Automated Server Configuration, Deployment, and Maintenance for Multiple Environments

12/6/2023

CSC 2510 Dev-Ops

BY: Behnjamin Barlow

……………………………………………………**TOPICS**……………………………………………………

Purpose:

- SwollenHippo Enterprises operates a two-tier architecture with a web server responsible for serving a simple static webpage and a database server managing essential data. Additionally, adopting a new web application that is hosted on a Git repository to have a seamless and secure process that not only sets up and maintains the server environment but also ensures that the latest version of the web application is deployed automatically across the Dev, Test, and Prod environments.

Setting up Servers:

- For this you will need to create 3 servers with specific settings listed below. You could use a platform like “Google Cloud Platform” to create one just create an account and get started you can navigate to console button in the top right then the engine tab to get to an overveiw of your servers. At the top you can hit “ create instance “ to make a server.

    - mainserver -
        - **Region:** Any of your choosing (ex: us-central1(Iowa))
        - **Computing Power:** 0.5-2 vCPU (1 shared core) w/ 2gb memory
        - **Operating system:** Debian GNU/Linux 11(10gb)
        - Leave rest alone
    - devserver -
        - **Region:** Any of your choosing (ex: us-central1(Iowa))
        - **Computing Power:** 0.5-2 vCPU (1 shared core) w/ 2gb memory
        - **Operating system:** Debian GNU/Linux 11(10gb)
        - **Allow:** HTTP and HTTPS traffic
        - Leave rest alone
    - webserver -
        - **Region:** Any of your choosing (ex: us-central1(Iowa))
        - **Computing Power:** 0.5-2 vCPU (1 shared core) w/ 2gb memory
        - **Operating system:** Debian GNU/Linux 11(10gb)
        - **Allow:** HTTP and HTTPS traffic
        - Leave rest alone

STARTING SERVER:

Once servers are made you can hit the triple dots icon on the right and select “start/stop instance” then once it starts click on the SSH icon under the “connect” column to start it up in the browser window, just authorize the ssh connection and Congrats you’re in your server!!

Setting up mainserver:

MainServer: This is your command center for the other two servers and will use ansible to control the proccesses nessisary.

- **Installing packages:**
  - Install ansible using the command
    - **sudo yum install ansible**
  - Install git using the command
    - **sudo yum install git**
  - Install nano using the command
    - **sudo yum install nano**

- **Setting up Ansible:**
  - After installing ansible which you can think of like a folder that hold playbooks which preform specific actions used for automation
  - You will need to go to the host file using the command:
    - **sudo nano /etc/ansible/hosts**
  - This will give you access ansible hosts file and add servers(IP adress) to be able to interact with them. Frist scroll all the way to the bottom of the file and paste in the correct contents of **ansiblehosts.txt** make sure to have no extra white spaces(not empty lines those are fine) as it can cause errors. Then fill it in with the proper info and you can now leave that file with (ctrl + x)->(y)->(enter)
  - Then use this command to return to home directory for the next step
    - **Cd**

- **Setting up Root Login:**
  - This will allow you to have root access on all servers from the main server
  - You Will need to navigate to the sshd_config file using:
    - **sudo nano /etc/ssh/sshd_config**
  - You will need to change some settings in this file scroll till you see **“PermitRootLogin no”** and make sure you change it to **“yes”** you may se many **permitRootLogins** just make sure they all say **“yes”**

- Now scroll till you see **“PasswordAuthentication no”** and change it to **“yes”** now leave that file with (ctrl + x)->(y)->(enter)
- Then use this command to return to home directory for the next step
  - **Cd**

◊ **Setting up git:**

- Create github account if you haven’t already!!
- This will allow you to have a secure connection to git and be able to clone your repo with the following link: **“git@github.com:ttu-bburchfield/ swollenhippofinal.git”**
- Use the command
  - **ssh-keygen -t ed25519 -C “example@email.com”**
- using the email you used to create your git hub account NEXT hit ENTER you want the file to be stored in the defualt position then you are propted with passphrase you can hit ENTER to not have one or choose to make one.
- After that open up your github account in the browser and navigate through to profile icon in top right then (**settings**)-> then on the left (**SSH and GPG keys**). Here hit **“new ssh key”** you may give it any name you want but I suggest something to know which server it belongs to then you will see a comment block where you ssh key goes so lets grab that!
- Navitage back to the server and type command:
  - **sudo cat .ssh/id_ed25519.pub**
- the (**.pub**) it only nessisary if you didn’t make a password otherwise take it off. Now copy the results and paste that into the comment block on github where your ssh key goes and hit create.
- Now to ensure your key is working and to add it to your servers section of trusted hosts use the following two commands:
  - **eval “$(ssh-agent –s)”**
  - **ssh-add .ssh/id_ed25519   or   ssh-add .ssh/id_ed25519.pub**
- Congrats Your server is now set up & you can interact with github servers safely!!

