


Scale seamlessly with Droplet Autoscale Pools, now available →

BlogDocsGet SupportContact Sales



Products ▾Solutions ▾Developers ▾Partners ▾PricingLog in ▾Sign up ▾

TutorialsQuestionsProduct DocsCloud Chats

Q Search Community

Step 1 — Creating a New User

Step 2 — Creating a New Droplet

Step 3 — Granting Administrative Privileges

Step 4 — Setting Up a Basic Firewall

Step 5 — Enabling External Access for Your Regular User


Where To Go From Here?

// TUTORIAL //

Initial Server Setup with Debian 11

Published on October 30, 2021

- DebianGetting StartedInitial Server SetupDebian 11



[Brian Boucheron](#)



Not using Debian 11?

Choose a different version or distribution.

Debian 11 ▾



Introduction

When you first create a new Debian 11 server, there are a few configuration steps that you should take early on as part of the basic setup. This will increase the security and usability of your server and will

give you a solid foundation for subsequent actions.

In this tutorial, we will learn how to log into our server as the **root** user, create a new user with admin privileges, and set up a basic firewall.

Step 1 – Logging in as Root

To log into your server, you will need to know your **server’s public IP address**. You will also need the password or, if you installed an SSH key for authentication, the private key for the **root** user’s account. If you have not already logged into your server, you may want to follow our guide on [how to connect to your Droplet with SSH](#), which covers this process in detail.

If you are not already connected to your server, go ahead and log in as the **root** user using the following command (substitute the highlighted portion of the command with your server’s public IP address):

Copy

```
$ ssh root@ your_server_ip
```

Accept the warning about host authenticity if it appears. If you are using password authentication, provide your **root** password to log in. If you are using an SSH key that is passphrase protected, you may be prompted to enter the passphrase the first time you use the key each session. If this is your first time logging into the server with a password, you may also be prompted to change the **root** password.

About Root

The **root** user is the administrative user in a Linux environment that has very broad privileges. Because of the heightened privileges of the **root** account, you are *discouraged* from using it on a regular basis. This is because part of the power inherent with the **root** account is the ability to make very destructive changes, even by accident.

The next step is to set up an alternative user account with a reduced scope of influence for day-to-day work. Later, we’ll explain how to gain increased privileges for those times when you need them.

Step 2 – Creating a New User

Once you are logged in as **root**, we’re prepared to add the new user account that we will use to log in from now on.

This example creates a new user called **sammy**, but you should replace it with a username that you like:

Copy

```
# adduser sammy
```

You will be asked a few questions, starting with the account password.

Enter a strong password and, optionally, fill in any of the additional information you would like. This is not required and you can just hit **ENTER** in any field you wish to skip.

Next, we’ll set up this new user with admin privileges.

Step 3 – Granting Administrative Privileges

Now, we have created a new user account with regular account privileges. However, we may sometimes need to do administrative tasks with it.

To avoid having to log out of our normal user and log back in as the **root** account, we can set up what is known as *superuser* or **root** privileges for our normal account. This will allow our normal user to run commands with administrative privileges by putting the word **sudo** before the command.

To add these privileges to our new user, we need to add the new user to the **sudo** group. By default, on Debian 11, users who belong to the **sudo** group are allowed to use the `sudo` command.

As **root**, run this command to add your new user to the **sudo** group (substitute the highlighted word with your new user):

Copy

```
# usermod -aG sudo sammy
```

Now, when logged in as your regular user, you can type `sudo` before commands to run the command with superuser privileges.

Step 4 – Setting Up a Basic Firewall

Debian servers can use firewalls to make sure only certain connections to specific services are allowed. In this guide, we will install and use the UFW firewall to help set firewall policies and manage exceptions.

We can use the `apt` package manager to install UFW. Update the local index to retrieve the latest information about available packages and then install the UFW firewall software by typing:

Copy

```
# apt update
# apt install ufw
```

Note: If your servers are running on DigitalOcean, you can optionally use [DigitalOcean Cloud Firewalls](#) instead of the UFW firewall. We recommend using only one firewall at a time to avoid conflicting rules that may be difficult to debug.

Firewall profiles allow UFW to manage named sets of firewall rules for installed applications. Profiles for some common software are bundled with UFW by default and packages can register additional profiles with UFW during the installation process. OpenSSH, the service allowing us to connect to our server now, has a firewall profile that we can use.

You list all available application profiles by typing:

Copy

```
# ufw app list
```

Output

```
Available applications:
. . .
OpenSSH
. . .
```

We need to make sure that the firewall allows SSH connections so that we can log back in next time. We can allow these connections by typing:

Copy

```
# ufw allow OpenSSH
```

Afterwards, we can enable the firewall by typing:

Copy

```
# ufw enable
```

Type `y` and press `ENTER` to proceed. You can see that SSH connections are still allowed by typing:

```
# ufw status
```

```
Output
Status: active

To Action From
--
OpenSSH ALLOW Anywhere
OpenSSH (v6) ALLOW Anywhere (v6)
```

As **the firewall is currently blocking all connections except for SSH**, if you install and configure additional services, you will need to adjust the firewall settings to allow acceptable traffic in. You can learn some common UFW operations in [our UFW essentials guide](#).

Step 5 – Enabling External Access for Your Regular User

Now that we have a regular user for daily use, we need to make sure we can SSH into the account directly.

Note: Until verifying that you can log in and use `sudo` with your new user, we recommend staying logged in as **root**. This way, if you have problems, you can troubleshoot and make any necessary changes as **root**. If you are using a DigitalOcean Droplet and experience problems with your **root** SSH connection, you can also [log into the Droplet using the DigitalOcean Console](#).

The process for configuring SSH access for your new user depends on whether your server’s **root** account uses a password or SSH keys for authentication.

