Updated Sep 27, 2023

Milestone 4.1 - Automation with pyvmomi

> 💡Any systems administration activity that is characterized by a good deal of pointing and clicking, selection of options, typing of free text such as IP addresses, names of VMs and arbitrary commands is prone to error. Automation makes systems administration tasks more consistent and easily repeatable. VSphere has a vast array of automation options including ansible, powercli, the REST API, Terraform and likely others. We are going to use python3 in conjunction with the pyvmomi library to learn about automation of enterprise virtualization tasks.
>
> You will need to brush off your python skills.

Resources

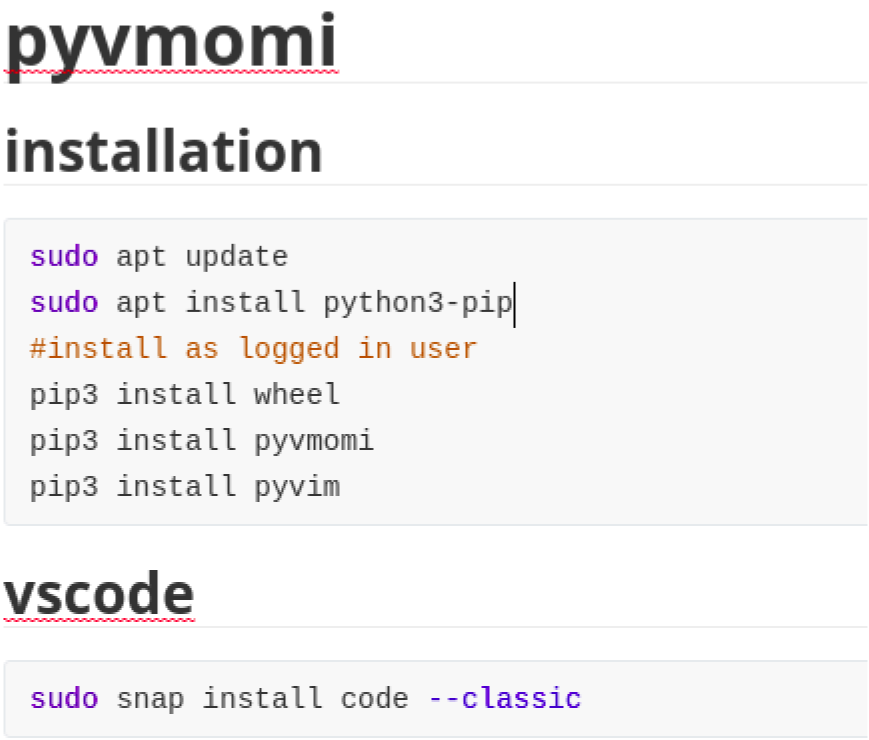
The following pyvmomi samples are great and provide simple driver programs to invoke some common vsphere operations. You are encouraged to import and use functions from the tools directory in your own programs.

- https://github.com/vmware/pyvmomi-community-samples
    - These samples have all the vsphere examples you will need to complete your program. You will just need to integrate it and modify as you see fit.
- This is a great tutorial for making a basic python program with a menu.
- Demonstration of interactive login using pyvmomi
- Instructor Demonstration (without the code review)
- Another awesome resource

Prerequisites

The following tech journal excerpt shows a method to install the dependencies for pyvmomi and visual studio code on mgmt1. If you've not enabled a method to get to mgmt01 full screen with copy/paste it is time to do so. See early suggestions on Chrome Remote Desktop.

