Prepare Lab Environment

1. Create and Manage Linux VMs with the Azure CLI 2.0

1.1 Open Azure Cloud Shell

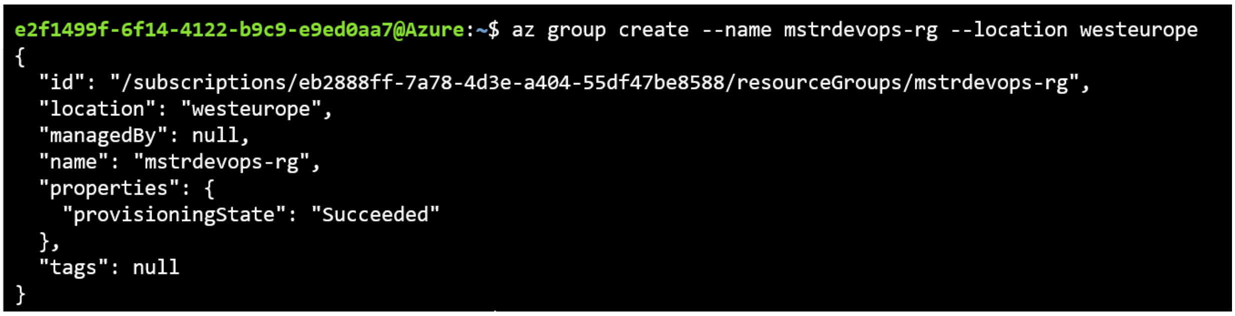
Select **Try It** in the upper-right corner of a code block.	
Open Cloud Shell in your browser.	
Select the **Cloud Shell** button on the menu in the upper-right corner of the Azure portal.	

If you choose to install and use the CLI locally, this tutorial requires that you are running the Azure CLI version 2.0.30 or later. Run az --version to find the version. If you need to install or upgrade, see Install Azure CLI 2.0.

1.2 Create resource group

Create a resource group with the az group create command.

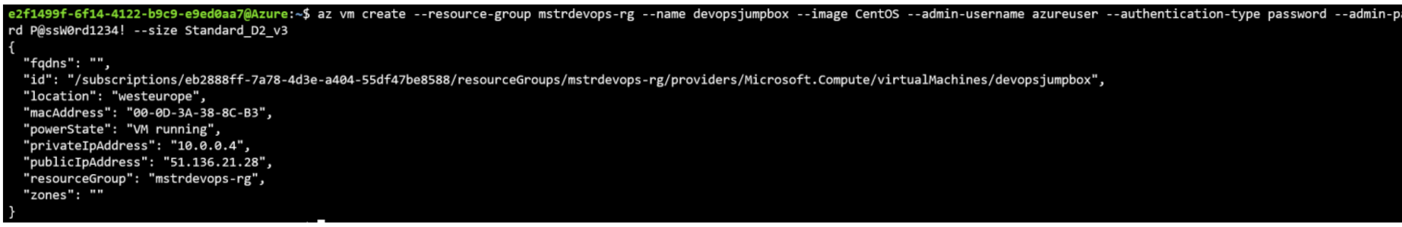
```
az group create --name mstrdevops-rg --location westeurope
```



```
e2f1499f-6f14-4122-b9c9-e9ed0aa7@Azure:~$ az group create --name mstrdevops-rg --location westeurope
{
  "id": "/subscriptions/eb2888ff-7a78-4d3e-a404-55df47be8588/resourceGroups/mstrdevops-rg",
  "location": "westeurope",
  "managedBy": null,
  "name": "mstrdevops-rg",
  "properties": {
    "provisioningState": "Succeeded"
  },
  "tags": null
}
```

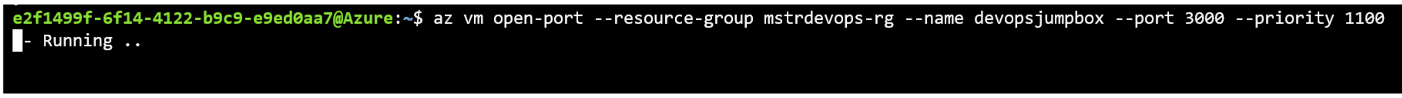
1.3 Create Linux VM and Allow Firewall Ports

```
az vm create --resource-group mstrdevops-rg --name devopsjumpbox --image CentOS --
admin-username azureuser --authentication-type password --admin-password
P@ssW0rd1234! --size Standard_D2_v3
```



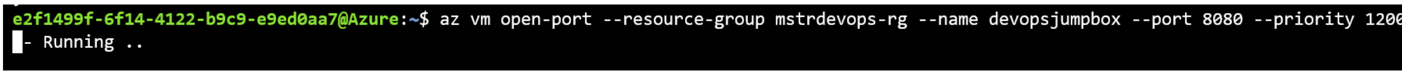
```
e2f1499f-6f14-4122-b9c9-e9ed0aa7@Azure:~$ az vm create --resource-group mstrdevops-rg --name devopsjumpbox --image CentOS --admin-username azureuser --authentication-type password --admin-p
rd P@ssW0rd1234! --size Standard_D2_v3
{
  "fqdns": "",
  "id": "/subscriptions/eb2888ff-7a78-4d3e-a404-55df47be8588/resourceGroups/mstrdevops-rg/providers/Microsoft.Compute/virtualMachines/devopsjumpbox",
  "location": "westeurope",
  "macAddress": "00-0D-3A-38-8C-B3",
  "powerState": "VM running",
  "privateIpAddress": "10.0.0.4",
  "publicIpAddress": "51.136.21.28",
  "resourceGroup": "mstrdevops-rg",
  "zones": ""
}
```

```
az vm open-port --resource-group mstrdevops-rg --name devopsjumpbox --port 3000 --
priority 1100
```



```
e2f1499f-6f14-4122-b9c9-e9ed0aa7@Azure:~$ az vm open-port --resource-group mstrdevops-rg --name devopsjumpbox --port 3000 --priority 1100
█- Running ..
```

```
az vm open-port --resource-group mstrdevops-rg --name devopsjumpbox --port 8080 --
priority 1200
```



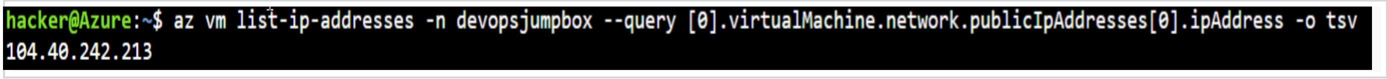
```
e2f1499f-6f14-4122-b9c9-e9ed0aa7@Azure:~$ az vm open-port --resource-group mstrdevops-rg --name devopsjumpbox --port 8080 --priority 1200
█- Running ..
```

```
az vm open-port --resource-group mstrdevops-rg --name devopsjumpbox --port 8081 --
priority 1300
```