◊ **Placing Files:**

- For **MAIN** servers you will need to clone git repo using the command:
  - **git clone –c git@github.com:BHBarlow/DevOpsFinal.git**
- You will wanna make sure your github ssh key is set up for this
- This clone will place all the files where they are needed to be in the server

◊ **Running Files:**

- **DO THIS AT THE VERY END!!!!**
- With main.sh use this command to make sure file has right permissions
  - **sudo chmod 755 main.sh**

    - Followed by this command to get into the cron system
        - **EDITOR=nano crontab –e**
    - Then once inside you will paste the following line at the top
        - **`* * * * * ~/main.sh`**
    - This will run this file every minute you can add the words “dev” or “test” after it to only schedule that servers playbook to run if you don’t include a parameter it will run both here is an example
        - **`* * * * * ~/main.sh dev`**
        - **`* * * * * ~/main.sh test`**

- ◊ **<u>Changing file run occurrence:</u>**
    - **USES CRON** the lines above are examples
    - **<u>Minute (0-59):</u>**
        - The first asterisk represents the minute field. It indicates the minute at which the cron job will run, ranging from 0 to 59. For example, if the minute field is set to *, the job will execute every minute.
    - **<u>Hour (0-23):</u>**
        - The second asterisk denotes the hour field, specifying the hour at which the cron job will run. It ranges from 0 to 23 in a 24-hour format. If set to *, the job will run every hour.
    - **<u>Day of the Month (1-31):</u>**
        - The third asterisk is associated with the day of the month field, defining the day on which the cron job will execute. This field ranges from 1 to 31. If set to *, the job will run every day of the month.
    - **<u>Month (1-12):</u>**
        - The fourth asterisk signifies the month field, indicating the month during which the cron job will run. It ranges from 1 to 12, with 1 corresponding to January and 12 to December. If set to *, the job will run every month.
    - **<u>Day of the Week (0-6 or 7):</u>**
        - The fifth asterisk is associated with the day of the week field, specifying the day on which the cron job will execute. It uses numbers 0 to 6, where 0 and 6 typically represent Sunday, and 1 to 5 correspond to Monday through Friday. Some systems also accept 7 as Sunday. If set to *, the job will run every day of the week

<u>Setting up devserver:</u>

<u>DevServer:</u> This is your server that hold data and also acts as a test server.

- **Installing packages:**
    - Install git using the command
        - **sudo apt install git**

- **Setting up Root Login:**
    - This will allow you to have root access on all servers from the main server
    - You Will need to navigate to the sshd_config file using:
        - **sudo nano /etc/ssh/sshd_config**
    - You will need to change some settings in this file scroll till you see **“PermitRootLogin no”** and make sure you change it to **“yes”** you may se many **permitRootLogins** just make sure they all say **“yes”**
    - Now scroll till you see **“PasswordAuthentication no**” and change it to **“yes”** now leave that file with (ctrl + x)->(y)->(enter)
    - Then use this command to return to home directory for the next step
        - **Cd**
    - Now use the command
        - **Sudo passwd**
    - To change the basic password for this server. At the same time I recommend going into the ansiblehosts.txt file and the part where it tells you to copy to your correct location it should say “PASSWORD” and you should fill all those in with the password of your choosing.

- **Setting up git:**
    - Create github account if you haven’t already!!
    - This will allow you to have a secure connection to git and be able to clone your repo with the following link: **“git@github.com:ttu-bburchfield/ swollenhippofinal.git”**
    - Use the command
        - **ssh-keygen -t ed25519 -C “example@email.com”**
    - using the email you used to create your git hub account NEXT hit ENTER you want the file to be stored in the defualt position then you are propted with password and you will need to make one and remember it!
    - After that open up your github account in the browser and navigate through to profile icon in top right then (**settings**)-> then on the left (**SSH and GPG keys**). Here hit **“new ssh key”** you may give it any name you want but I suggest something to know which server it belongs to then you will see a comment block where you ssh key goes so lets grab that!
    - Navitage back to the server and type command:
        - **sudo cat .ssh/id_ed25519**
    - Now to ensure your key is working and to add it to your servers section of trusted hosts use the following two commands:
        - **eval “$(ssh-agent –s)”**

        - **ssh-add .ssh/id_ed25519**
    - Congrats Your server is now set up & you can interact with github servers safely!!

Setting up Webserver:

WebServer: This is your web server this one is considerd “production” server and is whats out on the web

- **Installing packages:**
    - Install git using the command
        - **sudo apt install git**

- **Setting up Root Login:**
    - This will allow you to have root access on all servers from the main server
    - You Will need to navigate to the sshd_config file using:
        - **sudo nano /etc/ssh/sshd_config**
    - You will need to change some settings in this file scroll till you see **“PermitRootLogin no”** and make sure you change it to **“yes”** you may se many **permitRootLogins** just make sure they all say **“yes”**
    - Now scroll till you see **“PasswordAuthentication no”** and change it to **“yes”** now leave that file with (ctrl + x)->(y)->(enter)
    - Then use this command to return to home directory for the next step
        - **Cd**
    - Now use the command
        - **Sudo passwd**
    - To change the basic password for this server. At the same time I recommend going into the **ansiblehosts.txt** file and the part where it tells you to copy to your correct location it should say **“PASSWORD”** and you should fill all those in with the password of your choosing.

- **Setting up git:**
    - Create github account if you haven’t already!!
    - This will allow you to have a secure connection to git and be able to clone your repo with the following link: **“git@github.com:ttu-bburchfield/ swollenhippofinal.git”**
    - Use the command
        - **ssh-keygen -t ed25519 -C “example@email.com”**
    - using the email you used to create your git hub account NEXT hit ENTER you want the file to be stored in the defualt position then you are propted with password and you will need to make one and remember it!
    - After that open up your github account in the browser and navigate through to profile icon in top right then (**settings**)-> then on the left (**SSH and GPG keys**). Here hit **“new**

ssh key” you may give it any name you want but I suggest something to know which server it belongs to then you will see a comment block where you ssh key goes so lets grab that!

- Navitage back to the server and type command:
    - **sudo cat .ssh/id_ed25519**
- Now to ensure your key is working and to add it to your servers section of trusted hosts use the following two commands:
    - **eval “$(ssh-agent –s)”**
    - **ssh-add .ssh/id_ed25519**
- Congrats Your server is now set up & you can interact with github servers safely!!

◊ **Contact Info:**

- Phone: 931-472-9818
- Email: bhbarlow42@tntech.edu

EOF