How to manage a VPS

May 30, 2023 · 16 min read

By the end of this guide you should know how to set up, configure, and secure a VPS or dedicated server. The guide is useful for launching a Watcher instance or any blockchain node. It is required to have some basic knowledge of Linux.

Initial VPS Setup

1. Work with VPS Provider

A VPS is the most common way to run any blockchain infrastructure. It is a very efficient way to run a Watcher because it offers guaranteed uptime, redundancy in the case of hardware failure, and a static IP address that is required to use the installed Watcher instance during integration with the OMG Network. It is also possible to run a Watcher locally if you're an individual developer.

The guide is using Digital Ocean as an example of VPS. However, Amazon EC2, Google Cloud, OVH, and Linode are also popular choices. It's recommended to choose a provider that supports a Docker daemon and a Postgres database.

1.1 Set up an Account and Project

You need to have an account to start working with Digital Ocean. After your account is set up, you can create your first project.

1.2 Create a Droplet

Digital Ocean uses Droplets to create a new server, either standalone or as part of a larger, cloud-based infrastructure. To create a new Droplet, click on the `New Droplet` button.

There are a few things you need to configure before creating a Droplet. Consider using the following values:

1. Image: Ubuntu 16.04.6 (LTS) x64 or any other image that is listed on supported platforms list.
2. Plan: Basic with the pricing that matches the minimum hardware requirements.
3. Block storage: —
4. Datacenter Region: depends on your preference.
5. VPC Network: —
6. Additional options: —
7. Authentication: password. It is recommended to use SSH for increased security. The password option was chosen because some of the VPS providers don't provide SSH authentication out of the box. You'll learn how to set up the SSH further in the guide.
8. The number of Droplets: 1.
9. Hostname: an identifying name for your host.
10. Tags: any tags that will help to organize your servers, such as watcher, omg-network, etc.

To finish the process, click the `Create Droplet` button. The creation of a Droplet might take a few minutes. As a result, you will see your server details.

2. Connect to VPS

To connect to VPS, use `ssh` command from your terminal:

```
ssh root@$REMOTE_SERVER
```

The SSH Client is available starting from Windows 10 version. For earlier versions, please use PuTTY or other alternatives to connect to a remote server:

```
ssh root@$REMOTE_SERVER
```

By default, OpenSSH Client is an optional feature. You need to install it if you're using it the first time with the following commands:

1. Go to `Settings > Apps`, select `Manage optional features`.
2. Click the `Add a feature` button.
3. Find `OpenSSH Client`, click `Install`.
4. Test if the client works with the `ssh` command in your terminal.

When you connect to the server the first time, it will show the following message:

```
The authenticity of host '$REMOTE_SERVER' can't be established.
ECDSA key fingerprint is SHA256: ...
Are you sure you want to continue connecting (yes/no)? yes
```

Type `yes`. This will prompt you to enter a password. Note, all Linux systems don't reveal passwords when you type them, thus complete the process and click Enter. If your credentials are correct, you will see the following message:

```
Warning: Permanently added '$REMOTE_SERVER' (ECDSA) to the list of known hosts.
root@$REMOTE_SERVER's password:
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.4.0-169-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

0 packages can be updated.
0 updates are security updates.
```

VPS Security (basic)

3. Change a Password

Some of the server providers send the default login and password of the root user to your email. You must always change this password to a strong one. To change the password, use the `passwd` command. This will prompt you to type and repeat a new password. If everything is correct, you will see a success message:

```
passwd
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
```

4. Replace Root Login

4.1 Create a New User

It's not recommended to do any changes with the system under a root user due to security concerns. However, you can create a regular user and give `super user` rights to allow executing commands at root access without having all of the root access to modify system files. To create a new user, use the following command:

```
adduser $USER
```

Fill `$USER` with the name of the user you want to use instead of root. Then repeat a password for this user twice, and you can skip the other parts by pressing Enter:

```
adduser $USER
Adding user `$USER' ...
Adding new group `$USER' (1000) ...
Adding new user `$USER' (1000) with group `$USER' ...
Creating home directory `/home/$USER' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for $USER
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] y
```

4.2 Give Super User Rights To a New User

After a new user is created, one needs to have root access rights. This is accomplished with the following command:

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

This will add our user to a sudo group:

```
Adding user `rick' to group `sudo' ...
Adding user rick to group sudo
Done.
```

To test a connection with a new user, logout from your existing session and log in with the name of the user you've just created:

```
logout && ssh $USER@$REMOTE_SERVER
```

> Make sure to have both root and your new user passwords saved before doing any of the steps below.

