**DevOps**

Assignment-1

Version Control

Name – Pihu Khare

SAP ID – 500110072

Roll no. - R2142221371

Batch – 3 (CCVT Non-Hons)

Demonstrate usage of Subversion, Mercurial with commands, explanation, and screenshots/screen recordings.

**Subversion (SVN)**

Apache Subversion (often referred to as SVN) is a version control system that allows you to manage and track changes to your files and directories over time.

Below is a step-by-step guide to install and use Subversion.

**On Linux:**

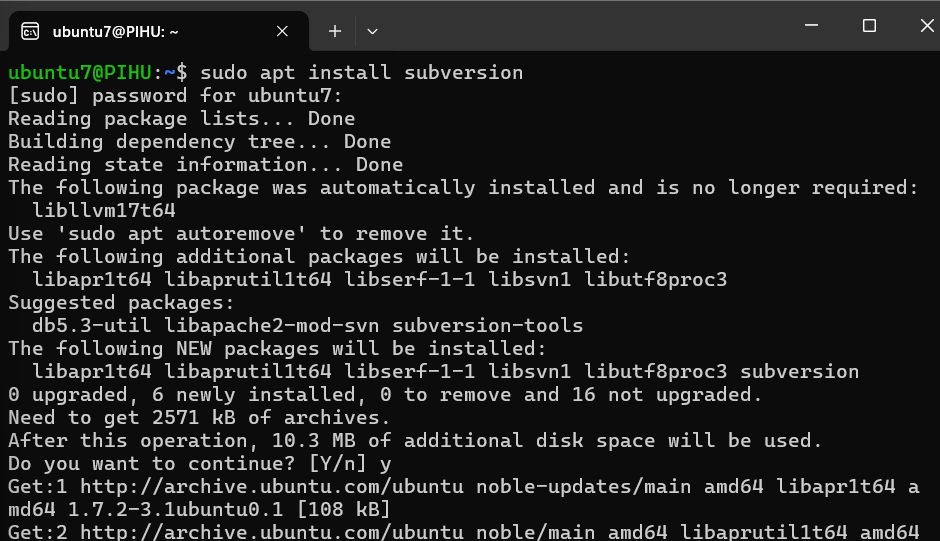
1. Update your package index:

* *sudo apt update # For Debian/Ubuntu*

1. Setting Up SVN Server with svnserve

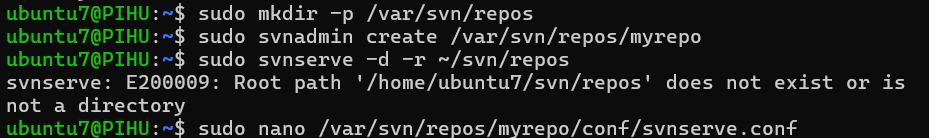
Install SVN

* *sudo apt update*
* *sudo apt install subversion*

mkdi

1. Create a Repository

* *sudo mkdir -p /var/svn/repos*
* *sudo svnadmin create /var/svn/repos/myrepo*



1. Configure svnserve

* Edit the svnserve.conf file to configure access:

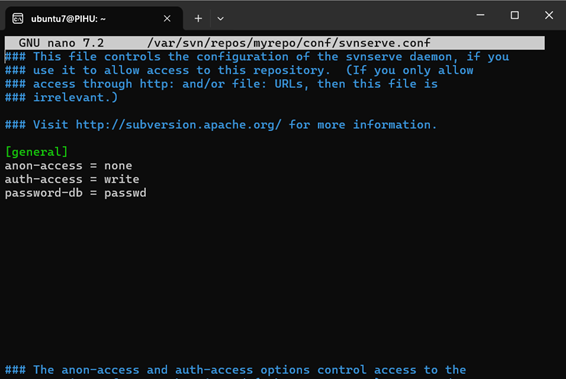
*sudo nano /var/svn/repos/myrepo/conf/svnserve.conf*

