School Of Computer Sciences

Dev Ops

Submitted By:

Name: Ayush Sharma

SAP ID: 500106907

Roll No.: R2142220869

Batch – 3 CCVT

Submitted To: Dr. Prateek Raj Gautam

VERSION CONTROL ASSIGNMENT-1

SUBVERSION

SVN is another name for subversion software. The version control system is free source. We may examine the prior iteration of the file and monitor changes over time by using subversion.

Version control systems come in two varieties:

- **Centralised Version Control System (CSCV**): All of the versions are kept on a single central server.
- Every user has a copy of the entire repository thanks to the Distributed Version Control System (DVSC).

Why is Version Control used?

- We can revert to the earlier version if necessary. To keep track of all the changes and preserve the history.
- We are able to combine new functionalities.

How to Install and Configure SVN on Windows:

1. Install TortoiseSVN (SVN) after downloading it.
2. Restart your computer after the installation.
3. Check the installation.

"svn –version"

```
C:\Users\ayush>svn --version
svn, version 1.14.5 (r1922182)
   compiled Nov 30 2024, 08:20:48 on x86-microsoft-windows

Copyright (C) 2024 The Apache Software Foundation.
This software consists of contributions made by many people;
see the NOTICE file for more information.
Subversion is open source software, see http://subversion.apache.org/

The following repository access (RA) modules are available:

* ra_svn : Module for accessing a repository using the svn network protocol.
  - with Cyrus SASL authentication
  - handles 'svn' scheme
* ra_local : Module for accessing a repository on local disk.
  - handles 'file' scheme
* ra_serf : Module for accessing a repository via WebDAV protocol using serf.
  - using serf 1.3.10 (compiled with 1.3.10)
  - handles 'http' scheme
  - handles 'https' scheme

The following authentication credential caches are available:

* Wincrypt cache in C:\Users\ayush\AppData\Roaming\Subversion
```

SVN Commands:

1. For initializing the repository.
   "svnadmin create ~/svn_repo/my_project"

```
C:\Users\ayush>svnadmin create E:\Ayush_repo
```

2. Checkout the repository
   "svn checkout file:///path/to/svn_repo/my_project"

```
C:\Users\ayush>svn checkout file:///E:/Ayush_repo Ayush-project
Checked out revision 0.
```

3. Add files to the directory
   - Navigate to your working directory.
   - Add new files.
   - Commit the changes.

```
C:\Users\ayush>cd Ayush-project

C:\Users\ayush\Ayush-project>echo "Hello svn" > Ayush_file.txt

C:\Users\ayush\Ayush-project>svn add Ayush_file.txt
A         Ayush_file.txt

C:\Users\ayush\Ayush-project>

C:\Users\ayush\Ayush-project>svn commit -m "Initial commit by Ayush
Adding         Ayush_file.txt
Transmitting file data .done
Committing transaction...
Committed revision 1.
```

4. Update your working copy and view the logs
   - svn update
   - svn log

```
C:\Users\ayush\Ayush-project>svn update
Updating '.':
At revision 1.

C:\Users\ayush\Ayush-project>svn log
------------------------------------------------------------------------
r1 | ayush | 2025-02-16 17:40:49 +0530 (Sun, 16 Feb 2025) | 1 line

Initial commit by Ayush
------------------------------------------------------------------------
```

5. Reverting the changes
    - svn revert file.txt

```
C:\Users\ayush\Ayush-project>svn revert Ayush_file.txt
```

6. Creating the branch and merge the changes.
    - **svn copy file:///C:/svn_repos/my_repo/trunk**
    - **file:///C:/svn_repos/my_repo/branches/faeature-branch -m "Creating feature branch"**
    - **svn merge file:///C:/svn_repos/my_repo/branches/faeature-branch**

```
C:\Users\ayush\Ayush-project>svn mkdir file:///C:/Ayush_repo/branches -m "Creating Branches Directory"
svn: E170013: Unable to connect to a repository at URL 'file:///C:/Ayush_repo'
svn: E180001: Unable to open repository 'file:///C:/Ayush_repo'

C:\Users\ayush\Ayush-project>svn mkdir file:///E:Ayush_repo/branches -m "Creating Branches "
Committing transaction...
Committed revision 2.
C:\Users\ayush\Ayush-project>svn copy file:///E:/Ayush_repo/trunk file:///E:/Ayush_repo/branches/new-features -m "Creating Branches by Ayush"
Committing transaction...
Committed revision 5.
```

HG MERCURIAL

A distributed version control system (DVCS) called Mercurial was created to manage projects of all sizes effectively. Mercurial's functionality is comparable to Git, but it prioritises usability and simplicity. Written in Python, Mercurial is renowned for its cross-platform interoperability, strong performance, and simple instructions.

