fedora
remix

Learning Linux

Creating a Fedora Remix

Travis Michette

Table of Contents

Fedora Remix Creation Quickstart 1
Quickstart 2
Appendix A: Fedora Remix 10
A.1. Fedora Remix Application Repositories 10
A.2. Fedora Remix Extensions and Additional Projects 11

Fedora Remix Creation Quickstart

This quickstart will provide a quick method to download a pre-built Fedora Remix as well as creating your own Fedora Remix Building system. The https://github.com/tmichett/Fedora_Remix has been created with Ansible playbooks that will setup a Fedora Remix Building system using the Fedora Remix installed from the liveCD provided (https://drive.google.com/drive/folders/1UAT07AJIrTdMk3ke_QE6S6vz-qn6qGtR). The requirements are relatively simply to build and the system has Fedora Remix customization scripts and self-launching menu to assist with the initial setup. The quickstart and documentation are also available at the following link: https://tmichett.github.io/Fedora_Remix/.

Building your own Fedora Remix - Overview

1. Download the Fedora Remix ISO and install locally to hard drive
    a. Setup a local user and ensure it is in the SUDOERS file
    b. Run some the Fedora Remix customize scripts to create SSH keys and SUDO with no password
2. Clone https://github.com/tmichett/Fedora_Remix
3. Run the Ansible playbooks with the "sudo" command
    a. Prepare_Web_Files.yml
    b. Prepare_Fedora_Remix_Build.yml
4. Modify the Kickstart files in **/livecd-creator/FedoraRemix**
    a. FedoraRemix.ks - To control various setup components and run commands
    b. FedoraRemixPackages.ks - To control RPM packages that are installed as part of your Remix
5. Run the **Remix_Build_Script.sh**

Error Regarding kickstart.py

With newer Fedora releases, there is an error with the **kickstart.py** package and URLGrabber. Located on the "FedoraRemixTools" application, I've included a button to "FIX" the issue with the **kickstart.py** package. After completing the fix, you will no longer get the issue.

Listing 1. Python kickstart.py Errors

```
Traceback (most recent call last):
  File "/usr/sbin/livecd-creator", line 265, in <module>
    sys.exit(main())

... OUTPUT OMITTED ...
```

Quickstart

This quickstart provides detailed information on how to download, install, and create your custom Fedora Remix. It depends first on having the Fedora Remix installed locally and access to the Github repository and playbooks contained withing the repository.

1. Download the "Latest" FedoraRemix LiveISO and create a bootable USB disk.
    - https://drive.google.com/drive/folders/1UAT07AJIrTdMk3ke_QE6S6vz-qn6qGtR
        - Use Balena Etcher or Fedora Media Writer to create the bootable USB device for installation on a physical machine.
        - Attach the ISO image to your virtual machine and select it as the boot device so it can be installed
2. Insert the disk into a computer or connect to the virtual machine and select

