GCIS-667 Cloud Networks

Jenkins

Running Jenkins on docker and Kubernetes
Kubectl and minikube.

Team Members:

Varshitha

Pranay Reddy

Shanmukha

Kubectl

Step 1: Install kubectl from windows platform

kubectl is a command-line utility for running tasks against Kubernetes clusters. Utilizing kubectl enables the deployment of apps, the inspection and management of cluster resources, and the viewing of logs. kubectl is available for installation on a wide number of Linux, macOS, and Windows platforms. In this case, we prefer the Windows operating system.

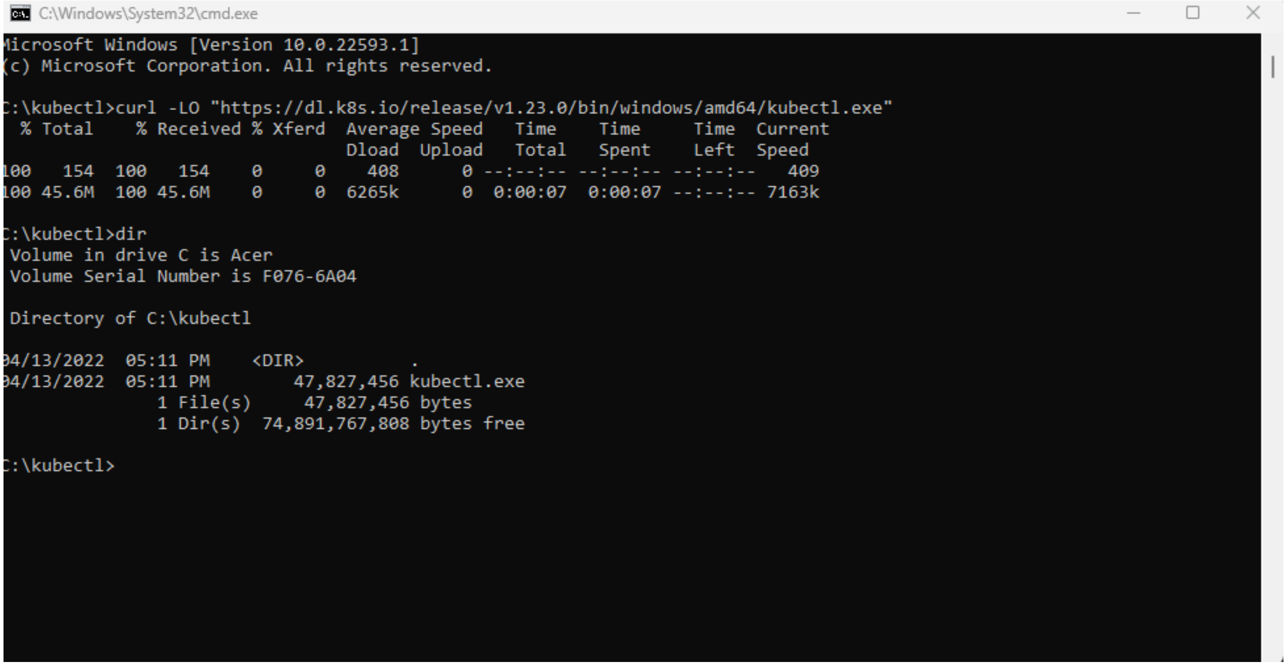
Step 2: Install and Set Up kubectl on Windows

The following methods exist for installing kubectl on Windows:

- Install kubectl binary with curl on Windows
- Install on Windows using Chocolatey or Scoop

Step 3: Install kubectl binary with curl on Windows

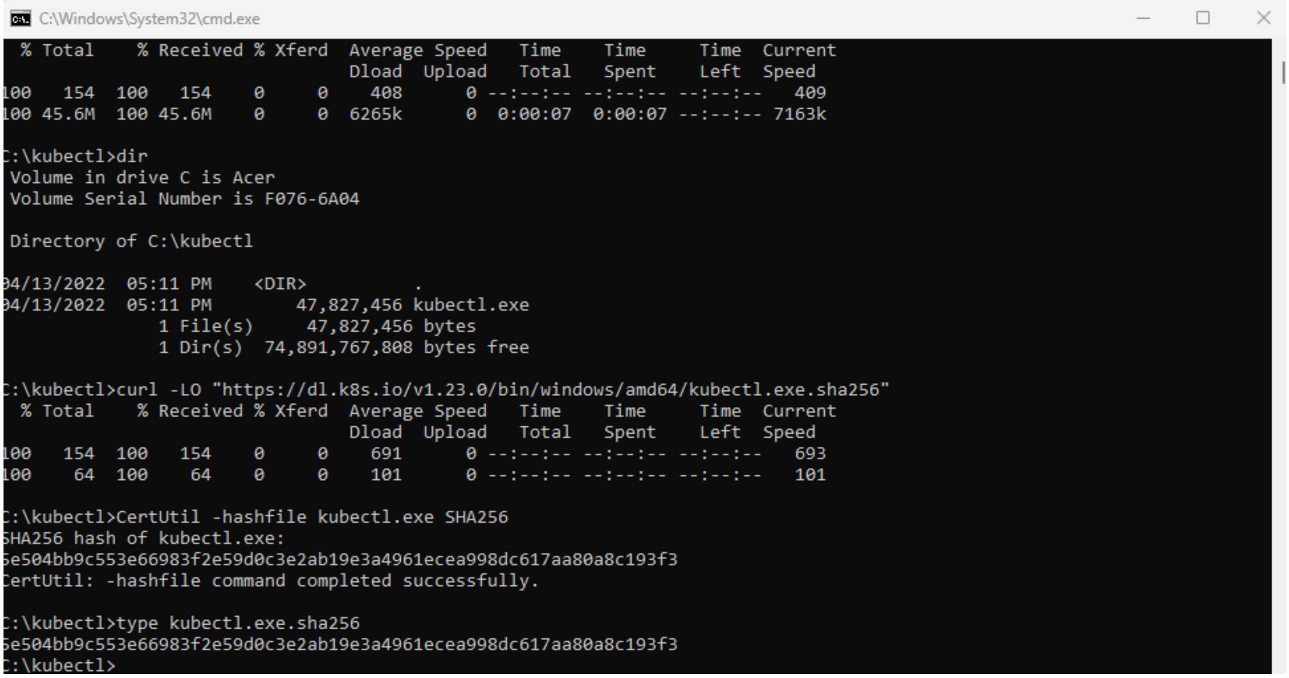
Two ways to install kubectl on windows:
We can either download the latest release of kubectl or we can do curl install from command prompt
Download the latest release v1.23.0.
curl -LO https://dl.k8s.io/release/v1.23.0/bin/windows/amd64/kubectl.exe
Here we did install from cmd



```
Microsoft Windows [Version 10.0.22593.1]
(c) Microsoft Corporation. All rights reserved.

C:\kubectl>curl -LO "https://dl.k8s.io/release/v1.23.0/bin/windows/amd64/kubectl.exe"
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   154  100   154    0     0    408      0 --:--:-- --:--:-- --:--:--   409
100 45.6M  100 45.6M    0     0  6265k      0  0:00:07  0:00:07 --:--:-- 7163k

C:\kubectl>dir
 Volume in drive C is Acer
 Volume Serial Number is F076-6A04

 Directory of C:\kubectl

04/13/2022  05:11 PM    <DIR>          .
04/13/2022  05:11 PM        47,827,456 kubectl.exe
               1 File(s)     47,827,456 bytes
               1 Dir(s)  74,891,767,808 bytes free

C:\kubectl>
```

Step 4: Validate the binary (optional)

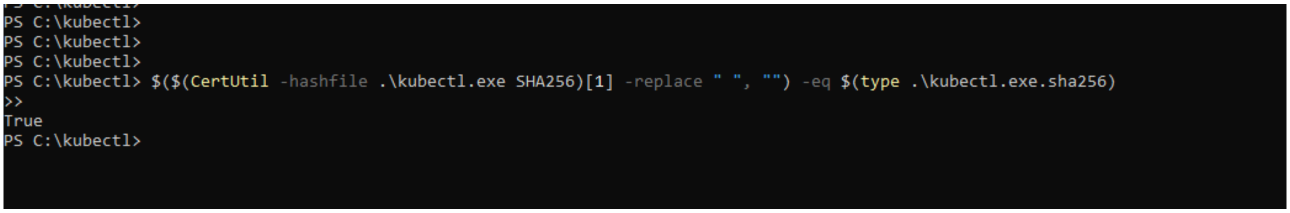
- Download the kubectl checksum file:
  curl -LO https://dl.k8s.io/v1.23.0/bin/windows/amd64/kubectl.exe.sha256
  Validate the kubectl binary against the checksum file:
- Using Command Prompt to manually compare CertUtil's output to the checksum file which is downloaded:
  CertUtil -hashfile kubectl.exe SHA256
  type kubectl.exe.sha256



```
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   154  100   154    0     0    408      0 --:--:-- --:--:-- --:--:--   409
100 45.6M  100 45.6M    0     0  6265k      0  0:00:07  0:00:07 --:--:-- 7163k

C:\kubectl>dir
 Volume in drive C is Acer
 Volume Serial Number is F076-6A04

 Directory of C:\kubectl

04/13/2022  05:11 PM    <DIR>          .
04/13/2022  05:11 PM        47,827,456 kubectl.exe
               1 File(s)     47,827,456 bytes
               1 Dir(s)  74,891,767,808 bytes free

C:\kubectl>curl -LO "https://dl.k8s.io/v1.23.0/bin/windows/amd64/kubectl.exe.sha256"
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   154  100   154    0     0    691      0 --:--:-- --:--:-- --:--:--   693
100    64  100    64    0     0    101      0 --:--:-- --:--:-- --:--:--   101

C:\kubectl>CertUtil -hashfile kubectl.exe SHA256
SHA256 hash of kubectl.exe:
5e504bb9c553e66983f2e59d0c3e2ab19e3a4961ecea998dc617aa80a8c193f3
CertUtil: -hashfile command completed successfully.

C:\kubectl>type kubectl.exe.sha256
5e504bb9c553e66983f2e59d0c3e2ab19e3a4961ecea998dc617aa80a8c193f3
C:\kubectl>
```

Step 5: PowerShell to automate the verification using the -eq operator

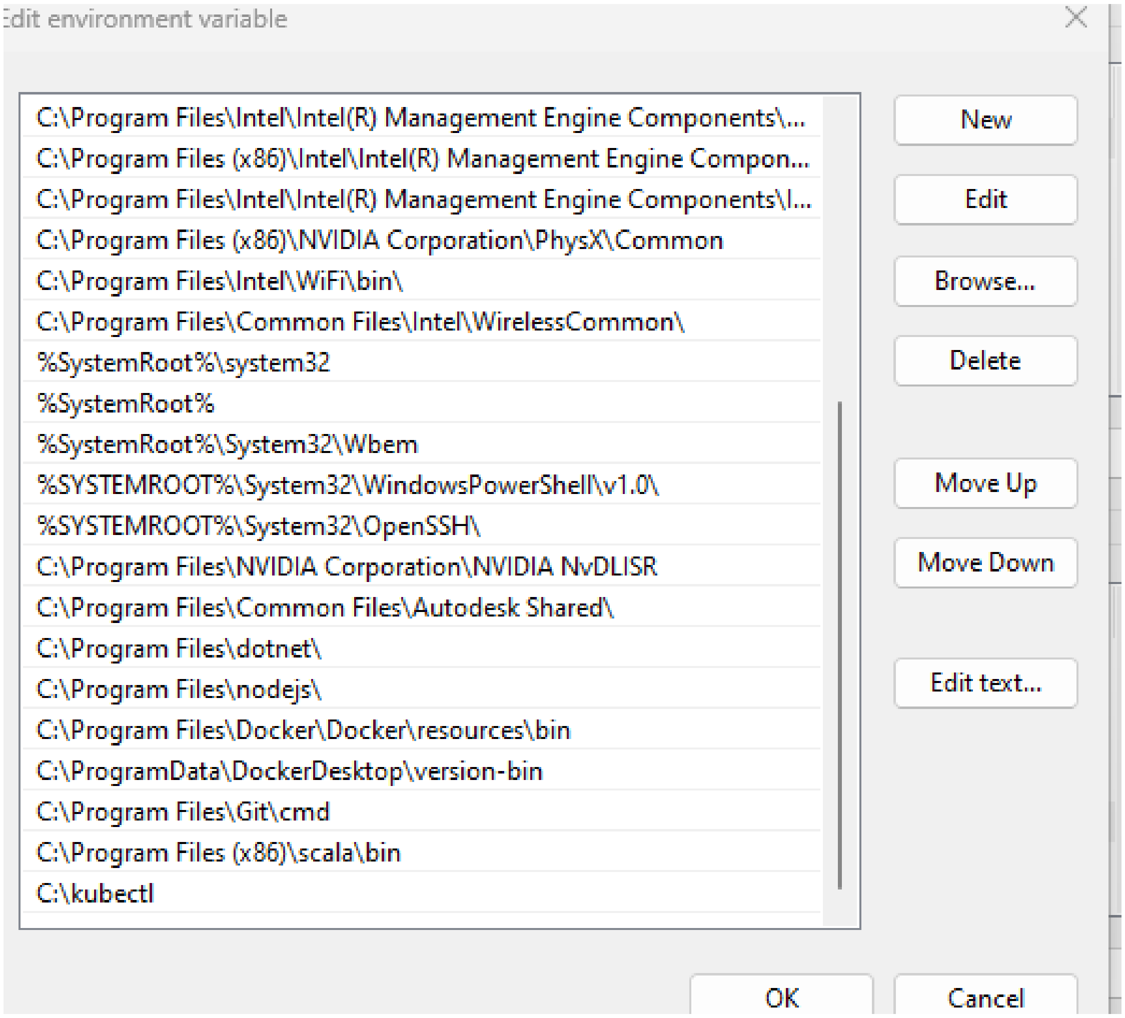
Using PowerShell to automate the verification using the -eq operator to get a True or False result:
$($(CertUtil -hashfile .\kubectl.exe SHA256)[1] -replace " ", "") -eq $(type .\kubectl.exe.sha256)



```
PS C:\kubectl>
PS C:\kubectl>
PS C:\kubectl>
PS C:\kubectl> $($(CertUtil -hashfile .\kubectl.exe SHA256)[1] -replace " ", "") -eq $(type .\kubectl.exe.sha256)
>>
True
PS C:\kubectl>
```

