#### DevOps Tooling Website Documentation

Links

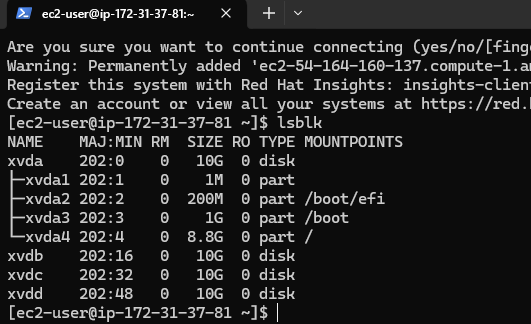
[SELF STUDY](https://docs.google.com/document/d/1hVlFnTPj5zYaiLiESrPeVNd_30PGewvuNsLoSNHHejg/edit?usp=sharing)

[Linkedin](https://www.linkedin.com/posts/adebayo-awolola_devops-webdevelopment-mysql-activity-7255817786927587328-8BzK?utm_source=share&utm_medium=member_desktop)

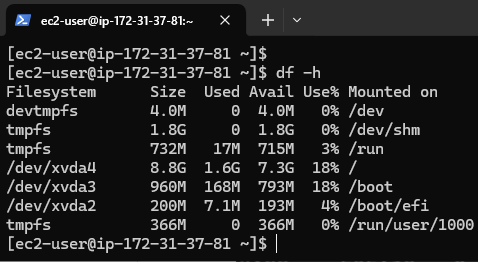
#### 

#### Step 1 — Prepare NFS Server

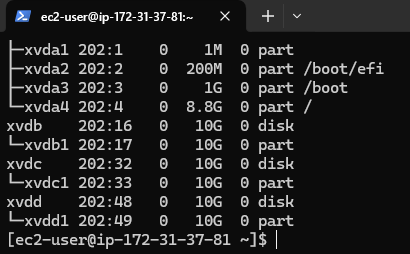
1. Launch an EC2 instance that will serve as "NFS Server". Create 3 volumes in the Attach all three volumes one by one to your Web Server EC2 instance same AZ as your Web Server EC2, each of 10 GiB.
2. Open up the Linux terminal to begin configuration
3. Use [lsblk](https://man7.org/linux/man-pages/man8/lsblk.8.html) command to inspect what block devices are attached to the server. Notice names of your newly created devices. All devices in Linux reside in /dev/ directory. Inspect it with ls /dev/ and make sure you see all 3 newly created block devices there - their names will likely be xvdb, xvdc, xvdd.



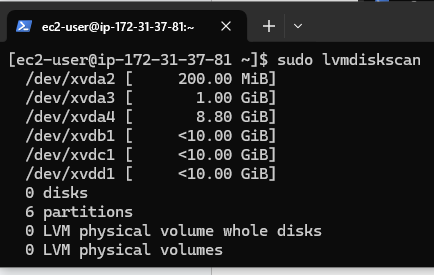
1. Use [df -h](https://en.wikipedia.org/wiki/Df_(Unix)) command to see all mounts and free space on your server



1. Use gdisk utility to create a single partition on each of the 3 disks
2. Use lsblk utility to view the newly configured partition on each of the 3 disks.



1. Install [lvm2](https://en.wikipedia.org/wiki/Logical_Volume_Manager_(Linux)) package using sudo yum install lvm2. Run sudo lvmdiskscan command to check for available partitions.



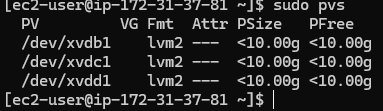
1. Use [pvcreate](https://linux.die.net/man/8/pvcreate) utility to mark each of 3 disks as physical volumes (PVs) to be used by LVM

sudo pvcreate /dev/xvdb1

sudo pvcreate /dev/xvdc1

sudo pvcreate /dev/xvdd1

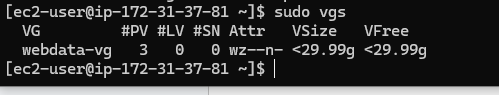
1. Verify that your Physical volume has been created successfully by running sudo pvs