If the Root Account Uses Password Authentication

If you logged in to your **root** account *using a password*, then password authentication is enabled for SSH. You can SSH to your new user account by opening up a new terminal session and using SSH with your new username:

```
$ ssh sammy @ your_server_ip
```

After entering your regular user’s password, you will be logged in. Remember, if you need to run a command with administrative privileges, type `sudo` before it like this:

```
$ sudo command_to_run
```

You will be prompted for your regular user password when using `sudo` for the first time each session (and periodically afterwards).

To enhance your server’s security, **we strongly recommend setting up SSH keys instead of using password authentication**. Follow our guide on [setting up SSH keys on Debian 11](#) to learn how to configure key-based authentication.

If the Root Account Uses SSH Key Authentication

If you logged in to your **root** account *using SSH keys*, then password authentication is *disabled* for SSH. You will need to add a copy of your local public key to the new user's `~/.ssh/authorized_keys` file to log in successfully.

Since your public key is already in the **root** account's `~/.ssh/authorized_keys` file on the server, we can copy that file and directory structure to our new user account in our existing session with the `cp` command. Afterwards, we can adjust ownership of the files using the `chown` command.

Make sure to change the highlighted portions of the command below to match your regular user's name:

Copy

```
# cp -r ~/.ssh /home/ sammy
# chown -R sammy : sammy /home/ sammy /.ssh
```

The `cp -r` command copies the entire directory to the new user's home directory, and the `chown -R` command changes the owner of that directory (and everything inside it) to the specified `username:groupname` (Debian creates a group with the same name as your username by default).

Now, open up a new terminal session and log in via SSH with your new username:

Copy

```
$ ssh sammy @ your_server_ip
```

You should be logged in to the new user account without using a password. Remember, if you need to run a command with administrative privileges, type `sudo` before it like this:

Copy

```
$ sudo command_to_run
```

You will be prompted for your regular user password when using `sudo` for the first time each session (and periodically afterwards).

Where To Go From Here?

At this point, you have a solid foundation for your server. You can install any of the software you need on your server now.

Thanks for learning with the DigitalOcean Community. Check out our offerings for compute, storage, networking, and managed databases.

[Learn more about our products →](#)

About the authors



[Brian Boucheron](#) Author

Still looking for an answer?

Ask a question

Search for more help

Was this helpful?

Yes

No



Comments

Leave a comment

B *I* U ☒ 📎 🖼️ ✎ H₁ H₂ H₃ ☰ ☱ “” ⓘ 🗎 <>

Leave a comment...

▲ ▼

This textbox defaults to using **Markdown** to format your answer.

You can type **!ref** in this text area to quickly search our full set of tutorials, documentation & marketplace offerings and insert the link!

Sign In or Sign Up to Comment



This work is licensed under a Creative Commons Attribution-NonCommercial- ShareAlike 4.0 International License.

Try DigitalOcean for free

Click below to sign up and get **\$200 of credit** to try our products over 60 days!

Sign up

Popular Topics

- AI/ML
- Ubuntu
- Linux Basics
- JavaScript
- Python
- MySQL
- Docker
- Kubernetes

All tutorials →

Talk to an expert →

Congratulations on unlocking the whale ambience easter egg!

Click the whale button in the bottom left of your screen to toggle some ambient whale noises while you read.

[Reset easter egg to be discovered again](#)
[Permanently dismiss and hide easter egg](#)



Thank you to the [Glacier Bay National Park & Preserve](#) and [Merrick079](#) for the sounds behind this easter egg.



Interested in whales, protecting them, and their connection to helping prevent climate change? We recommend checking out the [Whale and Dolphin Conservation](#).



Become a contributor for community

Get paid to write technical tutorials and select a tech-focused charity to receive a matching donation.

Sign Up →



DigitalOcean Documentation

Full documentation for every DigitalOcean product.

Learn more →



Resources for startups and SMBs

The Wave has everything you need to know about building a business, from raising funding to marketing your product.

Learn more →

Get our newsletter

Stay up to date by signing up for DigitalOcean’s Infrastructure as a Newsletter.

Email address

Submit

New accounts only. By submitting your email you agree to our [Privacy Policy](#)

The developer cloud

Scale up as you grow — whether you're running one virtual machine or ten thousand.

View all products

Get started for free

Sign up and get \$200 in credit for your first 60 days with DigitalOcean.*

Get started

*This promotional offer applies to new accounts only.

Company

- About
- Leadership
- Blog
- Careers
- Customers
- Partners
- Referral Program
- Affiliate Program
- Press
- Legal
- Privacy Policy
- Security
- Investor Relations
- DO Impact
- Nonprofits

Contact

- Support
- Sales
- Report Abuse
- System Status

Products

- Overview
- Droplets
- Kubernetes
- Functions
- App Platform
- GPU Droplets
- 1-Click Models
- GenAI Platform
- Bare Metal GPUs
- Load Balancers
- Managed Databases
- Spaces
- Block Storage
- API
- Uptime
- Identity Access Management
- Cloudways

Resources

- Community Tutorials
- Community Q&A
- CSS-Tricks
- Write for DOnations
- Currents Research
- Hatch Startup Program
- Wavemakers Program
- Compass Council
- Open Source
- Newsletter Signup
- Marketplace
- Pricing
- Pricing Calculator
- Documentation
- Release Notes
- Code of Conduct
- Shop Swag

Solutions

- Website Hosting
- VPS Hosting
- Web & Mobile Apps
- Game Development
- Streaming
- VPN
- SaaS Platforms
- Cloud Hosting for Blockchain
- Startup Resources

Share your ideas



© 2025 DigitalOcean, LLC. [Sitemap](#).

