

Lab 1: Changing Username and Installing Docker Engine

Part A: Changing Username (Commands Only)

1. Create the new user (ajitlab):
sudo useradd -m ajitlab
2. Set password for the new user:
sudo passwd ajitlab
3. Add ajitlab to the wheel (sudo) group:
sudo usermod -aG wheel ajitlab
4. Change the original username (ajith to ajith1):
sudo usermod -l ajith1 ajith
5. Move the old home directory to the new name:
sudo mv /home/ajith /home/ajith1

```
Start Hypriand with command Hypriand
ajith A ~
>> sudo useradd -m ajith1
[sudo] password for ajith:
ajith A ~
>> sudo passwd ajith1
New password:
Retype new password:
passwd: password updated successfully
ajith A ~
>> sudo usermod -aG wheel ajith1
ajith A ~
>> id ajith1
uid=1003(ajith1) gid=1003(ajith1) groups=1003(ajith1),998(wheel)
ajith A ~
>> ls /home
ajith  ajith1  labuser  tempuser
ajith A ~
>> sudo usermod -l ajithlab ajith1
ajith A ~
>> sudo mv /home/ajith1 /home/ajithlab
ajith A ~
>> sudo chown -R ajithlab:ajithlab /home/ajithlab
chown: invalid group: 'ajithlab:ajithlab'
ajith A ~
>> id ajithlab
uid=1003(ajithlab) gid=1003(ajith1) groups=1003(ajith1),998(wheel)
ajith A ~
>> sudo chown -R ajithlab:<primarygroup> /home/ajithlab
bash: primarygroup: No such file or directory
ajith A ~
>> sudo groupadd ajithlab
ajith A ~
>> sudo usermod -g ajithlab ajithlab
ajith A ~
>> sudo chown -R ajithlab:ajithlab /home/ajithlab
ajith A ~
>> sudo chown -R ajithlab:ajithlab /home/ajithlab
ajith A ~
>> id ajithlab
uid=1004(ajithlab) gid=1004(ajithlab) groups=1004(ajithlab),998(wheel)
ajith A ~
drwx----- - ajith 17 Oct 09:41 ajith
drwx----- - ajithlab 17 Oct 09:41 ajithlab
drwx----- - labuser 17 Oct 09:36 labuser
drwx----- - tempuser 15 Oct 15:10 tempuser
ajith A ~
```

```
~ - flatpak-spawn - Konsole
New Tab Split View
Copy Paste Find...

ajith A ~
>> id ajith1
uid=1003(ajith1) gid=1003(ajith1) groups=1003(ajith1),998(wheel)

ajith A ~
>> ls /home
ajith ajith1 labuser tempuser

ajith A ~
>> sudo usermod -l ajithlab ajith1

ajith A ~
>> sudo mv /home/ajith1 /home/ajithlab

ajith A ~
>> sudo chown -R ajithlab:ajithlab /home/ajithlab
chown: invalid group: 'ajithlab:ajithlab'

ajith A ~
>> id ajithlab
uid=1003(ajithlab) gid=1003(ajith1) groups=1003(ajith1),998(wheel)

ajith A ~
>> sudo chown -R ajithlab:<primarygroup> /home/ajithlab
bash: primarygroup: No such file or directory

ajith A ~
>> sudo groupadd ajithlab

ajith A ~
>> sudo usermod -g ajithlab ajithlab

ajith A ~
>> sudo chown -R ajithlab:ajithlab /home/ajithlab

ajith A ~
>> sudo chown -R ajithlab:ajithlab /home/ajithlab

ajith A ~
>> id ajithlab
uid=1003(ajithlab) gid=1004(ajithlab) groups=1004(ajithlab),998(wheel)
ls -l /home
drwx----- - ajith 17 Oct 09:41 ajith
drwx----- - ajithlab 17 Oct 09:41 ajithlab
drwx----- - labuser 17 Oct 09:36 labuser
drwx----- - tempuser 15 Oct 15:10 tempuser

ajith A ~
>> whoami
ajith

ajith A ~
>> id ajithlab
uid=1003(ajithlab) gid=1004(ajithlab) groups=1004(ajithlab),998(wheel)

ajith A ~
>> su - ajithlab
su: warning: cannot change directory to /home/ajith1: No such file or directory
[ajithlab@archlinux ajith]$ whoami
ajithlab
[ajithlab@archlinux ajith]$ pwd
/home/ajith
[ajithlab@archlinux ajith]$
```

6. Update the ownership of the new home directory:

sudo chown -R ajith1:ajitlab /home/ajith1

7. Verify user IDs:

id ajith1

8. List home directory contents:

ls -l /home

9. Switch to the new user:

su - ajitlab

10. Confirm current user:

whoami

Part B: Installing and Testing Docker Engine (Commands Only)