1. Use [vgcreate](https://linux.die.net/man/8/vgcreate) utility to add all 3 PVs to a volume group (VG). Name the VG **webdata-vg**

sudo vgcreate webdata-vg /dev/xvdb1 /dev/xvdc1 /dev/xvdd1

1. Verify that your VG has been created successfully by running sudo vgs



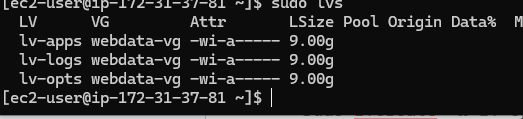
1. Use [lvcreate](https://linux.die.net/man/8/lvcreate) utility to create 3 logical volumes. lv-**apps,lv-opts**, and **logs-lv** ***of 10G***. **NOTE**: apps-lv will be used to store data for the Website while, logs-lv will be used to store data for logs.

sudo lvcreate -n lv-apps -L 9G webdata-vg &&

sudo lvcreate -n lv-opts -L 9G webdata-vg &&

sudo lvcreate -n lv-logs -L 9G webdata-vg

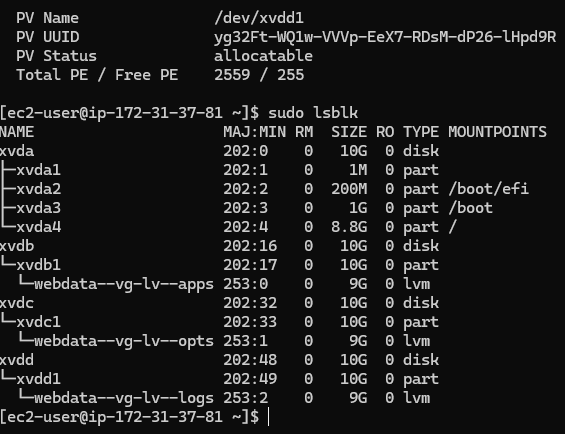
1. Verify that your Logical Volume has been created successfully by running sudo lvs



1. Verify the entire setup

sudo vgdisplay -v #view complete setup - VG, PV, and LV

sudo lsblk



1. Use mkfs.xfs to format the logical volumes with xfs filesystem

sudo mkfs -t xfs /dev/webdata-vg/lv-opts

sudo mkfs -t xfs /dev/webdata-vg/lv-apps &&

sudo mkfs -t xfs /dev/webdata-vg/lv-logs

1. Create **/**/mnt/apps directory to store website files sudo mkdir -p /mnt/apps
2. Create /mnt/opts directory to store website files sudo mkdir -p /mnt/opts
3. Create **/mnt/logs** to store backup of log data sudo mkdir -p /mnt/logs
4. Mount **/mnt/apps** on lv-**apps** logical volume

sudo mount /dev/webdata-vg/lv-apps /mnt/apps

sudo mount /dev/webdata-vg/lv-opts /mnt/opts

sudo mount /dev/webdata-vg/lv-logs /mnt/logs

1. Update /etc/fstab file so that the mount configuration will persist after restart of the server.

The UUID of the device will be used to update the /etc/fstab file;

sudo blkid

Opts: "bq50Qs-fl6R-bAke-022o-E2mS-9d1Y-ePiTE4"

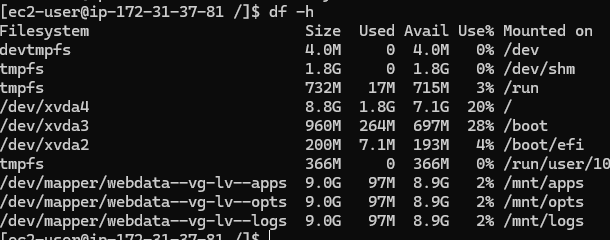
Apps:"ygbi0c-h5EX-WNWS-U0xc-RenJ-JYcx-mUVhR0""

Logs: “yg32Ft-WQ1w-VVVp-EeX7-RDsM-dP26-lHpd9R"

sudo vi /etc/fstab

Update /etc/fstab in this format using your own UUID and remember to remove the leading and ending quotes.

