

P465 - GIT, SSH, PULL-REQUESTS

WORKING COLLABORATIVELY

30 Minutes

SSH

- * Remote shell
- * Remote command execution
- * Encrypted network tunneling
- * Highly configurable
- * client runs ssh (or scp) and remote system runs sshd
- * ssh uses asymmetric encryption keys, labeling one the private key and the other public key. The default encryption algorithm is RSA.

Encryption

- * Uses an encryption key value to transform data such that the original form of the data can only be derived from a corresponding decryption key.
- * The amount of effort required to guess the correct decryption key (or an equivalent value) is a measure of the strength of the encryption.
- * Symmetrical encryption algorithms, like AES, use the same key value for encryption and decryption.
- * Asymmetrical encryption algorithms, like RSA, use a pair mathematically entangled values such that if data is encrypted with one of the pair, only the other value can decrypt the data.

ssh configuration

- * ssh configuration information maintained in \$HOME/.ssh
- * config - default options for connecting to remote systems
- * known_hosts - list of remote systems (hosts) for which we have host keys
- * authorized_keys - list of public keys allowed to access the account
- * Private keys and corresponding public keys

ssh-keygen

- * ssh-keygen is used to create pair of keys using random seed values
- * You are encouraged to encrypt your private key with a symmetric encryption key. This prevents someone using a copy of your private key.
- * Public keys are in clear text but often encoded into an ASCII representation and can have a short comment attached.

ssh config file

- * The config file allows you to set your ssh default options per host
- * Host - short name or alias for a remote system; starts a stanza specifying the options for that system
- * HostName - this option specifies the FQDN of the host
- * User - username to connect to the Host
- * IdentityFile - Path the private ssh key used to authenticate to the Host
- * ForwardAgent - Yes/No option to allow the ssh-agent to forward keys

ssh known_hosts

- * File contains a list of hostnames and the corresponding host public key.
- * Each remote system has its own pair of ssh keys
- * ssh connections authenticates the remote system by proving it has the corresponding private host key.
- * If a host does not have an public key in the known_hosts file, ssh will prompt if you trust the public key being offered by the remote system and if so adds it to the known_hosts file.

ssh authorized_keys file

- * In the \$HOME/.ssh of the remote system you will find the authorized_keys file
- * This file contains all the user public keys that are allowed to access this account on the remote system.
- * The ssh client uses the user's unencrypted private key that corresponds to the public key in the remote authorized_keys file.

ssh-agent

- * Encrypted private keys must be decrypted before being used.
- * ssh-agent is a personal daemon that will load the decrypted private keys (ssh-add command).
- * Each time ssh connects, it will query the agent for any keys and attempt to use those to connect.
- * Agents can forward the keys to a remote system allowing the remote system access to the private key as if it was local
- * Windows users: <https://interworks.com/blog/2021/09/15/setting-up-ssh-agent-in-windows-for-passwordless-git-authentication/>

ssh setup

- * On your client, create a ssh key pair (ssh-keygen) if you do not already have one
- * On your client, load the private key (ssh-add) into the ssh agent.
- * Update the client ./ssh/config file with any default ssh connection values
- * Connect to the remote system using password authentication. Copy the contents of the public key into the remote system authorized_keys file.
- * Logout of the remote and log back in using ssh key authentication.

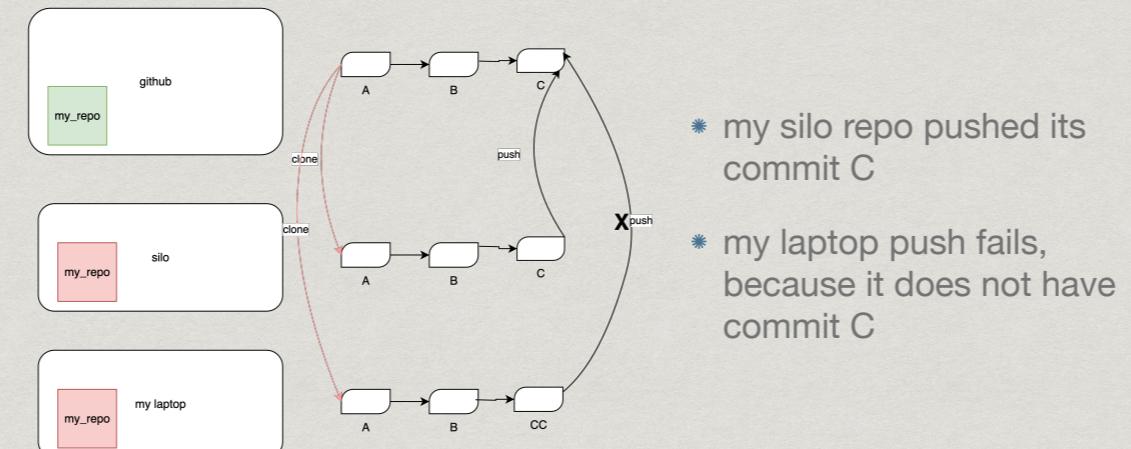
Git overview

- * Source code version control system
- * git clones are entire copies of all the changes in the source code
- * A bare git repository has just the git data files and does not have "checked out" versions of the project files.
- * A working copy of a git project, has all the project files available and keeps the git data files in the project directories .git directory

git commits

- * git records changes to the repository as commits
- * A commit can consist of changes of many files
- * Commits are required to have a comment describing the changes and are attributed to the person committing the changes.
- * The files to be part of a commit must be identified (staged) using the **git add command** prior to using the **git commit command**.

Multiple working repositories



Branching

- * git supports creating branches
- * Generally a repository will have a main branch and development commits are done to a branch off the main branch
- * When a development branch is complete, it is merged into the main branch. This creates a merge commit.
- * Rebase takes the commits from your feature branch and replays them on top of the target branch (like main), creating new commits with the same changes but different commit hashes. This results in a linear history that appears as if your changes were made directly on top of the latest version of the target branch, without any merge commits.

Github overview

- * GitHub is a website that hosts git repositories
- * GitHub provides additional capabilities when working and collaborating on git repositories
- * GitHub adds support for issue tracking, a Wiki, forum, workflows, and more

Pull-Requests

- * Pull-requests are a feature of GitHub that allows a commit (usually a branch merge) to be reviewed and approved by repository collaborators before the commit is applied.
- * Pull-requests support discussion and additional commits to be made before the eventual merge of the commits.

Team Repo (Team Best In-Class)

- * Create a team repository in GitHub
- * Initially only need README.md and .gitignore files.
- * Each member needs to setup a working copy of the repo in silo
- * Each member will add their individual name and email address to the README.md file in their working copy, commit it, and then push the commit to the team repo in GitHub.
- * Make sure each member of your team is a collaborator as well as the instructor and your assigned TA.
- * Post the URL for the tam in the Canvas assignment.