**Implementing a Git Server on Raspberry Pi**

*Towson University*

May 12, 2020

Alexandra Chambers - achamb19@students.towson.edu

Alexander Dao - adao1@students.towson.edu

Turhan Kimbrough - tkimbr1@students.towson.edu

Vince Lepatan – [vlepat1@students.towson.ed](mailto:vlepat1@students.towson.edu)u

|  |  |
| --- | --- |
| Group Member | Work description |
| Alex Chambers | * Tested environment * Proofread all documents |
| Alex Dao | * Purchased equipment * Managed presentation materials |
| Turhan Kimbrough | * Led the development process * Created demonstration videos * Configured Git |
| Vince Lepatan | * Found supporting online documents for Raspberry Pi configuration * Create SSH keys for git |

**Objectives & Motivation:** This project will demonstrate the flexibility of a modern Operating System by extending its functionality to provide services to client computers. We will extend the base image of the Raspbian Operating System to support Git, mimicking a production server. The implementation of this Operating System image will facilitate the group’s understanding of various networking services, along with the development of basic system administration skills. This project offers a practical way to learn about the components of the client-server model, along with its relevance to modern Operating Systems.

**Project Overview:** The Raspberry Pi will allow users to create and access their private git repository. Users can access the Raspberry Pi remotely and simultaneously via network connection. This provides a free private git repository but must be turned on to be accessed.

**Project description:** The Raspberry Pi 4 is an inexpensive single-board computer using the ARM Cortex-A72 architecture. Primarily deployed using a variant of the GNU/Linux Operating system, the Raspberry Pi 4 has developed a platform attracting professionals and hobbyists alike. In this project, the Raspberry Pi 4 will be configured to support protocols and software tools used in the industry such as Secure Shell (SSH), ngrok, and Git.

**Original Plan & Timeline:**

* Pre-proposal writing stage - Discuss with group on possible projects and outcomes
* Research books and sites for types of projects for Raspberry Pi
  + Purchase:
    - Raspberry Pi 4+
    - BreadBoard
    - Resistors/LED
    - Raspberry Pi for dummies book
* **(02/25/2020)**
  + Project Proposal Due
* **(03/13/2020)** (The due date for the last Lab)
  + Continue additional research on coding and inner workings of Raspberry Pi
  + Look into other possible functions for the LED
  + Continue to learn more about C language
* **(03/24/2020)**
  + Get together and work on code to get the LED running
* **(03/31/2020)**
  + Run tests on code with the LED
  + Make changes as needed
  + Start writing Progress Report
* **(04/6/2020)**
  + Progress Report Due
  + Start Final Package Report
* **(05/2/2020)**
  + Debug, make last edits, before presentation
* **(05/4/2020) - (05/15/2020)**
  + Presentation Date (Before, pick a presentation date before)
* **(05/10/2020)**
  + Make changes that is suggested by the professor
* **(05/12/2020)**
  + Project Final Package Due

**Implementation details:** SSH is an application for computer ports supporting cryptographically secure connections over the transmission control protocol (TCP). The next application, ngrok, is a tunneling service which allows for local machines to be exposed and accessed via the internet. Lastly, Git is a content-tracking application typically used for tracing changes in source code and files relevant to software development. SSH and ngrok were used for remote collaboration between group members while implementing/testing Git functionality. A brief list of steps are provided for configuring the Raspberry Pi:

(On local machine)

* Using a Linux, Mac, or Windows personal computer, flash the latest version of an ARM-based Operating System on a microSD card
* Once the OS contents are on the microSD card, place an empty file with no file extensions called ‘ssh’ (this will enable ssh on boot for the Raspberry Pi)
* Place another file into the boot directory called wpa\_supplicant.conf (this is where you will add credentials for connecting to WiFi) and place the contents below inside the file:

ctrl\_interface=DIR=/var/run/wpa\_supplicant GROUP=netdev

update\_config=1

country=US

network={

ssid="<Wireless LAN info here>"

psk="<Wireless LAN password here>"

}

(On Raspberry Pi)

* Place the microSD card into the Raspberry Pi and connect it to a power source (it should automatically boot and listen for remote connections)

(Back to local host)

