**Create new User and grant admin permission (wheel/sudo)**

Default admin group is wheel on CentOS and Rehl. However default group is sudo on Ubuntu

Follow the steps below to create a sudo user on your CentOS server.

1. **Start by logging in to your server as the root user from different machine**

ssh root@server\_ip\_address

1. **Create a new user account using the useradd/adduser command.**

useradd username : password need to set seperatly

adduser username : will ask pwd

Replace username with the user name that you want to create.

Once a new user created, it’s entry automatically added to the ‘**/etc/passwd**‘ file. The file is used to store users information and the entry should be.

1. **Use the passwd command to set a password for the new user.**

passwd username

You will be prompted to confirm the password. Make sure you use a strong  
password.

Changing password for user username.

New password:

Retype new password:

passwd: all authentication tokens updated successfully.

1. Add the new user to the wheel group.

usermod -aG wheel username

or

usermod -aG sudo username

**How to use Sudo**

Switch to the newly created user:

su - username

To use sudo, simply prefix the command with sudo and space.

sudo [COMMAND]

### Add a User to Multiple Groups

The ‘**-G**‘ option is used to add a user to additional groups. Each group name is separated by a comma, with no intervening spaces.

Here in this example, we are adding a user ‘**tecmint**‘ into multiple groups like **admins**, **webadmin** and **developer**.

[root@tecmint ~]

# useradd -G admins,webadmin,developers tecmint

---- EC2 ssh key setup

Add a new user to the EC2 Linux instance

1.    [Connect to your Linux instance using SSH](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AccessingInstancesLinux.html).

2.    Use the adduser command to add a new user account to an EC2 instance (replace new\_user with the new account name). The following example creates an associated group, home directory, and an entry in the /etc/passwd file of the instance:

$ sudo adduser new\_user

Note: If you add the new\_user to an Ubuntu instance, include the --disabled-password option to avoid adding a password to the new account:

$ sudo adduser new\_user --disabled-password

3.    Change the security context to the new\_user account so that folders and files you create will have the correct permissions:

$ sudo su - new\_user

Note: When you run the sudo su - new\_user command, the name at the top of the command shell prompt changes to reflect the new user account context of your shell session.

4.    Create a .ssh directory in the new\_user home directory:

mkdir .ssh

5.    Use the chmod command to change the .ssh directory's permissions to 700. Changing the permissions restricts access so that only the new\_user can read, write, or open the .ssh directory.

chmod 700 .ssh

6.    Use the touch command to create the authorized\_keys file in the .ssh directory:

touch .ssh/authorized\_keys

7.    Use the chmod command to change the .ssh/authorized\_keys file permissions to 600. Changing the file permissions restricts read or write access to the new\_user.

chmod 600 .ssh/authorized\_keys

1. Switch to the new account so that the directory and file that you will create will have the proper ownership.
2. [ec2-user ~]$ **sudo su - *newuser***

[newuser ~]$

Notice that the prompt changes from ec2-user to newuser to indicate that you have switched the shell session to the new account.

1. Add the SSH public key to the user account. First create a directory in the user's home directory for the SSH key file, then create the key file, and finally paste the public key into the key file.
   1. Create a .ssh directory in the newuser home directory and change its file permissions to 700 (only the owner can read, write, or open the directory).
   2. [newuser ~]$ **mkdir .ssh**

[newuser ~]$ **chmod 700 .ssh**

**Important**

Without these exact file permissions, the user will not be able to log in.

* 1. Create a file named authorized\_keys in the .ssh directory and change its file permissions to 600 (only the owner can read or write to the file).
  2. [newuser ~]$ **touch .ssh/authorized\_keys**

[newuser ~]$ **chmod 600 .ssh/authorized\_keys**

**Important**

Without these exact file permissions, the user will not be able to log in.

* 1. Open the authorized\_keys file using your favorite text editor (such as **vim** or **nano**).

[newuser ~]$ **nano .ssh/authorized\_keys**

Paste the public key for the key pair into the file and save the changes. For example:

ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQClKsfkNkuSevGj3eYhCe53pcjqP3maAhDFcvBS7O6V

hz2ItxCih+PnDSUaw+WNQn/mZphTk/a/gU8jEzoOWbkM4yxyb/wB96xbiFveSFJuOp/d6RJhJOI0iBXr

lsLnBItntckiJ7FbtxJMXLvvwJryDUilBMTjYtwB+QhYXUMOzce5Pjz5/i8SeJtjnV3iAoG/cQk+0FzZ

qaeJAAHco+CY/5WrUBkrHmFJr6HcXkvJdWPkYQS3xqC0+FmUZofz221CBt5IMucxXPkX4rWi+z7wB3Rb

BQoQzd8v7yeb7OzlPnWOyN0qFU0XA246RA8QFYiCNYwI3f05p6KLxEXAMPLE

The user should now be able to log into the newuser account on your instance using the private key that corresponds to the public key that you added to the authorized\_keys file.