1.4 Connect to VM

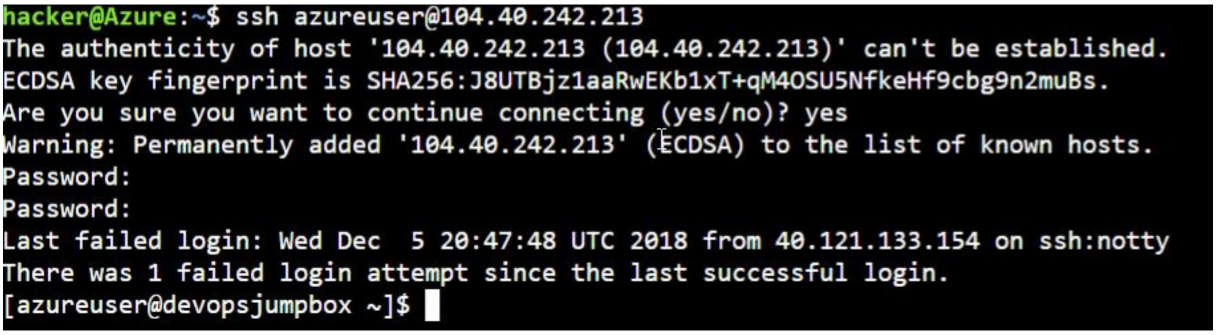
You can now connect to the VM with SSH from your local computer. Replace the example IP address with the `publicIpAddress`. To get public IP Address run the command.

```
az vm list-ip-addresses -n devopsjumpbox --
query[0].virtualMachine.network.publicIpAddresses[0].ipAddress -o tsv
```



```
hacker@Azure:~$ az vm list-ip-addresses -n devopsjumpbox --query [0].virtualMachine.network.publicIpAddresses[0].ipAddress -o tsv
104.40.242.213
```

```
ssh azureuser@104.40.242.213
```

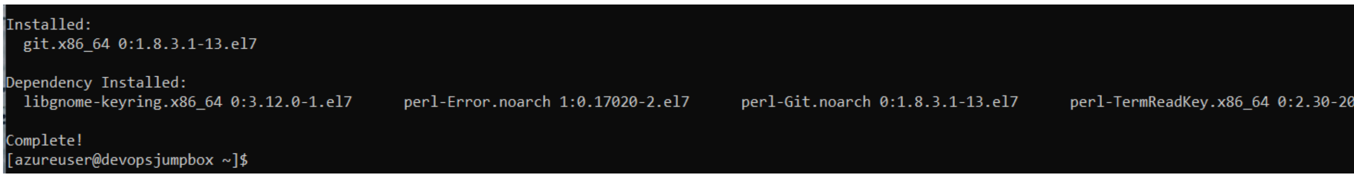


```
hacker@Azure:~$ ssh azureuser@104.40.242.213
The authenticity of host '104.40.242.213 (104.40.242.213)' can't be established.
ECDSA key fingerprint is SHA256:J8UTBjz1aaRwEKb1xT+qM4OSU5NfkeHf9cbg9n2muBs.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '104.40.242.213' (ECDSA) to the list of known hosts.
Password:
Password:
Last failed login: Wed Dec  5 20:47:48 UTC 2018 from 40.121.133.154 on ssh:notty
There was 1 failed login attempt since the last successful login.
[azureuser@devopsjumpbox ~]$
```

2. Initial Setup and Running App on Local Machine

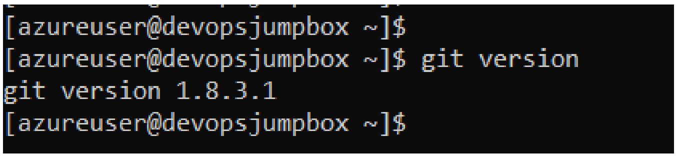
2.1 Install Git on Linux VM (Jumpbox)

```
sudo yum install git
```



```
Installed:
  git.x86_64 0:1.8.3.1-13.el7

Dependency Installed:
  libgnome-keyring.x86_64 0:3.12.0-1.el7      perl-Error.noarch 1:0.17020-2.el7      perl-Git.noarch 0:1.8.3.1-13.el7      perl-TermReadKey.x86_64 0:2.30-20

Complete!
[azureuser@devopsjumpbox ~]$
```

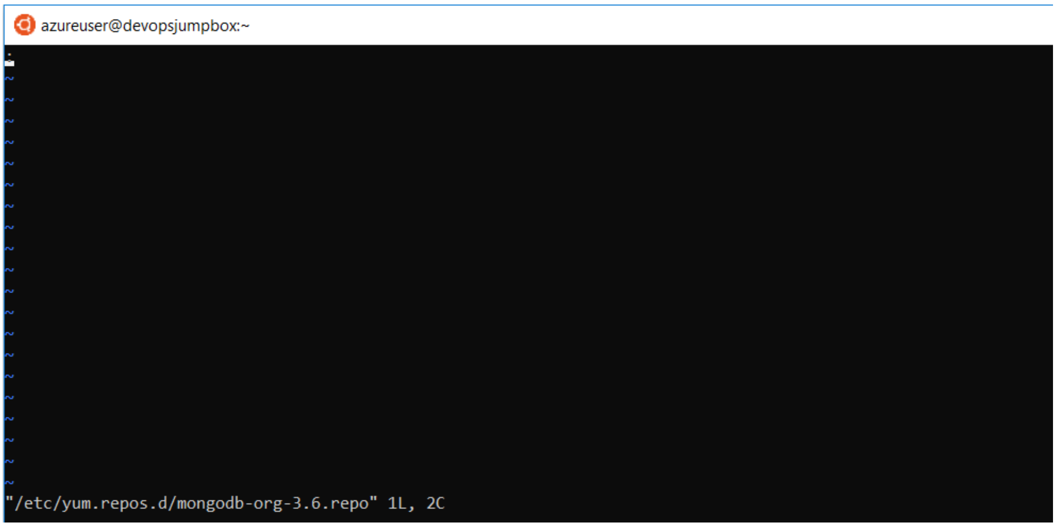
```
git version
```



```
[azureuser@devopsjumpbox ~]$
[azureuser@devopsjumpbox ~]$ git version
git version 1.8.3.1
[azureuser@devopsjumpbox ~]$
```

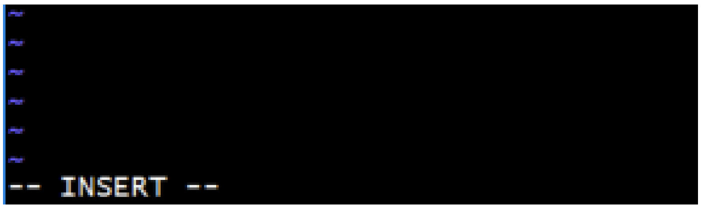
2.2 Install MongoDB on Jumpbox

```
sudo vi /etc/yum.repos.d/mongodb-org-3.6.repo
```



```
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"/etc/yum.repos.d/mongodb-org-3.6.repo" 1L, 2C
```

Select “i” on your keyboard. You’ll see the bottom of the window showing INSERT mode.