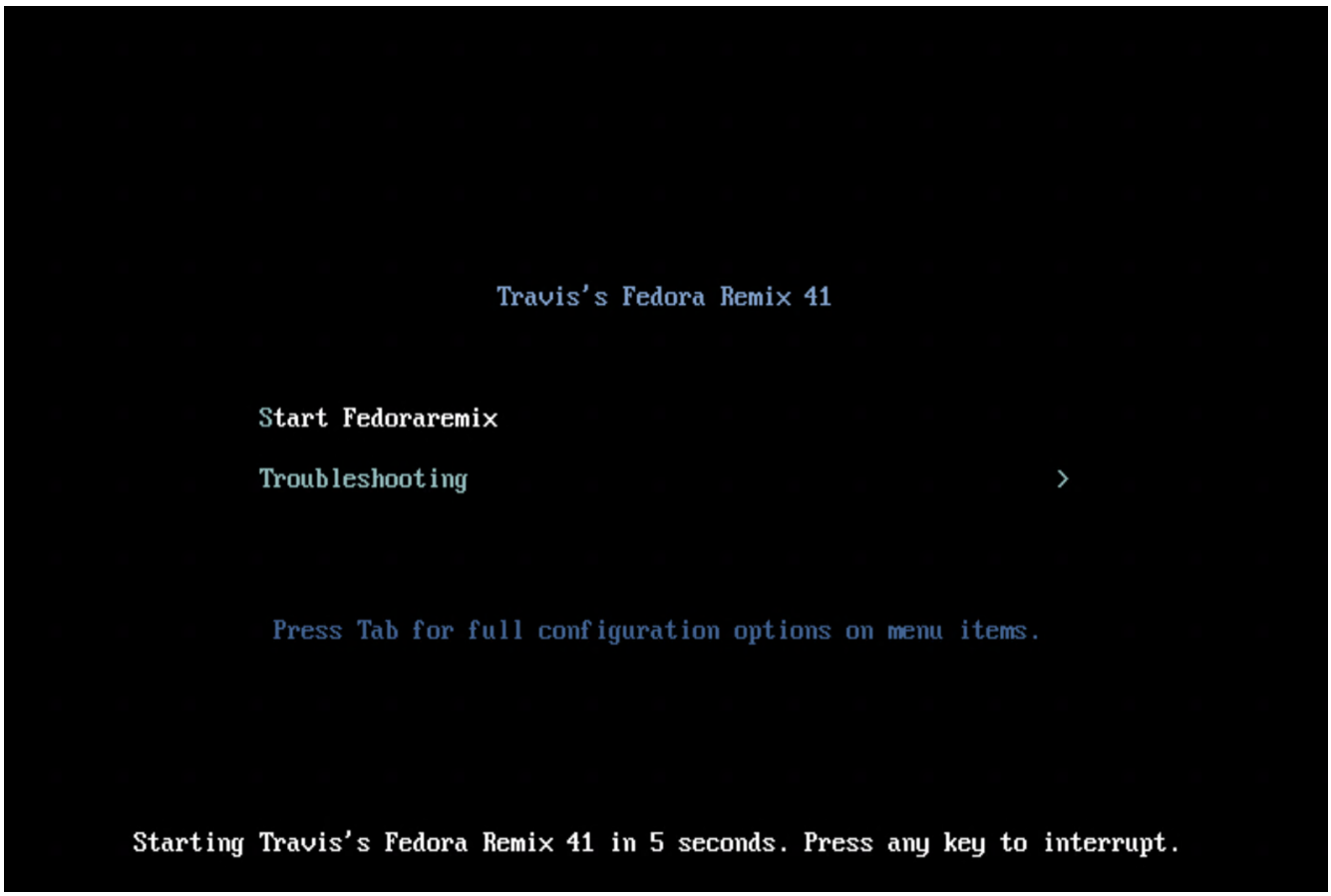

Figure 1. GRUB Boot Menu

3. Select "Start Fedoraremix" and hit Enter
4. Once booted to the main screen, click on the icon in the top left beside of "Apps" to bring up the application dock. Select the icon from the dock