Step 6: Create the environment for kubectl

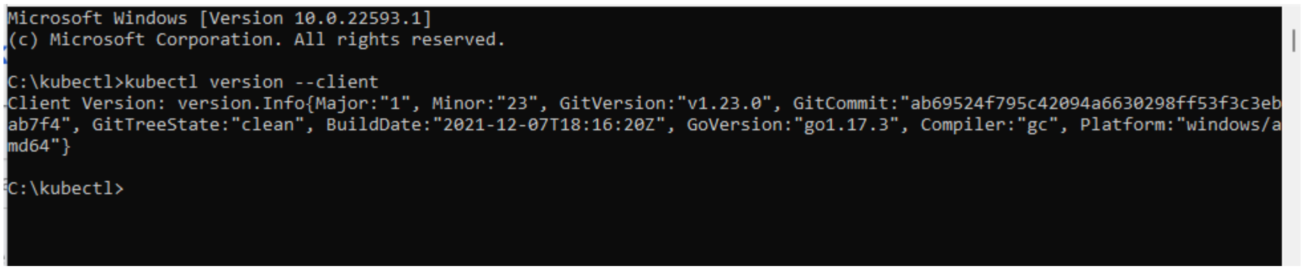
Using the environment variable, create an environment for kubectl. We will also be able to alter the path, so save the path as kubectl as shown in the figure below.



C:\Program Files\Intel\Intel(R) Management Engine Components\...
C:\Program Files (x86)\Intel\Intel(R) Management Engine Compon...
C:\Program Files\Intel\Intel(R) Management Engine Components\I...
C:\Program Files (x86)\NVIDIA Corporation\PhysX\Common
C:\Program Files\Intel\WiFi\bin\
C:\Program Files\Common Files\Intel\WirelessCommon\
%SystemRoot%\system32
%SystemRoot%
%SystemRoot%\System32\Wbem
%SYSTEMROOT%\System32\WindowsPowerShell\v1.0\
%SYSTEMROOT%\System32\OpenSSH\
C:\Program Files\NVIDIA Corporation\NVIDIA NvDLISR
C:\Program Files\Common Files\Autodesk Shared\
C:\Program Files\dotnet\
C:\Program Files\nodejs\
C:\Program Files\Docker\Docker\resources\bin
C:\ProgramData\DockerDesktop\version-bin
C:\Program Files\Git\cmd
C:\Program Files (x86)\scala\bin
C:\kubectl

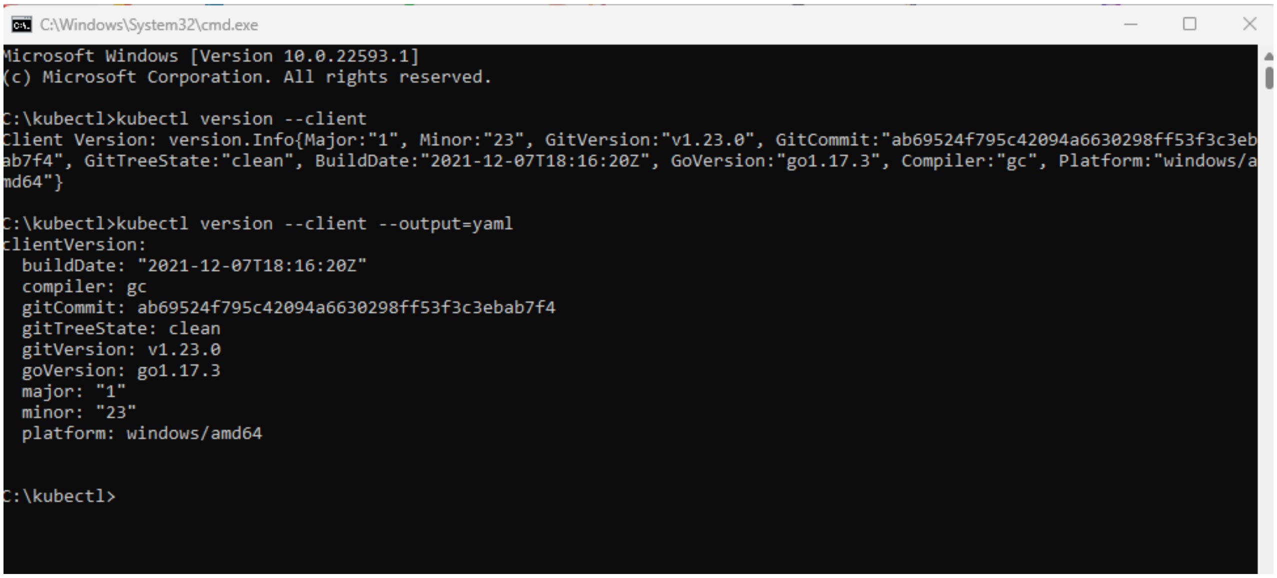
Step 7: Verify that the version of kubectl is same to the one downloaded

- Test whether we installed the right version of kubectl which we downloaded earlier
  kubectl version –client



```
Microsoft Windows [Version 10.0.22593.1]
(c) Microsoft Corporation. All rights reserved.

C:\kubectl>kubectl version --client
Client Version: version.Info{Major:"1", Minor:"23", GitVersion:"v1.23.0", GitCommit:"ab69524f795c42094a6630298ff53f3c3eb
ab7f4", GitTreeState:"clean", BuildDate:"2021-12-07T18:16:20Z", GoVersion:"go1.17.3", Compiler:"gc", Platform:"windows/a
md64"}

C:\kubectl>
```

- To view the detailed view of the version we downloaded, below is the cmd kubectl version --client --output=yaml



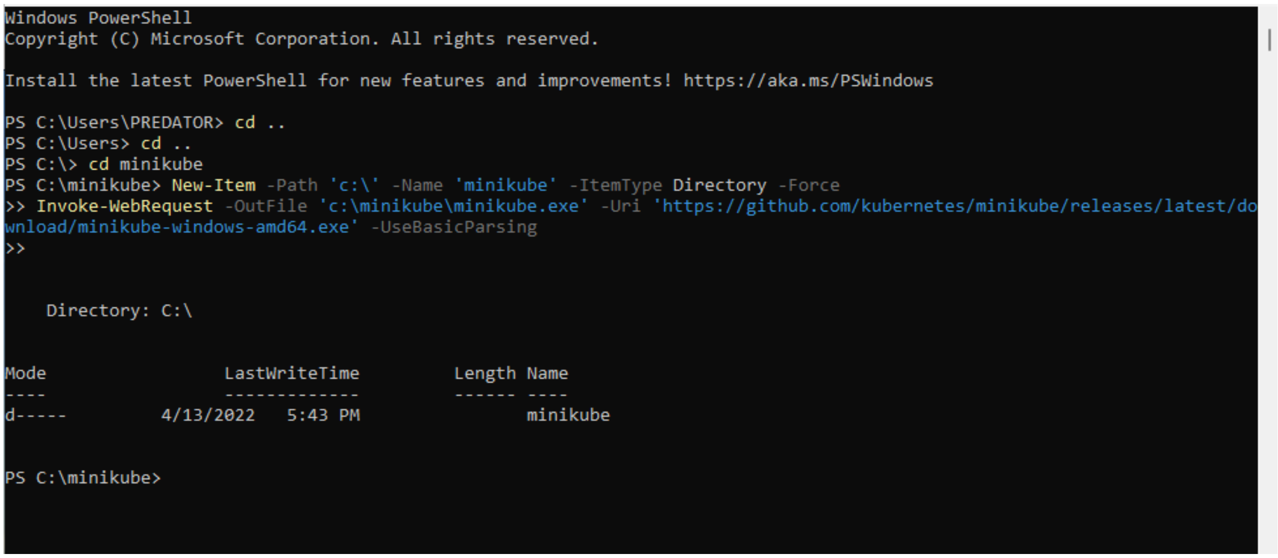
C:\Windows\System32\cmd.exe

```
Microsoft Windows [Version 10.0.22593.1]
(c) Microsoft Corporation. All rights reserved.

C:\kubectl>kubectl version --client
Client Version: version.Info{Major:"1", Minor:"23", GitVersion:"v1.23.0", GitCommit:"ab69524f795c42094a6630298ff53f3c3eb
ab7f4", GitTreeState:"clean", BuildDate:"2021-12-07T18:16:20Z", GoVersion:"go1.17.3", Compiler:"gc", Platform:"windows/a
md64"}

C:\kubectl>kubectl version --client --output=yaml
clientVersion:
  buildDate: "2021-12-07T18:16:20Z"
  compiler: gc
  gitCommit: ab69524f795c42094a6630298ff53f3c3ebab7f4
  gitTreeState: clean
  gitVersion: v1.23.0
  goVersion: go1.17.3
  major: "1"
  minor: "23"
  platform: windows/amd64


C:\kubectl>
```

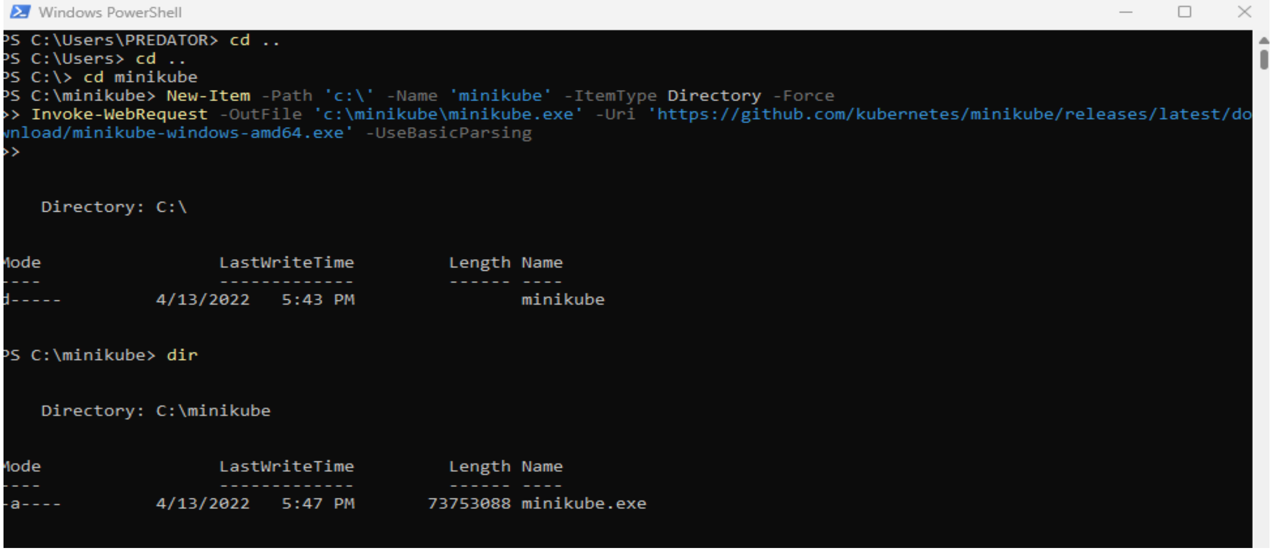
minikube

minikube is local Kubernetes, focusing on making it easy to develop Kubernetes. All we need is a Docker container or a Virtual Machine environment.