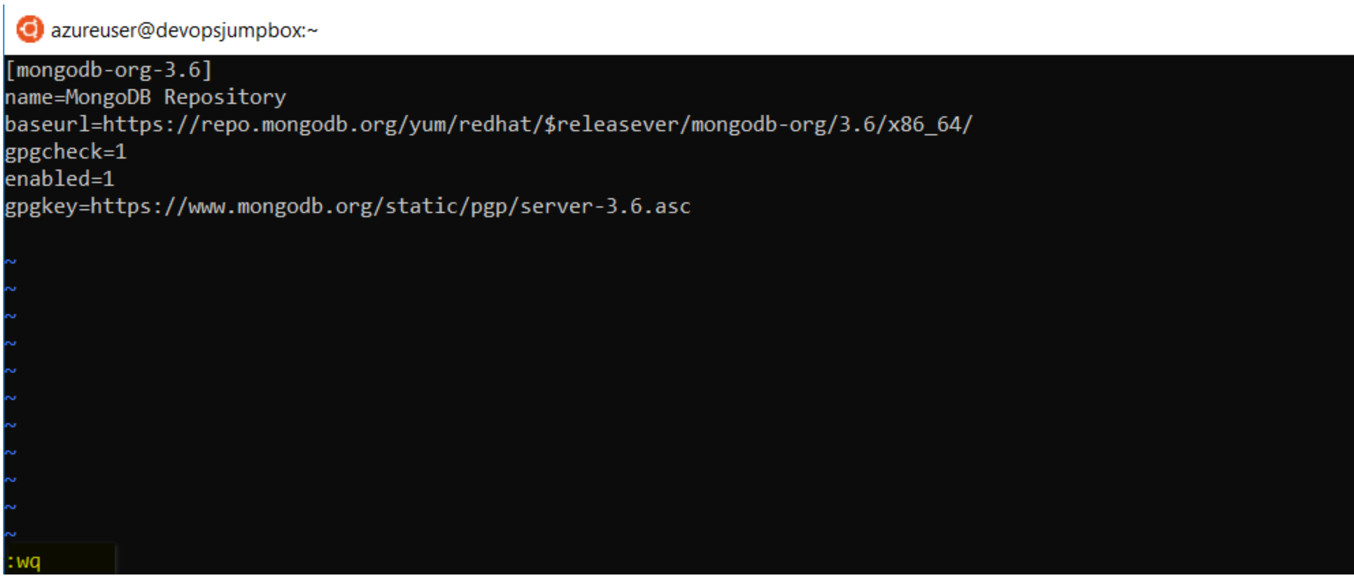
```
~
~
~
~
~
~
-- INSERT --
```

NOTE: Type the following into the editor, as you may have errors with copying and pasting.

```
[mongodb-org-3.6]
name=MongoDB Repository
baseurl=https://repo.mongodb.org/yum/redhat/$releasever/mongodb-org/3.6/x86_64/
gpgcheck=1
enabled=1
gpgkey=https://www.mongodb.org/static/pgp/server-3.6.asc
```

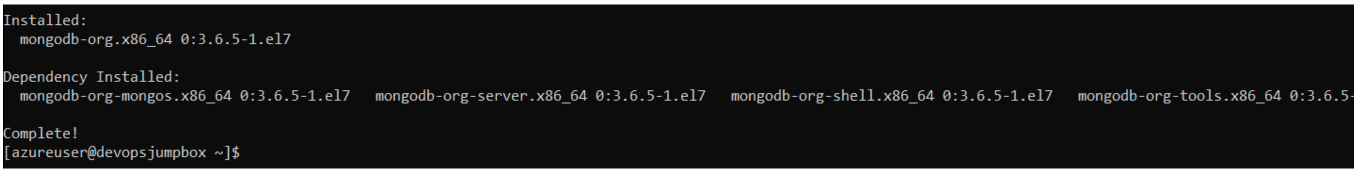
When you are finished typing, hit the Esc key and type ":wq" and hit the Enter key to save the changes and close the file.

```
<Esc>
:wq!
<Enter>
```



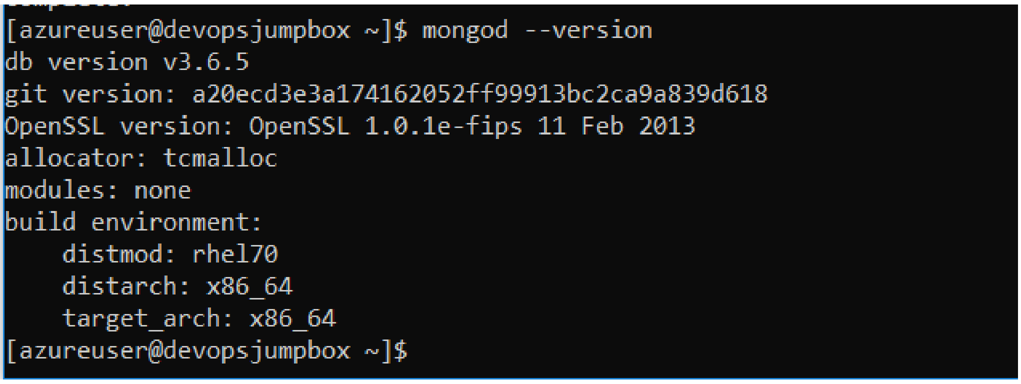
```
[mongodb-org-3.6]
name=MongoDB Repository
baseurl=https://repo.mongodb.org/yum/redhat/$releasever/mongodb-org/3.6/x86_64/
gpgcheck=1
enabled=1
gpgkey=https://www.mongodb.org/static/pgp/server-3.6.asc

~
~
~
~
~
~
~
~
~
~
~
:wq
```

```
sudo yum install -y mongodb-org
```



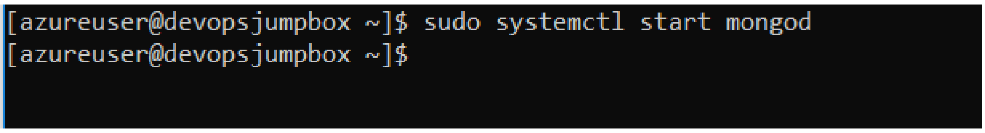
```
Installed:
  mongodb-org.x86_64 0:3.6.5-1.el7

Dependency Installed:
  mongodb-org-mongos.x86_64 0:3.6.5-1.el7   mongodb-org-server.x86_64 0:3.6.5-1.el7   mongodb-org-shell.x86_64 0:3.6.5-1.el7   mongodb-org-tools.x86_64 0:3.6.5-

Complete!
[azureuser@devopsjumpbox ~]$
```

```
mongod -version
```



```
[azureuser@devopsjumpbox ~]$ mongod --version
db version v3.6.5
git version: a20ecd3e3a174162052ff99913bc2ca9a839d618
OpenSSL version: OpenSSL 1.0.1e-fips 11 Feb 2013
allocator: tcmalloc
modules: none
build environment:
    distmod: rhel70
    distarch: x86_64
    target_arch: x86_64
[azureuser@devopsjumpbox ~]$
```