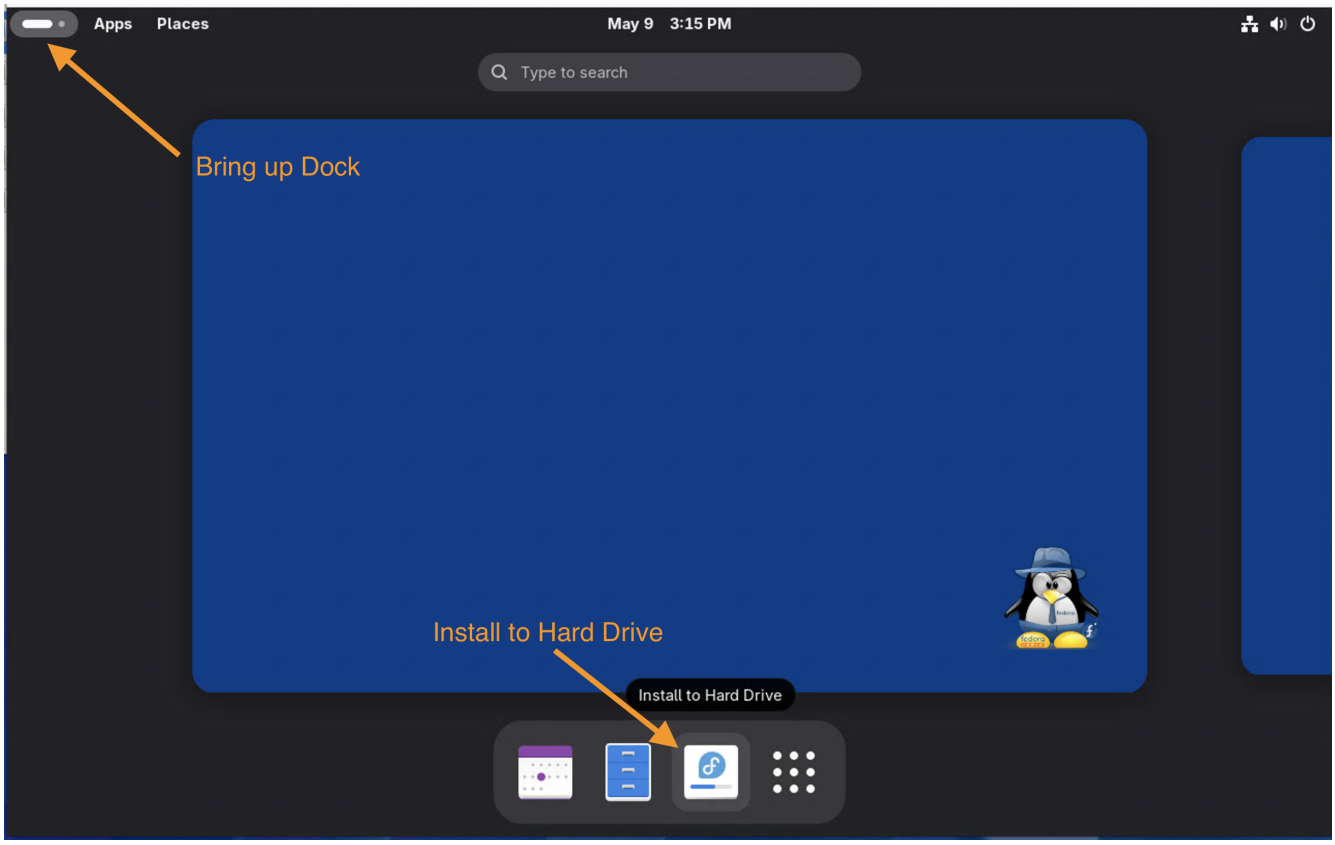

Figure 2. Fedora Remix LiveCD Desktop

5. The Anaconda graphical installer will appear. Select your desired language and optionally select the keyboard layout.

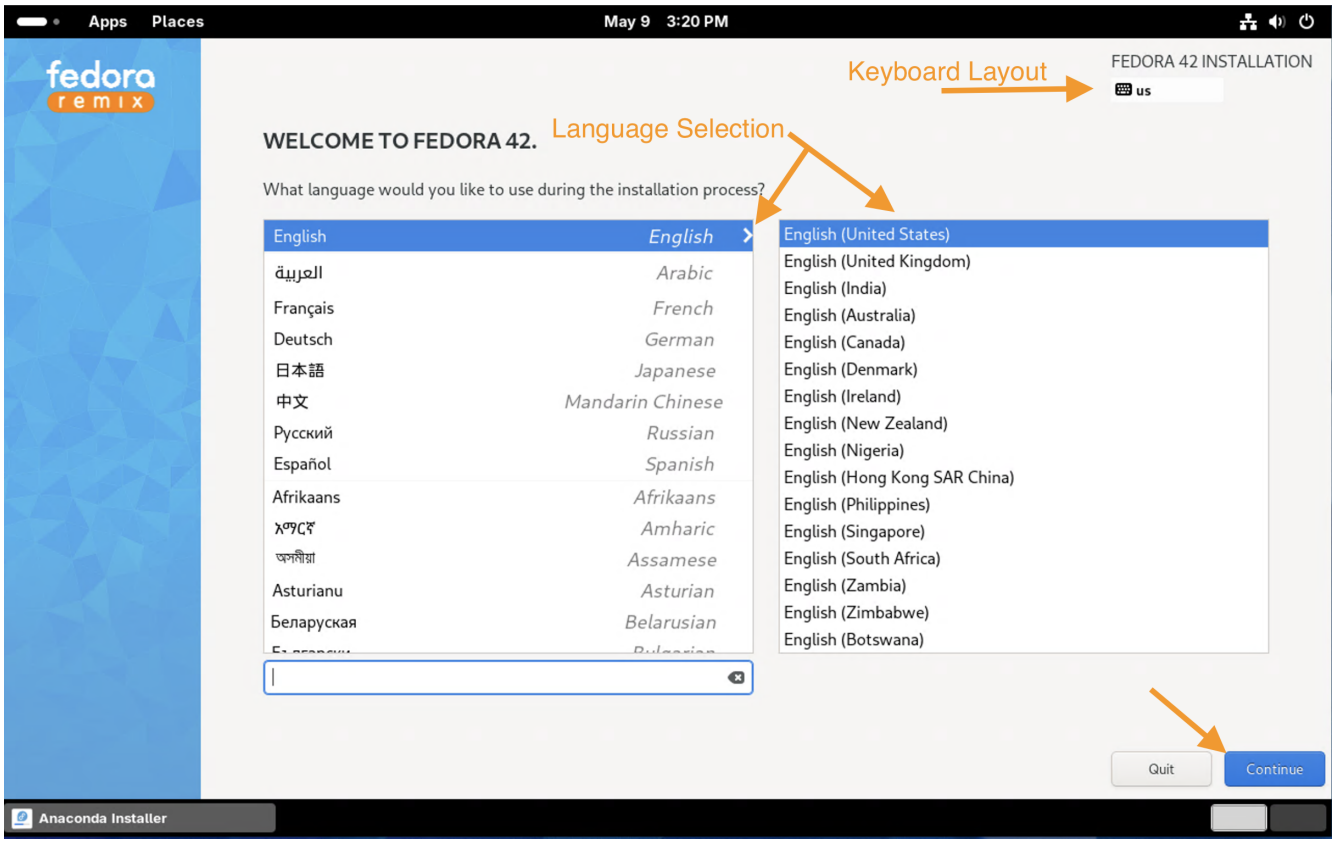