Updated Sep 27, 2023



pyvmomi

installation

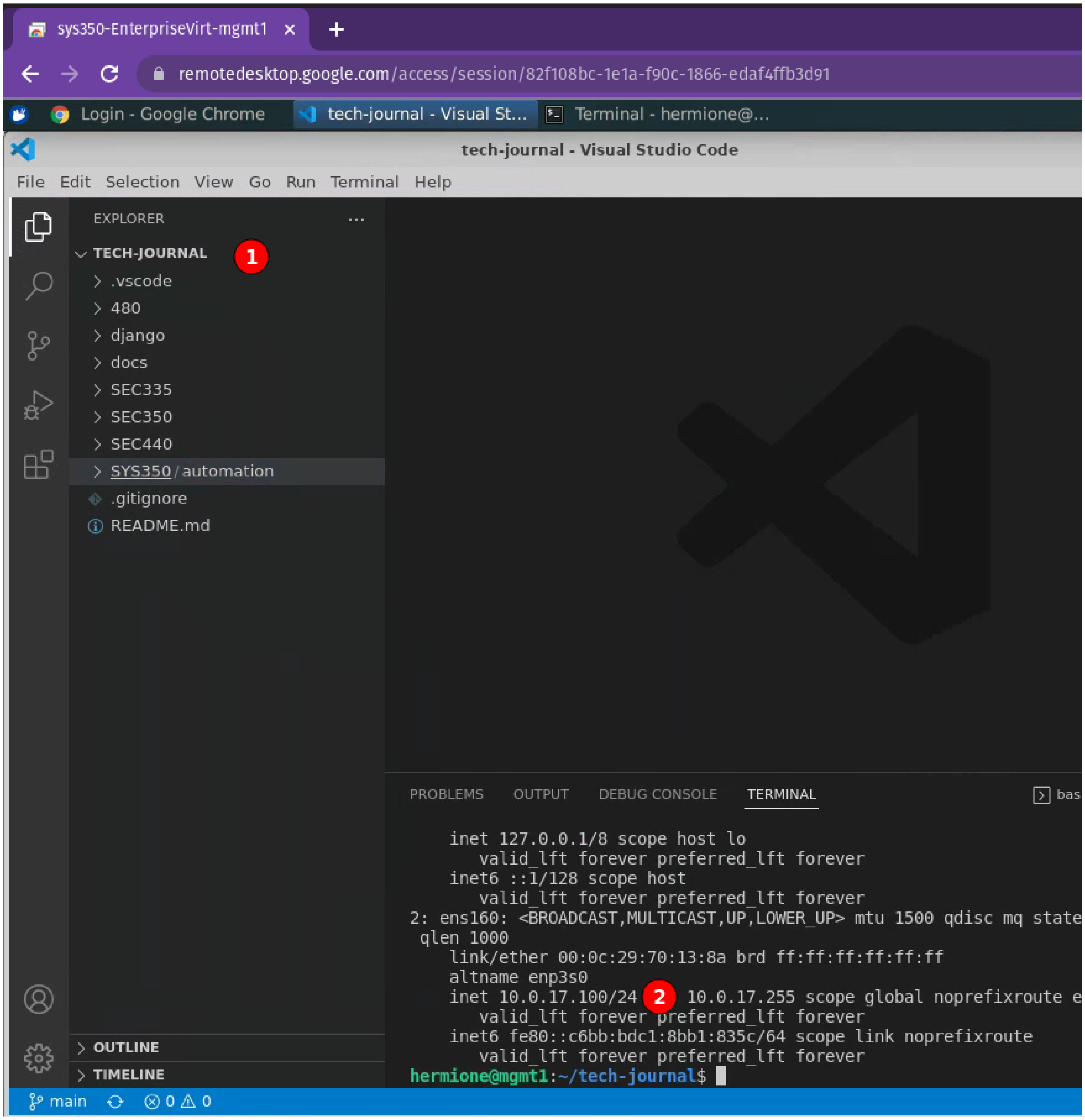
```
sudo apt update
sudo apt install python3-pip
#install as logged in user
pip3 install wheel
pip3 install pyvmomi
pip3 install pyvim
```

vscode

```
sudo snap install code --classic
```

Updated Sep 27, 2023

Deliverable 1. Figure out how to clone your course git repository. Open the base directory in visual studio code(1), and cover in your video demo a view similar to the one below that also shows a terminal with your mgmt1 IP address(2)


sys350-EnterpriseVirt-mgmt1
remotedesktop.google.com/access/session/82f108bc-1e1a-f90c-1866-edaf4ffb3d91
Login - Google Chrome
tech-journal - Visual St...
Terminal - hermione@...
tech-journal - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
TECH-JOURNAL
1
.vscode
480
django
docs
SEC335
SEC350
SEC440
SYS350 / automation
.gitignore
README.md
OUTLINE
TIMELINE
main
0 0
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
bas
2

```
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state
 qlen 1000
    link/ether 00:0c:29:70:13:8a brd ff:ff:ff:ff:ff:ff
    altname enp3s0
    inet 10.0.17.100/24     10.0.17.255 scope global noprefixroute e
       valid_lft forever preferred_lft forever
    inet6 fe80::c6bb:bdc1:8bb1:835c/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
hermione@mgmt1:~/tech-journal$ 
```

