Deploy the application on Linux server using Nginx web server from GitHub repo by using script.

Task-

1. Pic the project required files and folders from GitHub repo
2. Deploy it to a linux server which have the public IP. So, via that public IP the application will be accessible

Step 1-

Login to the Linux server and verify the running ports on it. Verify if the nginx is installed or not. Alternatively we can do this by checking the nginx status “>> systemctl status nginx”.

>> sudo ss -nptl

A screen shot of a computer

Description automatically generated

There are no nginx installed on the server. Go ahead to install nginx web server.

* A web server is a computer system that stores, manages, and sends web content like images, videos, web pages, and website applications over the internet. Web servers are made up of hardware and software, and are a key part of the internet infrastructure that lets users browse web content.
* i.e. the webservers is a software. Once we install any of the web server software(nginx, apache tomcat, etc..) on any server the server becomes web server. That the webserver software uses that server’s hardware resources to run and to serve the application.

Update packages on the server.

>> sudo apt update

A screen shot of a computer

Description automatically generated

Install Nginx software.

>> sudo apt install nginx

Note- during installation it’s asking for the confirmation (yes/No)

A screenshot of a computer

Description automatically generated

Once installed successfully check it’s status and port.

A screenshot of a computer program

Description automatically generated

So, we see that the nginx is installed successfully and running on port 80.

As expected because most of the web server need to server applications on internet via http request that is 80.

Step 2-

Let’s browse it.

A screenshot of a computer

Description automatically generated

This suggest that can’t reach this page. Let’s try locally is the application is running.

>> curl localhost

A screenshot of a computer program

Description automatically generated

Here we see the application is running and it’s serving a default welcome page. Also if we see the above error page “can’t reach to that site” it’s says that to check the firewall.

It means on that server which we are trying to reach that server firewall is not allowing port 80.

On azure vm the virtual firewall is NSG (until we manually configure any other firewall).

Check NSG rule.

A screenshot of a computer

Description automatically generated

Port 80 is not allowed. Allow port 80.

A screenshot of a computer

Description automatically generated

Browse the site once the port is added successfully.

A screenshot of a computer

Description automatically generated

Step 3-

Now we need to remove this default page and place out application pages.

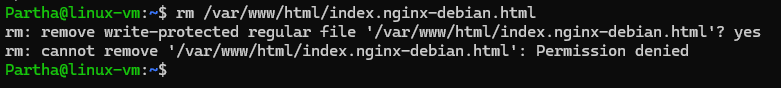
The Nginx web server keeps it’s default application file path to “/var/www/html”.

A computer screen with white text

Description automatically generated

Remove the default html page.

>> rm /var/www/html/index.nginx-debian.html



Can not remove due to permission issue because the nginx directory is different. It’s not in our directory.

>> sudo rm /var/www/html/index.nginx-debian.html



A screenshot of a computer

Description automatically generated

Step 4-

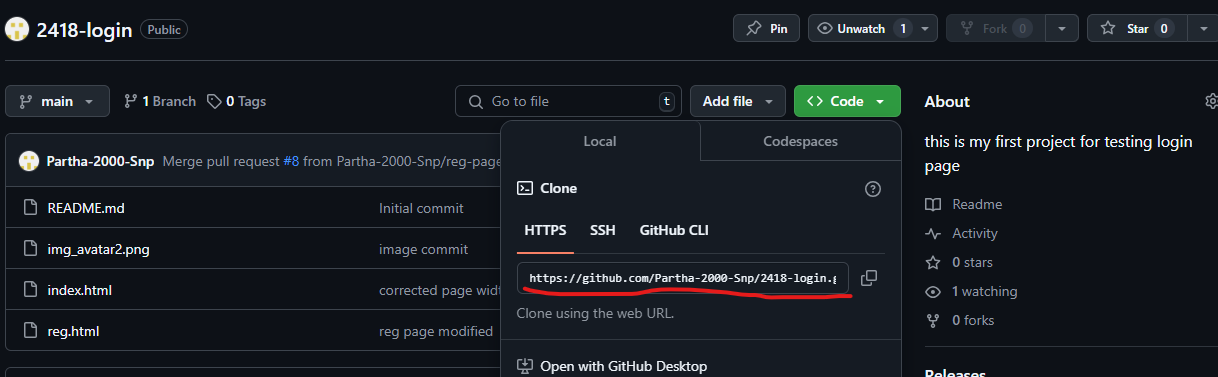
Now place the required application to nginx path.