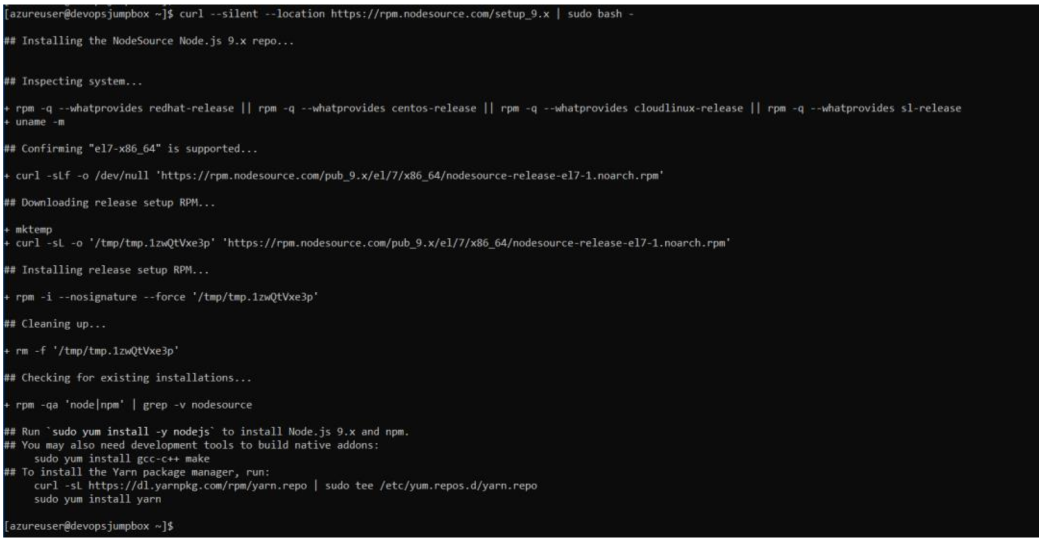
```
sudo systemctl start mongod
```



```
[azureuser@devopsjumpbox ~]$ sudo systemctl start mongod
[azureuser@devopsjumpbox ~]$
```

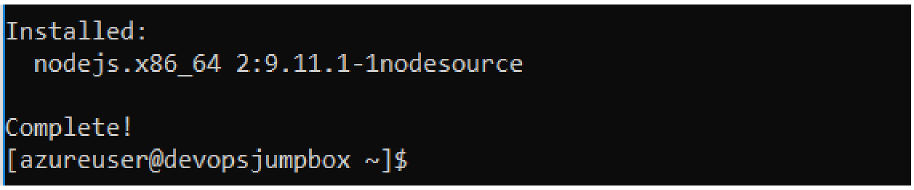
2.3 Install NodeJS on Jumpbox

```
curl --silent --location https://rpm.nodesource.com/setup_9.x | sudo bash -
```



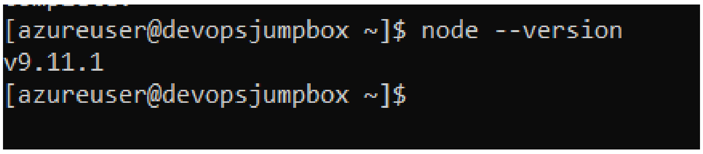
```
[azureuser@devopsjumpbox ~]$ curl --silent --location https://rpm.nodesource.com/setup_9.x | sudo bash -

## Installing the NodeSource Node.js 9.x repo...


## Inspecting system...

+ rpm -q --whatprovides redhat-release || rpm -q --whatprovides centos-release || rpm -q --whatprovides cloudlinux-release || rpm -q --whatprovides sl-release
+ uname -m

## Confirming "el7-x86_64" is supported...

+ curl -sLf -o /dev/null 'https://rpm.nodesource.com/pub_9.x/el/7/x86_64/nodesource-release-el7-1.noarch.rpm'

## Downloading release setup RPM...

+ mktemp
+ curl -sL -o '/tmp/tmp.1zwQtVxe3p' 'https://rpm.nodesource.com/pub_9.x/el/7/x86_64/nodesource-release-el7-1.noarch.rpm'

## Installing release setup RPM...

+ rpm -i --nosignature --force '/tmp/tmp.1zwQtVxe3p'

## Cleaning up...

+ rm -f '/tmp/tmp.1zwQtVxe3p'

## Checking for existing installations...

+ rpm -qa 'node|npm' | grep -v nodesource

## Run `sudo yum install -y nodejs` to install Node.js 9.x and npm.
## You may also need development tools to build native addons:
     sudo yum install gcc-c++ make
## To install the Yarn package manager, run:
     curl -sL https://dl.yarnpkg.com/rpm/yarn.repo | sudo tee /etc/yum.repos.d/yarn.repo
     sudo yum install yarn

[azureuser@devopsjumpbox ~]$
```

```
sudo yum -y install nodejs
```



```
Installed:
  nodejs.x86_64 2:9.11.1-1nodesource

Complete!
[azureuser@devopsjumpbox ~]$
```

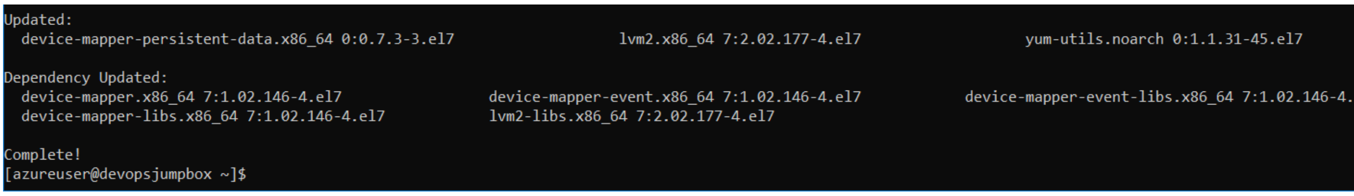
```
node --version
```



```
[azureuser@devopsjumpbox ~]$ node --version
v9.11.1
[azureuser@devopsjumpbox ~]$
```

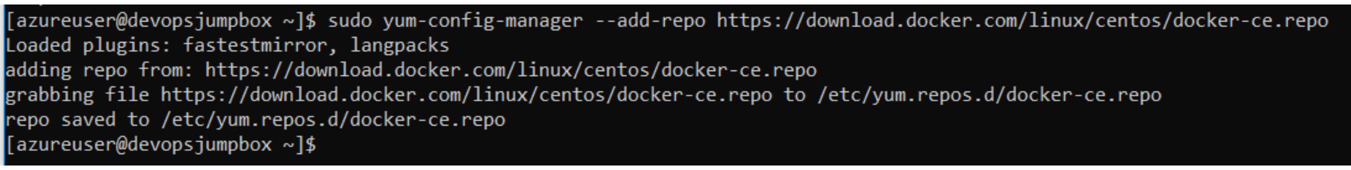
2.4 Install DockerCE on Jumpbox

```
sudo yum install -y yum-utils device-mapper-persistent-data lvm2
```



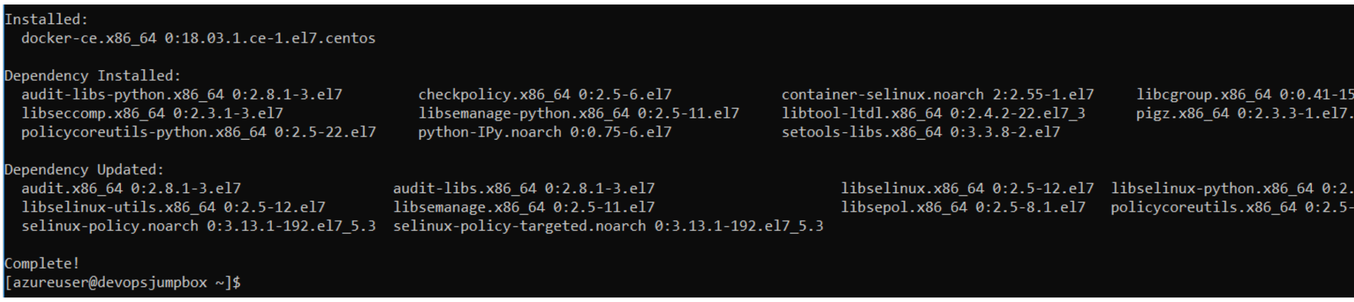
```
Updated:
  device-mapper-persistent-data.x86_64 0:0.7.3-3.el7                   lvm2.x86_64 7:2.02.177-4.el7                   yum-utils.noarch 0:1.1.31-45.el7

Dependency Updated:
  device-mapper.x86_64 7:1.02.146-4.el7                device-mapper-event.x86_64 7:1.02.146-4.el7           device-mapper-event-libs.x86_64 7:1.02.146-4.
  device-mapper-libs.x86_64 7:1.02.146-4.el7           lvm2-libs.x86_64 7:2.02.177-4.el7

Complete!
[azureuser@devopsjumpbox ~]$
```

```
sudo yum-config-manager --add-repo https://download.docker.com/linux/centos/docker-ce.repo
```