* Make the following changes :

anon-access = none # Disable anonymous access

auth-access = write # Allow authenticated users to write

password-db = passwd # Use the passwd file for authentication



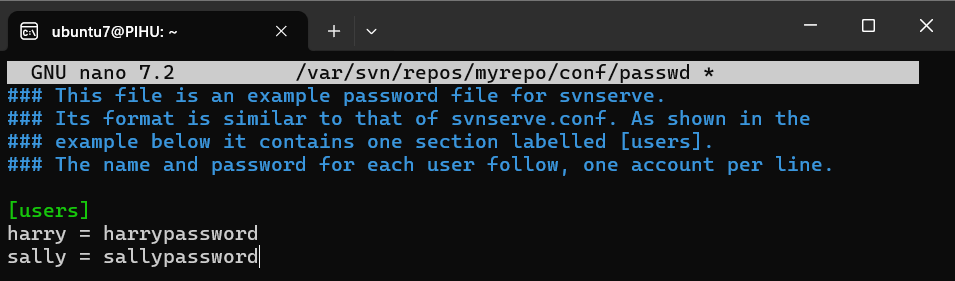
1. Set Up Users

* Edit the **passwd** file to add users:
* Add a user:

harry = harrypassword

sally = sallypassword



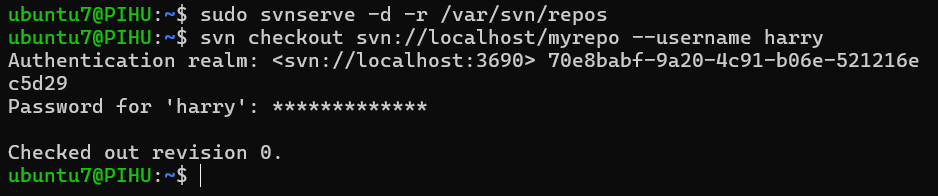


1. Start svnserve

* Start the svnserve daemon:
* *sudo svnserve -d -r /var/svn/repos*

1. Check Out a Repository

* *svn checkout svn://localhost/myrepo --username harry*



1. Add Files to the Repository

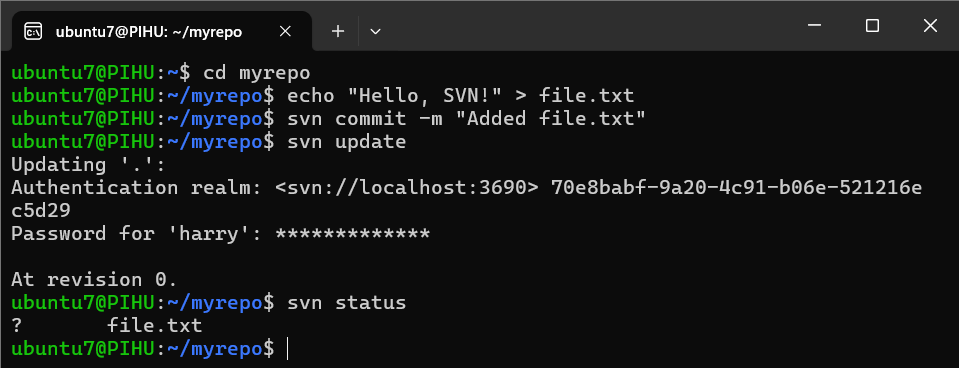
* *cd myrepo*
* echo "Hello, SVN!" > file.txt
* svn add file.txt
* svn commit -m "Added file.txt"