1. Test the configuration and reload the daemon  
   sudo mount -a  
   sudo systemctl daemon-reload
2. Verify your setup by running df -h, output must look like this



1. Install NFS server, configure it to start on reboot and make sure it is u and running

sudo yum -y update

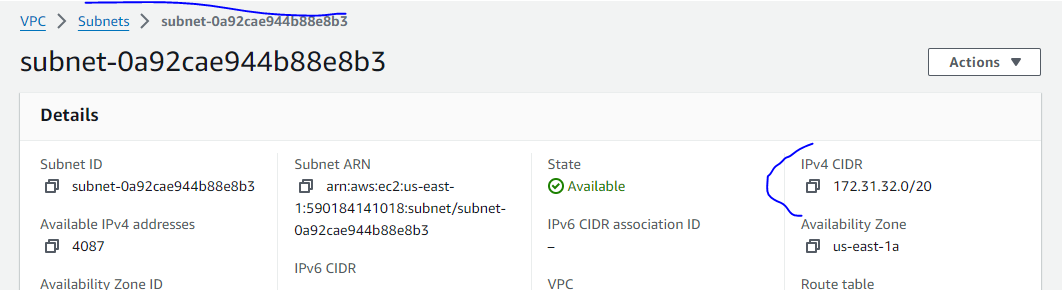
sudo yum install nfs-utils -y

sudo systemctl start nfs-server.service

sudo systemctl enable nfs-server.service

sudo systemctl status nfs-server.service

1. Export the mounts for webservers' subnet cidr to connect as clients. For simplicity, you will install your all three Web Servers inside the same subnet, but in production set up you would probably want to separate each tier inside its own subnet for higher level of security. To check your subnet cidr - open your EC2 details in AWS web console and locate 'Networking' tab and open a Subnet link



set up permission that will allow our Web servers to read, write and execute files on NFS:

sudo chown -R nobody: /mnt/apps

sudo chown -R nobody: /mnt/logs

sudo chown -R nobody: /mnt/opt

sudo chmod -R 777 /mnt/apps

sudo chmod -R 777 /mnt/logs

sudo chmod -R 777 /mnt/opt

sudo systemctl restart nfs-server.service

Configure access to NFS for clients within the same subnet (example of Subnet CIDR - 172.31.32.0/20 ):

sudo vi /etc/exports

/mnt/apps 172.31.32.0/20(rw,sync,no\_all\_squash,no\_root\_squash)

/mnt/logs 172.31.32.0/20(rw,sync,no\_all\_squash,no\_root\_squash)

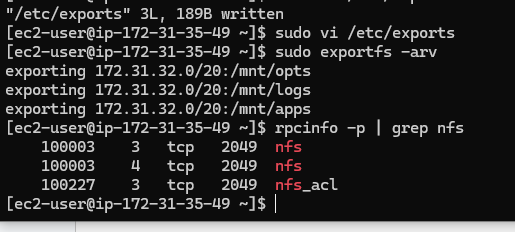
/mnt/opts 172.31.32.0/20(rw,sync,no\_all\_squash,no\_root\_squash)

Esc + :wq!

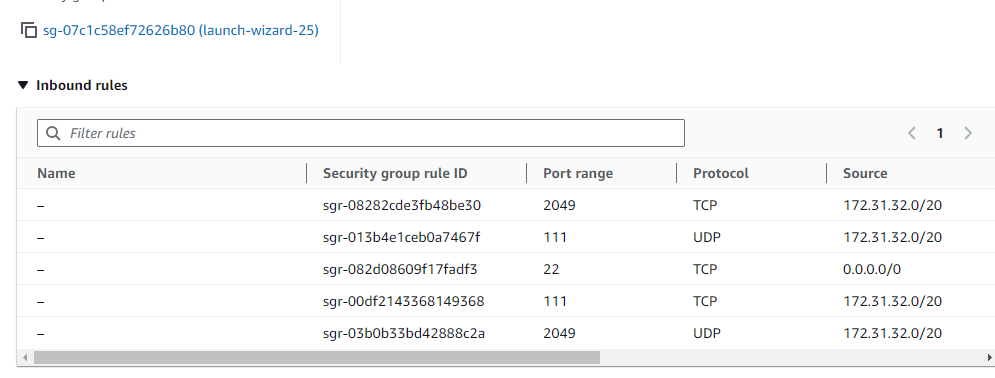
sudo exportfs -arv

1. Check which port is used by NFS and open it using Security Groups (add new Inbound Rule)

rpcinfo -p | grep nfs



**Important note:** In order for NFS server to be accessible from your client, you must also open following ports: TCP 111, UDP 111, UDP 2049