* Make a free ngrok account at <https://ngrok.com/> (more on this later)
* Create a remote terminal session via ssh by typing ssh [pi@XX.XX.XX](mailto:pi@XX.XX.XX).XXXX where X’s represent the IP address (default credentials are Username: Pi, Password: Raspberry)

(On Raspberry Pi remotely)

* Install git with sudo apt-get install git
* Install ngrok with:

sudo wget <https://dl.ngrok.com/ngrok_2.0.19_linux_arm.zip> (download zip)

sudo unzip ngrok\_2.0.19\_linux\_arm.zip (unzip folder)

Login to website and get authentication token

./ngrok authtoken <authentication token>

After the steps above, the Raspberry Pi 4 has all the necessary components for the project. A demonstration video provides details on how these technologies cooperate.

**Challenges & problems:** The one notable challenge during the implementation of this project was when the group attempted to configure SSH for a public IP address. Unfortunately, because of the strict security protocols set by the service provider Xfinity, connections to SSH via public IP were always registered as attacks. This produced false-positives 100% of the time, and would occur even when the firewall was turned off. Another challenge that has come up was with the use of Ngrok. Ngrok only allows for a limited number of users to remotely login to the Raspberry Pi under the basic subscription plan.

**Experimental Setting:** The Raspberry Pi 4 device is running on the Raspbian Buster Operating System (based on Debian Buster). Raspbian Buster is the official Operating System supported by the Raspberry Pi Foundation. Aside from the installation of Raspbian itself, all configurations were done through a remote command-line interface on all group members’ personal computers.

**Result Analysis**: Starting the project, we were expecting to use the Raspberry Pi along with an LED, due to certain circumstances, we changed out project mid-semester and opted for creating a private Git server on the Raspberry Pi instead. We were worried about the sudden change in project and in the beginning, had issues with connectivity and wondered if we could do this in the allowed scheduled time. However, even after all of the struggles, we were successfully able to run through all of the deliverables in a timely manner. We were able to push and pull documents from the private Git server to different users remotely while connecting to the Raspberry Pi via SSH. As long as the Raspberry Pi was connected to a local Wifi, as well as plugged-in, anyone of our members could connect to the server easily.

**Future Work:** In the future, we would like to implement the LED driver and while also using the git functionality built into the Pi. Since some research has been conducted gy the group for the LED driver, we feel like this is the best option for our next step.

**Updated Timeline:**

* **(04/6/2020)**

-Looking into resources to host a private Git server on Raspberry Pi (Vince)

* **(04/11/2020)**

-Connect to the the private Git Server on Raspberry Pi (Turhan)

* **(04/15/2020)**
  + Progress report Due
  + Integrate SSH with Git so everyone can test the environment (Vince)
* **(04/17/2020)**
  + Test Git environment to push and pull (Alexander)
  + Record a demo video to explain the functionality of server (Alex & Turhan)
* **(05/2/2020)**
  + Debug, make last edits, before presentation (Everyone over Discord)
* **(05/4/2020) - (05/15/2020)**
  + Presentation Date (Before, pick a presentation date before)
* **(05/10/2020 or anytime after presentation date)**
  + Make changes that is suggested by the professor (Everyone)
* **(05/12/2020)**
  + Project Final Package Due (Vince)

**Repository URL:** <https://gitlab.com/adao1/something-with-raspberry-pi>

**Deliverables:**

* Raspberry pi with remote access & Git Server
* Demo video to demonstrate the functionality of the git server
* User guide with configuration procedures

**References:**

* [**https://www.instructables.com/id/GitPi-A-Private-Git-Server-on-Raspberry-Pi/**](https://www.instructables.com/id/GitPi-A-Private-Git-Server-on-Raspberry-Pi/)
* [**https://medium.com/@byteshiva/ssh-into-remote-linux-by-using-ngrok-b8c49b8dc3ca**](https://medium.com/@byteshiva/ssh-into-remote-linux-by-using-ngrok-b8c49b8dc3ca)
* [**https://medium.com/@maheshsenni/setting-up-a-raspberry-pi-without-keyboard-and-mouse-headless-9359e0926807**](https://medium.com/@maheshsenni/setting-up-a-raspberry-pi-without-keyboard-and-mouse-headless-9359e0926807)