

How to Create a Sudo User in Linux

Learn how to create a sudo user in Linux with step-by-step instructions. Enhance system security while maintaining administrative capabilities on your Linux system.

Contents

| | | |
|----|---|----|
| 01 | Introduction | 3 |
| 02 | Prerequisites | 3 |
| 03 | Create a Sudo User | 3 |
| 04 | Create a Sudo User on Red Hat-Based Distributions (AlmaLinux, Rocky Linux, CentOS) | 3 |
| 05 | Create a Sudo User on Debian and Ubuntu | 6 |
| 06 | Create a Sudo User on Arch Linux | 7 |
| 07 | Test Sudo User Privileges in Linux | 9 |
| 08 | Manage Password Prompts for a Sudo User | 11 |
| 09 | Conclusion | 15 |

Introduction

Creating a `sudo` user allows a non-root user to perform administrative tasks with elevated privileges. Instead of logging in as root, the user authenticates with their password, and the system securely logs each command they run. This approach improves system security by limiting root access and providing better accountability. Most Linux distributions follow a similar process for adding `sudo` users, with slight group names or package requirements variations.

This article explains how to create a `sudo` user on major Linux distributions, verify their privileges, and apply optional security settings to control `sudo` behavior.

Prerequisites

Before you begin, you need to:

- Have access to a [Linux instance](#) as a non-root sudo user.

Create a Sudo User

In this section, you will create a new user and grant them `sudo` privileges. While the exact steps may vary slightly between distributions, most follow a similar process. Start by choosing the section that matches your Linux distribution.

Create a Sudo User on Red Hat-Based Distributions (AlmaLinux, Rocky Linux, CentOS)

On Red Hat-based systems, users gain `sudo` access by being added to the `wheel` group, which is predefined in the instance **sudoers** configuration. This group

based approach helps centralize administrative permissions and improve security by controlling who can execute privileged commands.

1. Create a new user.

CONSOLE

```
$ sudo useradd -m example_user
```

In the above command, the `-m` option ensures a home directory is created at `/home/example_user`. Replace `example_user` with your desired username.

2. Set a password for the user.

CONSOLE

```
$ sudo passwd example_user
```

You'll be prompted to enter and confirm a secure password. The system stores this encrypted password in `/etc/shadow`, which is readable only by privileged users.

3. Add the user to the `wheel` group.

CONSOLE

```
$ sudo usermod -aG wheel example_user
```

The above command grants `sudo` access by adding the user to the `wheel` group.

- The `-a` option appends the user to the group without removing them from any existing groups.
- The `-G` flag specifies the group name.

4. Open the `/etc/sudoers` file with `visudo` to verify that the `wheel` group has `sudo` privileges.

CONSOLE

```
$ sudo visudo
```

Unlike a regular text editor, `visudo` performs syntax validation before saving changes. This helps prevent configuration errors that could lock you out of the system.

5. Locate the following line and make sure it is uncommented.

```
INI
...
## Allows people in group wheel to run all commands
%wheel  ALL=(ALL)        ALL
...
```

The above directive allows users in the `wheel` group to run any command with `sudo`. Its components mean:

- `%wheel`: Applies to the wheel group (% indicates a group).
- `First ALL`: Applies to all hosts.
- `(ALL)`: Allows running commands as any user.
- `Last ALL`: Allows running any command.

Note

Always use `visudo` to edit the sudoers file. Editing it directly with a standard text editor can result in syntax errors that break `sudo` functionality.

6. Switch to the newly created user account.

```
CONSOLE
$ sudo su - example_user
```

The hyphen (-) ensures that the user's full environment is loaded, including updated PATH variables.

7. Run the `whoami` command with `sudo` to verify administrative access.

```
CONSOLE
```

```
$ sudo whoami
```

Output:

```
root
```

The output `root` confirms that the configuration is correct and the user has `sudo` privileges.

Create a Sudo User on Debian and Ubuntu

Debian-based systems, such as Ubuntu and its derivatives, manage administrative privileges using the `sudo` group rather than `wheel`. In this section, you will create a new user, assign `sudo` access, and verify that the configuration works as expected.

1. Create a new user.

```
CONSOLE
```

```
$ sudo adduser example_user
```

The `adduser` utility is interactive and user-friendly. It creates the **home directory**, prompts for a **password**, and optionally collects user **metadata** like full name and contact information.

2. Add the user to the `sudo` group.

```
CONSOLE
```

```
$ sudo usermod -aG sudo example_user
```

By default, members of the `sudo` group have administrative privileges on Debian and Ubuntu systems. This configuration is defined in the `/etc/sudoers` file and included by default.

3. Switch to the newly created user account.

```
CONSOLE
$ sudo su - example_user
```

The `-` flag ensures the new user environment is fully loaded, including the updated group membership.

4. Run the `whoami` utility using `sudo` to print the effective user name.

```
CONSOLE
$ sudo whoami
```

Output:

```
root
```

If the output returns `root`, the user has successfully been granted `sudo` privileges.

Create a Sudo User on Arch Linux

Arch Linux's environment requires manual configuration of administrative access. In this section, you'll create a new user, assign them to the `wheel` group, and configure the `sudoers` file to enable `sudo` privileges.

1. Create a new user.

```
CONSOLE
$ sudo useradd --create-home example_user
```

In the above command, the `--create-home` flag ensures the system creates a home directory for the user. Replace `example_user`.

2. Set a password for the user.

CONSOLE

```
$ sudo passwd example_user
```

When prompted, enter and confirm a strong password. The system securely stores the password in encrypted form in the `/etc/shadow` file, which is readable only by the `root` user.

3. Add the user to the `wheel` group.

CONSOLE

```
$ sudo usermod --append --groups wheel example_user
```

Arch Linux uses the `wheel` group to control `sudo` access, similar to Red Hat-based systems.