Figure 3. Anaconda Installer - Language and Keyboard

6. Complete the required and optional installation components.

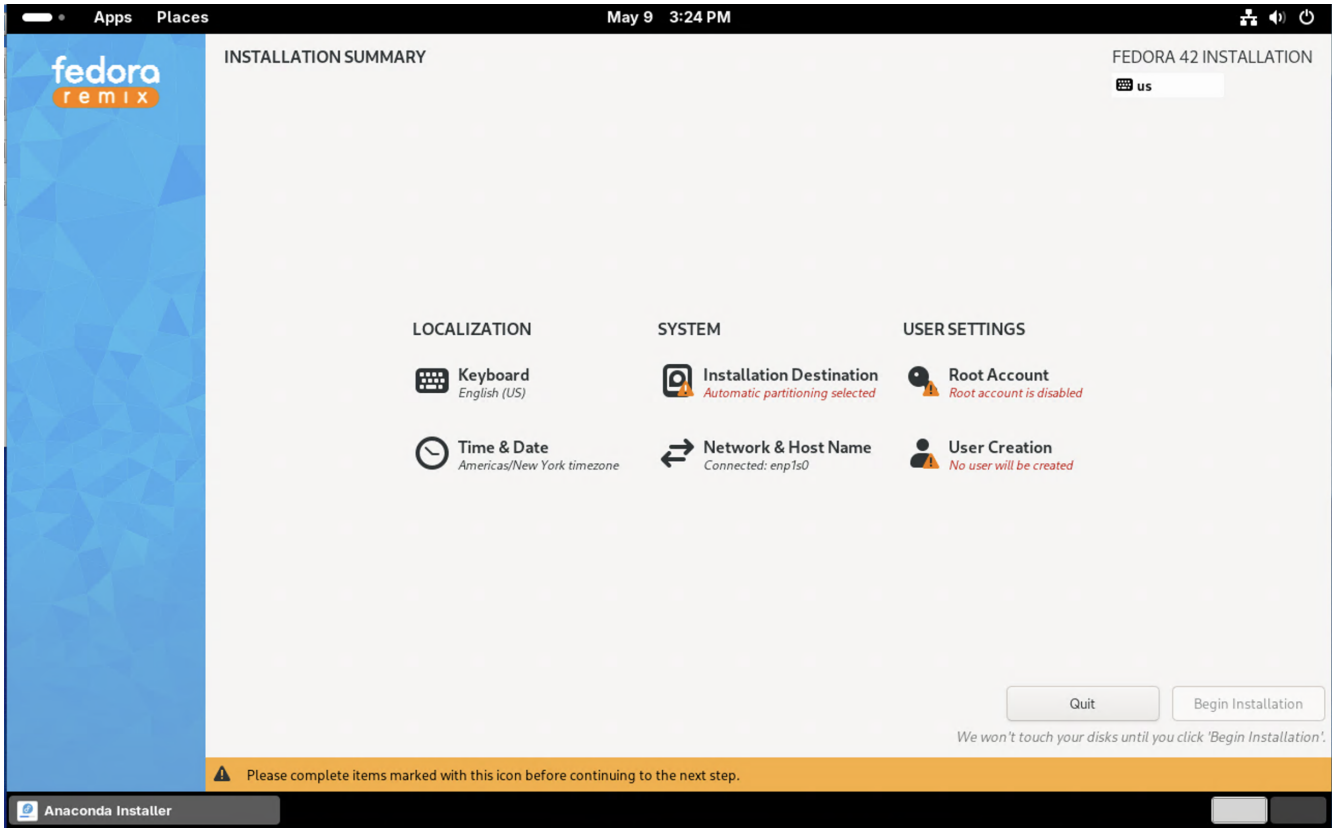

Figure 4. Anaconda Installer - Installation Summary

a. Select "Installation Destination" and choose an appropriate installation device.