1. Update Your Working Copy

* *svn update*

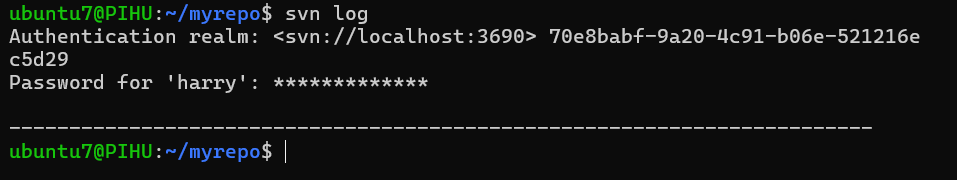
1. View Repository Status

* *svn status*



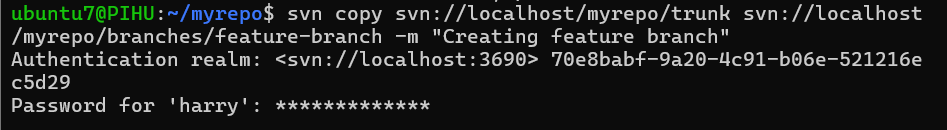
1. View Change History

* *svn log*



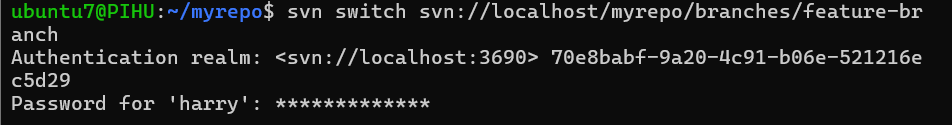
1. Create a Branch for new development

* *svn copy svn://localhost/myrepo/trunk svn://localhost/myrepo/branches/feature-branch -m "Creating feature branch"*



1. Switch to a Branch

* *svn switch svn://localhost/myrepo/branches/feature-branch*

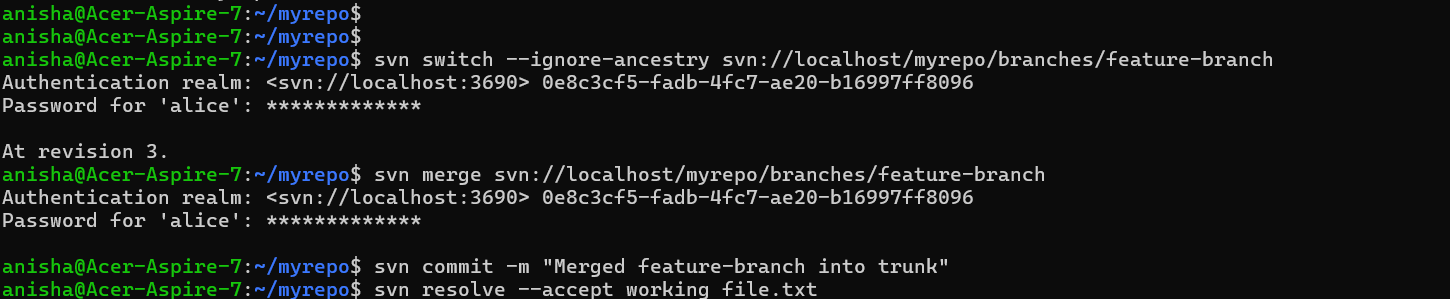
**

1. Merge Changes

* *svn merge svn://localhost/myrepo/branches/feature-branch*
* *svn commit -m "Merged feature-branch into trunk"*

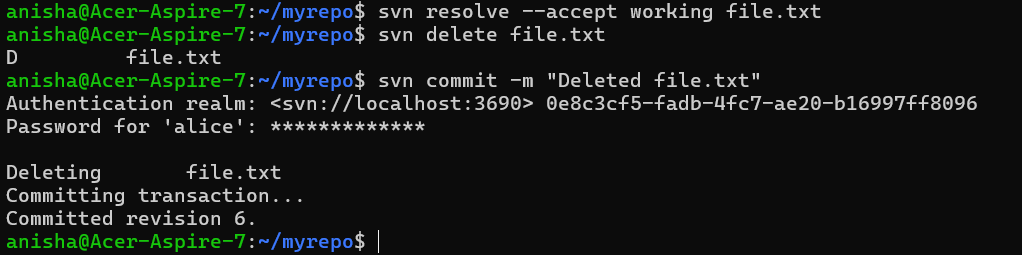
1. Resolve conflicts

* *svn resolve --accept working file.txt*



1. Delete a File

* *svn delete file.txt*
* *svn commit -m "Deleted file.txt"*



**Mercurial:**

Mercurial is a distributed version control system (DVCS) designed for efficient handling of projects of all sizes. It is similar to Git in functionality but emphasizes simplicity and ease of use. Mercurial is written in Python and is known for its intuitive commands, robust performance, and cross-platform compatibility. It is widely used in both open-source and enterprise projects.