Step 8: create a folder and run cmd as cd minikube in the same directory



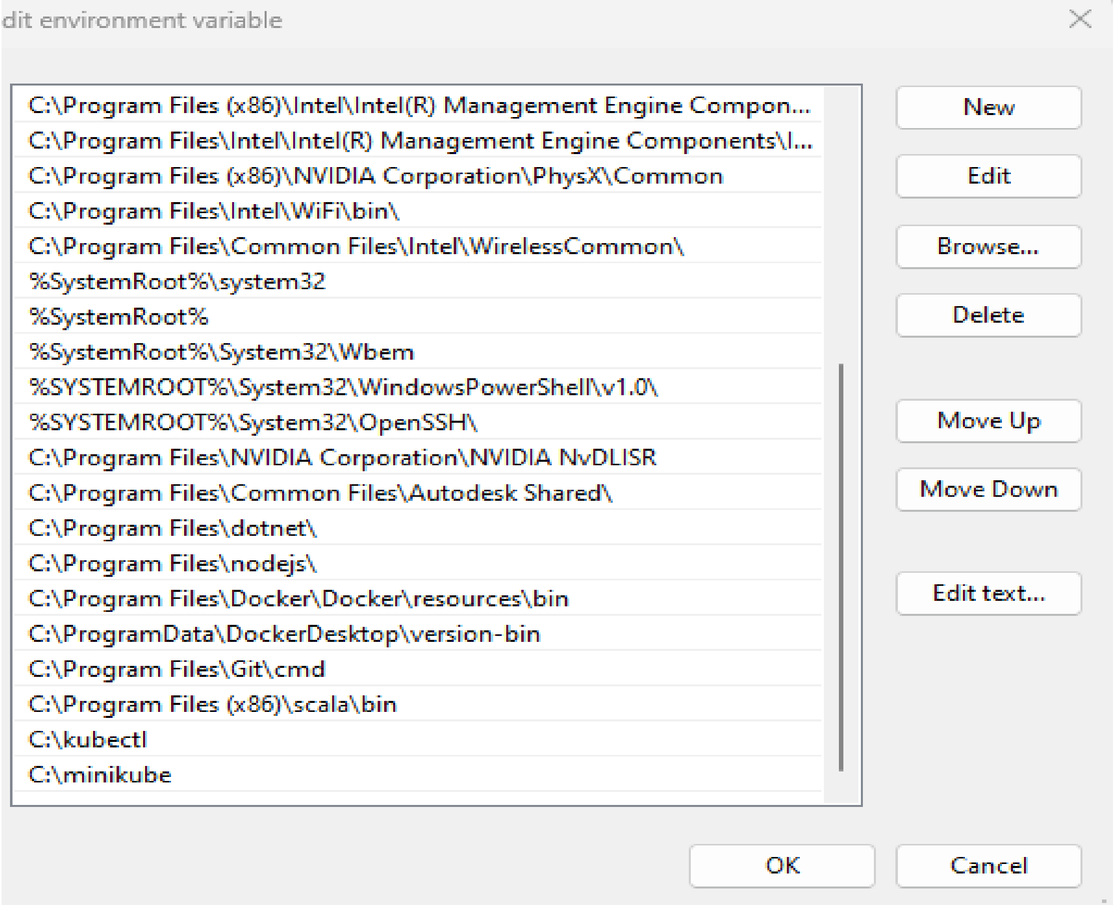
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\PREDATOR> cd ..
PS C:\Users> cd ..
PS C:\> cd minikube
PS C:\minikube> New-Item -Path 'c:\' -Name 'minikube' -ItemType Directory -Force
>> Invoke-WebRequest -OutFile 'c:\minikube\minikube.exe' -Uri 'https://github.com/kubernetes/minikube/releases/latest/do
wnload/minikube-windows-amd64.exe' -UseBasicParsing
>>


    Directory: C:\


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----         4/13/2022   5:43 PM                minikube


PS C:\minikube>
```



```
PS C:\Users\PREDATOR> cd ..
PS C:\Users> cd ..
PS C:\> cd minikube
PS C:\minikube> New-Item -Path 'c:\' -Name 'minikube' -ItemType Directory -Force
>> Invoke-WebRequest -OutFile 'c:\minikube\minikube.exe' -Uri 'https://github.com/kubernetes/minikube/releases/latest/do
wnload/minikube-windows-amd64.exe' -UseBasicParsing
>>


    Directory: C:\


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----         4/13/2022   5:43 PM                minikube


PS C:\minikube> dir


    Directory: C:\minikube


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----         4/13/2022   5:47 PM       73753088 minikube.exe
```

Step 9: Create the environment for minikube

Using the environment variable, create an environment for minikube. We will also be able to alter the path, so save the path as minikube as shown in the figure below.



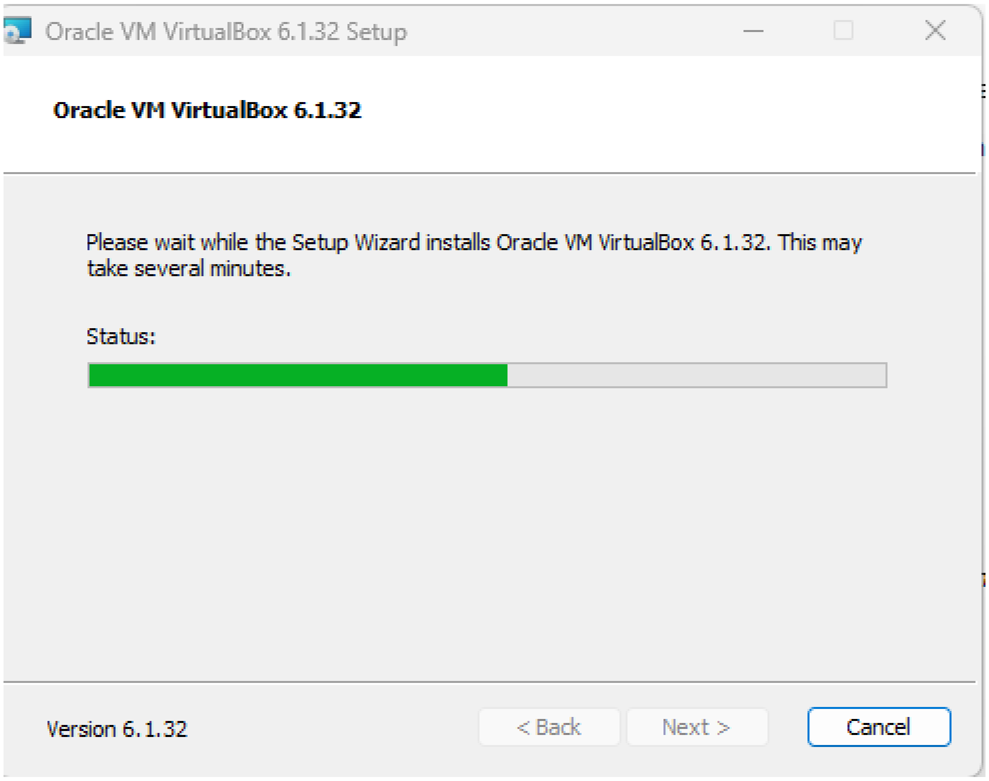
C:\Program Files (x86)\Intel\Intel(R) Management Engine Compon...
C:\Program Files\Intel\Intel(R) Management Engine Components\I...
C:\Program Files (x86)\NVIDIA Corporation\PhysX\Common
C:\Program Files\Intel\WiFi\bin\
C:\Program Files\Common Files\Intel\WirelessCommon\
%SystemRoot%\system32
%SystemRoot%
%SystemRoot%\System32\Wbem
%SYSTEMROOT%\System32\WindowsPowerShell\v1.0\
%SYSTEMROOT%\System32\OpenSSH\
C:\Program Files\NVIDIA Corporation\NVIDIA NvDLISR
C:\Program Files\Common Files\Autodesk Shared\
C:\Program Files\dotnet\
C:\Program Files\nodejs\
C:\Program Files\Docker\Docker\resources\bin
C:\ProgramData\DockerDesktop\version-bin
C:\Program Files\Git\cmd
C:\Program Files (x86)\scala\bin
C:\kubectl
C:\minikube

Virtualbox

VirtualBox is minikube's original driver. It may not provide the fastest start-up time, but it is the most stable driver available for users of Microsoft Windows Home.

Step 10: Installing the Oracle VM VirtualBox

Oracle VM VirtualBox, the world's most popular open source, cross-platform, virtualization software, enables developers to deliver code faster by running multiple operating systems on a single device. IT teams and solution providers use VirtualBox to reduce operational costs and shorten the time needed to securely deploy applications on-premises and to the cloud.



Oracle VM VirtualBox 6.1.32 Setup

Oracle VM VirtualBox 6.1.32

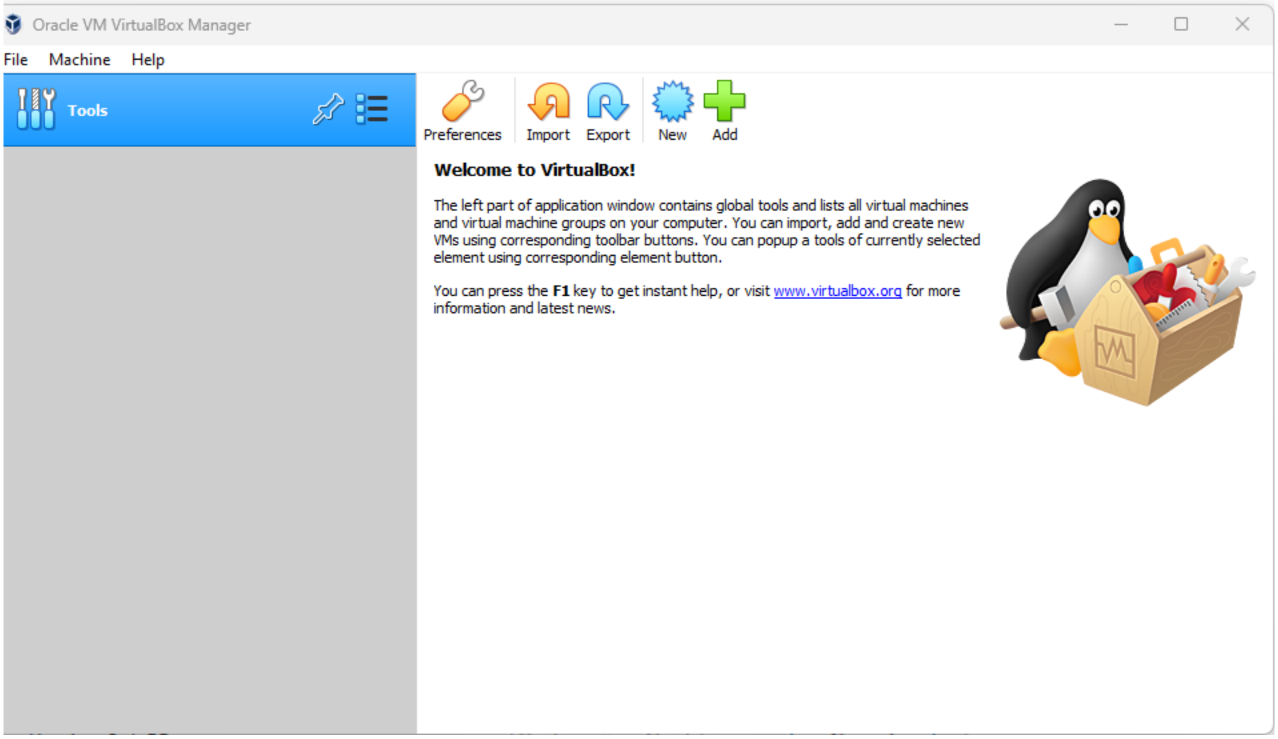
Please wait while the Setup Wizard installs Oracle VM VirtualBox 6.1.32. This may take several minutes.

Status:

Version 6.1.32

Step 11: How the Oracle VM VirtualBox installed on windows

The following screenshot shows how Oracle VM VirtualBox, installed and running on Windows Server in a virtual machine window.



Welcome to VirtualBox!

The left part of application window contains global tools and lists all virtual machines and virtual machine groups on your computer. You can import, add and create new VMs using corresponding toolbar buttons. You can popup a tools of currently selected element using corresponding element button.

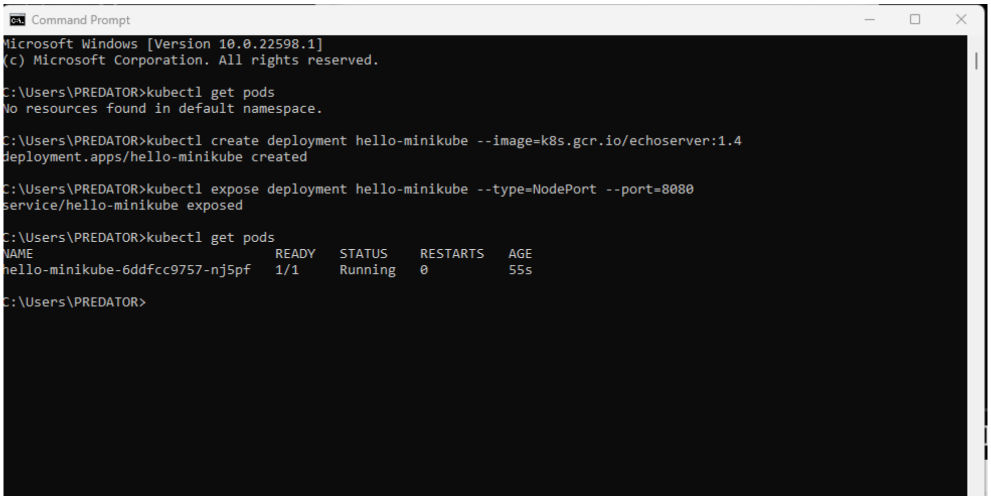
You can press the **F1** key to get instant help, or visit www.virtualbox.org for more information and latest news.

Step 12:

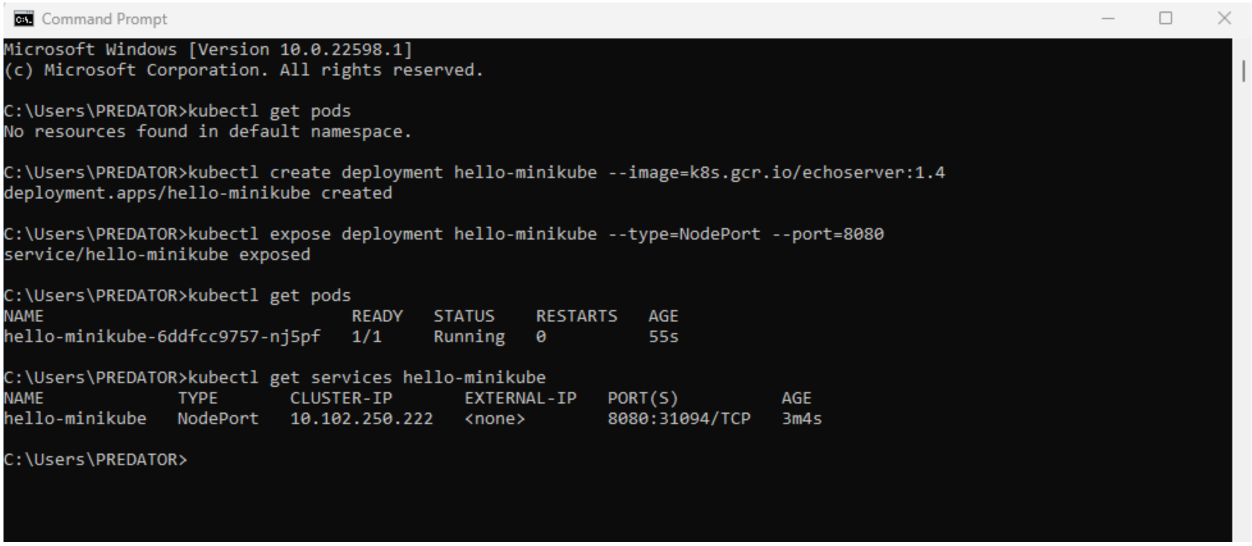
To see pods what we created we use kubectl get pods

We have to create deployment and we have to give a name.

We use command kubectl create deployment hello-minikube –image-k8s.gcr.io/echoserver:1.4



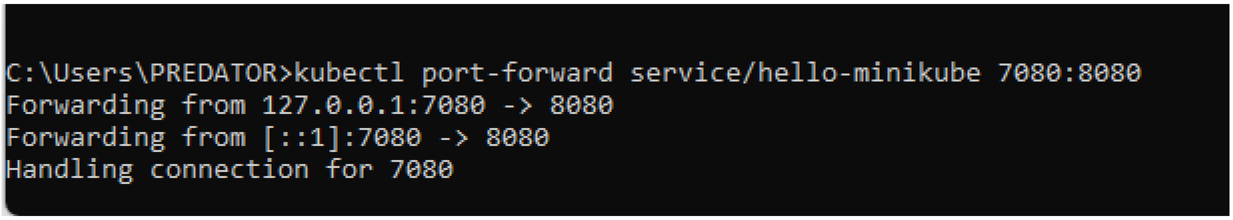
```
Microsoft Windows [Version 10.0.22598.1]
(c) Microsoft Corporation. All rights reserved.