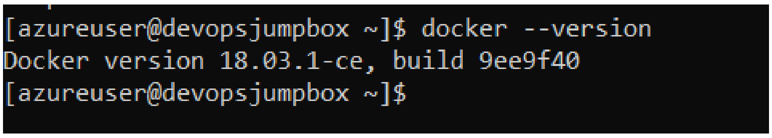
```
[azureuser@devopsjumpbox ~]$ sudo yum-config-manager --add-repo https://download.docker.com/linux/centos/docker-ce.repo
Loaded plugins: fastestmirror, langpacks
adding repo from: https://download.docker.com/linux/centos/docker-ce.repo
grabbing file https://download.docker.com/linux/centos/docker-ce.repo to /etc/yum.repos.d/docker-ce.repo
repo saved to /etc/yum.repos.d/docker-ce.repo
[azureuser@devopsjumpbox ~]$
```

```
sudo yum install docker-ce
```



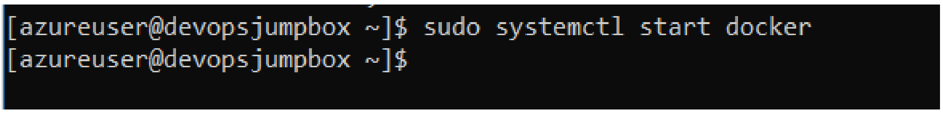
```
Installed:
  docker-ce.x86_64 0:18.03.1.ce-1.el7.centos

Dependency Installed:
  audit-libs-python.x86_64 0:2.8.1-3.el7         checkpolicy.x86_64 0:2.5-6.el7            container-selinux.noarch 2:2.55-1.el7     libcgroup.x86_64 0:0.41-15
  libseccomp.x86_64 0:2.3.1-3.el7                libsemanage-python.x86_64 0:2.5-11.el7    libtool-ltdl.x86_64 0:2.4.2-22.el7_3      pigz.x86_64 0:2.3.3-1.el7.
  policycoreutils-python.x86_64 0:2.5-22.el7     python-IPy.noarch 0:0.75-6.el7            setools-libs.x86_64 0:3.3.8-2.el7

Dependency Updated:
  audit.x86_64 0:2.8.1-3.el7                  audit-libs.x86_64 0:2.8.1-3.el7                          libselinux.x86_64 0:2.5-12.el7  libselinux-python.x86_64 0:2.
  libselinux-utils.x86_64 0:2.5-12.el7        libsemanage.x86_64 0:2.5-11.el7                          libsepol.x86_64 0:2.5-8.1.el7   policycoreutils.x86_64 0:2.5-
  selinux-policy.noarch 0:3.13.1-192.el7_5.3  selinux-policy-targeted.noarch 0:3.13.1-192.el7_5.3

Complete!
[azureuser@devopsjumpbox ~]$
```

```
docker --version
```



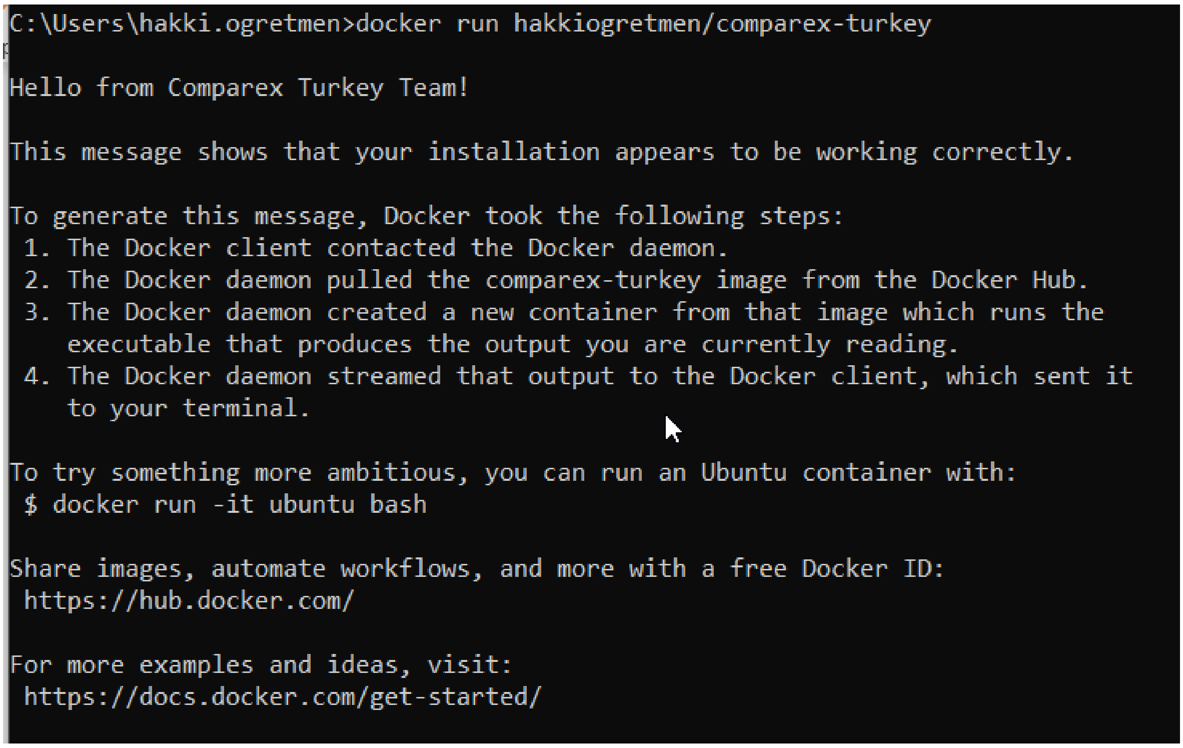
```
[azureuser@devopsjumpbox ~]$ docker --version
Docker version 18.03.1-ce, build 9ee9f40
[azureuser@devopsjumpbox ~]$
```

```
sudo systemctl start docker
```