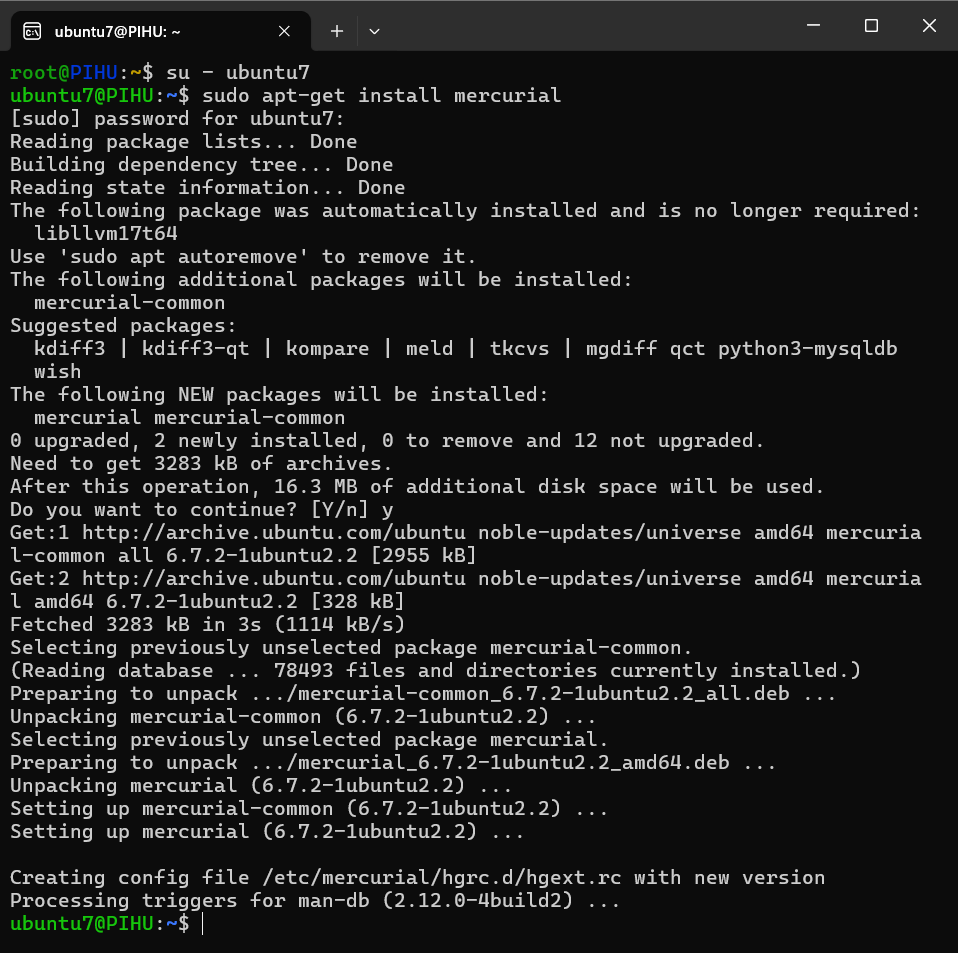
Key Features of Mercurial:

* **Distributed Version Control:** Every developer has a full copy of the repository, enabling ofline work and independent branching.
* **Lightweight and Fast:** Efficient handling of large projects and binary files.
* **Simple and Intuitive Commands:** Commands are easy to learn and use, with a consistent syntax.
* **Cross-Platform**: Works on Windows, macOS, and Linux.
* **Extensible:** Supports pluginsfor additional functionality.
* **Secure:** Uses SHA-1 hashing for data integrity and supports signed commits.