C:\Users\PREDATOR>kubectl get pods
No resources found in default namespace.

C:\Users\PREDATOR>kubectl create deployment hello-minikube --image=k8s.gcr.io/echoserver:1.4
deployment.apps/hello-minikube created

C:\Users\PREDATOR>kubectl expose deployment hello-minikube --type=NodePort --port=8080
service/hello-minikube exposed

C:\Users\PREDATOR>kubectl get pods
NAME                              READY   STATUS    RESTARTS   AGE
hello-minikube-6ddfcc9757-nj5pf   1/1     Running   0          55s

C:\Users\PREDATOR>
```



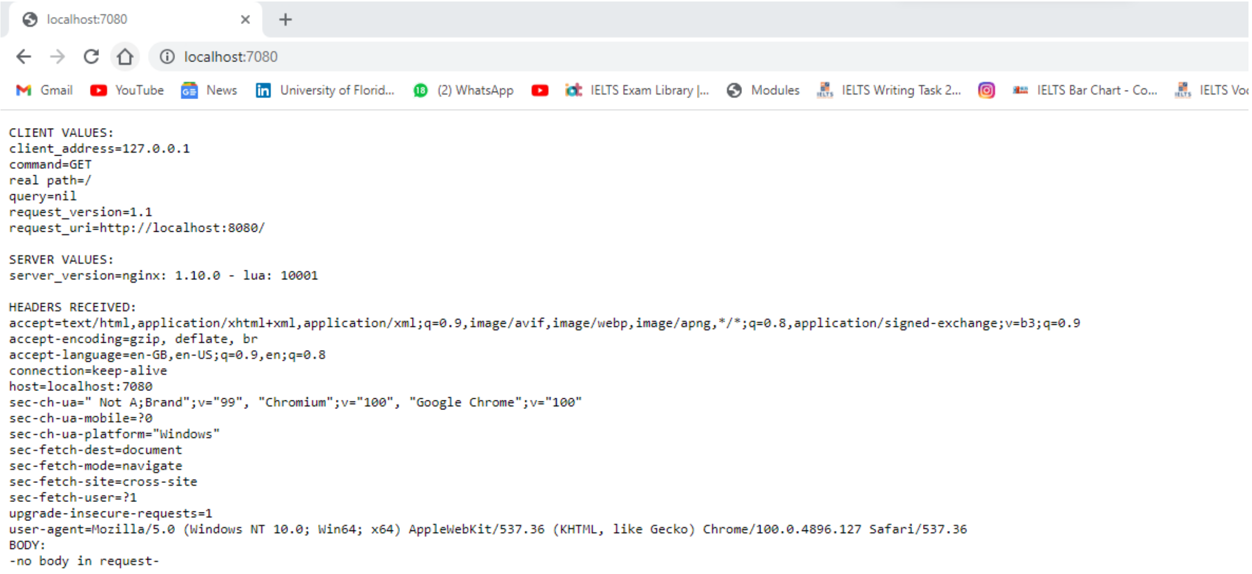
```
Microsoft Windows [Version 10.0.22598.1]
(c) Microsoft Corporation. All rights reserved.

C:\Users\PREDATOR>kubectl get pods
No resources found in default namespace.

C:\Users\PREDATOR>kubectl create deployment hello-minikube --image=k8s.gcr.io/echoserver:1.4
deployment.apps/hello-minikube created

C:\Users\PREDATOR>kubectl expose deployment hello-minikube --type=NodePort --port=8080
service/hello-minikube exposed

C:\Users\PREDATOR>kubectl get pods
NAME                              READY   STATUS    RESTARTS   AGE
hello-minikube-6ddfcc9757-nj5pf   1/1     Running   0          55s

C:\Users\PREDATOR>kubectl get services hello-minikube
NAME             TYPE       CLUSTER-IP       EXTERNAL-IP   PORT(S)          AGE
hello-minikube   NodePort   10.102.250.222   <none>        8080:31094/TCP   3m4s

C:\Users\PREDATOR>
```

We will port forward

7080-8080



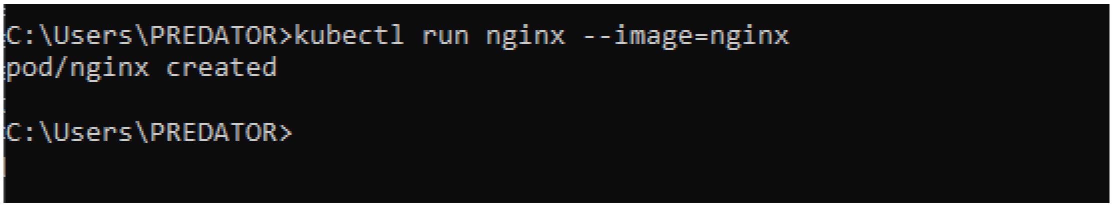
```
C:\Users\PREDATOR>kubectl port-forward service/hello-minikube 7080:8080
Forwarding from 127.0.0.1:7080 -> 8080
Forwarding from [::1]:7080 -> 8080
Handling connection for 7080
```



```
CLIENT VALUES:
client_address=127.0.0.1
command=GET
real path=/
query=nil
request_version=1.1
request_uri=http://localhost:8080/

SERVER VALUES:
server_version=nginx: 1.10.0 - lua: 10001

HEADERS RECEIVED:
accept=text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
accept-encoding=gzip, deflate, br
accept-language=en-GB,en-US;q=0.9,en;q=0.8
connection=keep-alive
host=localhost:7080
sec-ch-ua=" Not A;Brand";v="99", "Chromium";v="100", "Google Chrome";v="100"
sec-ch-ua-mobile=?0
sec-ch-ua-platform="Windows"
sec-fetch-dest=document
sec-fetch-mode=navigate
sec-fetch-site=cross-site
sec-fetch-user=?1
upgrade-insecure-requests=1
user-agent=Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/100.0.4896.127 Safari/537.36
BODY:
-no body in request-
```

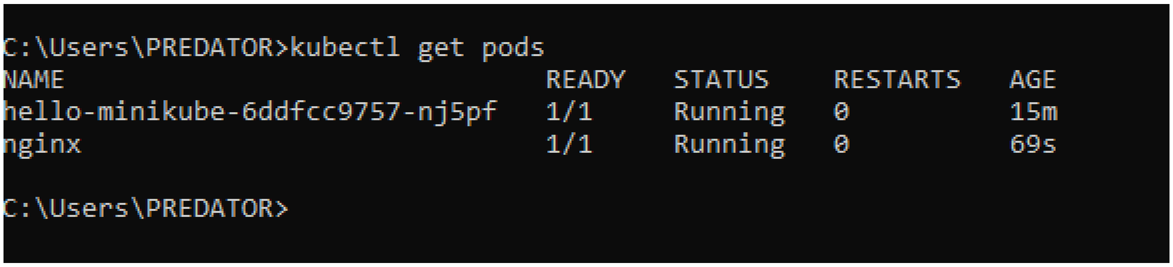
To check server is running we have see in localhost:7080

We have Created nginx



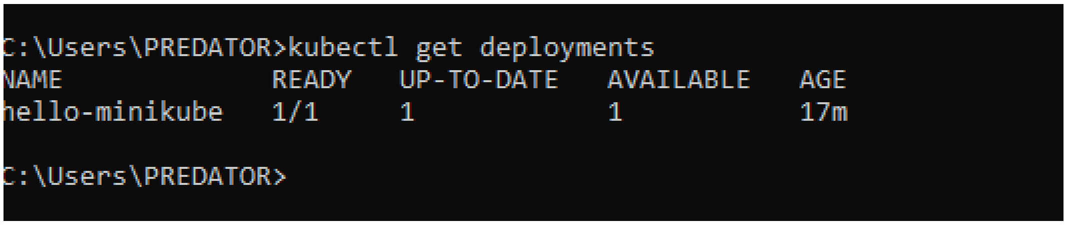
```
C:\Users\PREDATOR>kubectl run nginx --image=nginx
pod/nginx created

C:\Users\PREDATOR>
```

Two pods are running

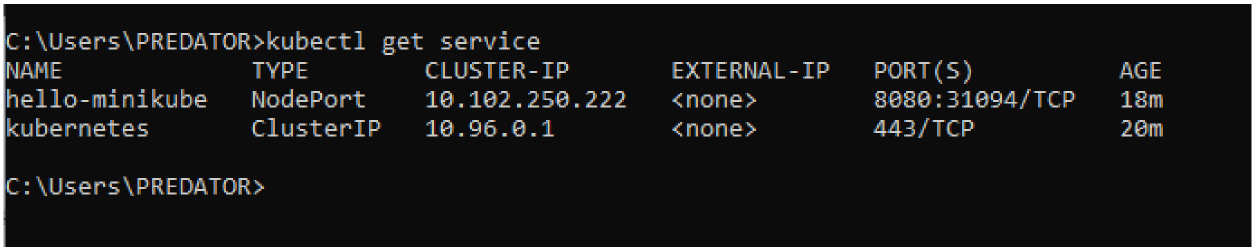


```
C:\Users\PREDATOR>kubectl get pods
NAME                              READY   STATUS    RESTARTS   AGE
hello-minikube-6ddfcc9757-nj5pf   1/1     Running   0          15m
nginx                             1/1     Running   0          69s

C:\Users\PREDATOR>
```

Get deployments

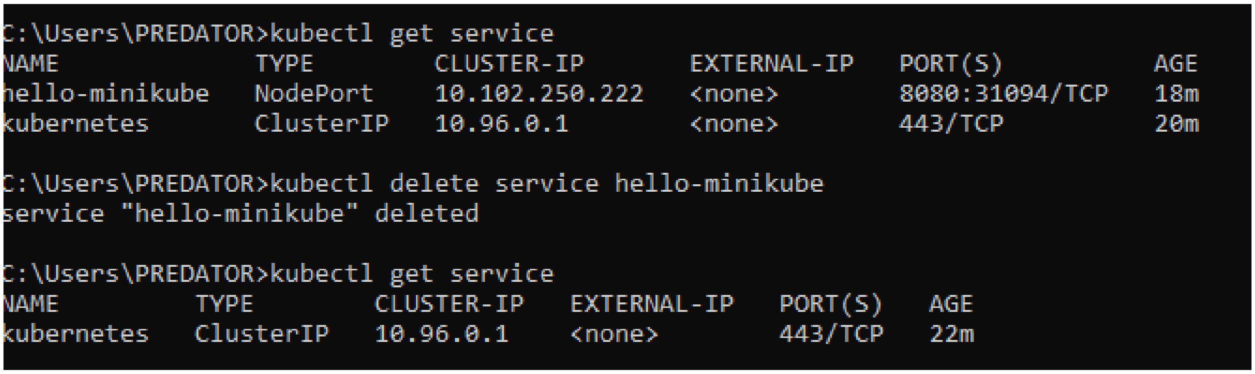


```
C:\Users\PREDATOR>kubectl get deployments
NAME             READY   UP-TO-DATE   AVAILABLE   AGE
hello-minikube   1/1     1            1           17m

C:\Users\PREDATOR>
```



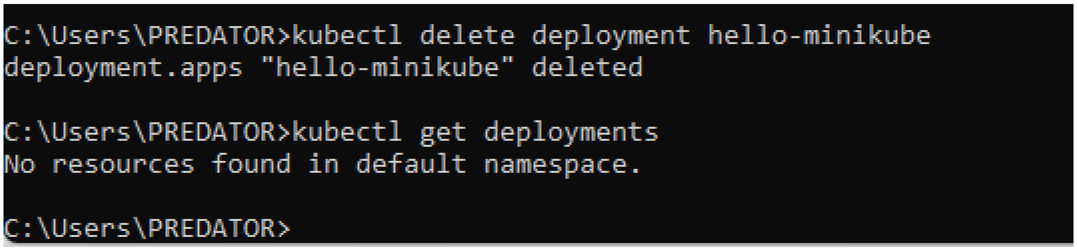
```
C:\Users\PREDATOR>kubectl get service
NAME             TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)          AGE
hello-minikube   NodePort    10.102.250.222   <none>        8080:31094/TCP   18m
kubernetes       ClusterIP   10.96.0.1        <none>        443/TCP          20m

C:\Users\PREDATOR>
```

To Delete service



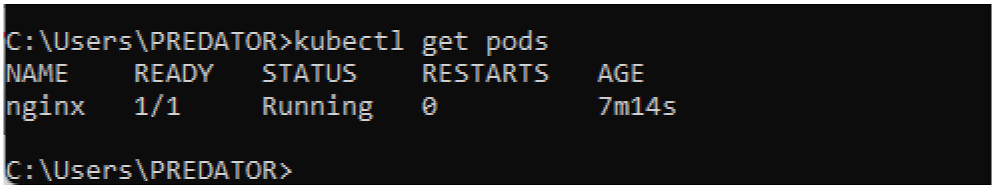
```
C:\Users\PREDATOR>kubectl get service
NAME             TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)          AGE
hello-minikube   NodePort    10.102.250.222   <none>        8080:31094/TCP   18m
kubernetes       ClusterIP   10.96.0.1        <none>        443/TCP          20m

C:\Users\PREDATOR>kubectl delete service hello-minikube
service "hello-minikube" deleted

C:\Users\PREDATOR>kubectl get service
NAME         TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
kubernetes   ClusterIP   10.96.0.1    <none>        443/TCP   22m
```

To Delete deployment



```
C:\Users\PREDATOR>kubectl delete deployment hello-minikube
deployment.apps "hello-minikube" deleted

C:\Users\PREDATOR>kubectl get deployments
No resources found in default namespace.

