VE482 Lab Report

Lab 2 - Fall 2021

Name: Mingxuan Lu
ID: 518021911166
Email: mingxuan.lu@sjtu.edu.cn

Minix 3

- We use pkgin to manage software in Minix3:
  - install: `pkgin install <package>`
  - remove: `pkgin remove <package>`
  - update: `pkgin update`
- Purpose of the commands are listed as follows:
  - `ifconfig` : configure the kernel-resident network interfaces and display basic net information, such as LAN ip, netmask
  - `adduser` : add a new user to the system
  - `passwd` : change the user's password or initiate a password

Working on a remote server

- Setup SSH

```
pkgin install openssh
vim /usr/pkg/etc/ssh/sshd_config
# change Port 22 to Port 2222
passwdc
```

- `22` is the default SSH port.
- We can run `cd ~/.ssh` to check files related to SSH
  - `id_rsa` : the private key
  - `id_rsa.pub` : the public key
  - `authorized_keys` : a list of public keys. Whenever a user tries to connect with SSH, its public key will be compared with the public keys in the file. The user can connect the host without entering the password if matched.
  - `known_hosts` : Personal Certificate Authority. It is the list of all SSH server host public keys that you have determined are accurate.
- First, the user generates a pair of public and private keys. Then, the public key is sent to the host and stored as authorized keys. Whenever a user tries to connect with the host through SSH, his private key will be compared with the list of authorized keys. The user will connect to the host if matched.

  By default `PasswordAuthentication` is set to **yes**.

  You'll need to **explicitly** set `PasswordAuthentication no` to allow only Public Key Authentication.
- `ssh root@192.168.65.129 -p 2222` , `192.168.65.129` is the IP address of Minix 3.

Git

- `help` : help manual for Git commands
- `init` : initiate an empty local Git repository
- `branch` : list all existing branches of the current repository
- `checkout` : switch to another branch
- `add` : add all changes in the current directory for later commitment
- `commit` : commit the staged changes with a piece of commit message

- `diff` : display the difference between the current directory and the stage of the last commitment
- `fetch` : fetch a specific branch
- `pull` : fetch from and integrate with another repository or a local branch
- `push` : Update remote refs along with associated objects
- `clone` : Clone a repository into a new directory
- `merge` : Join two or more development histories together
- `reset` : Reset current HEAD to the specified state
- `tag` : Create, list, delete or verify a tag object signed with GPG
- `log` : Show commit logs