On Linux:

1. Use your package manager

* *sudo apt-get install mercurial*



1. Initialize a Repository:

* *hg init my\_project*
* *cd my\_project*

1. Add Files:

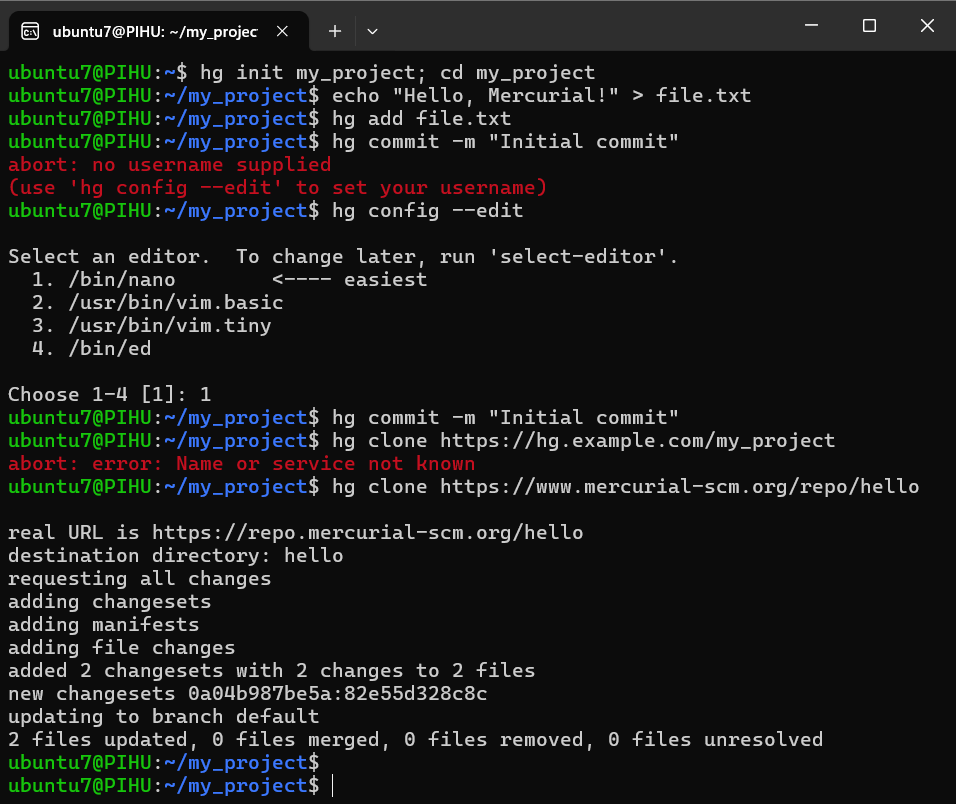
* *echo "Hello, Mercurial!" > file.txt*
* *hg add file.txt*

1. Commit Changes:

* *hg commit -m "Initial commit"*

1. Clone a Remote Repository:

* *hg clone* [*https://hg.example.com/my\_project*](https://hg.example.com/my_project)