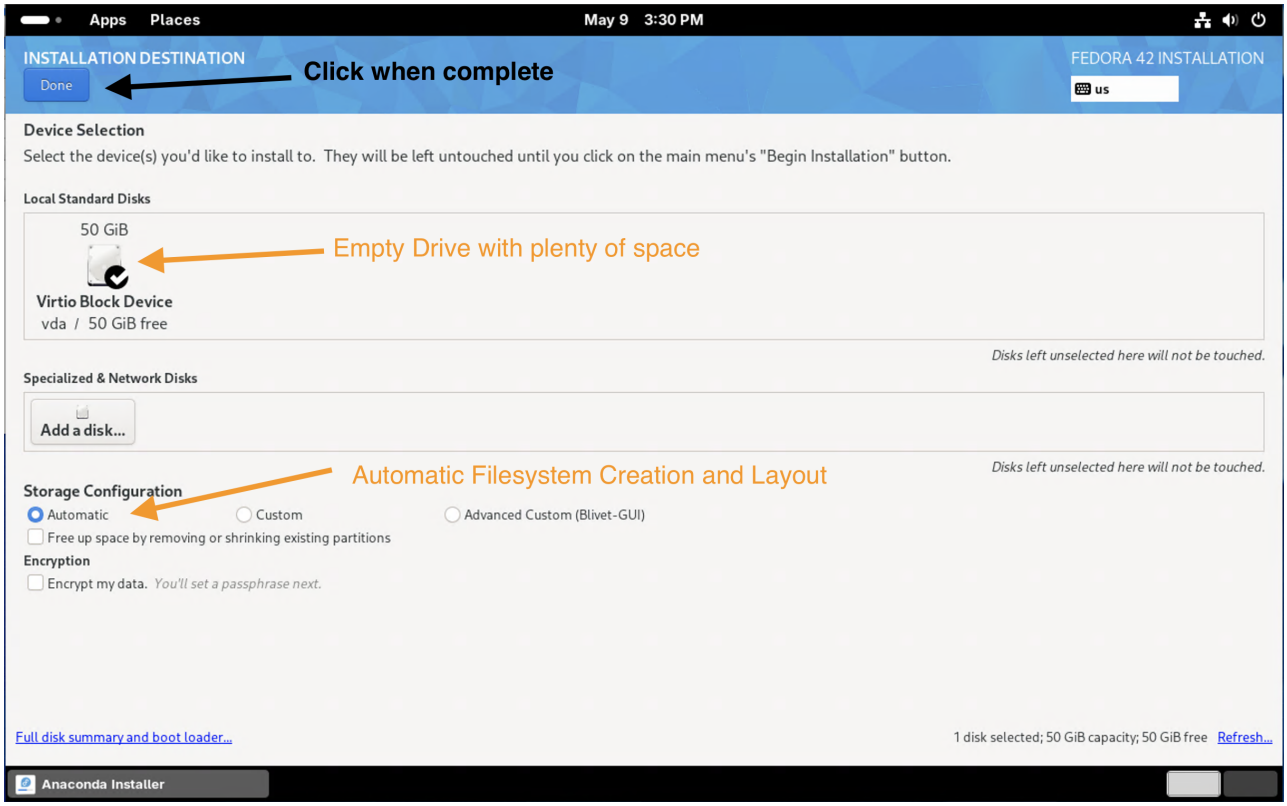

Figure 5. Setting Installation Location

b. Create a user account by clicking "User Creation". Complete the required information and then click done.