4. Install the `vi` editor, which the `visudo` utility requires.

CONSOLE

```
$ sudo pacman --sync vi
```

5. Update the `sudoers` file.

CONSOLE

```
$ sudo visudo
```

Find the following line and ensure it is uncommented.

INI

```
...  
%wheel ALL=(ALL) ALL  
...
```

This enables all users in the `wheel` group to use the `sudo` utility.

6. Switch to the newly created user.

```
CONSOLE
```

```
$ sudo su - example_user
```

Then verify `sudo` access:

```
CONSOLE
```

```
$ sudo whoami
```

Output:

```
root
```

This confirms that `sudo` is configured for the new user.

Test Sudo User Privileges in Linux

After setting up a `sudo` user, verify that the account has administrative privileges. This section walks you through several tests to confirm proper `sudo` functionality.

Run Basic Commands with Sudo

Run these commands to ensure the user can perform administrative tasks.

1. View the `root` user's home directory.

```
CONSOLE
```

```
$ sudo ls -la /root
```

This command displays hidden and regular files in `/root`, which only `root` or users with `sudo` privileges can access.

2. Update the server package lists.

- On Debian-based systems (Ubuntu, Debian):

CONSOLE

```
$ sudo apt update
```

- On Red Hat-based systems (RHEL, CentOS, Fedora):

CONSOLE

```
$ sudo dnf makecache
```

- On Arch Linux:

CONSOLE

```
$ sudo pacman -Sy
```

Package management commands require `root` privileges because they modify system files and packages. If these commands work without errors, your `sudo` setup is functioning.

Verify Access to Protected Files

Confirm that your `sudo` user can access sensitive files restricted to `root`.

CONSOLE

```
$ sudo cat /etc/shadow
```

The `/etc/shadow` file contains encrypted user passwords and is critical for system security. It's only readable by `root` or users with `sudo` privileges, so viewing it confirms your `sudo` privileges are working.

Check Sudo Logs

`sudo` maintains detailed logs of all activity, which is crucial for security auditing and accountability.

- On Debian-based systems, run:

CONSOLE

```
$ sudo grep sudo /var/log/auth.log
```

- On Red Hat systems, run:

CONSOLE

```
$ sudo grep sudo /var/log/secure
```

These logs record when users run `sudo`, who ran it, and which commands they executed. This audit trail is a key security advantage of using `sudo` instead of directly operating as the `root` user.

Manage Password Prompts for a Sudo User

In this section, you will configure how and when `sudo` prompts for a password, allowing you to fine-tune the balance between security and convenience.

Configure Password Timeout

By default, `sudo` caches your credentials for **15 minutes**. Follow the steps below to customize this timeout to suit your workflow.

1. Open the `sudoers` file using the `visudo` utility.

```
CONSOLE
```

```
$ sudo visudo
```

2. Add or modify the following line.

```
INI
```

```
Defaults          timestamp_timeout=30
```

This example sets the timeout to **30 minutes**. This value determines how long sudo remembers your authentication after a successful password entry.

Special values for `timestamp_timeout`:

- `0`: Require the password for every `sudo` command.
- `-1`: Never prompt again during the session.
- Positive integers (for example, `5`, `15`, `30`): Define the authentication timeout in minutes.

Behind the scenes, `sudo` creates timestamp files in `/var/run/sudo/` or `/var/lib/sudo/` to track when users last authenticated.

Set Up Passwordless Sudo

For automation tasks or controlled environments, you can allow specific **users** or **groups** to execute `sudo` commands without being prompted for a password.

1. Open the `sudoers` file.

```
CONSOLE
```

```
$ sudo visudo
```

2. Add a rule for the user.

```
INI
```

```
example_user ALL=(ALL) NOPASSWD: ALL
```

Or configure it for an entire group:

```
INI
```

```
%wheel ALL=(ALL) NOPASSWD: ALL
```

Warning

Enabling password-less `sudo` reduces security by eliminating authentication prompts. Use it only in secure environments with fully trusted users.

Limit Command Execution

Following the principle of least privilege, restrict which commands users can execute with `sudo`.

1. Edit the `sudoers` file.

```
CONSOLE
```

```
$ sudo visudo
```

2. Define explicit command permissions.

```
INI
```

```
example_user ALL=(ALL) /usr/bin/apt update, /usr/bin/apt  
upgrade
```

This configuration allows the user to run only the specified commands with `sudo`. Use absolute paths, as `sudo` does not resolve commands through `$PATH`.

To locate a command's full path, run:

```
CONSOLE
```

```
$ which <command>
```

For example:

```
CONSOLE
```

```
$ which mkdir
```

Output:

```
/usr/bin/mkdir
```

Create Command Aliases

In multi-user environments, command aliases can simplify the privilege definitions.

1. Open the `sudoers` file.

```
CONSOLE
```

```
$ sudo visudo
```

2. Define command groups and assign them.

```
INI
```

```
# Command Aliases
Cmnd_Alias UPDATES = /usr/bin/apt update, /usr/bin/apt
upgrade
Cmnd_Alias SERVICES = /usr/bin/systemctl restart apache2

# User Privileges
example_user ALL=(ALL) UPDATES, SERVICES
```

Command aliases let you group related commands under a single name, making the `sudoers` file more organized and maintainable as it grows.

Conclusion

You have successfully created and configured a `sudo` user across major Linux distributions. You granted administrative privileges by assigning the user to the appropriate group (`wheel` or `sudo`) and verified access through protected commands. You also reviewed ways to enforce secure privilege escalation by customizing the **sudoers** file, such as restricting commands, enabling password-less access, and setting timeout and retry limits. With these configurations, your system now has properly managed `sudo` access, enhancing usability and security.



VULTR