1. Install the Docker package:
sudo pacman -S docker
2. Check Docker version (daemon will be stopped):
sudo docker version

```
Start Hyprland with command Hyprland

ajith@ ~$ sudo pacman -S docker
[sudo] password for ajith:
resolving dependencies...
looking for conflicting packages...

Packages (3) containerd-2.1.4-1  runc-1.3.2-1  docker-1:28.5.1-1
Total Installed Size: 194.67 MiB

:: Proceed with installation? [Y/n] y
(2/3) checking keys in keyring
(2/3) checking package integrity
(3/3) loading package files
(2/3) checking for file conflicts
(2/3) checking available disk space
:: Processing package changes...
(1/3) installing runc
Optional dependencies for runc
  criu: checkpoint support
(2/3) installing containerd
(2/3) installing docker
Optional dependencies for docker
  btrfs-progs: btrfs backend support
  pigz: parallel gzip compressor support
  docker-buildx: extended build capabilities
:: Running post-transaction hooks...
(1/4) Creating system user accounts...
(2/4) Reloading system manager configuration...
(3/4) Reloading device manager configuration...
(4/4) Arming ConditionNeedsUpdate...

ajith@ ~$ sudo docker version
Client:
Version: 28.5.1
API version: 1.51
Go version: go1.25.2 X:nodwarf5
Git commit: e180ab8ab8
Built: Wed Oct 8 17:17:33 2025
OS/Arch: linux/amd64
Context: default
Cannot connect to the Docker daemon at unix:///var/run/docker.sock. Is the docker daemon running?

ajith@ ~$ sudo systemctl start docker.service

ajith@ ~$ sudo systemctl status docker.service
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; disabled; preset: disabled)
   Active: active (running) since Fri 2025-10-17 09:56:17 UTC; 7s ago
   Invocation: 90618da0afa3495195bda2470d5eba0d
   TriggeredBy: docker.socket
   Docs: https://docs.docker.com
   Main PID: 14214 (dockerd)
   Tasks: 14
   Memory: 26.2M (peak: 28.8M)
   CPU: 553ms
```

```
Docs: https://docs.docker.com
Main PID: 14214 (dockerd)
Tasks: 14
Memory: 26.2M (peak: 28.8M)
CPU: 553ms
CGroup: /system.slice/docker.service
└─14214 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Oct 17 09:56:16 archlinux dockerd[14214]: time="2025-10-17T09:56:16.578048169Z" level=info msg="Creating a containerd client" address=/run/containerd/containerd.sock timeout=1m0s
Oct 17 09:56:16 archlinux dockerd[14214]: time="2025-10-17T09:56:16.624793845Z" level=info msg="Loading containers: start."
Oct 17 09:56:16 archlinux dockerd[14214]: time="2025-10-17T09:56:16.958607430Z" level=info msg="Loading containers: done."
Oct 17 09:56:16 archlinux dockerd[14214]: time="2025-10-17T09:56:16.982223039Z" level=warning msg="Not using native diff for overlay2, this may cause degraded performance for building image
s: kernel has CONFIG_OVERLAY_FS_REDIRECT_DIR enabled" storage-driver=overlay2
Oct 17 09:56:16 archlinux dockerd[14214]: time="2025-10-17T09:56:16.982438112Z" level=info msg="Docker daemon" commit=f8215cc266 containerd-snapshotter=false storage-driver=overlay2 version
=28.5.1
Oct 17 09:56:16 archlinux dockerd[14214]: time="2025-10-17T09:56:16.982619071Z" level=info msg="Initializing buildkit"
Oct 17 09:56:17 archlinux dockerd[14214]: time="2025-10-17T09:56:17.010622281Z" level=info msg="Completed buildkit initialization"
Oct 17 09:56:17 archlinux dockerd[14214]: time="2025-10-17T09:56:17.019552397Z" level=info msg="Daemon has completed initialization"
Oct 17 09:56:17 archlinux dockerd[14214]: time="2025-10-17T09:56:17.019679299Z" level=info msg="API listen on /run/docker.sock"
Oct 17 09:56:17 archlinux systemd[1]: Started Docker Application Container Engine.

$ sudo systemctl enable docker.service
Created symlink '/etc/systemd/system/multi-user.target.wants/docker.service' → '/usr/lib/systemd/system/docker.service'.

$ sudo docker run hello-world
docker: permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Head "http://%2Fvar%2Frun%2Fdocker.sock/_ping": dial unix /var/run/docker.sock:
connect: permission denied

Run 'docker run --help' for more information

$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
17ec7b3c9d7: Pull complete
Digest: sha256:6dc565aa630927052111f823c303948cf83670a3903ffa3849f1488ab517f891
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
   (amd64)
3. The Docker daemon created a new container from that image which runs the
   executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
   to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/
```

3. Start the Docker daemon service:

sudo systemctl start docker.service

4. Check the service status (should show 'active (running)'):

sudo systemctl status docker.service

5. Run the test image:

sudo docker run hello-world