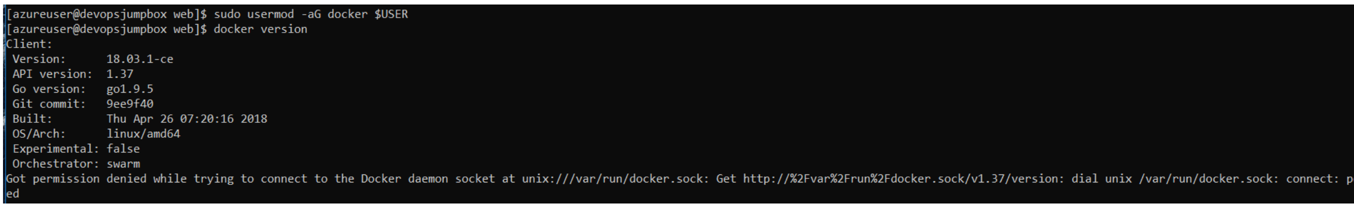
```
[azureuser@devopsjumpbox ~]$ sudo systemctl start docker
[azureuser@devopsjumpbox ~]$
```

```
sudo docker run hakkiogretmen/comparex-turkey
```



```
C:\Users\hakki.ogretmen>docker run hakkiogretmen/comparex-turkey

Hello from Comparex Turkey Team!

This message shows that your installation appears to be working correctly.

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

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

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

For more examples and ideas, visit:
 https://docs.docker.com/get-started/
```

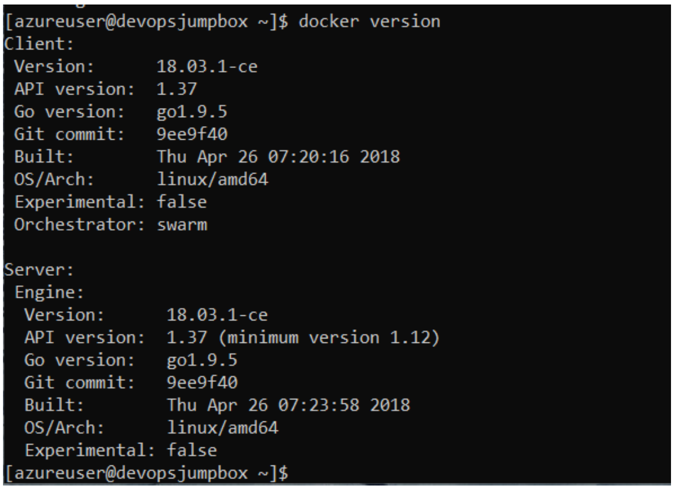
```
sudo usermod -aG docker $USER
```



```
[azureuser@devopsjumpbox web]$ sudo usermod -aG docker $USER
[azureuser@devopsjumpbox web]$ docker version
Client:
 Version:      18.03.1-ce
 API version:  1.37
 Go version:   go1.9.5
 Git commit:   9ee9f40
 Built:        Thu Apr 26 07:20:16 2018
 OS/Arch:      linux/amd64
 Experimental: false
 Orchestrator: swarm
Got permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Get http://%2Fvar%2Frun%2Fdocker.sock/v1.37/version: dial unix /var/run/docker.sock: connect: p
ed
```

In order for the user permission changes to take effect, exit the SSH session by typing ‘exit’, then press <Enter>. Repeat connect from establish the SSH session

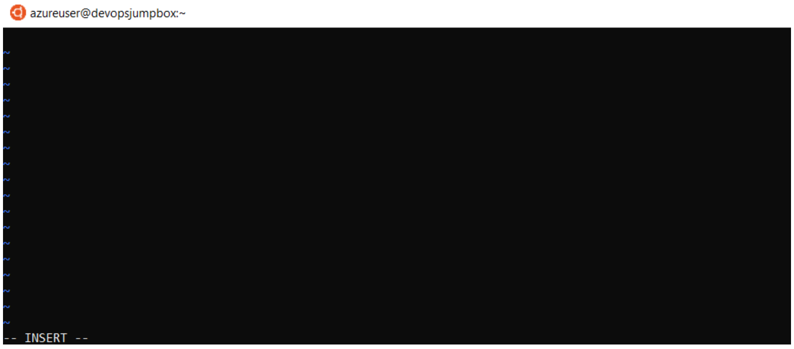
```
ssh azureuser@104.40.242.213
docker version
```



```
[azureuser@devopsjumpbox ~]$ docker version
Client:
 Version:      18.03.1-ce
 API version:  1.37
 Go version:   go1.9.5
 Git commit:   9ee9f40
 Built:        Thu Apr 26 07:20:16 2018
 OS/Arch:      linux/amd64
 Experimental: false
 Orchestrator: swarm

Server:
 Engine:
  Version:      18.03.1-ce
  API version:  1.37 (minimum version 1.12)
  Go version:   go1.9.5
  Git commit:   9ee9f40
  Built:        Thu Apr 26 07:23:58 2018
  OS/Arch:      linux/amd64
  Experimental: false
[azureuser@devopsjumpbox ~]$
```

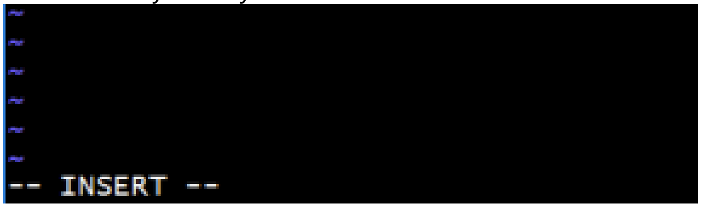
2.5 Install Kubectl on Jumpbox

```
sudo vi /etc/yum.repos.d/kubernetes.repo
```



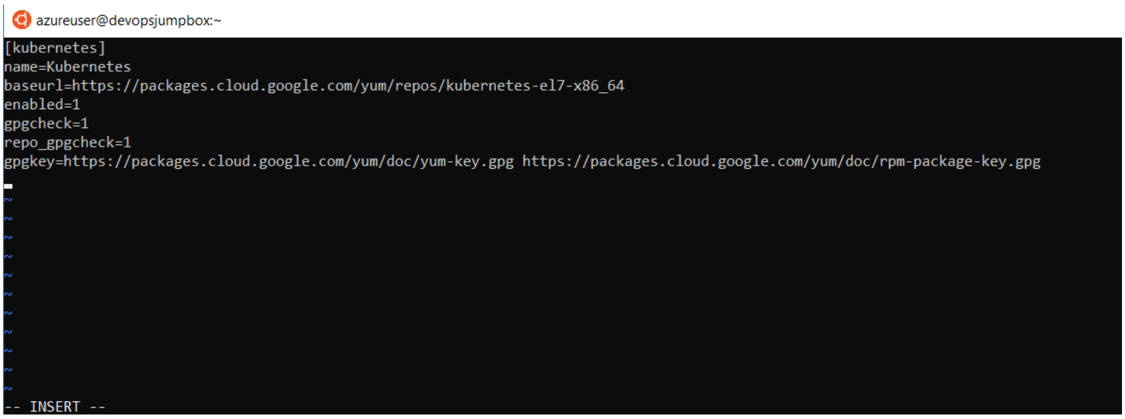
```
azureuser@devopsjumpbox:~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
-- INSERT --
```

Select "i" on your keyboard. You'll see the bottom of the window showing INSERT mode.



```
~
~
~
~
~
~
-- INSERT --
```

```
[kubernetes]
name=Kubernetes
baseurl=https://packages.cloud.google.com/yum/repos/kubernetes-el7-x86_64
enabled=1
gpgcheck=1
repo_gpgcheck=1
gpgkey=https://packages.cloud.google.com/yum/doc/yum-key.gpg
https://packages.cloud.google.com/yum/doc/rpm-package-key.gpg
```