C:\Users\PREDATOR>
```

Then after deleting all service it running Only nginx

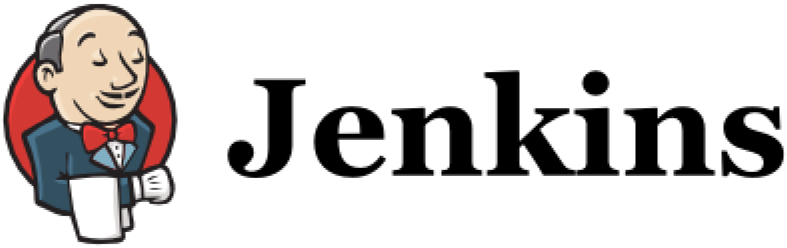


```
C:\Users\PREDATOR>kubectl get pods
NAME    READY   STATUS    RESTARTS   AGE
nginx   1/1     Running   0          7m14s

C:\Users\PREDATOR>
```

Pulling Jerkins from docker hub

Jenkins

In the Java programming language, Jenkins is an open source Continuous Integration/Continuous Delivery and Deployment (CI/CD) automation tool for DevOps, which means that it can be used by anyone. It is used to make CI/CD workflows, which are called pipelines.

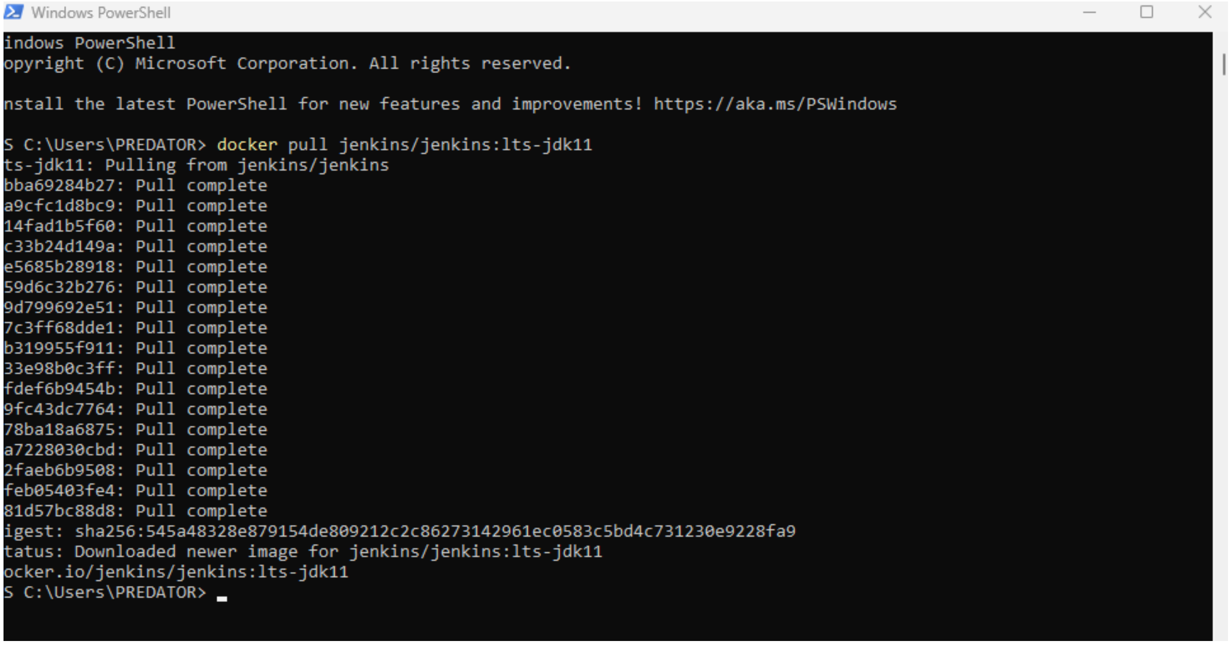


Pipelines automate the testing and reporting of small changes in a large code base, and they make it easier to merge different branches of the code into a single main branch. Defects in a code base can also be found quickly with these tools. They build software, run automated tests on their builds, prepare the code base for deployment (delivery), and then deploy the code to containers and virtual machines, as well as bare metal and cloud servers. There are a lot of paid versions of Jenkins. This word only refers to the open source that comes from the upstream.

Steps for installing Jenkins on docker

1 we have to open power shell

2 run a command `docker pull jenkins/jenkins:lts-jdk11`



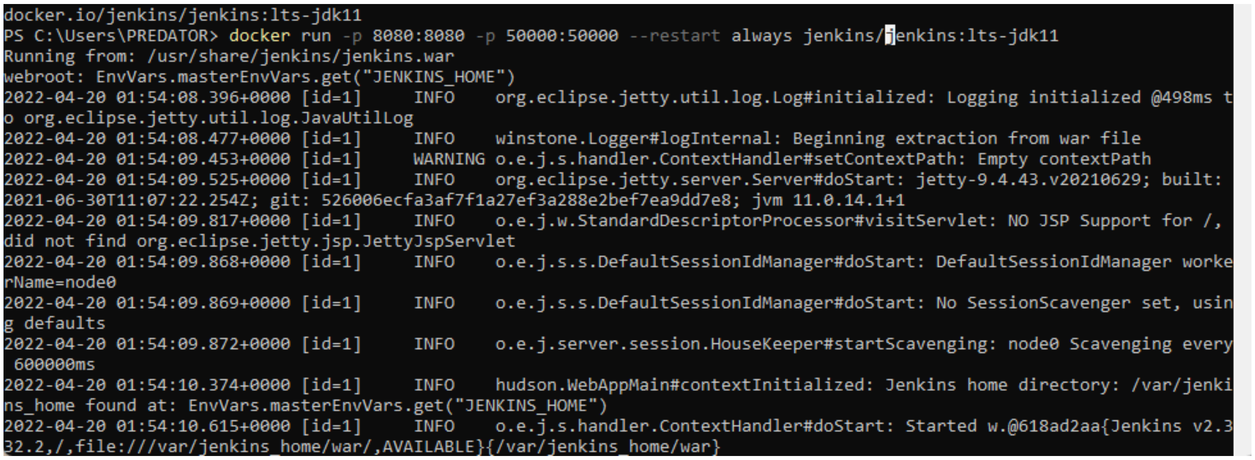
```
indows PowerShell
opyright (C) Microsoft Corporation. All rights reserved.

nstall the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

S C:\Users\PREDATOR> docker pull jenkins/jenkins:lts-jdk11
ts-jdk11: Pulling from jenkins/jenkins
bba69284b27: Pull complete
a9cfc1d8bc9: Pull complete
14fad1b5f60: Pull complete
c33b24d149a: Pull complete
e5685b28918: Pull complete
59d6c32b276: Pull complete
9d799692e51: Pull complete
7c3ff68dde1: Pull complete
b319955f911: Pull complete
33e98b0c3ff: Pull complete
fdef6b9454b: Pull complete
9fc43dc7764: Pull complete
78ba18a6875: Pull complete
a7228030cbd: Pull complete
2faeb6b9508: Pull complete
feb05403fe4: Pull complete
81d57bc88d8: Pull complete
igest: sha256:545a48328e879154de809212c2c86273142961ec0583c5bd4c731230e9228fa9
tatus: Downloaded newer image for jenkins/jenkins:lts-jdk11
ocker.io/jenkins/jenkins:lts-jdk11
S C:\Users\PREDATOR> _
```

Then we have running Jenkins on docker

We have create a image so

We used command

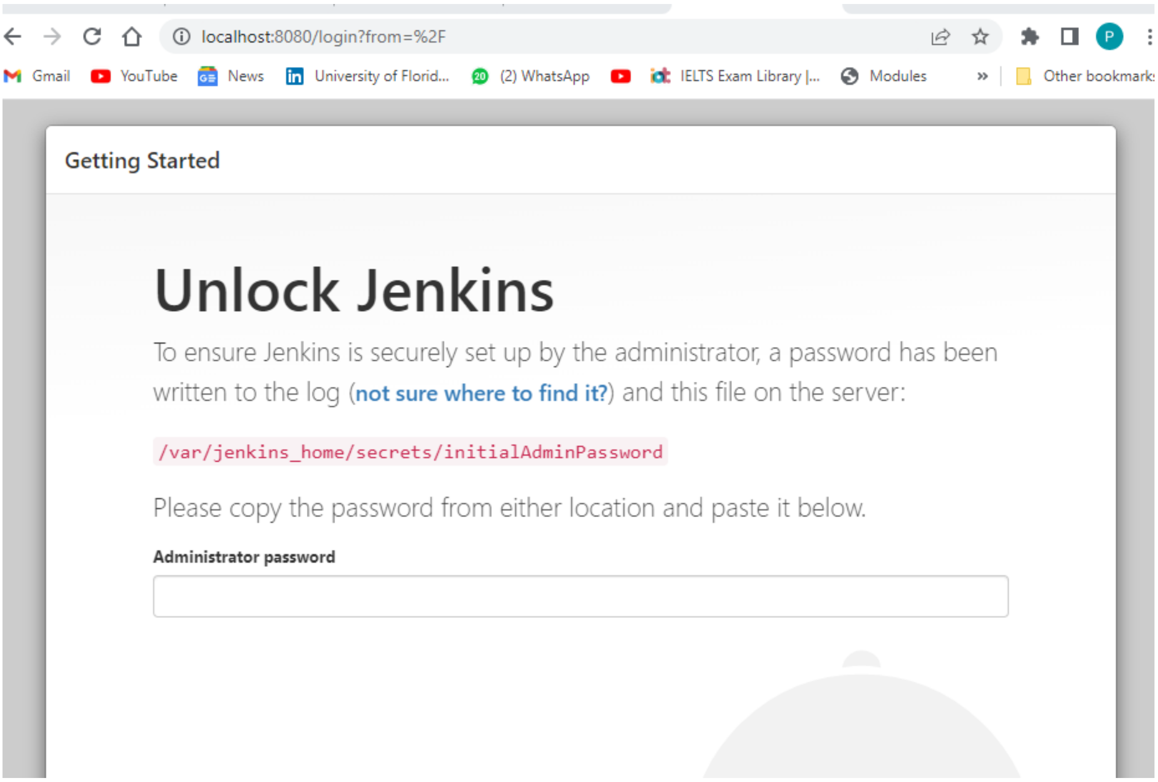
docker run -p 8080:8080 -p 50000:50000 --restart always jenkins/jenkins:lts-jdk11