Mercurial's characteristics include:

- Distributed version control allows for independent branching and offline work because each developer has a complete copy of the repository.
- **Fast and lightweight**: Effective management of big projects and binary files.
- **Cross-Platform**: It is compatible with Linux, macOS, and Windows.
- Plugins are supported for extra functionality, making it extensible.
- Simple and Intuitive Commands: Commands are straightforward to learn and use, with a consistent syntax.

Installing & setting up Mercurial on Windows:

1. Download and install Mercurial (TortoiseHg)
2. After the installation restart your system.
3. Verify the installation.

```
PS C:\Users\ayush> hg --version
Mercurial Distributed SCM (version 6.5.1)
(see https://mercurial-scm.org for more information)

Copyright (C) 2005-2023 Olivia Mackall and others
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
PS C:\Users\ayush>
```

Mercurial Commands

1. Creating and Initializing the Repository
   “hg init my-hg-repo”

```
PS C:\Users\ayush> hg init Ayush-hg-repo
PS C:\Users\ayush> cd Ayush-hg-repo
PS C:\Users\ayush\Ayush-hg-repo>
```

2. Adding files and committing the files
   hg add file.txt
   hg commit -m “Added newfile.txt”

```
PS C:\Users\ayush\Ayush-hg-repo> hg pull
pulling from default
abort: repository default not found
PS C:\Users\ayush\Ayush-hg-repo> hg pull "C:\Users\ayush\my-hg-repo"
pulling from C:\Users\ayush\my-hg-repo
no changes found
PS C:\Users\ayush\Ayush-hg-repo>
```

3. Cloning, Updating and Reverting
   "hg clone https://example.com/repo
   hg pull
   hg update
   hg log"

```
PS C:\Users\ayush> cd Ayush-hg-repo
PS C:\Users\ayush\Ayush-hg-repo> hg pull "C:\Users\ayush\my-hg-repo"
pulling from C:\Users\ayush\my-hg-repo
no changes found
PS C:\Users\ayush\Ayush-hg-repo> hg update
0 files updated, 0 files merged, 0 files removed, 0 files unresolved
PS C:\Users\ayush\Ayush-hg-repo> hg log
changeset:   0:d71d0e70619b
tag:         tip
user:        Ayush Sharma <ayushvashisth2004@gmail.com>
date:        Sun Feb 16 21:35:36 2025 +0530
summary:     Initial commit
```

```
PS C:\Users\ayush\Ayush-hg-repo> hg revert Ayush_file.txt
no changes needed to Ayush_file.txt
PS C:\Users\ayush\Ayush-hg-repo>
```

4. **Branching and Merging**
   "hg branch new-feature
   hg merge"

```
PS C:\Users\ayush\Ayush-hg-repo> hg branch Ayush-feature
marked working directory as branch Ayush-feature
(branches are permanent and global, did you want a bookmark?)
PS C:\Users\ayush\Ayush-hg-repo>
```

```
PS C:\Users\ayush\Ayush-hg-repo> hg commit -m "Creating a new feature branch by Ayush"
PS C:\Users\ayush\Ayush-hg-repo> echo "This is an update from Ayush-feature branch" >> Ayush_file.txt
PS C:\Users\ayush\Ayush-hg-repo> hg commit -m "Updated Ayush_file.txt in Ayush-feature branch"
PS C:\Users\ayush\Ayush-hg-repo> hg update default
1 files updated, 0 files merged, 0 files removed, 0 files unresolved
PS C:\Users\ayush\Ayush-hg-repo> hg merge Ayush-feature
1 files updated, 0 files merged, 0 files removed, 0 files unresolved
(branch merge, don't forget to commit)
PS C:\Users\ayush\Ayush-hg-repo> hg commit -m "Merged Ayush-feature into default"
PS C:\Users\ayush\Ayush-hg-repo>
```