#### Install MySQL on your DB Server EC2

1. Install MySQL server
2. Create a database and name it tooling
3. Create a database user and name it webaccess
4. Grant permission to webaccess user on tooling database to do anything only from the webservers subnet cidr

sudo yum update

sudo yum install mysql-server

Verify that the service is up and running by using sudo systemctl status mysqld, if it is not running, restart the service and enable it so it will be running even after reboot:

sudo systemctl restart mysqld

sudo systemctl enable mysqld

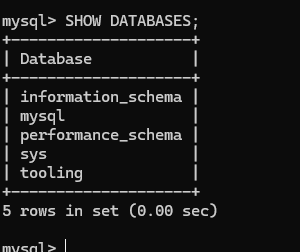
#### Install MySQL server

Create a database and name it tooling

sudo mysql

CREATE DATABASE tooling;

CREATE USER 'webaccess'@'172.31.32.0/20' IDENTIFIED BY 'mypass';



GRANT ALL ON tooling.\* TO 'webaccess'@'172.31.32.0/20';

FLUSH PRIVILEGES;

SHOW DATABASES;

use tooling;

select host, user from mysql.user;

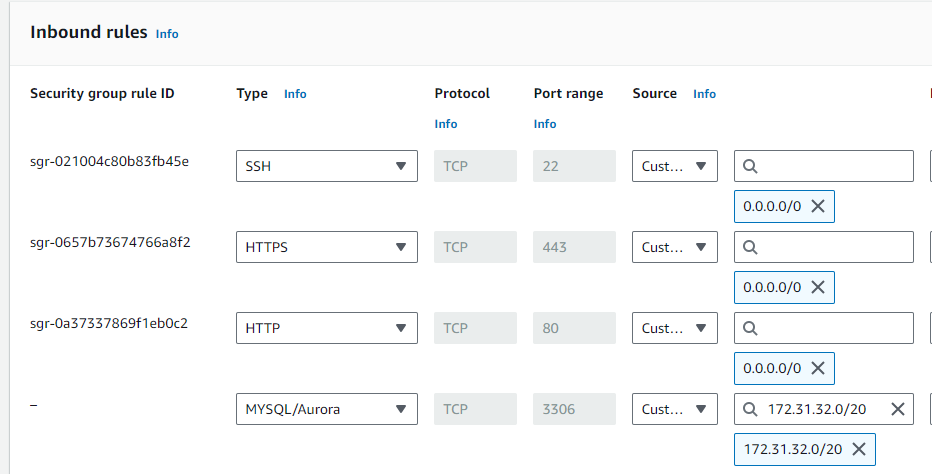
Exit

#### **Set Bind Address and restart MySQL**

sudo vi /etc/mysql/mysql.conf.d/mysqld.cnf

sudo systemctl restart mysql

sudo systemctl status mysql



**Hint:** Do not forget to open MySQL port 3306 on DB Server EC2. For extra security, you shall allow access to the DB server **ONLY** from your Web Servers CIDR, so in the Inbound Rule configuration specify source as /20

1. Install MySQL client and test that you can connect from your Web Server to your DB server by using mysql-client

sudo yum install mysql

sudo mysql -u admin -p -h <DB-Server-Private-IP-address>

1. Verify if you can successfully execute SHOW DATABASES; command and see a list of existing databases.

#### Prepare the Web Servers

We need to make sure that our web servers can serve the same content from shared storage solutions, such as NFS servers and MySQL databases. You already know that one DB can be accessed for reads and writes by multiple clients. For storing shared files that our Web Servers will use - we will utilize NFS and mount previously created Logical Volume lv-apps to the folder where Apache stores files to be served to the users (/var/www).