Here we can do it in two ways.

1. Clone the repository to our directory and then copy or move it to nginx path
2. Directly copy to nginx path.

Let us try to directly copy to nginx path.

Copy the github url to clone.



>> git clone url destination

>> sudo git clone [https://github.com/Partha-2000-Snp/2418-login.git /var/www/html/](https://github.com/Partha-2000-Snp/2418-login.git%20/var/www/html/)

A screenshot of a computer

Description automatically generated

>> ls /var/www/html

A black background with white text

Description automatically generated

Check the application availability.

A screenshot of a login page

Description automatically generated

Step 5-

Now write a script in such a way that. The script will be present along with application code on github. So, in feature any time need to setup the application. We will just take that script file to linux server and run it. It should do all the configurations and make the application ready for users.

To setup this we need to write a new file to repo. As we know that the main branch of the repository. Will always be protected and we can’t directly perform any operation on that. To do any changes we must have our own branch. From there we need to go for pull request to marge the code to main branch.

Let us quickly have a ticket for this change.

A screenshot of a computer

Description automatically generated

Create new branch for the change.

A screenshot of a computer

Description automatically generated

Now clone the project to our directory for easy use.

>> git clone url

A black screen with text and symbols

Description automatically generated with medium confidence

Now we need to authenticate our server

>> git config --global user.name github-username

>> git config --global user.email github\_email\_ID



Step 6-

Create the script file and write script.

>> vi application-nxinx-setup.sh

Here we need to mention all the steps of setup by cmd.

Note – few of the cmd need confirmation to proceed (yes/no) as we saw during the time of Nginx installation. We want to avoid those. So in cmd we can mention “-y” that means yes for all.

Ex- sudo apt install nginx -y

Note- here we need to enable port 80 on NSG rule manually because the cmd we are running in vm from there we can’t change networking rule. For this we need to go for az cmd on cloud shell or bash if we want to do by cmd.

 **NSG rules are managed at the Azure platform level**.

 The VM doesn't have control over the **Azure networking configuration**, like NSG rules, load balancers, or public IP configurations.

But we can enable local port on vm by cmd.

If we want thet something will not get stuck for permission or any or due to recursive. We can apply fore remove by using “-rf” (remove force).

Code-

#starting acknowadgement text

echo "starting with nginx setup"

#update the system

echo "system updae in progress"

sudo apt update -y

echo "update completed"

#install nginx

echo "installing nginx"

sudo apt install nginx -y

echo "installation process completed"

#clear nginx default html

echo "clearing default nginx page"

sudo rm -rf /var/www/html

#clone the project repo to nginx destination

echo "cloning the project"

sudo git clone https://github.com/Partha-2000-Snp/2418-login.git /var/www/html/

#All done msg with ackno to enable port 80 on NSG

echo "setup completed you can proceed to enable port 80 on server NSG or firewall an proceed browsing the application."

Step 7-

A screen shot of a computer

Description automatically generated

Now move the file to stagging area from working area.

>> git add filename

A screen shot of a computer program

Description automatically generated

Setup the branch.

>> git branch

A screen shot of a computer

Description automatically generated

Switch to dev branch

>> git checkout dev

A screen shot of a computer

Description automatically generated

Commit the changes to local repo.

>> git commit -m "scriptfile created"

A screen shot of a computer program

Description automatically generated

Push to github dev branch (sync local to remote).

>> git push --all

Github username and access token

A computer screen with white text

Description automatically generated

Now create pull request and marge to main branch.

A screenshot of a computer

Description automatically generated

Main branch updated. Now close the issue.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Step 8-

Do verification by running the script is that working as expected. For this I am taking a totally new server and do setup.

Login to new server.

Clone repository.

A computer screen shot of a black screen

Description automatically generated

Got the script file but it’s not in executable format need to set executable permissions.

>> sudo chmod 777 filename

Ex-

sudo chmod 777 application-nxinx-setup.sh

A screen shot of a computer

Description automatically generated

Note- we just cloned the project to our directory to get the script file. Cloning to nginx repo and all done by script.

A computer screen with white text

Description automatically generated

Test locally

A computer screen with white text

Description automatically generated

Working, now enable the port 80 on nsg and browse.

A screenshot of a computer

Description automatically generated

All Done !....