
    ▪ Booting up control plane ...
    ▪ Configuring RBAC rules ...\ E0216 18:47:41.011388     645 kubeadm.go:270] unable to create cluster role binding, some add
ons might not work: apply sa: sudo /var/lib/minikube/binaries/v1.23.1/kubectl create clusterrolebinding minikube-rbac --cluster
role=cluster-admin --serviceaccount=kube-system:default --kubeconfig=/var/lib/minikube/kubeconfig: Process exited with status 1
stdout:

stderr:
error: failed to create clusterrolebinding: clusterrolebindings.rbac.authorization.k8s.io "minikube-rbac" already exists

   Verifying Kubernetes components...
    ▪ Using image gcr.io/k8s-minikube/storage-provisioner:v5
   Enabled addons: storage-provisioner, default-storageclass
   Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
```

- Step 6 – Check the sample applications and make sure .sh and touch scripts are running in **Kuard.**



```
latest: Pulling from library/alpine
c9b1b535fdd9: Already exists
Digest: sha256:ab00606a42621fb68f2ed6ad3c88be54397f981a7b70a79db3d1172b11c4367d
Status: Downloaded newer image for alpine:latest
 ---> e7d92cdc71fe
Step 12/14 : USER nobody:nobody
 ---> Running in 5511c17df1a4
Removing intermediate container 5511c17df1a4
 ---> f131c8332a29
Step 13/14 : COPY --from=build /go/bin/kuard /kuard
 ---> f9b7a5a2a349
Step 14/14 : CMD [ "/kuard" ]
 ---> Running in b329c450e6ff
Removing intermediate container b329c450e6ff
 ---> a3d361b37314
Successfully built a3d361b37314
Successfully tagged kuard:latest
```



```
#
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
#     http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.

set -o errexit
set -o nounset
set -o pipefail

echo "Verbose: ${VERBOSE}"
GO_FLAGS=
if [[ "${VERBOSE:-0}" = "1" ]]; then
  echo "Building with VERBOSE"
  GO_FLAGS="-x"
  set -o xtrace
fi

if [ -z "${PKG}" ]; then
    echo "PKG must be set"
    exit 1
fi
if [ -z "${ARCH}" ]; then
    echo "ARCH must be set"
    exit 1
fi
if [ -z "${VERSION}" ]; then
    echo "VERSION must be set"
```

```
   minikube v1.25.1 on Ubuntu 20.04
   Using the docker driver based on existing profile
   Starting control plane node minikube in cluster minikube
   Pulling base image ...
   Updating the running docker "minikube" container ...
   Preparing Kubernetes v1.23.1 on Docker 20.10.12 ...
    • kubelet.housekeeping-interval=5m
   Verifying Kubernetes components...
```

6. (Optional)testing a Hello world application.

➔ Clone the application to the same rep where the Kuard is and the Tanzu has been accessed at-least once on the core.

```
remote: Total 137 (delta 0), reused 0 (delta 0), pack-reused 137
Receiving objects: 100% (137/137), 155.86 KiB | 3.46 MiB/s, done.
Resolving deltas: 100% (20/20), done.
ankit@LAPTOP-S2U1QMGB:/mnt/c/kuard/hello$ ls
simple-go-web-app
```

➔ Switch back to Linux and run the Tanzu to touch the application.
Install dependencies.

```
Installing github.com/uudashr/gopkgs/v2/cmd/gopkgs@latest
(C:\Users\ankit\go\bin\gopkgs.exe) SUCCEEDED
Installing github.com/ramya-rao-a/go-outline@latest
(C:\Users\ankit\go\bin\go-outline.exe) SUCCEEDED
Installing github.com/cweill/gotests/gotests@latest
(C:\Users\ankit\go\bin\gotests.exe) SUCCEEDED
Installing github.com/fatih/gomodifytags@latest
(C:\Users\ankit\go\bin\gomodifytags.exe) SUCCEEDED
Installing github.com/josharian/impl@latest
(C:\Users\ankit\go\bin\impl.exe) SUCCEEDED
Installing github.com/haya14busa/goplay/cmd/goplay@latest
(C:\Users\ankit\go\bin\goplay.exe) SUCCEEDED
Installing github.com/go-delve/delve/cmd/dlv@latest
(C:\Users\ankit\go\bin\dlv.exe) SUCCEEDED
Installing honnef.co/go/tools/cmd/staticcheck@latest
(C:\Users\ankit\go\bin\staticcheck.exe) SUCCEEDED
Installing golang.org/x/tools/gopls@latest (C:\Users\ankit\go\bin\gopls.exe)
SUCCEEDED
All tools successfully installed. You are ready to Go. :)
```

7. Addition step 7 – Delete the amazon instance, Docker clean as well as the Tanzu instances.

- As all the expectations have been met, I have gone ahead and deleted all the **amazon EC2 instances.**
- Deleted all of the Volumes that were used to deploy the apps and the services.

- Finally deleted the security group for the above **Bucket** as well as the **server.**
- Deleted the minikube resources set up for the deployment.

```
docker system prune -a
WARNING! This will remove:
  - all stopped containers
  - all networks not used by at least one container
  - all images without at least one container associated to them
  - all build cache
Are you sure you want to continue? [y/N] y
Deleted Containers:
0c034ae63a0c3b118e5b92170cde19f807378b03aa344b4aba452fa684451f64
4ce88a695e4c3632e161ffb824a406bb00d17c96fcb7a458efbce6750ccde0a1
79e5503c668611a54de623258303a427dcc6f5f231e5fe28725007b42e6316da
Deleted Networks:
minikube

Deleted Images:
untagged: hello-world:latest
untagged: hello-world@sha256:97a379f4f88575512824f3b352bc03cd75e239179eea0fecc38e597b2209f49a
deleted: sha256:feb5d9fea6a5e9606aa995e879d862b825965ba48de054caab5ef356dc6b3412
deleted: sha256:e07ee1baac5fae6a26f30cabfe54a36d3402f96afda318fe0a96cec4ca393359
```

- Purged the **tanzu** Services

```
deleted: sha256:d55b3a11c2c16903280e578808cfd5c0fa3dedf513a1ee7dd895f0bb622ece5b

deleted: sha256:ae76d11f561d551ee4c904dc7a5ad487923f4c4a68f9871bccedf865cd76c0d2

deleted: sha256:9de65d1e8b2782409b2420bf9347003a43e91bb65c1e4c8fbd7d098d6234f359

deleted: sha256:e0f8e3acb2bf7fe9384463ae7009179d299b211e7cf17c2bf9d8e5e248cfe5b0

deleted: sha256:0e64bafdc7ee828d0f3995bebfa388ced52a625ad2969eeb569f4a83db56d505
```

Total reclaimed space: 1.144GB