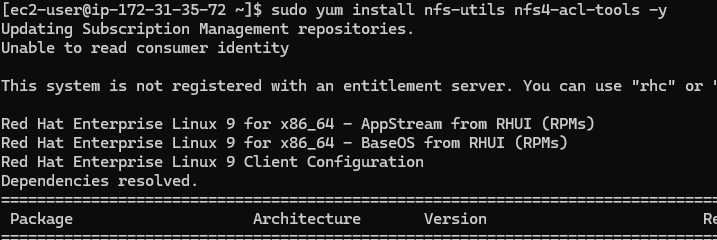
This approach will make our Web Servers stateless, which means we will be able to add new ones or remove them whenever we need, and the integrity of the data (in the database and on NFS) will be preserved.

During the next steps we will do following:

* Configure NFS client (this step must be done on all three servers)
* Deploy a Tooling application to our Web Servers into a shared NFS folder
* Configure the Web Servers to work with a single MySQL database

1. Launch a new EC2 instance with RHEL 8 Operating System
2. Install NFS client

sudo yum install nfs-utils nfs4-acl-tools -y



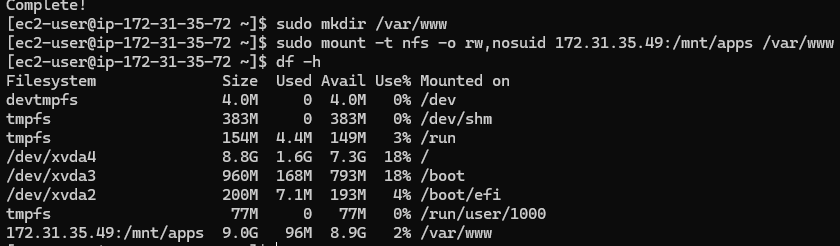
1. Mount /var/www/ and target the NFS server's export for apps

sudo mkdir /var/www

sudo mount -t nfs -o rw,nosuid <NFS-Server-Private-IP-Address>:/mnt/apps /var/www

1. Verify that NFS was mounted successfully by running df -h. Make sure that the changes will persist on Web Server after reboot:

sudo vi /etc/fstab



add following line

<NFS-Server-Private-IP-Address>:/mnt/apps /var/www nfs defaults 0 0

1. Install [Remi's repository](http://www.servermom.org/how-to-enable-remi-repo-on-centos-7-6-and-5/2790/), Apache and PHP

sudo yum install httpd -y

sudo dnf install https://dl.fedoraproject.org/pub/epel/epel-release-latest-9.noarch.rpm

sudo dnf install dnf-utils http://rpms.remirepo.net/enterprise/remi-release-8.rpm

sudo dnf module reset php

sudo dnf module enable php:remi-7.4

sudo dnf install php php-opcache php-gd php-curl php-mysqlnd

sudo systemctl start php-fpm

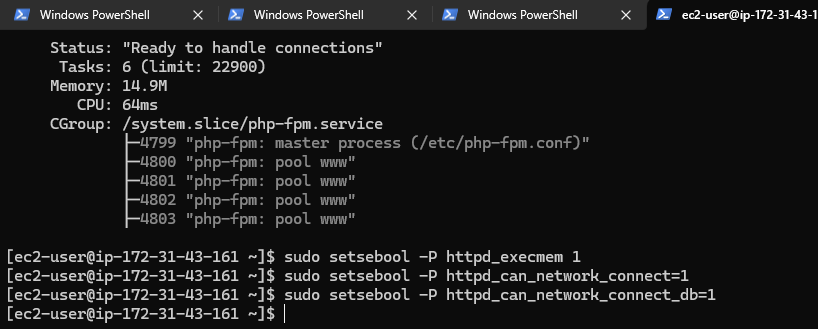
sudo systemctl enable php-fpm

sudo systemctl status php-fpm

sudo setsebool -P httpd\_execmem 1 # Allows the Apache HTTP server (httpd) to execute memory that it can also write to. This is often needed for certain types of dynamic content and applications that may need to generate and execute code at runtime.

sudo setsebool -P httpd\_can\_network\_connect=1 # Allows the Apache HTTP server to make network connections to other servers.