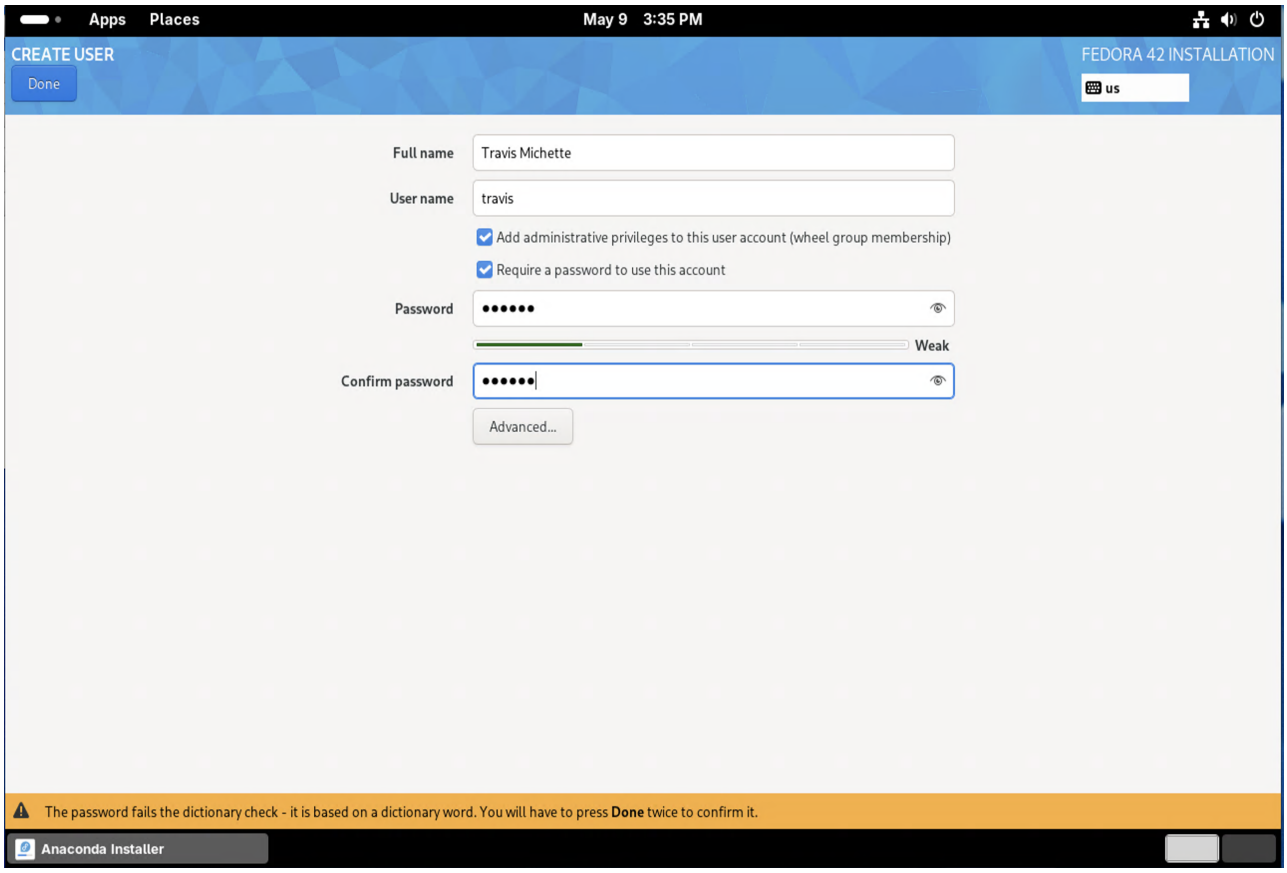

Figure 6. User Creation

Root Account

After you have specified the installation location and an initial user, you may proceed with the installation. Creating and enabling the **root** user is optional and not required. If you create the root user, likewise, you are not required to create a regular user account.

Setup Complete

After the installation is completed, you will need to reboot your system and perform package updates and any other needed tasks.

7. Clone the Fedora Remix Repository

```
travis@localhost-fedoraremix-live [~] | (05/09/25 @ 15:55)
$ git clone https://github.com/tmichett/Fedora_Remix.git
Cloning into 'Fedora_Remix'...
remote: Enumerating objects: 1949, done

... OUTPUT OMITTED ...

Resolving deltas: 100% (1042/1042), done.
```

8. Prepare the Fedora Remix build environment by running the Python scripts from the **Ansible** directory as the root user.

Listing 2. Create /livecd-creator/FedoraRemix Assets

```
travis@localhost-fedoraremix-live [~/Fedora_Remix/Ansible] | (05/09/25 @ 15:57)  |
main ✓ |
$ sudo python Prepare_Fedora_Remix_Build.py

... OUTPUT OMITTED ...

Copying ../Remix_Buid_Script.sh to /livecd-creator/FedoraRemix/Remix_Buid_Script.sh
Setup complete!
```

Listing 3. Create httpd server and repository assets

```
travis@localhost-fedoraremix-live [~/Fedora_Remix/Ansible] | (05/09/25 @ 15:59)  |
main ✓ |
$ sudo python Prepare_Web_Files.py
Installing packages: httpd
Running command: dnf install -y httpd

... OUTPUT OMITTED ...

Setup complete!
```

9. Perform System and Package customizations by modifying the **FedoraRemix.ks** kickstart file and the **FedoraRemixPackages.ks** file.

   a. The setup scripts or playbooks created **/livecd-creator/FedoraRemix** directory. This is where the kickstart files live that you will modify as well as the directory where the FedoraRemix.iso will reside once built.