4.3. Disable Root Login

To prevent brute force login attempts to the root account, it's recommended to disable root login. This is accomplished by changing configs in the `sshd_config` file using `nano` or `vi` text editors:

```
sudo nano /etc/ssh/sshd_config
```

To prevent from login into root user, scroll to the `#Authentication` section and change the `PermitRootLogin` key from `yes` to `no` as follows:

```
# Authentication:
LoginGraceTime 120
PermitRootLogin no
StrictModes yes
```

Press `ctrl+o` (Linux/Windows) or `control+o` (macOS) to save and `Enter` to confirm the changes respectively. Then exit the file with `ctrl+x` or `control+x`.

To apply the changes, restart SSH with the following command:

```
sudo service ssh restart
```

Now if you try to log into your server with root via SSH, it won't work and return the `Permission denied, please try again` message.

5. Change the Default Port

Another security measure to prevent people from making different types of attacks is to make it harder to find your SSH access port. You can change it in the `sshd_config` file by using the method from the previous step:

```
sudo nano /etc/ssh/sshd_config
```

Scroll to the `# What ports, IPs and protocols we listen for` section and change the port number from `22` to any number higher than 100 or ideally 1000. Save the changes, restart the SSH, and logout from the server. If you try to access your server with default connections, you'll receive a `Connection refused` error. You'll need to specify a port number each time you login into the server as follows:

```
ssh $USER@$REMOTE_SERVER -p $PORT
```

VPS Security (medium)

6. Change Authentication Method

6.1 Generate SSH Keys

As was mentioned earlier, SSH keys allow a more secure method of authenticating to your server. To establish such a method, you need to have a pair of public and private keys on your laptop. Note, you should check for existing SSH keys and make a backup if they are available but you don't want to use them for this specific server.

Open a terminal and run the following command:

```
cd ~/.ssh
```

If you see `No such file or directory`, you don't have any SSH keys on your local computer. If you have a `.shh` folder but you're not sure if you have any existing keys, use this command:

```
ls id_*
```

If you have existing SSH keys but you don't want to use them, you can make a backup with the `cp` command:

```
mkdir key_backup && cp id_rsa* key_backup
```

If you don't have any keys, create a new pair from your terminal:

```
ssh-keygen -t rsa
```

Open a command prompt and run the following command:

```
cd %userprofile%/.ssh
```

If you see `No such file or directory`, you don't have any SSH keys on your local computer. If you have a `.shh` folder but you're not sure if you have any existing keys, follow this command:

```
dir id_*
```

If you have existing SSH keys, but you don't want to use them, you can make a backup with the `copy` command:

```
mkdir key_backup
copy id_rsa* key_backup
```

If you don't have any keys, create a new pair from your command prompt:

```
ssh-keygen -t rsa
```

You will be asked to choose the path to save the keys and a passphrase that will be used during login to the server. You can press `Enter` for both of the options to have a default path and no passphrase accordingly. If you're concerned that your SSH keys can be hacked or compromised, you might consider setting a password. The entire process will look as follows:

```
ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key ($ID_RSA_DIR):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in $ID_RSA_DIR.
Your public key has been saved in $ID_RSA_DIR.
The key fingerprint is:
SHA256: ...
The key's randomart image is:
+---[RSA 2048]----+
|++.o             |
|* o    .         |
|E+  o o          |
|. ...+ o         |
|  o.o.o S .      |
|.ooo.o . +       |
```

```
|ooooo o.o        |
| .== =.=..       |
|o*B=ooO=o        |
+----[SHA256]-----+
```