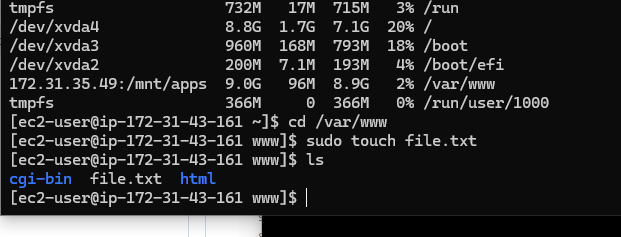
sudo setsebool -P httpd\_can\_network\_connect\_db=1 # allows the Apache HTTP server to connect to remote database servers.

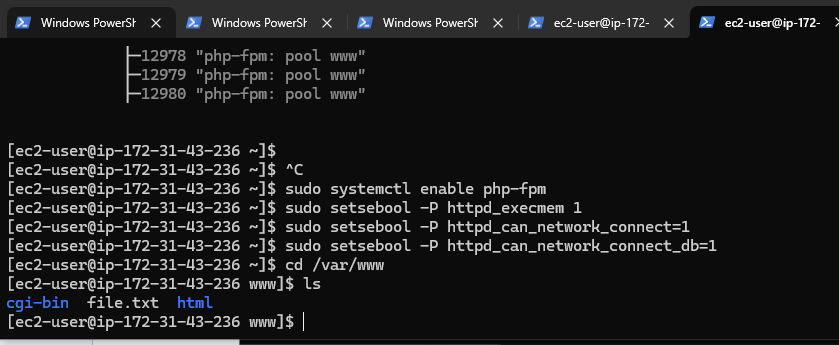


**Repeat steps 1-5 for another 2 Web Servers.**

1. Verify that Apache files and directories are available on the Web Server in /var/www and also on the NFS server in /mnt/apps. If you see the same files - it means NFS is mounted correctly. You can try to create a new file touch test.txt from one server and check if the same file is accessible from other Web Servers.

The test file was created on webserver 2 and can be accessed on webserver 1 and 3.



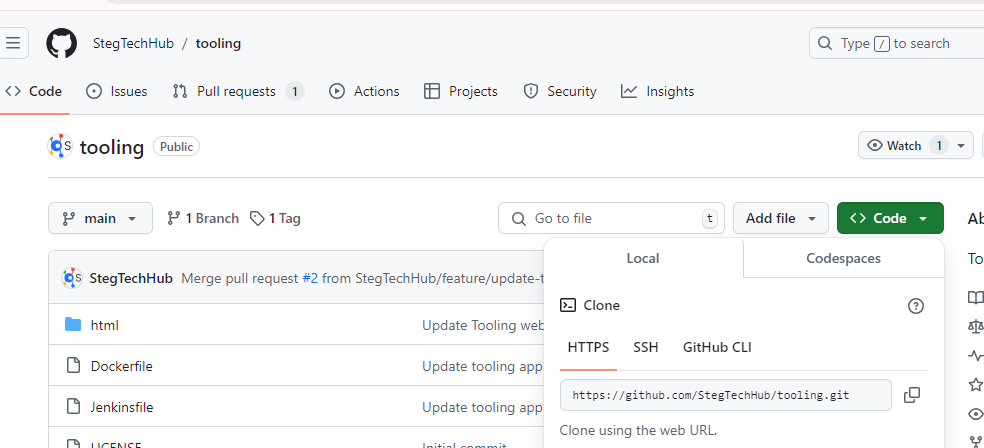


1. Locate the Apache log folder on the Web Server and mount it to the NFS server's log export. Repeat the step on all webservers to ensure the mount point persists after reboot.