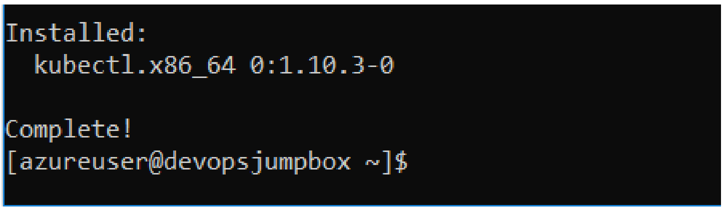


```
azureuser@devopsjumpbox:~
[kubernetes]
name=Kubernetes
baseurl=https://packages.cloud.google.com/yum/repos/kubernetes-el7-x86_64
enabled=1
gpgcheck=1
repo_gpgcheck=1
gpgkey=https://packages.cloud.google.com/yum/doc/yum-key.gpg https://packages.cloud.google.com/yum/doc/rpm-package-key.gpg

~
~
~
~
~
~
~
~
~
~
~
-- INSERT --
```

When you are finished typing, hit the Esc key and type ":wq!" and hit the Enter key to save the changes and close the file.

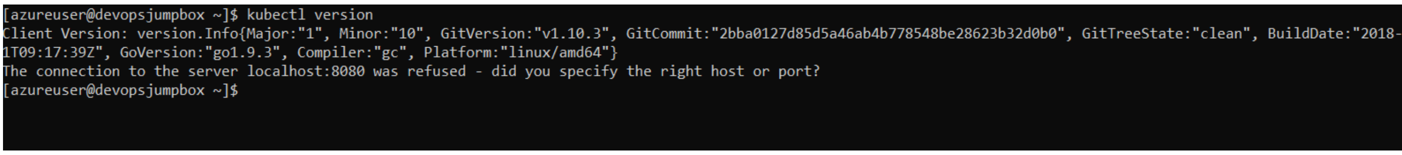
```
<Esc>
:wq!
<Enter>
```

```
sudo yum install kubectl
```



```
Installed:
  kubectl.x86_64 0:1.10.3-0

Complete!
[azureuser@devopsjumpbox ~]$
```

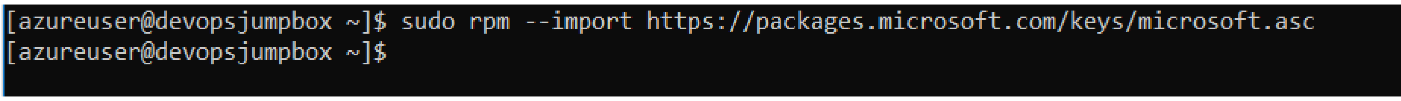
```
kubectl version
```



```
[azureuser@devopsjumpbox ~]$ kubectl version
Client Version: version.Info{Major:"1", Minor:"10", GitVersion:"v1.10.3", GitCommit:"2bba0127d85d5a46ab4b778548be28623b32d0b0", GitTreeState:"clean", BuildDate:"2018-
1T09:17:39Z", GoVersion:"go1.9.3", Compiler:"gc", Platform:"linux/amd64"}
The connection to the server localhost:8080 was refused - did you specify the right host or port?
[azureuser@devopsjumpbox ~]$
```

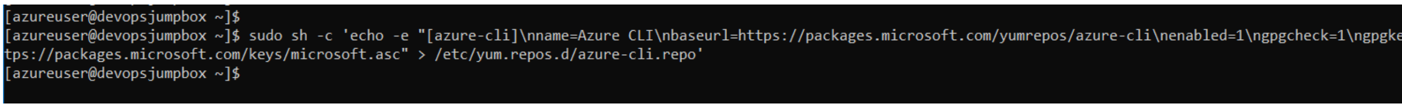
2.6 Install AzureCLI on Jumpbox

```
sudo rpm --import https://packages.microsoft.com/keys/microsoft.asc
```



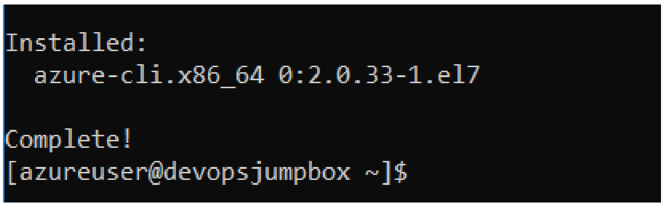
```
[azureuser@devopsjumpbox ~]$ sudo rpm --import https://packages.microsoft.com/keys/microsoft.asc
[azureuser@devopsjumpbox ~]$
```

```
sudo sh -c 'echo -e "[azure-cli]\nname=Azure
CLI\nbaseurl=https://packages.microsoft.com/yumrepos/azure-
cli\nenabled=1\ngpgcheck=1\ngpgkey=https://packages.microsoft.com/keys/microsoft.a
sc" > /etc/yum.repos.d/azure-cli.repo'
```



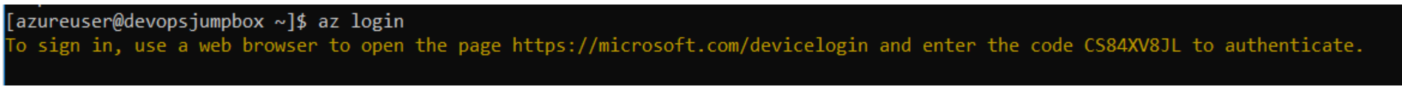
```
[azureuser@devopsjumpbox ~]$
[azureuser@devopsjumpbox ~]$ sudo sh -c 'echo -e "[azure-cli]\nname=Azure CLI\nbaseurl=https://packages.microsoft.com/yumrepos/azure-cli\nenabled=1\ngpgcheck=1\ngpgke
tps://packages.microsoft.com/keys/microsoft.asc" > /etc/yum.repos.d/azure-cli.repo'
[azureuser@devopsjumpbox ~]$
```

```
sudo yum install -y azure-cli
```

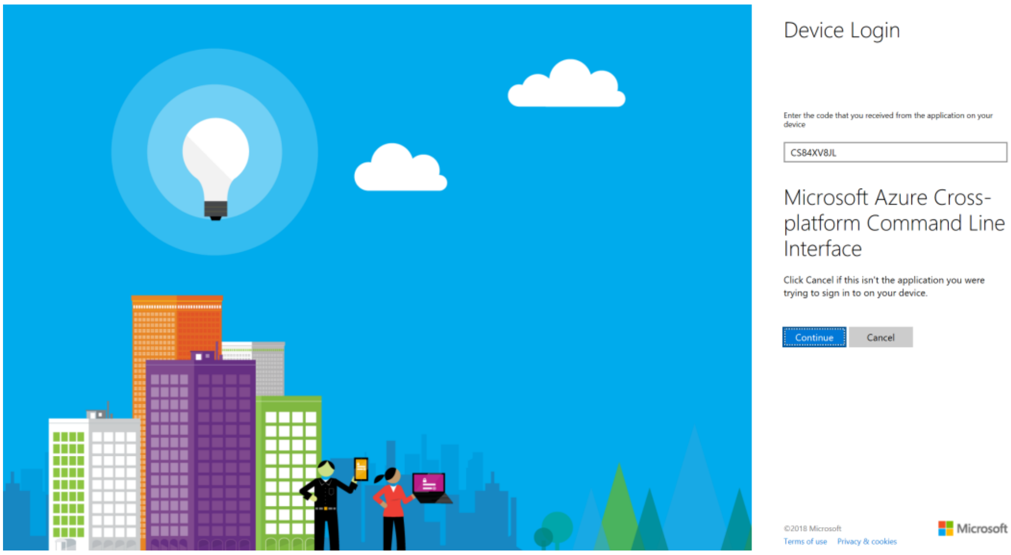


```
Installed:
  azure-cli.x86_64 0:2.0.33-1.el7

Complete!
[azureuser@devopsjumpbox ~]$
```

```
      az login
```



```
[azureuser@devopsjumpbox ~]$ az login
To sign in, use a web browser to open the page https://microsoft.com/devicelogin and enter the code CS84XV8JL to authenticate.
```



