Installing Legends Tracker

# Linode

1. Setup a new Linode using Ubuntu 20.04 LTS with a public IP address
2. Update all installed packages
   1. sudo apt update
   2. sudo apt upgrade
3. Create a new user and add to sudo
   1. adduser <username>
   2. usermod -aG sudo <username>

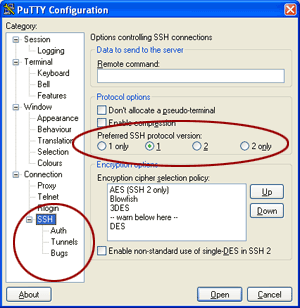
## Harden Linode – SSH

Many of the following instructions are sourced from these sites:

<https://www.informaticar.net/security-hardening-ubuntu-20-04/>

<https://linuxhint.com/secure-ssh-server-ubuntu/>

1. Block root from ssh logins
   1. sudo nano /etc/ssh/sshd\_config
   2. Change PermitRootLogin yes 🡺 PermitRootLogin no
2. Change SSH to listen on a different port
   1. Pick any port number above 1023 (I used 1059)
   2. Uncomment #Port 22 by removing the #
   3. Change the 22 to your chosen port number
3. Change maximum number of tries to login
   1. Uncomment #MaxAuthTries x
   2. Change the number of failed attempts before the lock comes in
4. Protocol 2
   1. Enable a more secure Protocol
   2. Add Protocol 2 on a new line towards the top of the file
   3. After enabling this, you may need to adjust Putty



1. Session Timeout
   1. Uncomment #ClientAliveInterval xx
   2. Set the number of seconds before you want the user to be logged out due to inactivity
2. Restrict log in to certain users
   1. Add a new line AllowUsers <username1> <username2> … <usernamex>
   2. Add a space between user names
3. Restart ssh to apply the changes
   1. sudo service ssh restart

## Harden Linode – Firewall

1. Check ufw status
   1. sudo ufw status
2. Allow all outgoing, but block all incoming by default
   1. sudo ufw default allow outgoing
   2. sudo ufw default deny incoming
3. Allow incoming ssh
   1. sudo allow <custom port #>
4. Start the firewall
   1. sudo ufw enable

## Harden Linode – Disable Root Login

1. Stop the root user from being able to log in at all
   1. sudo password -l root

## Harden Linode – MFA

1. Install Google Authenticator
   1. sudo apt install libpam-google-authenticator
2. Enable the authernticator
   1. sudo nano /etc/pam.d.sshd
   2. Add auth required pam\_google\_authenticator.so
   3. Save and exit
3. Update the SSH configuration
   1. sudo nano /etc/ssh/sshd\_config
   2. Change ChallengeResponseAuthentication no 🡺 ChallengeResponseAuthentication yes
4. Configure the authenticator
   1. Type google-authenticator [ENTER]
   2. Time based tokens: y
   3. Scan the barcode, record the emergency tokens
   4. Update the .google\_authenticator file: y
   5. Disallow multiple uses: y
   6. Permit skew: n
   7. Enable rate limiting: y
5. Restart SSH
   1. sudo service ssh restart

## Harden Linode – Fail2Ban

1. Install Fail2ban
   1. sudo apt-get install fail2ban
   2. Create the local jail file
   3. sudo nano /etc/fail2ban/jail.local
   4. Copy the following to the file
   5. Replace xxx.xxx.xxx.xxx with your public IP address
   6. Set the port # to the custom SSH port number set above
   7. Save and exit

[DEFAULT]

bantime = 8h

ignoreip = 127.0.0.1/8 xxx.xxx.xxx.xxx

ignoreself = true

[sshd]

enabled = true

port = 222

filter = sshd

logpath = /var/log/auth.log

maxretry = 3

1. Restart Fail2Ban
   1. sudo systemctl restart fail2ban

Manage Fail2Ban

To check for banned offenders:

sudo iptables -L f2b-sshd --line-numbers

To remove bans:

sudo iptables -D fail2ban-ssh <Chain num>

# Setup LAMP

Instructions sourced from:

<https://www.linuxbabe.com/ubuntu/install-lamp-stack-ubuntu-20-04-server-desktop>