sudo vi /etc/fstab

172.31.35.49:/mnt/logs /var/log/httpd nfs defaults 0 0

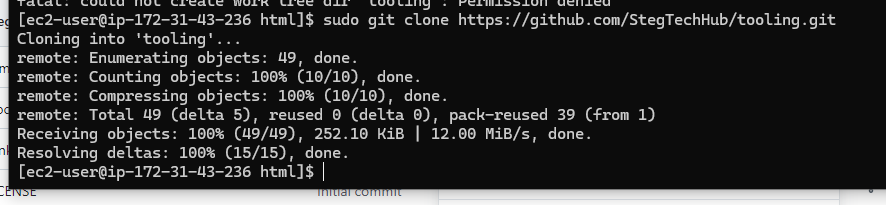
1. Fork the tooling source code from [StegHub Github Account](https://github.com/StegTechHub/tooling.git) to your Github account. (Learn how to fork a repo [here](https://youtu.be/f5grYMXbAV0))

****

1. Deploy the tooling website's code to the Webserver. Ensure that the **html** folder from the repository is deployed to /var/www/html. Change directory using cd /var/www/html.

sudo yum install git -y to install git on any of your webservers and use the command.

Git clone https://github.com/StegTechHub/tooling.git

****

**Note: Acces the website on a browser**

* **Ensure TCP port 80 is open on the Web Server.**
* **If 403 Error occur, check permissions to the /var/www/html folder and also disable SELinux**