1. Push Changes to Remote:

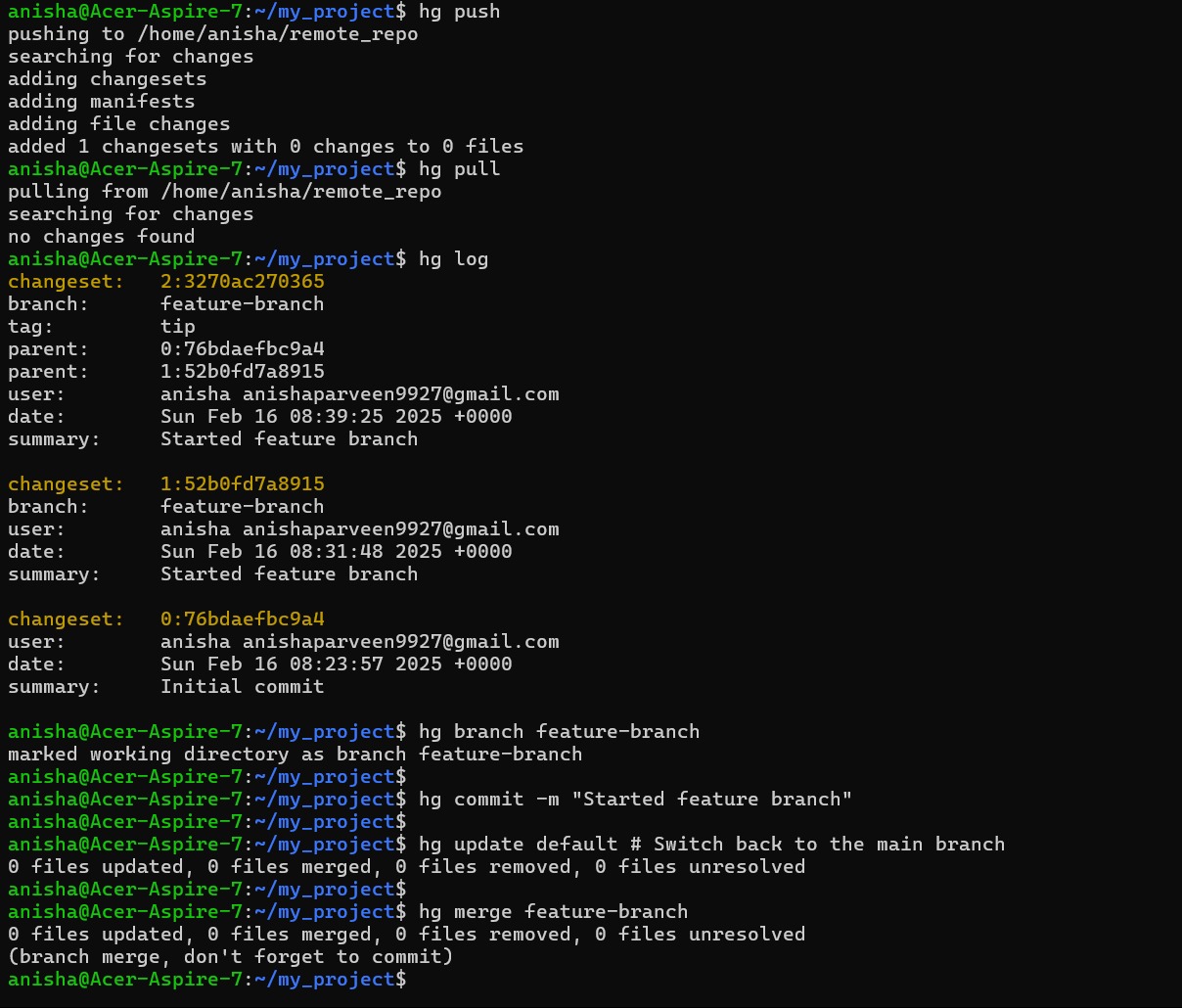
* *hg push*

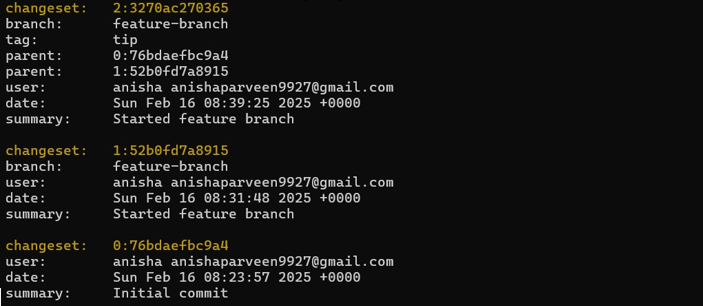
1. Pull Changes from Remote:

* *hg pull*

1. View Commit History:

* *hg log*





1. Create and Switch Branches:

* *hg branch feature-branch*
* *hg commit -m "Started feature branch"*
* *hg update default # Switch back to the main branch*

1. Merge Branches:

* *hg merge feature-branch*
* *hg commit -m "Merged feature-branch int*