Listing 4. FedoraRemix.ks

```
... OUTPUT OMITTED ...

## Add Fedora Dynamic MotD Script
cd /usr/bin
wget http://localhost/files/fedora-dynamic-motd.sh
chmod +x /usr/bin/fedora-dynamic-motd.sh
echo /usr/bin/fedora-dynamic-motd.sh >> /etc/profile


## Customize BASH Prompts and Shell
mkdir /opt/bash
cd /opt/bash
```

```
wget http://localhost/files/bashrc.append
## Install Gitprompt
git clone https://github.com/tmichett/bash-git-prompt.git /opt/bash-git-prompt
--depth=1

... OUTPUT OMITTED ...
```

Listing 5. FedoraPackages.ks

```
... OUTPUT OMITTED ...

## Image Editing and Manipulation
inkscape
gimp
krita
netpbm-progs
scribus

## Video Editing and Manipulation
kdenlive

## Container Tools
buildah
skopeo
podman-machine

... OUTPUT OMITTED ...
```

10. Execute the **Remix_Build_Script.sh** file to kick off the build. Depending on system and Internet connection speed and customizations to the kickstart file, this process could take up to an hour.

Listing 6. Launching the Script

```
travis@localhost-fedoraremix-live [/livecd-creator/FedoraRemix] | (05/09/25 @ 16:09)
$ time ./Remix_Buid_Script.sh
/usr/lib/python3.13/site-packages/pykickstart/commands/partition.py:461:
KickstartParseWarning: A partition with the mountpoint / has already been defined.
  warnings.warn(_("A partition with the mountpoint %s has already been defined.") %
pd.mountpoint, KickstartParseWarning)
fedora                                           19 MB/s |  35 MB     00:01
updates                                         3.8 MB/s | 5.8 MB     00:01
google-chrome                                   7.5 kB/s | 1.9 kB     00:00

... OUTPUT OMITTED ...

Pass 4: Checking reference counts
```

```
Pass 5: Checking group summary information
_FedoraRemix: 371986/1324512 files (0.2% non-contiguous), 4248977/5294080 blocks

 e2fsck 1.47.1 (20-May-2024)

 0

real    34m41.254s
user    143m42.195s
sys     5m11.054s
```

Error Regarding kickstart.py

With newer Fedora releases, there is an error with the **kickstart.py** package and URLGrabber. Located on the "FedoraRemixTools" application, I've included a button to "FIX" the issue with the **kickstart.py** package. After completing the fix, you will no longer get the issue.

Listing 7. Python kickstart.py Errors

```
Traceback (most recent call last):
  File "/usr/sbin/livecd-creator", line 265, in <module>
    sys.exit(main())
             ~~~~^^
  File "/usr/sbin/livecd-creator", line 195, in main
    ks = imgcreate.read_kickstart(options.kscfg)
  File "/usr/lib/python3.13/site-packages/imgcreate/kickstart.py", line
54, in read_kickstart
    ksfile = urlgrabber.urlgrab(path, filename=tmpks)
  File "/usr/lib/python3.13/site-packages/urlgrabber/grabber.py", line
808, in urlgrab
    return default_grabber.urlgrab(url, filename, **kwargs)
           ~~~~~~~~~~~~~~~~~~~~~~~^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/lib/python3.13/site-packages/urlgrabber/grabber.py", line
1205, in urlgrab
    (url,parts) = opts.urlparser.parse(url, opts)
                  ~~~~~~~~~~~~~~~~~~~~^^^^^^^^^^^^
  File "/usr/lib/python3.13/site-packages/urlgrabber/grabber.py", line
873, in parse
    pathname = pathname2url(url)
  File "/usr/lib64/python3.13/urllib/request.py", line 1679, in
pathname2url
    return quote(pathname, encoding=encoding, errors=errors)
  File "/usr/lib64/python3.13/urllib/parse.py", line 924, in quote
    raise TypeError("quote() doesn't support 'encoding' for bytes")
TypeError: quote() doesn't support 'encoding' for bytes
```

Appendix A: Fedora Remix

This Fedora Remix has been built and assembled from many projects and continues to evolve and become simplified. The various Github repositories are meant to allow easier updates to individual Fedora Remix components and to allow certain components to be used more in a utility fashion where they can be used without impacting the Fedora Remix project.