```
docker.io/jenkins/jenkins:lts-jdk11
PS C:\Users\PREDATOR> docker run -p 8080:8080 -p 50000:50000 --restart always jenkins/jenkins:lts-jdk11
Running from: /usr/share/jenkins/jenkins.war
webroot: EnvVars.masterEnvVars.get("JENKINS_HOME")
2022-04-20 01:54:08.396+0000 [id=1]     INFO    org.eclipse.jetty.util.log.Log#initialized: Logging initialized @498ms t
o org.eclipse.jetty.util.log.JavaUtilLog
2022-04-20 01:54:08.477+0000 [id=1]     INFO    winstone.Logger#logInternal: Beginning extraction from war file
2022-04-20 01:54:09.453+0000 [id=1]     WARNING o.e.j.s.handler.ContextHandler#setContextPath: Empty contextPath
2022-04-20 01:54:09.525+0000 [id=1]     INFO    org.eclipse.jetty.server.Server#doStart: jetty-9.4.43.v20210629; built:
2021-06-30T11:07:22.254Z; git: 526006ecfa3af7f1a27ef3a288e2bef7ea9dd7e8; jvm 11.0.14.1+1
2022-04-20 01:54:09.817+0000 [id=1]     INFO    o.e.j.w.StandardDescriptorProcessor#visitServlet: NO JSP Support for /,
did not find org.eclipse.jetty.jsp.JettyJspServlet
2022-04-20 01:54:09.868+0000 [id=1]     INFO    o.e.j.s.s.DefaultSessionIdManager#doStart: DefaultSessionIdManager worke
rName=node0
2022-04-20 01:54:09.869+0000 [id=1]     INFO    o.e.j.s.s.DefaultSessionIdManager#doStart: No SessionScavenger set, usin
g defaults
2022-04-20 01:54:09.872+0000 [id=1]     INFO    o.e.j.server.session.HouseKeeper#startScavenging: node0 Scavenging every
 600000ms
2022-04-20 01:54:10.374+0000 [id=1]     INFO    hudson.WebAppMain#contextInitialized: Jenkins home directory: /var/jenki
ns_home found at: EnvVars.masterEnvVars.get("JENKINS_HOME")
2022-04-20 01:54:10.615+0000 [id=1]     INFO    o.e.j.s.handler.ContextHandler#doStart: Started w.@618ad2aa{Jenkins v2.3
32.2,/,file:///var/jenkins_home/war/,AVAILABLE}{/var/jenkins_home/war}
```

After running command we it will install some files then it shows successfully connected.

Running jerkins in localhost 8080 port



Getting Started

Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log (not sure where to find it?) and this file on the server:

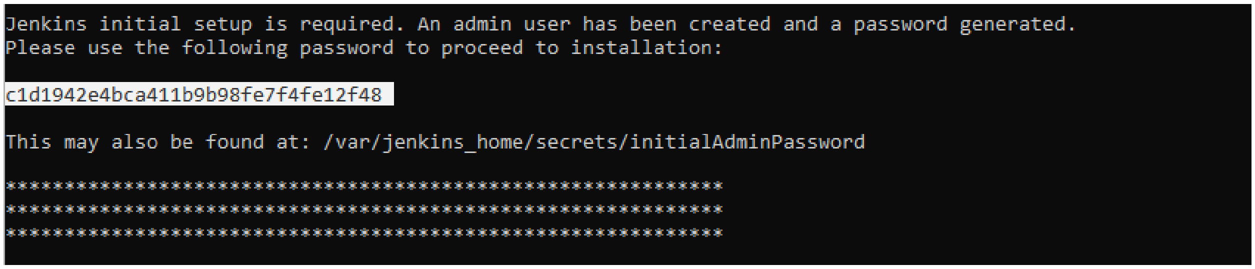
`/var/jenkins_home/secrets/initialAdminPassword`

Please copy the password from either location and paste it below.

Administrator password

We will see some administrative password to login to jerkins.

So in local server it generates random password

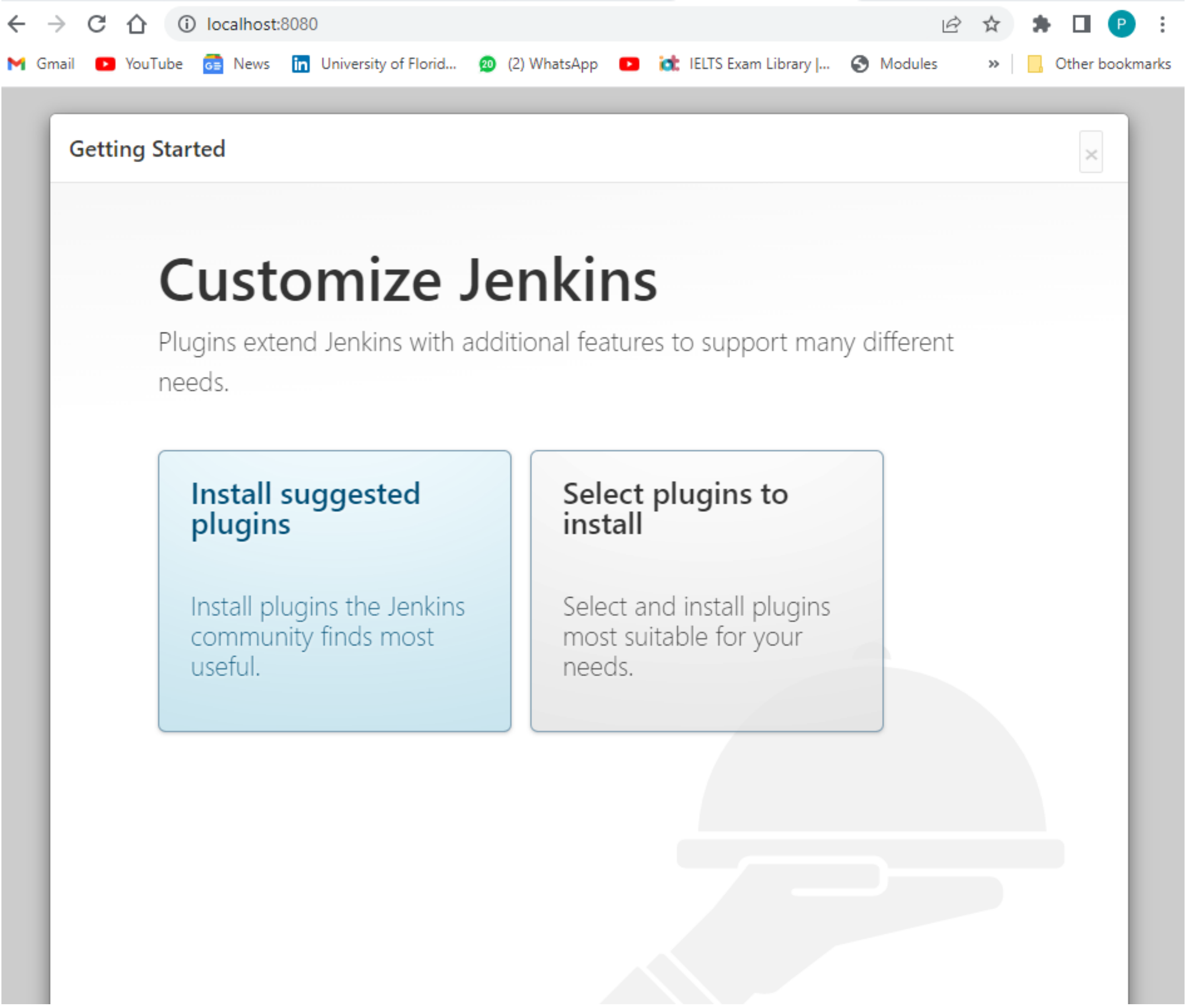


```
Jenkins initial setup is required. An admin user has been created and a password generated.
Please use the following password to proceed to installation:

c1d1942e4bca411b9b98fe7f4fe12f48

This may also be found at: /var/jenkins_home/secrets/initialAdminPassword

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

After login we have install plugins

There will be two options like

Install suggested plugins and select plugins to install



Getting Started

Customize Jenkins

Plugins extend Jenkins with additional features to support many different needs.

Install suggested plugins

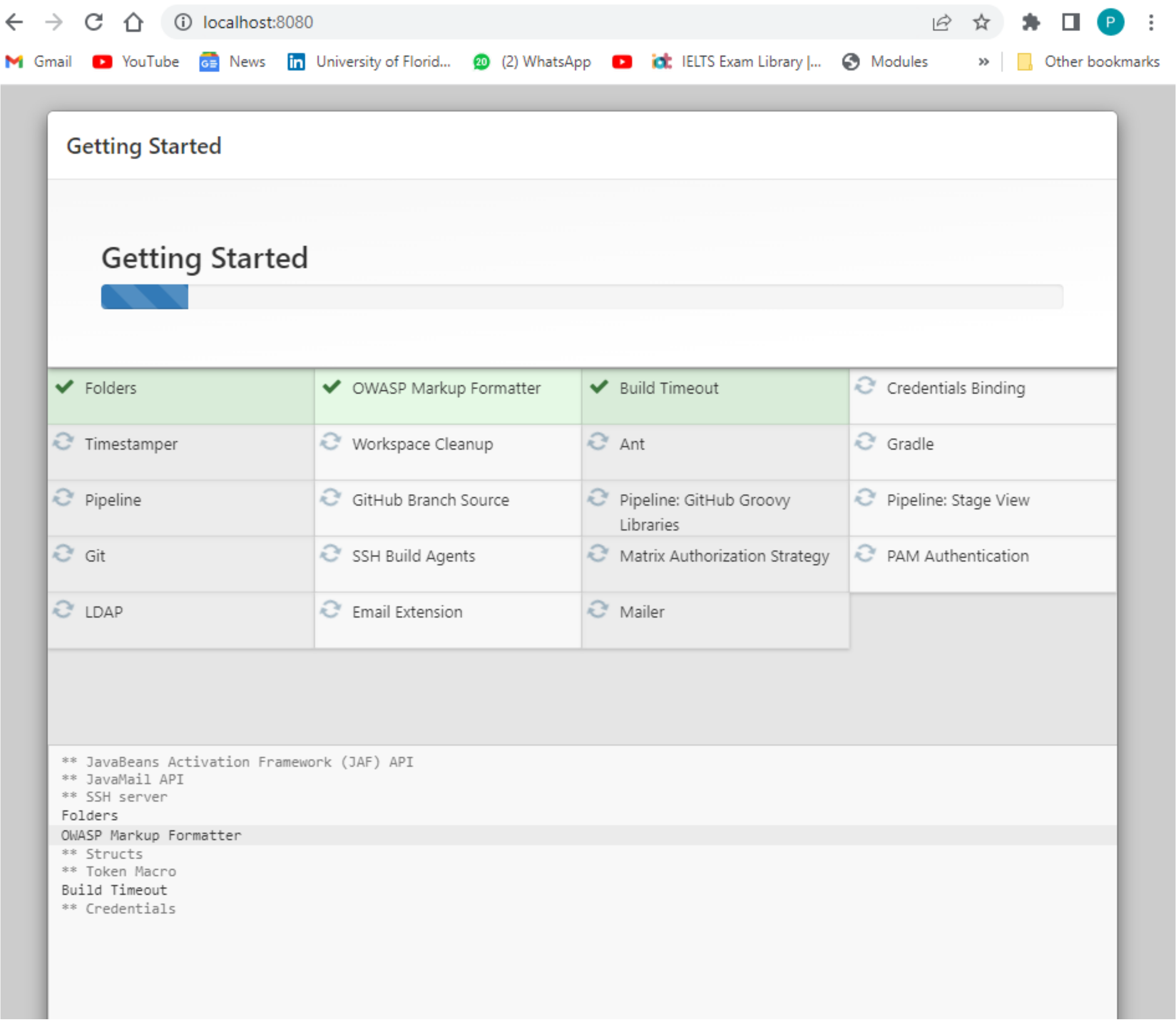
Install plugins the Jenkins community finds most useful.

Select plugins to install

Select and install plugins most suitable for your needs.

So we selected install suggested plugins.

Then it will start and install plugins according to them.



Getting Started

Getting Started

✓ Folders	✓ OWASP Markup Formatter	✓ Build Timeout	Credentials Binding
Timestamper	Workspace Cleanup	Ant	Gradle
Pipeline	GitHub Branch Source	Pipeline: GitHub Groovy Libraries	Pipeline: Stage View
Git	SSH Build Agents	Matrix Authorization Strategy	PAM Authentication
LDAP	Email Extension	Mailer	

```
** JavaBeans Activation Framework (JAF) API
** JavaMail API
** SSH server
Folders
OWASP Markup Formatter
** Structs
** Token Macro
Build Timeout
** Credentials
```

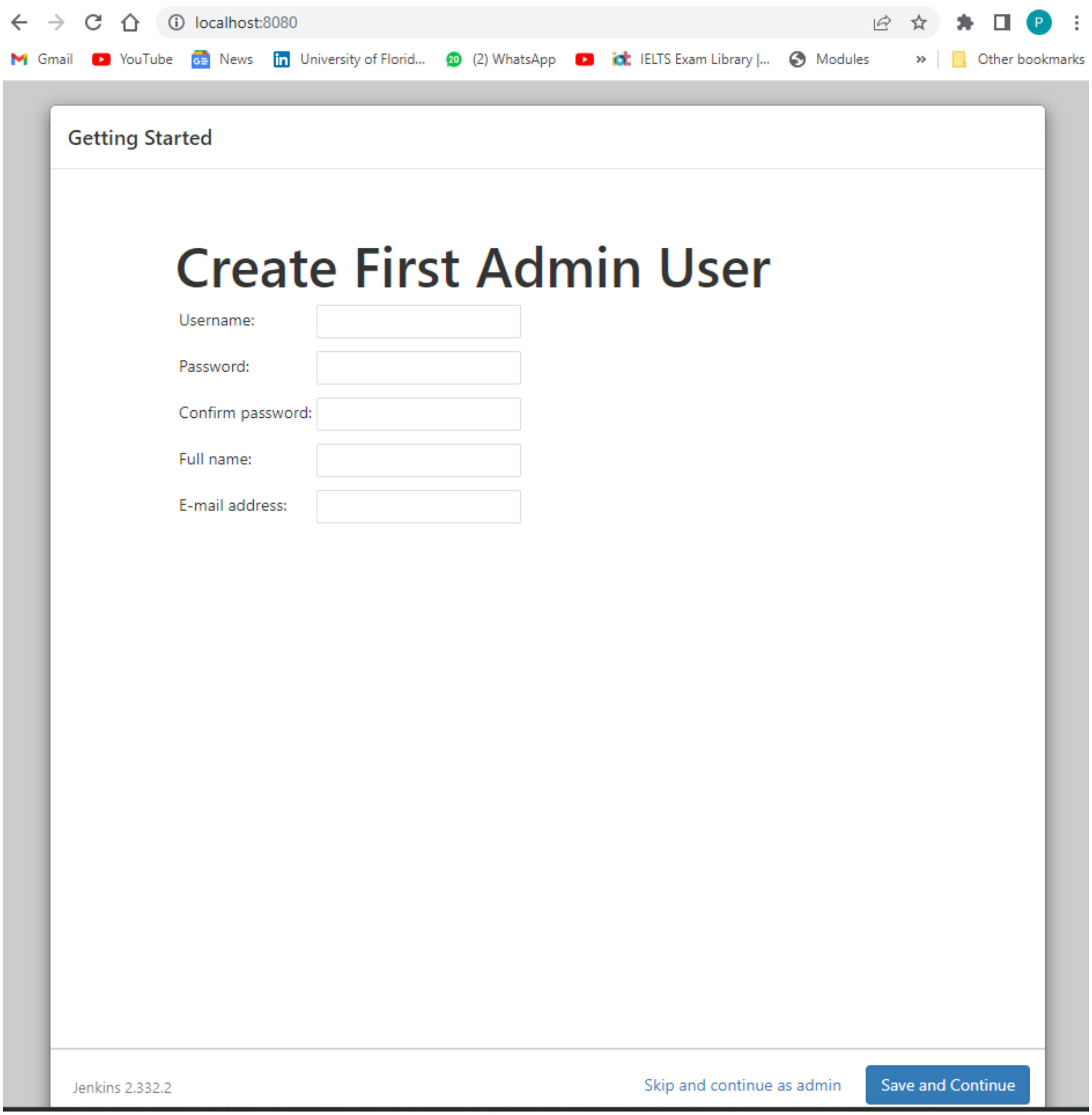
In next step it ask for create admin user

Similarly here we have two option they are

Sign up and skip and continue as admin.