6.2 Copy SSH Keys to Your Server

After you've generated the SSH keys, you need to copy the public key to your server. You can do that with the `ssh-copy-id` command:

```
ssh-copy-id $USER@$REMOTE_SERVER -p $PORT
```

Some of the versions of macOS may not support `ssh-copy-id` out of the box, so you need to install it first with `brew` or other alternatives:

```
brew install ssh-copy-id
```

Then you can copy the public key:

```
ssh-copy-id $USER@$REMOTE_SERVER -p $PORT
```

Windows currently doesn't support `ssh-copy-id` but you can use an alternative approach to achieve the same result. Run the following command from the Powershell as administrator:

```
cat ~/.ssh/id_rsa.pub | ssh $USER@$REMOTE_SERVER -p $PORT "umask 077; test -d .ssh || mkdir .ssh ; cat >>
.ssh/authorized_keys"
```

If the keys were added successfully, you will be prompt to log in a passphrase you set up during SSH keys generation:

```
Enter passphrase for key '$ID_RSA_DIR':
```

Otherwise, you will be logged in without a passphrase.

6.3 Disable Password Logins

Disabling password logins is the last step of basic security measures for your server. Make sure to verify one more time that authentication with SSH keys works before disabling password logins. You can accomplish that with the following command:

```
ssh $USER@$REMOTE_SERVER -p $PORT
```

You can disable password logins by changing the `sshd_config` file on your server as follows:

```
sudo nano /etc/ssh/sshd_config
```

Scroll to `# Change to no to disable tunneled clear text passwords` section and change `PasswordAuthentication` from `yes` to `no` as follows:

```
PasswordAuthentication no
```

Save the changes, close the file, and restart the SSH service:

```
sudo service ssh restart
```

7. Remove IPv6 listening

By default, Linux servers are configured to listen on IPv6 ports in addition to the standard IPv4 ports. It's a common practice to disable IPv6 because it's not widely used yet and can cause certain issues with SSH. To remove IPv6 listening, run the following command:

```
echo 'AddressFamily inet' | sudo tee -a /etc/ssh/sshd_config
```

Save the changes, close the file, and restart the SSH service. If you ever need IPv6 SSH back, remove the `AddressFamily inet` line.

VPS Security (advanced)

8. Set Up Firewall

A firewall is the last point of contact before anyone on the internet can get into your server. Getting a firewall up is crucial before deploying a server online. The example below demonstrates iptables as a way to set up Firewall rules. However, you may choose another software you're more comfortable with, such as `ufw`, `firewalld`, etc.

8.1 Check the Current Iptables Rules

```
sudo iptables -S
```

8.2 Install iptables-persistent

On Ubuntu, the easiest way to save iptables rules without a server reboot is to use the `iptables-persistent` package. You can install it with the following command:

```
sudo apt-get install iptables-persistent
```

During the installation, you will be prompt to save the current iptables rules. You can save them to be able to make a backup of the current configs in case something goes wrong.

> Make sure to have two active SSH connections to your server (two terminals) before following the rest of the guide. This will help to change configs back without rebooting the server.

8.3 Add Iptables Rules

Open iptables file:

```
nano /etc/iptables/rules.v4
```

Add the following values:

```
*filter
:INPUT DROP [0:0]
:FORWARD DROP [0:0]
:OUTPUT ACCEPT [985:1075980]
:f2b-sshd - [0:0]
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -p icmp -j ACCEPT
-A INPUT -i lo -j ACCEPT
# block invalid trafic
-A INPUT -m state --state INVALID -j REJECT
-A FORWARD -m state --state INVALID -j REJECT

# to ensure fail2ban works correctly after iptables restore
-A INPUT -p tcp -m multiport --dports $PORT -j f2b-sshd
# SSH
-A INPUT -p tcp --dport $PORT -j ACCEPT
```

```
COMMIT
```

If the file is not empty, replace it with the content above. Press `ctrl+o` (Linux/Windows) or `control+o` (macOS) to save and `Enter` to confirm the changes respectively. Then exit the file with `ctrl+x` or `control+x`.

> Note, `$PORT` is a port you're using to connect to the server via SSH. The default value is 22 but if you follow this guide, it should be different by now. See `step 5` for reference.

