Managing User Accounts on Your Linux Instance

Each Linux instance type launches with a default Linux system user account. For Amazon Linux 2 or the Amazon Linux, the user name is `ec2-user`. For Centos, the user name is `centos`. For Debian, the user name is `admin` or `root`. For Fedora, the user name is `ec2-user` or `fedora`. For RHEL, the user name is `ec2-user` or `root`. For SUSE, the user name is `ec2-user` or `root`. For Ubuntu, the user name is `ubuntu`. Otherwise, if `ec2-user` and `root` don't work, check with your AMI provider.

> **Note**
>
> Linux system users should not be confused with AWS Identity and Access Management (IAM) users. For more information, see IAM Users and Groups (https://docs.aws.amazon.com/IAM/latest/UserGuide/Using_WorkingWithGroupsAndUsers.html) in the *IAM User Guide*.

Using the default user account is adequate for many applications, but you may choose to add user accounts so that individuals can have their own files and workspaces. Creating user accounts for new users is much more secure than granting multiple (possibly inexperienced) users access to the `ec2-user` account, because that account can cause a lot of damage to a system when used improperly.

After you add the user account, you must set up access keys that allow the user to log in.

Prerequisites

Create a key pair for the user or use an existing key pair. For more information, see Creating a Key Pair Using Amazon EC2 (ec2-key-pairs.html#having-ec2-create-your-key-pair) . To retrieve a public key from an existing key pair, see Retrieving the Public Key for Your Key Pair on Linux (ec2-key-pairs.html#retrieving-the-public-key) .

To add a user account

1. Use the following **adduser** command to add the `newuser` account to the system (with an entry in the `/etc/passwd` file). This command also creates a group and a home directory for the account.

   ```
   [ec2-user ~]$ sudo adduser newuser
   ```

   [Ubuntu] When adding a user to an Ubuntu system, include the `--disabled-password` option with this command to avoid adding a password to the account.

   ```
   [ubuntu ~]$ sudo adduser newuser --disabled-password
   ```

2. Switch to the new account so that newly created files have the proper ownership.

   ```
   [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.

3. 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).

   ```
   [newuser ~]$ mkdir .ssh
   [newuser ~]$ chmod 700 .ssh
   ```

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

4. 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).

   ```
   [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.

5. 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 your 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.

To remove a user from the system

If a user account is no longer needed, you can remove that account so that it may no longer be used. When you specify the `-r` option, the user's home directory and mail spool are deleted. To keep the user's home directory and mail spool, omit the `-r` option.

```
[ec2-user ~]$ sudo userdel -r olduser
```