We have select here skip and continue as admin.

Then as save and continue .



Getting Started

Create First Admin User

Username:

Password:

Confirm password:

Full name:

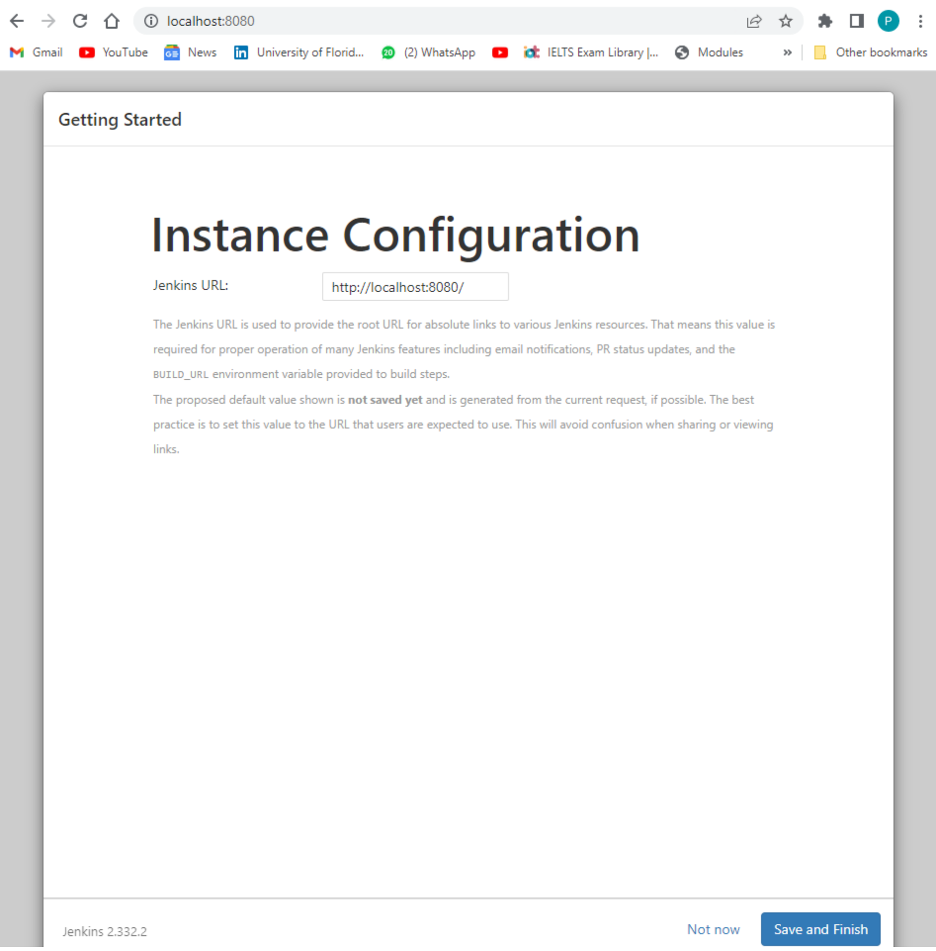
E-mail address:

Jenkins 2.332.2

In this step we can see instance configuration

Localhost:8080

Then save and finish



Instance Configuration

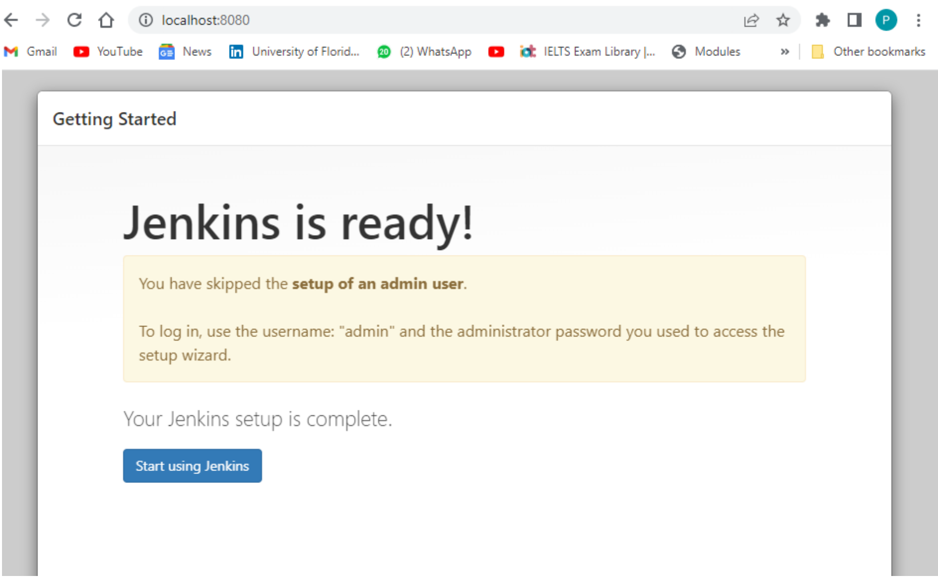
Jenkins URL: http://localhost:8080/

The Jenkins URL is used to provide the root URL for absolute links to various Jenkins resources. That means this value is required for proper operation of many Jenkins features including email notifications, PR status updates, and the `BUILD_URL` environment variable provided to build steps.

The proposed default value shown is **not saved yet** and is generated from the current request, if possible. The best practice is to set this value to the URL that users are expected to use. This will avoid confusion when sharing or viewing links.

In final step we can see Jenkins is ready to use

While clicking on start using Jenkins.



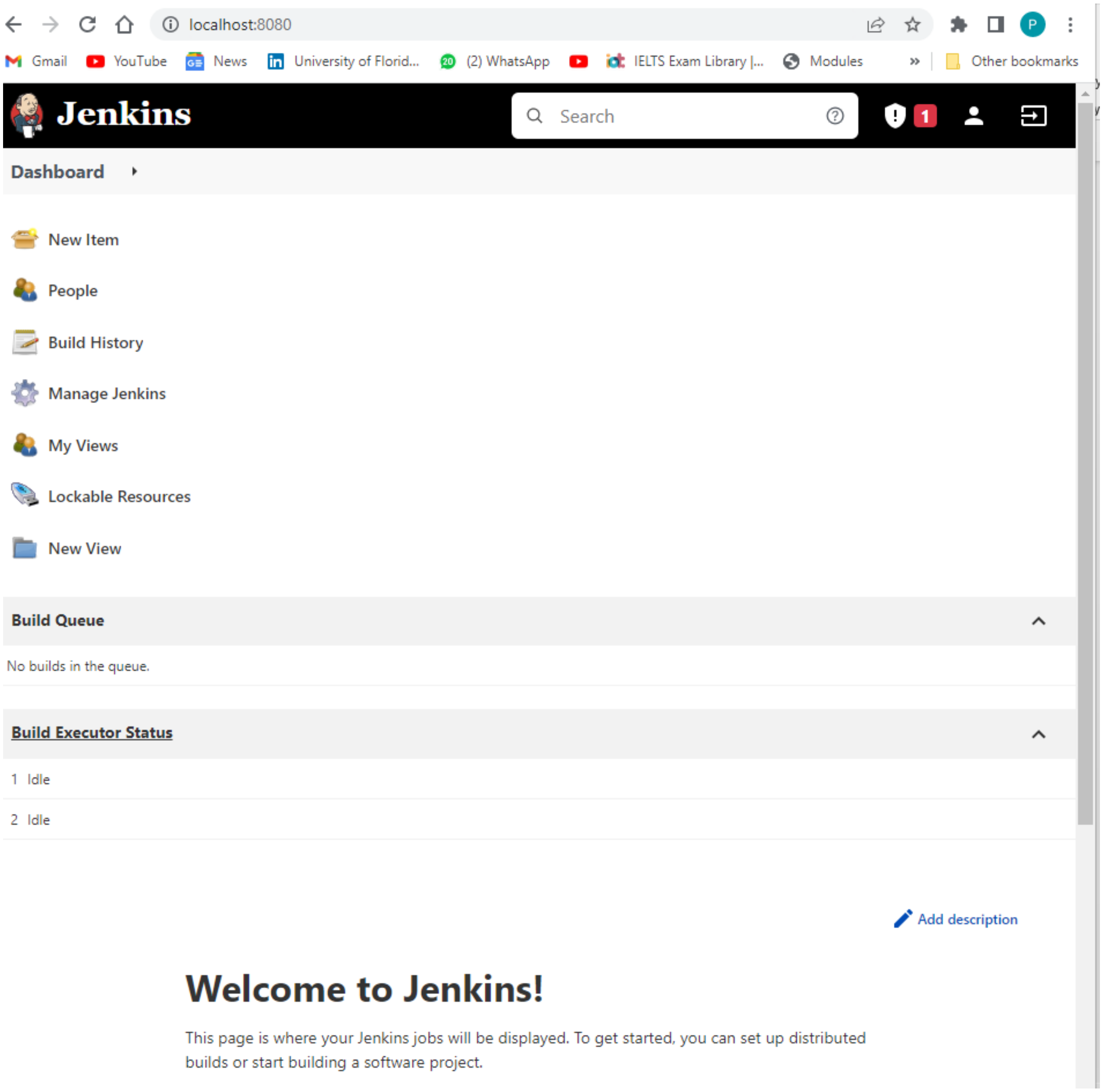
Jenkins is ready!

You have skipped the **setup of an admin user**.

To log in, use the username: "admin" and the administrator password you used to access the setup wizard.

Your Jenkins setup is complete.

We can see welcome to Jenkins



localhost:8080

Build Queue

No builds in the queue.

Build Executor Status

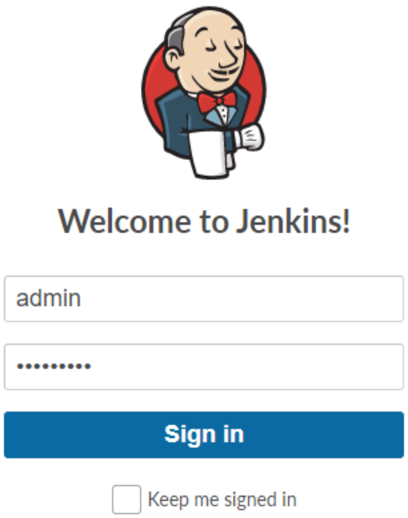
1 Idle

2 Idle

Welcome to Jenkins!

This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

We can see user login page.



Welcome to Jenkins!

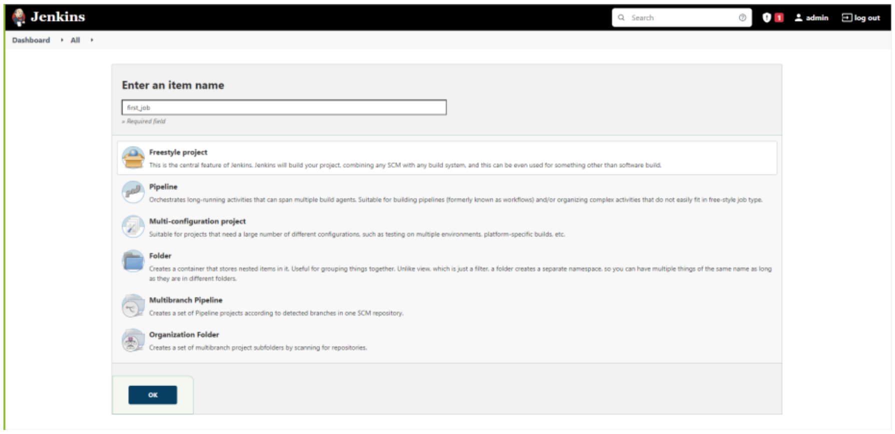
admin

•••••••••

Create project

We have to enter a project name.