8.4 Restore Iptables

`iptables-restore` is used to restore IP Tables from data specified on STDIN or in a file. The command should be used as follows:

```
sudo iptables-restore < /etc/iptables/rules.v4
```

8.5 Restart Docker Services

If you are using Docker or any other virtualization software, you may need to restart their services after doing an `iptables-restore`:

```
systemctl restart docker && systemctl restart containerd
```

8.6 Check the Result

To check if you set up everything properly, use `nmap` or `netcat` tools as follows:

```
nmap -sS -p $PORT -T4 $REMOTE_SERVER
```

Example output (for your server's port):

```
Starting Nmap 7.80 ( https://nmap.org ) at 2020-06-02 19:06 FLE Daylight Time
Nmap scan report for $REMOTE_SERVER
Host is up (0.34s latency).

PORT     STATE SERVICE
1111/tcp open  unknown

Nmap done: 1 IP address (1 host up) scanned in 1.73 seconds
```

Example output (for the arbitrary port):

```
Starting Nmap 7.80 ( https://nmap.org ) at 2020-06-02 19:04 FLE Daylight Time
Nmap scan report for $REMOTE_SERVER
Host is up (0.37s latency).

PORT     STATE    SERVICE
2222/tcp filtered EtherNetIP-1

Nmap done: 1 IP address (1 host up) scanned in 4.01 seconds
```

Alternatively, you can use `Firewall Rule Test` to achieve the same result.

9. Set Up Fail2Ban

Fail2ban scans log files and bans IPs that show malicious signs, such as too many password failures, seeking for exploits, etc. Fail2Ban is then used to update firewall rules to reject the IP addresses for a specified amount of time, although any arbitrary other action (e.g. sending an email) could also be configured.

9.1 Install Dependencies

To install the dependencies, use the `apt-get` command as follows:

```
sudo apt-get install fail2ban sendmail
```

> `sendmail` is an optional dependency used to send emails when new IP bans happen.

9.2 Check Fail2Ban Status

When the dependencies are installed correctly, Fail2Ban status should indicate `active (running)`. You can verify this with the following command:

```
service fail2ban status
```

9.3 Configure Fail2Ban

To define Fail2Ban rules, you need to create a configurations file:

```
sudo touch /etc/fail2ban/jail.local
```

Then, open the file in `nano` or `vi` text editor, paste the following values, and save the result:

```
sudo nano /etc/fail2ban/jail.local
```

```
[DEFAULT]
# "ignoreip" can be an IP address, a CIDR mask or a DNS host. Fail2ban will not
```

```
# ban a host which matches an address in this list. Several addresses can be
# defined using space separator.
ignoreip = 127.0.0.1/8 $REMOTE_SERVER

# "bantime" is the number of seconds that a host is banned.
bantime = 3600

# A host is banned if it has generated "maxretry" during the last "findtime"
# seconds.
findtime = 600
maxretry = 3

# Enable the SSH daemon jail.
[sshd]
enabled = true

# Enable Email alerts
destemail = user@example.com
sendername = Fail2Ban
sender = user@server
action = %(action_mwl)s
```

> For setting up custom configurations, refer to `Linode guide`.

After the changes are saved, restart the Fail2Ban service as follows:

```
sudo service fail2ban restart
```

Now, if anyone makes 3 failed attempts to log in to your server with the wrong SSH passphrase within 600 seconds, the corresponding IP will be banned for 3600 seconds.

9.4 Check Fail2Ban Status

After the Fail2Ban is configured, you can check its status with the following command:

```
sudo fail2ban-client status
```

Example output:

```
Status
|- Number of jail:    1
`- Jail list:    sshd
```

You can also poll the detailed status of individual jails, such as `sshd`:

```
sudo fail2ban-client status sshd
```

Example output:

```
Status for the jail: sshd
|- Filter
|  |- Currently failed:    0
|  |- Total failed:    0
|  `- File list:                /var/log/auth.log
`- Actions
|- Currently banned:    0
|- Total banned:    0
`- Banned IP list:
```

Tags: tech writing web3

Recent posts

Dev Portals Research

How to manage a VPS