The whoami command **allows Linux users to see the currently logged-in user**. The output displays the username of the effective user in the current shell. Additionally, whoami is useful in bash scripting to show who is running the script.

**whoami Command Syntax**

The syntax for the **whoami** command is:

whoami [option]

**whoami Command Options**

The **whoami** command comes with the two following options:

| **Option** | **Description** |
| --- | --- |
| --help | Shows a help message and exits. |
| --version | Shows the version information and exits. |

**whoami Command Examples**

There are several different ways to use **whoami**. The examples below explain common use cases of the **whoami** command.

**Basic whoami Usage**

When running **whoami** without any options, the output shows the name of the currently logged-in user.

To test the command, open the terminal and enter:

whoami



The output prints the name of the effective user.

**Verify the Effective User**

When you have more than one account, use **whoami** to verify the username after switching to a different account.

For example, switch to another user with the [su command](https://phoenixnap.com/kb/su-command-linux-examples):

su [account name]

su Command Terminal Output

Then, run the **whoami** command to verify the effective user:

whoami Command Terminal Output

Since you executed the command as a different user, the terminal shows another username.

**Check an Account for Sudo Permissions**

Use the **whoami** command to check if a user has **sudo** privileges. To do this, execute the command with [sudo](https://phoenixnap.com/kb/linux-sudo-command) and provide the password:

sudo whoami

sudo whoami Terminal Output

In case you don’t have sudo privileges, the output looks like this:

sudo whoami Terminal Output With no Sudo Privileges

**Note:** Learn how can you [create sudo user](https://phoenixnap.com/kb/how-to-create-sudo-user-on-ubuntu).

**Confirm Which User is Running a Script**

The **whoami** command in bash scripts shows which user is running the script. For example, use **whoami** to test if root is executing the script, and if so, print a warning message using [the echo command](https://phoenixnap.com/kb/echo-command-linux).

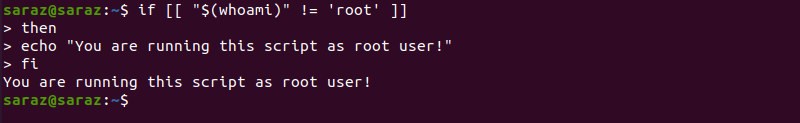
Use a simple bash if statement to test this:

if [[ "$(whoami)" != 'root' ]]

then

echo "You are running this script as root user!"

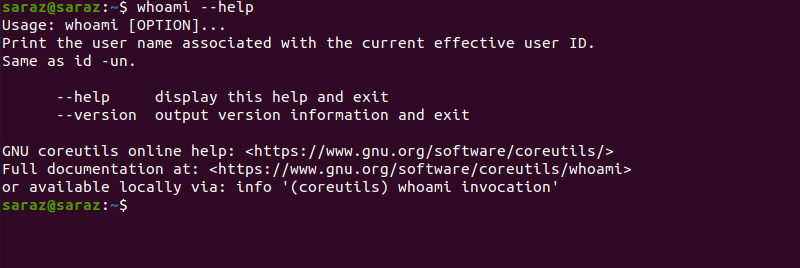
fi



**List All whoami Options**

Execute the following command to show the help message for **whoami** and see all available options:

whoami --help

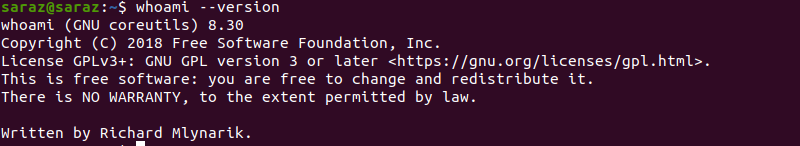


The output prints the usage of the **whoami** command and provides the alternative **id -un** command. Furthermore, the terminal also shows links to the online help page and the full documentation website.

**Check whoami Version and Exit**

To check the version of the **whoami** command, execute:

whoami --version

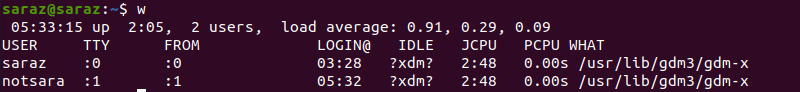


**whoami vs. Other Commands**

Some commands print the same output as **whoami**. The examples below provide the **whoami** alternatives.

**whoami vs. w**

While **whoami** only prints the username of the effective user, the [w command](https://phoenixnap.com/kb/w-command-in-linux) provides more details. That is, the **w** command shows where users are logged in from and what they are currently doing.



**whoami vs. logname**

Both **whoami** and **logname** show the name of the current user.



The difference is that while the **whoami** command shows the effective user, the **logname** command only returns the username.

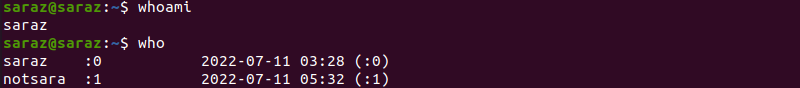
Test this by running the commands with **sudo**:



When using **sudo** with **logname**, the command returns the name of the current user. However, if you execute **sudo** with **whoami**, the command returns root.

**whoami vs. who**

[The who command](https://phoenixnap.com/kb/linux-who-command) returns info about all logged-in users (not only the effective users). The command provides not only usernames but also shows their terminal line and the time they logged in.



**whoami vs. id**

When executing the **id** command without any options, it returns the currently logged-in user details, like the user id, the group id, and the list of groups the user is in.

id Command Terminal Output

However, if you execute the **id** command with the **-un** option, the output is the same as with **whoami**.