Then we have select project type



Enter an item name

first_job

» Required field

Freestyle project
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

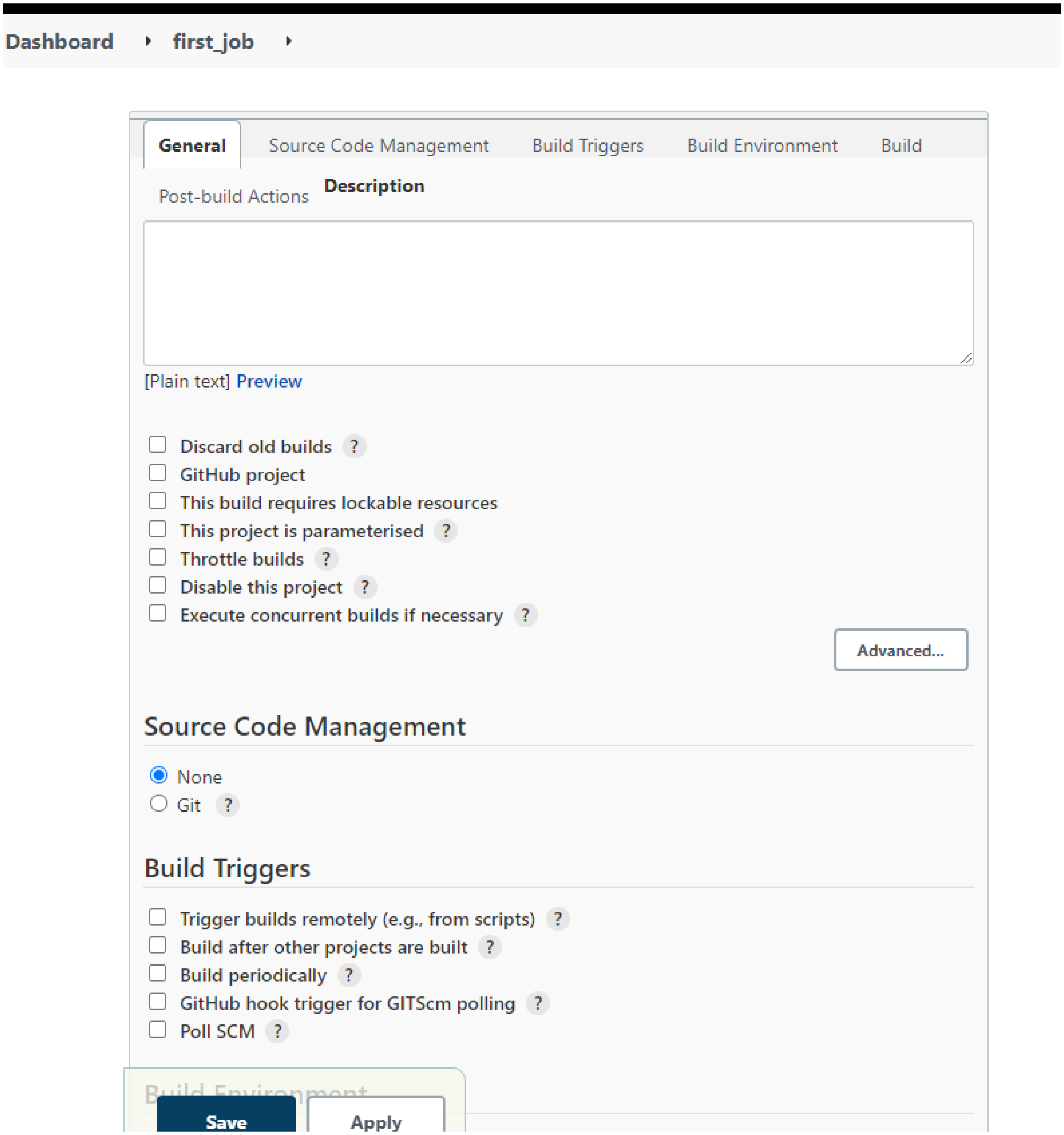
Folder
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

Multibranch Pipeline
Creates a set of Pipeline projects according to detected branches in one SCM repository.

Organization Folder
Creates a set of multibranch project subfolders by scanning for repositories.

We have to write some descriptions

Select source code management



Description

[Plain text] Preview

☐ Discard old builds ?
☐ GitHub project
☐ This build requires lockable resources
☐ This project is parameterised ?
☐ Throttle builds ?
☐ Disable this project ?
☐ Execute concurrent builds if necessary ?

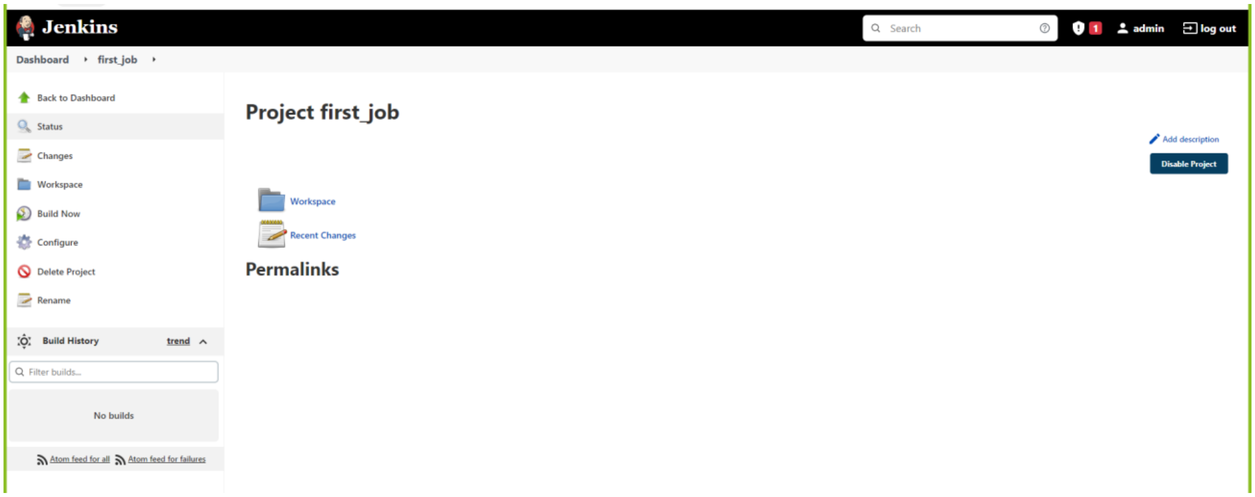
Source Code Management

● None
○ Git ?

Build Triggers

☐ Trigger builds remotely (e.g., from scripts) ?
☐ Build after other projects are built ?
☐ Build periodically ?
☐ GitHub hook trigger for GITScm polling ?
☐ Poll SCM ?

Build Environment

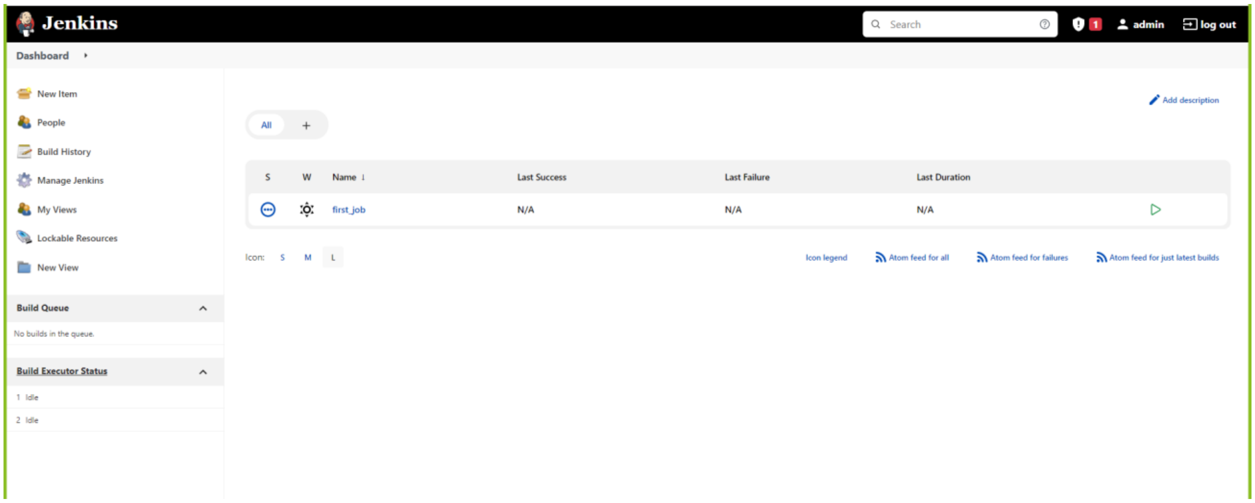


Project first_job

No builds

Dashboard

On dashboard we can see jobs what we created.



No builds in the queue.

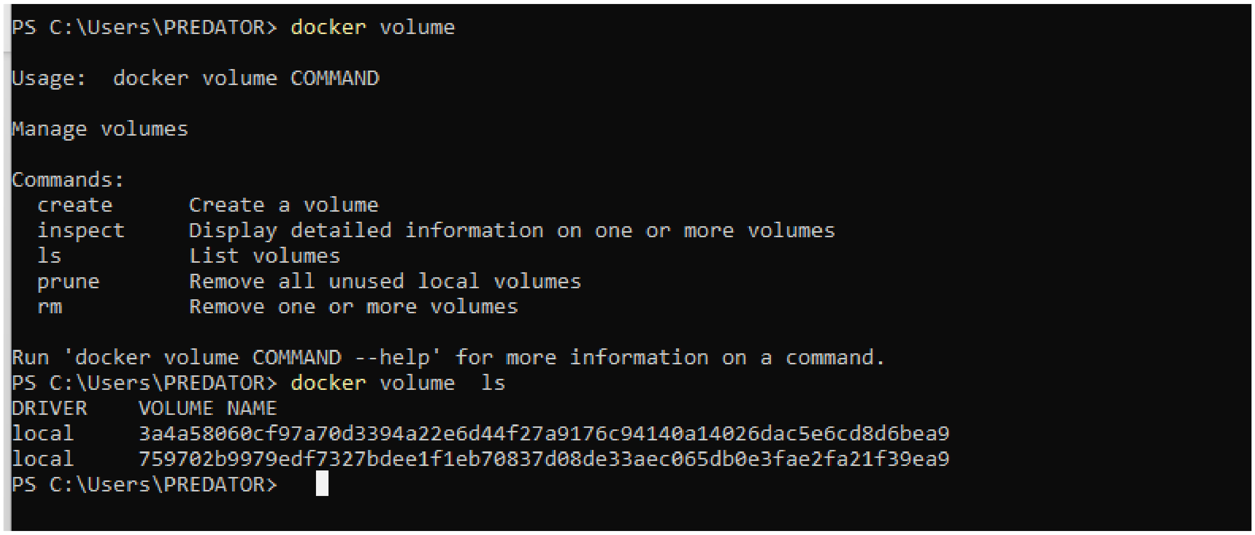
1 Idle

2 Idle

S	W	Name ↓	Last Success	Last Failure	Last Duration
		first_job	N/A	N/A	N/A

To see volume what we created

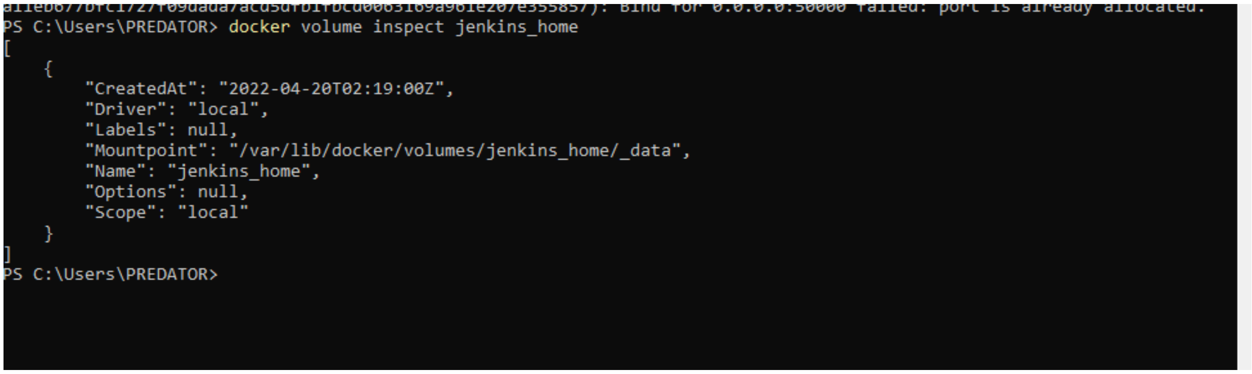
We have to run command docker volume



```
PS C:\Users\PREDATOR> docker volume

Usage:  docker volume COMMAND

Manage volumes

Commands:
  create      Create a volume
  inspect     Display detailed information on one or more volumes
  ls          List volumes
  prune       Remove all unused local volumes
  rm          Remove one or more volumes

Run 'docker volume COMMAND --help' for more information on a command.
PS C:\Users\PREDATOR> docker volume  ls
DRIVER    VOLUME NAME
local     3a4a58060cf97a70d3394a22e6d44f27a9176c94140a14026dac5e6cd8d6bea9
local     759702b9979edf7327bdee1f1eb70837d08de33aec065db0e3fae2fa21f39ea9
PS C:\Users\PREDATOR>
```

To inspect volume jenkins

We use command docker volume inspect jenkins_home



```
a11eb677bfc1727f09dada7acd5dfb1fbcd0063169a961e207e355857): Bind for 0.0.0.0:50000 failed: port is already allocated.
PS C:\Users\PREDATOR> docker volume inspect jenkins_home
[
    {
        "CreatedAt": "2022-04-20T02:19:00Z",
        "Driver": "local",
        "Labels": null,
        "Mountpoint": "/var/lib/docker/volumes/jenkins_home/_data",
        "Name": "jenkins_home",
        "Options": null,
        "Scope": "local"
    }
]
PS C:\Users\PREDATOR>
```

https://hub.docker.com/r/jenkins/jenkins

https://github.com/jenkinsci/docker/blob/master/README.md