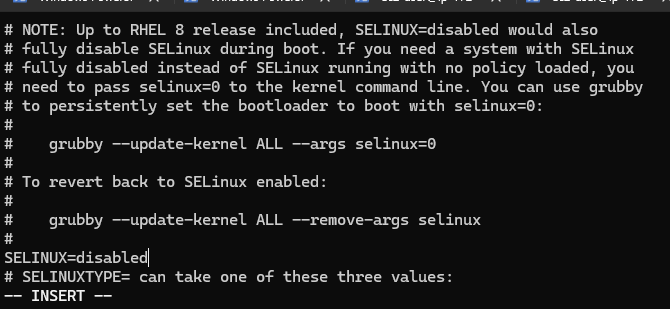
**sudo setenforce 0**

**To make the change permanent, open selinux file and set selinux to disable.**

**sudo vi /etc/sysconfig/selinux**

**SELINUX=disabled**

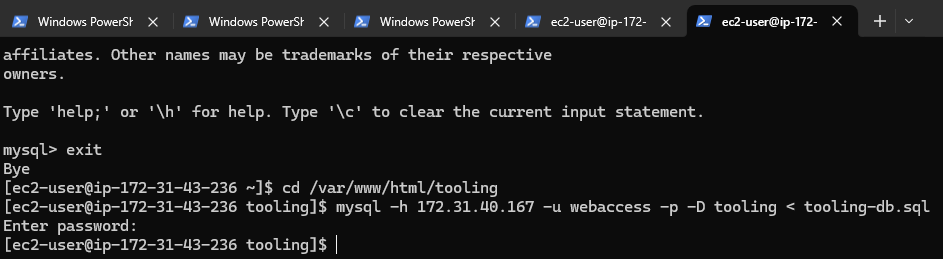
**sudo systemctl restart httpd**

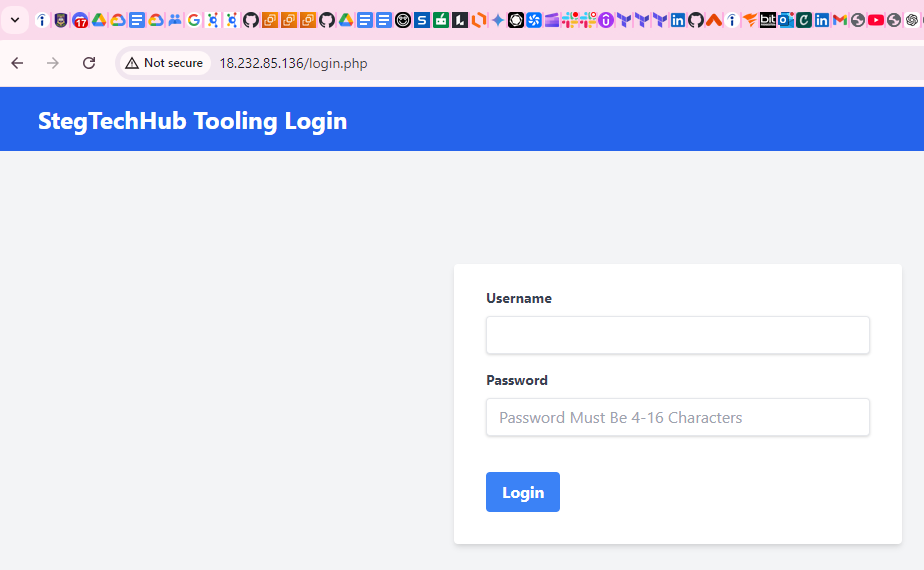
****

**Update the website's configuration to connect to the database (in /var/www/html/function.php file).**

**sudo vi /var/www/html/functions.php**

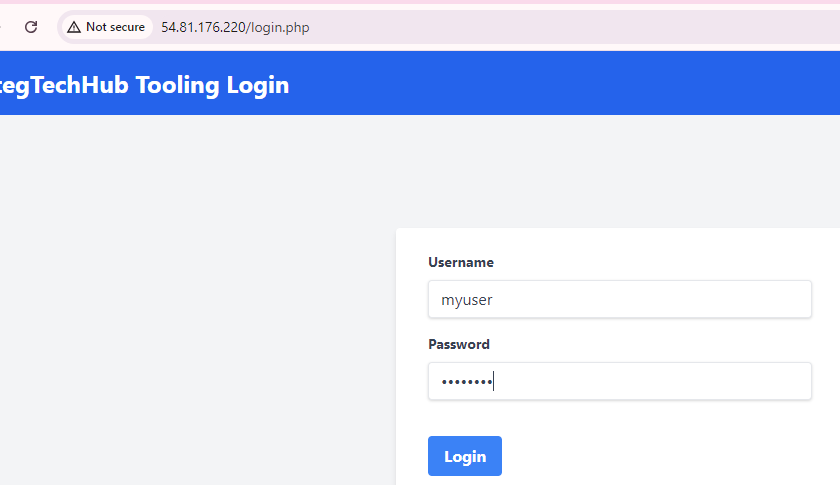
**Apply tooling-db.sql command sudo mysql -h <db-private-IP> -u <db-username> -p <db-password < tooling-db.sql on the webserver.**

****

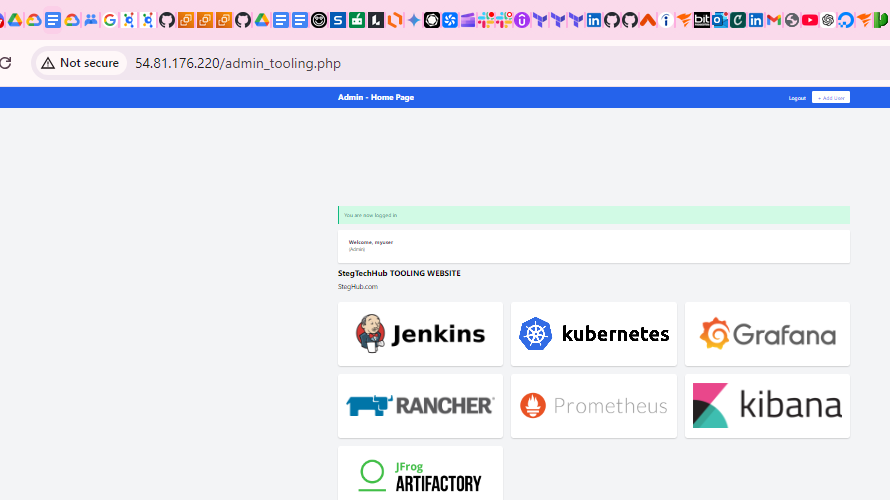
****

1. Create in MySQL a new admin user with username: myuser and password: password: on the DB server.

INSERT INTO 'users' ('id', 'username', 'password', 'email', 'user\_type', 'status') VALUES -> (1, 'myuser', '5f4dcc3b5aa765d61d8327deb882cf99', 'user@mail.com', 'admin', '1');

****

1. Open the website in your browser http://<Web-Server-Public-IP-Address-or-Public-DNS-Name>/index.php and make sure you can log in into the website with myuser user.

****