A.1. Fedora Remix Application Repositories

This project was initially 100% part of the Fedora Remix repository using various BASH scripts, Ansible Playbooks, Python tools, and various applications like YAD (Yet Another Dialog). As the project grew, some of the components have been separated out into their own repositories for use in the creation of the Fedora Remix.

- **Log Viewer** - (https://github.com/tmichett/log_viewer)

  The Log Viewer was developed to look at and analyze the output of the kickstart file. During development and updates to the Fedora Remix build process as part of Fedora 41/42, the kickstart and build files became extremely large and it was difficult to identify errors and successes within a huge file, so the Log Viewer was created to look for keywords in that file and assist with the building and troubleshooting process. Using Github actions and workflows, this builds the Python application as a Linux executable with all dependencies as well as produces an RPM for easy installation in Fedora Remix.

- **PXE Server** - (https://github.com/tmichett/PXEServer)

  The PXEServer repository was created to separate just the PXE-boot components from Fedora Remix. This may later be replaced as it will be a modular component and it was designed to allow a "lab" or "classroom-style" deployment where Fedora Remix would be installed to a single system and the PXE server could provide the Fedora Remix Live image across the network via PXE without ever installing anything to the remaining computers. This repository contains the items and automation for configuring PXE boot services and identifying the components.

- **Fedora Remix Customize** - (https://github.com/tmichett/FedoraRemixCustomize)

  The Fedora Remix Customize repository contains some of the user-based customizations. It was created as a separate repository for tracking playbooks and scripts that would further customize the environment. Eventually, this will be the location of many more customization utilities as the build and deployment process gets more streamlined.

- **Fedora Remix Tools** - (https://github.com/tmichett/Fedora_Remix_Tools)

  The Fedora Remix Tools repository brings a new take to the **Fedora Remix Apps** which was the first set of tools and customizations. The Fedora Remix Apps used YAD and was a set of BASH scripts and playbooks located in the main Fedora Remix repository in the YAD folder

(https://github.com/tmichett/Fedora_Remix/tree/main/YAD). The implementation and location of YAD made things more difficult to test and maintain as well as add new features. With the release of Fedora 42, a newer Python application has been developed. This application has a dynamic menu system that can be controlled by changes in a configuration file (config.yml) that allows simple text changes to change and modify the application look and feel. The newer Fedora Remix Tools application also provides a *command output window* which can be detached and looked at so a user knows if there were errors or issues running any of the customizations. The Fedora Remix Tools repository also uses Github actions and workflows to build an RPM which can be easily installed on the Fedora Remix system.

- **OhMyBash** - (https://github.com/tmichett/oh-my-bash)

  The **OhMyBash** repository was forked from the main repository but customized for Fedora Remix. This repository is used to install OhMyBash as part of the Fedora Remix built process and perform the various customizations to the terminal and BASH shell including configuration and creation of a custom theme developed for the Fedora Remix (https://github.com/tmichett/oh-my-bash/tree/master/themes/tm-fedora-remix).

A.2. Fedora Remix Extensions and Additional Projects

There are two additional projects inspired by the Fedora Remix work which will be coming soon to Fedora Remix or stand slightly apart from Fedora Remix. The idea is to look at making a Live boot USB more interactive and not waste an entire thumb drive as well as experiment with PXE and other distribution methods including BootC container images.

- **Fedora BootC** - (https://github.com/tmichett/Fedora_BootC)

  The Fedora BootC repository is in initial testing and development stages and is meant to be picked up later in 2025 to extend the customizations and create an image-based installer that can be used with the Fedora Remix PXE process that will allow a kickstart file to provision systems in lab or classroom-style environments rather than just a live image. This repository will also likely extend into research and initial starting point for creating a WSL image of Fedora Remix to run on Windows systems.

- **Fedora Remix Ventoy** - (https://github.com/tmichett/Fedora_Remix_Ventoy)

  The Ventoy application has been around and is available for Windows and Linux allowing users to build USB drives with multiple ISOs creating a multi-boot USB. Ventoy can also be customized with many components, but it must be installed and users must know how to use Ventoy. The Fedora Remix Ventoy repository has been created to allow an RPM installer to bake in Ventoy into the Fedora Remix Live image. It also has a Python GUI application that is meant to bring up some of the Ventoy customization components without needing to know all the commands, procedures, and processes in an attempt to make things more accessible.