

Welcome to the intern team at SwollenHippo Technologies. We are very excited that you chose to join our team this term and look forward to the awesome work you will produce.

Looking through your resume and transcripts that you provided, it appears you are familiar with DevOps and are an excellent candidate for our flagship project. This flagship project will be used to help us setup new servers in our development environment.

Background

Currently our development team requires the provisioning of new virtual machines on our Google Cloud Platform (GCP) account each time they engage on a new project. Unfortunately, our systems team, who is responsible for this task, has recently seen a loss of a number of employees due to retirement. This reduced team size unfortunately has resulted in an increase in the processing time for these requests.

When our systems team has the opportunity to process the request, they begin by logging into our internal ServiceNow IT management system. After authentication, the systems team member searches for and reviews the request from the developers. The request or ticket is then marked as in-progress by the systems team member. The team member will then log into our GCP account and create a new virtual machine (VM) using computer engine. These virtual machines are e2-medium servers running the latest version of Debian Linux with SSH and HTTP/HTTPS access enabled. Once the VM is created, the systems team member will review the ticket and verify what software packages must be installed.

The systems team member will then return to the GCP console and using the web-based SSH option, login to the server. The systems team member will then install each of the requested software packages and verify that they have been installed. A logfile is then manually created by the systems team member indicating which packages were installed and its version.

After the installation process has been completed, the systems team member returns back to the ServiceNow platform and collects information on what additional configurations must be completed for the request. Those additional configurations are then completed and notes regarding them are added to the previously created logfile. This logfile is then stored on the server for future verification and review.

Finally, the systems team member will return back to the ServiceNow platform and mark the ticket as Complete. All told, even small and sparse server setup processes ties up a systems team member for several hours.

After several discussions between the systems and development teams we have been able to identify a number of standard setups that could be used to streamline server setup. However, even with the standard setups, the process still takes several hours to complete.

Mission

We are tasking you with automating this process. We understand that this is a challenging assignment, but we have assigned you a mentor to act as a resource to assist you in direction if you need help. Your mentor, Ben Burchfield, is normally available during scheduled office hours as well as via Teams most times. Additionally, you are allowed access to any resource (**except other interns/students or people**), but we do ask that you try not to use generative AI without first speaking with your mentor as it could expose some trade secrets to the owner of the generative AI. You have until the time indicated in Teams to complete this mission.

Additional Details

You should use a private Git repository to contain your work and track changes as you go. The basis for the entirety of the process is a robust shell script that will allow the user to provide two parameters: 1) the external IP address of a virtual machine that was manually created in GCP, and 2) the TicketID of the request.

The log file being created will be stored at /configurationLogs with the name being the TicketID.log provided from the web service return (<https://www.swollenhippo.com/ServiceNow/systems/devTickets.php>).

The web service above will return details about the ticket including the requestor, the ticketID, the submissionDate, the softwarePackages, and additionalConfigs. An example return would look similar to this:

A web server request

```
{ticketID: 17065, requestor: Blake Smith, submissionDate: 17-April-2024, standardConfig: Web Server, softwarePackages: [{name:Apache 2, install:apache2},{name:JSON Parser, install:jq},{name:MariaDB, install:mariadb-server}],additionalConfigs: [{name>Create Example File, config: touch /var/etc/www/index.html}]}
```

A file server request

```
{ticketID: 17042, requestor: Joanne Longfellow, submissionDate: 13-April-2024, standardConfig: File Server, softwarePackages: [],additionalConfigs:[{name>Create Open Directory, config: mkdir /shared},{name:Update Permissions,config: chmod 777 /shared}]}
```

A database server request

```
{ticketID: 17066, requestor: Matt Silva, submissionDate: 17-April-2024, standardConfig: Database Server, softwarePackages: [{name:MariaDB, install:mariadb-server}],additionalConfigs: []}
```

The log file should include the following information at the top of the file: TicketID, DateTime of Start, Requestor, External IP Address of Server, Hostname, Standard Configuration

As each step is started you should include a line in the log that indicates Section (softwarePackage or additionalConfig), the task name, a timestamp of when the action started

Once the steps have been completed, the version of any software that was added by the script should be included in the log with this information: Version Check - software name, the resulting version information.