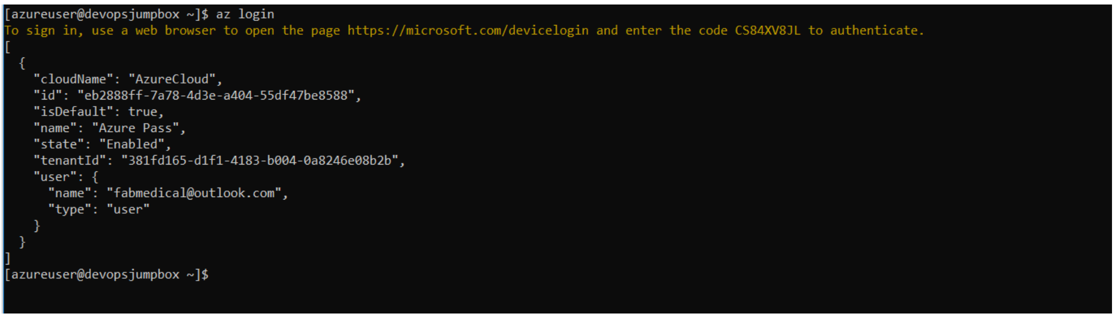
Device Login

Enter the code that you received from the application on your device

CS84XV8JL

Microsoft Azure Cross-platform Command Line Interface

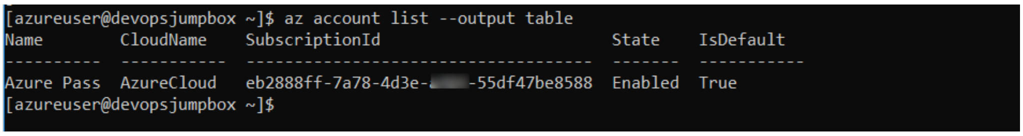
Click Cancel if this isn't the application you were trying to sign in to on your device.



```
[azureuser@devopsjumpbox ~]$ az login
To sign in, use a web browser to open the page https://microsoft.com/devicelogin and enter the code CS84XV8JL to authenticate.
[
  {
    "cloudName": "AzureCloud",
    "id": "eb2888ff-7a78-4d3e-a404-55df47be8588",
    "isDefault": true,
    "name": "Azure Pass",
    "state": "Enabled",
    "tenantId": "381fd165-d1f1-4183-b004-0a8246e08b2b",
    "user": {
      "name": "fabmedical@outlook.com",
      "type": "user"
    }
  }
]
[azureuser@devopsjumpbox ~]$
```

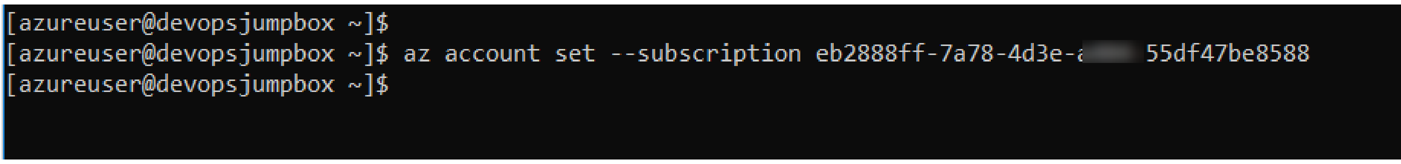
```
az account list --output table
```

Please note the subscriptionid value.



```
[azureuser@devopsjumpbox ~]$ az account list --output table
Name        CloudName    SubscriptionId                        State    IsDefault
----------  -----------  ------------------------------------  -------  -----------
Azure Pass  AzureCloud   eb2888ff-7a78-4d3e-[redacted]-55df47be8588  Enabled  True
[azureuser@devopsjumpbox ~]$
```

```
az account set --subscription "use the subscriptionid"
```



```
[azureuser@devopsjumpbox ~]$
[azureuser@devopsjumpbox ~]$ az account set --subscription eb2888ff-7a78-4d3e-[redacted] 55df47be8588
[azureuser@devopsjumpbox ~]$
```

2.7 Install Jenkins on Jumpbox

Jenkins is a Java application, so the first step is to install Java. Run the following command to install the OpenJDK 8 package:

```
sudo yum install -y java-1.8.0-openjdk-devel
```

Enable the Jenkins repository:

```
curl --silent --location http://pkg.jenkins-ci.org/redhat-stable/jenkins.repo |
sudo tee /etc/yum.repos.d/jenkins.repo
```

And add the repository to your system:

```
sudo rpm --import https://jenkins-ci.org/redhat/jenkins-ci.org.key
```

Install the latest stable version of Jenkins:

```
sudo yum install -y jenkins
```

To Change Jenkins port open Jenkins configuration File:

```
sudo nano /etc/sysconfig/jenkins
```

Change http port number:

```
JENKINS_PORT="8081"
```

Give permission to Jenkins user

```
sudo usermod -aG wheel jenkins
```

Give Jenkins user to docker deamon

```
sudo usermod -a -G docker jenkins
sudo systemctl restart docker
```

Make jenkins user sudoer for the lab environment.

Run visudo and at the bottom add a line similar to this;

```
sudo visudo
jenkins        ALL=(ALL)       NOPASSWD: ALL
```

Start Jenkins:

```
sudo systemctl start jenkins
```

Check the status:

```
systemctl status jenkins
```

Enable the Jenkins service to start on system boot:

```
sudo systemctl enable jenkins
```

Get initialAdminPassword:

```
sudo cat /var/lib/jenkins/secrets/initialAdminPassword
```

You should see a 32-character long alphanumeric password as shown below:

```
Output:
2115173b548f4e99a203ee99a8732a32
```

Copy the password from your terminal, open your browser, type http://104.40.242.213:8081 (Replace the example IP address with the **publicIpAddress** noted before) in your browser and paste copied password into the Administrator password field. Once you paste it click **continue**.

To Complete initial setup follow these steps:

- Choose Select plugins to install
- Search for GitHub in the text box across the top. Check the box for GitHub, then select Install
- Create the first admin user. Enter a username, such as admin, then provide your own secure password. Finally, type a full name and e-mail address.
- Select Save and Finish
- Once Jenkins is ready, select Start using Jenkins.