Lab 9 aNSWER sHEET

Term

# Error Handling in Python

## Learning Outcomes

Make a change to your backup script repository. We will be making this change simulating that you are collaborating on this project with other developers. This means we need to clone the repository and make our changes on a working branch. We will not be merging our changes with the master branch but instead pushing our working branch to the repository and opening a pull request (with yourself) requesting your changes be merged with the master branch along with a message of what changes you made.

|  |
| --- |
| Notes |
| * Dad Joke API   + Host: http://icanhazdadjoke.com |

|  |
| --- |
| Tips and Tricks |
| * Tips and tricks on the skills and tools |

## Commands and Options

|  |  |  |
| --- | --- | --- |
| Basic Name | Command | Brief |
| try | try: | tells python to attempt the code defined in the try handler |
| except | except: | define the exceptions (errors) you want handled and how you would like to handle them |
| else | else: | runs only if exception does not occur |
| finally | finally: | runs whether an exception occurs or not |
| shutdown interface | interface <interface>  shutdown | shutdown and interface on a cisco device |
| remove ssh from vty terminals | line vty 0 5  transport input none | Disable ssh from the virtual terminal lines. The 0 5 just states that there are 5 virtual terminal lines you can have more or less it depends on how many you configure. |
| clone repository | git clone | clones a remote repository to your local system |
| stage changes | git add <what\_to\_add> | Stage the changes you have made preparing to commit. |
| commit changes | git commit -m “your message here” | Commit changes to the current branch |
| create a branch | git branch <branch\_name> | create a working branch |
| checkout to a branch | git checkout <branch\_name> | naviage to a branch |
| push changes | git push | push your changes to the repository origin. |

## Error Handling for Network Device Backup Script

|  |
| --- |
| Lab Screenshots  Your screenshots should be clearly visible as I need to and be able to see the commands you entered along with the output. If I cannot see the commands, output, or the screenshot is too small for me to make it out it will NOT be counted.  For each requirement I need to see your code and the output of your code. |
| Setup Your Dev Environment   * Clone netmiko backup script from you GitHub repository * Issue the command to see the status of your repository * Configure global username and emails settings if not already configured otherwise verify these settings * Create a working branch with a name of your choice * Navigate to your working branch |
| C:\Users\inn\Desktop\Intern Projects\april\3\2.png  C:\Users\inn\Desktop\Intern Projects\april\3\4.png  C:\Users\inn\Desktop\Intern Projects\april\3\3.png  C:\Users\inn\Desktop\Intern Projects\april\3\4.png |
| Working on Branches   * Note: Make the changes ONLY to your new working branch NOT the master branch. * Import the following error exceptions class from the netmiko.ssh\_exception module.   + AuthenticationException   + SSHException   + NetMikoTimeoutException * Add error handling into your code using try/except statements to accept the following errors.   + AuthenticationException   + NetMikoTimeoutException   + SSHException * For each error print a message to the terminal letting the user know what device experienced what error. |
| C:\Users\inn\Desktop\Intern Projects\april\3\5.png  C:\Users\inn\Desktop\Intern Projects\april\3\6.png  C:\Users\inn\Desktop\Intern Projects\april\3\7.png  C:\Users\inn\Desktop\Intern Projects\april\3\8.png |
| Test your code   * Verify your error handling changes by performing the following   + Timeout: issue the shut command on the interface that netmiko connects to and execute the script   + SSH: Under “line vty” configuration enter the command “transport input none” to disable remote management over SSH.   + Authentication: Enter the incorrect username and password. * Execute your script to successfully backup the 1000v virtual router |
| C:\Users\inn\Desktop\Intern Projects\april\3\9.png  C:\Users\inn\Desktop\Intern Projects\april\3\10.png  C:\Users\inn\Desktop\Intern Projects\april\3\11.png  C:\Users\inn\Desktop\Intern Projects\april\3\12.png |
| Create a pull request   * While still on your working branch check the status of your repository using the appropriate command * If necessary, update and/or create your requirements.txt file * Stage your changes to the staging area (DO NOT INCLUDE YOUR VIRTUAL ENVIRONMENT!!! Don’t forget about the .gitignore file from lab 2!) * Commit your changes with a message of what changes you made * While still on your working branch push your changes to the repository origin * Create a pull request with yourself requesting your changes be merged to the master code base. |
|  |

# Feedback

|  |
| --- |
| Feedback Questionnaire |
| What did you think about the assignment? |
| It was really informative. |
| What was your favorite part? |
| The most favorite part was SSH module for Python Paramiko for which knowledge of Python as well as as networking was require. |
| What was your least favorite part? |
| Error handling was the least part as it was quite confusing. |
| How long did it take you to complete the assignment? |
| It took arounf 2,3 hours. |
| Did you find any errors whether they be technical or spelling? |
| No, I didn’t find any. |
| Did you learn anything? |
| Yes, in this lab I learned about error handling using python and came across different option of exception to deal with errors. |