After completing all steps, another web service should be called to indicate in ServiceNow that the ticket is complete. The results of this web service call should be included in the log file. The URL will be <https://www.swollenhippo.com/ServiceNow/systems/devTickets/completed.php?TicketID=xxxxx>. NOTE: the xxxxx in the URL should be replaced with the TicketID that the script is processing.

Example web service return

```
{outcome: TicketClosed}
```

Finally, a line should be written to the log file that indicates the finish date time.

Your shell script should exist on your local machine by connecting to the servers via SSH. You will then need to use another shell script to copy the script and run it on the GCP server. To make this work, you will need to add the gcp ssh key to each of the servers you are manually creating. Then to copy the newly created shell script, you will need to use scp. An example command to copy the shell file would be:

```
scp -i .ssh/gcp serverSetup.sh bburchfield@34.134.45.94:/home/  
bburchfield
```

In this command, I am using scp -i to connect to the server with my ssh key, then identifying the serverSetup.sh file to be copied to the remote server. The bburchfield@34.134.45.94 is my username and external IP address on the GCP server. The :/home/bburchfield identifies my home directory on the remote server. This script should then run the script on the remote server. **This is fairly advanced, so an example of this script is included in the instructions.**

Example of log file for ticketID 17065 (17065.log) above:

TicketID: 17065
Start DateTime: 17-Apr-2024 21:44
Requestor: Blake Smith
External IP Address: 149.149.23.105
Hostname: instance0765H1
Standard Configuration: Web Server

softwarePackage - Apache2 - 1713408536
softwarePackage - JSON Parser - 1713408552
softwarePackage - MariaDB - 1713408606
additionalConfig - Create Example File - 1713408608

Version Check - Apache2 - 12.07.01
Version Check - JSON Parser - 1.3
Version Check - MariaDB - 14.1.4

TicketClosed

Completed: 17-Apr-2024 21:52

Example of log file for ticketID 17042 (17042.log) above:

TicketID: 17042
Start DateTime: 17-Apr-2024 21:44
Requestor: Joanne Longfellow
External IP Address: 149.149.23.105
Hostname: instance0765H1
Standard Configuration: File Server

additionalConfig - Create Open Directory - 1713408608
additionalConfig - Update Permissions - 1713408610

TicketClosed

Completed: 17-Apr-2024 21:52

Example of log file for ticketID 17066 (17066.log) above:

TicketID: 17066
Start DateTime: 17-Apr-2024 21:44
Requestor: Matt Silva
External IP Address: 149.149.23.105
Hostname: instance0765H1
Standard Configuration: Database Server

softwarePackage - MariaDB - 1713408606

Version Check - MariaDB - 14.1.4

TicketClosed

Completed: 17-Apr-2024 21:52

Example Shell Script to Copy and run serverSetup.sh file on remote GCP server from your local machine

```
#!/bin/bash
# Description: Copies and runs automation script on GCP remote server
# Author: Ben Burchfield
# Date: 17 April 2024

strIP=$1
strTicketID=$2
strUsername=$3
eval "$(ssh-agent -s)"

# The location of my gcp ssh key is in the .ssh directory
# My gcp ssh key is called gcp
ssh-add .ssh/gcp

# Using scp, copies a shell script called serverSetup.sh from the
# current directory to the home directory on the remote server
# Note the two lines below is actually one line
scp -i .ssh/gcp serverSetup.sh "${strUsername}"@"${strIP}":/home/"$
{strUsername}"

ssh ${strUsername}@${strIP} "chmod 755 serverSetup.sh"
ssh ${strUsername}@${strIP} "./serverSetup.sh ${strIP} ${strTicketID}"
```

This script would need to be given execution rights and then could be ran like this:

./kickoff.sh 34.134.45.94 17065 bburchfield

.....where 34.134.45.94 is the external ip address of the remote server and bburchfield is the user on that server

Submission Details

For full credit on the assignment you must submit the following:

1. A copy of the history of your git repository in a text file called git.txt
2. A copy of your shell script used to process the requirements named severSetup.txt
3. A screenshot showing the servers you created in your GCP console
4. A copy of each of the log files that you create
5. Documentation/user manual for folks to use your shell script

Your grade will be based on the completeness and accuracy of your work.