## Apache

1. Install Apache
   1. sudo apt install -y apache2 apache2-utils
2. Allow through firewall
   1. The following will allow Apache tor respond on both port 80 (http) and 443 (https)
   2. sudo ufw allow "Apache Full"
3. Check firewall status
   1. Sudo ufw status
4. Check you can access the server
   1. Open a browser
   2. Type your IP address in the address bar and press [ENTER]
   3. You should see the default Apache2 page
5. Set the Apache user (www-data) as the owner of the document root
   1. sudo chown www-data:www-data /var/www/html/ -R
6. Set the server name
   1. sudo nano /etc/apache2/conf-available/servername.conf
   2. Add ServerName localhost
   3. Save and exit
   4. Enable the config file sudo a2enconf servername.conf
   5. Reload Apache sudo systemctl reload apache2
   6. Check status sudo apache2ctl -t

## MariaDB

1. Install MariaDB
   1. sudo apt install mariadb-server mariadb-client
2. Run the post-install security script to lock it down
   1. sudo mysql\_secure\_installation
   2. When asked for the current root password, just press [ENTER]
   3. Set root password: y
      1. Enter a secure root password
   4. Remove anonymous users: y
   5. Disallow root login remotely: y
   6. Remove test database and access: y
   7. Reload privilege table: y
3. Create a database
   1. sudo mariadb -u root
   2. create database legends;
   3. grant all privileges on legends.\* to 'legends'@'localhost' identified by 'ZFapZ2OjFTXfVNzEBYcrJmFqA188.;DR0';
   4. flush privileges;

## PHP 7

1. Install PHP 7.4
   1. sudo apt install php7.4 libapache2-mod-php7.4 php7.4-mysql php-common php7.4-cli php7.4-common php7.4-json php7.4-opcache php7.4-readline
2. Enable the PHP module in Apache
   1. sudo a2enmod php7.4
   2. Restart apache sudo systemctl restart apache2

# Configure Website

1. Configure DNS to point to server
2. Configure 2 x A-Records :
   1. lk.horscrust.com 🡺 IP Address
   2. [www.lk.horscrust.com](http://www.lk.horscrust.com) 🡺 IP Address
3. Configure a zone file for Apache
   1. sudo cp /etc/apache2/sites-available/000-default.conf /etc/apache2/sites-available/horscrust.com.conf
   2. sudo nano /etc/apache2/sites-available/horscrust.com.conf
   3. Set the following:

ServerName lk.horscrust.com

ServerAlias [www.lk.horscrust.com](http://www.lk.horscrust.com)

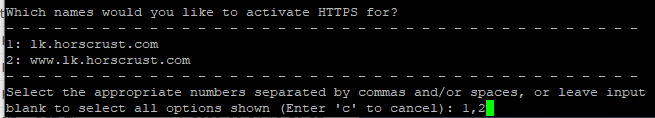
DocumentRoot /var/www/legendsc3

1. Make directory and set ownership
   1. sudo mkdir /var /www/legendsc3
   2. sudo chown www-data:www-data /var/www/legendsc3/ -R
2. Enable the new site
   1. sudo a2ensite horscrust.com.conf
3. Disable the default site
   1. sudo a2dissite 000-default.conf
4. Check config
   1. sudo apache2ctl configtest
   2. You should see Syntax OK
5. Restart Apache
   1. sudo systemctl restart apache2

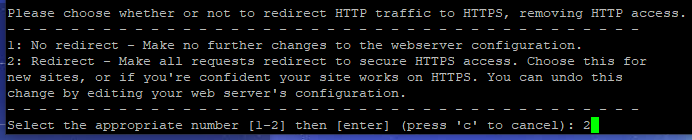
# Configure Lets Encrypt

Source: <https://www.digitalocean.com/community/tutorials/how-to-secure-apache-with-let-s-encrypt-on-ubuntu-20-04>

1. Install Certbot
   1. sudo apt install certbot python3-certbot-apache
2. Obtain SSL Certificate
   1. sudo certbot –apache
   2. Enter a valid email address
   3. Follow prompts
   4. The script should pick up the domains you want certificates for
   5. Enter the domains you want certificates for as a CSV



* 1. Enable redirection by entering 2



1. Verify Certificate auto-renewal
   1. sudo systemctl status certbot.timer
   2. Conduct a dry-run
      1. sudo certbot renew --dry-run
      2. If there are no errors, we are good to go!