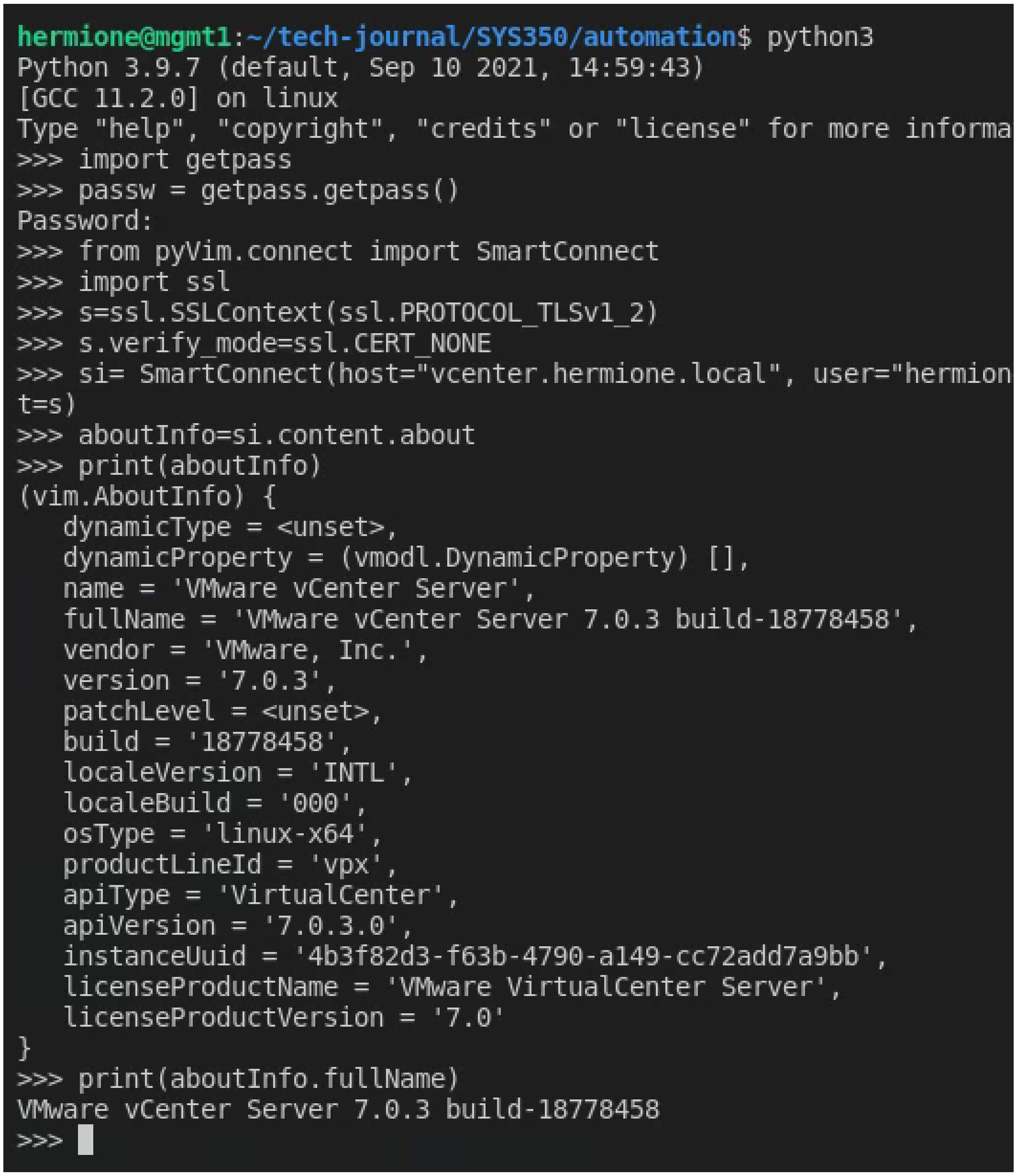
An interactive session

Watch the following video. Extend this interactive example into a python program that has the required features.

Updated Sep 27, 2023

`Deliverable 2. Demo your own interactive session with vcenter via pyvmomi. Print out an element of the aboutInfo object. Similar to the one below.`

Note: SmartConnect function is cut off a bit in screenshot - but need to use the passw variable and sslContext parameters. Review docs on the pyvim SmartConnect function!



```
hermione@mgmt1:~/tech-journal/SYS350/automation$ python3
Python 3.9.7 (default, Sep 10 2021, 14:59:43)
[GCC 11.2.0] on linux
Type "help", "copyright", "credits" or "license" for more informa
>>> import getpass
>>> passw = getpass.getpass()
Password:
>>> from pyVim.connect import SmartConnect
>>> import ssl
>>> s=ssl.SSLContext(ssl.PROTOCOL_TLSv1_2)
>>> s.verify_mode=ssl.CERT_NONE
>>> si= SmartConnect(host="vcenter.hermione.local", user="hermion
t=s)
>>> aboutInfo=si.content.about
>>> print(aboutInfo)
(vim.AboutInfo) {
   dynamicType = <unset>,
   dynamicProperty = (vmodl.DynamicProperty) [],
   name = 'VMware vCenter Server',
   fullName = 'VMware vCenter Server 7.0.3 build-18778458',
   vendor = 'VMware, Inc.',
   version = '7.0.3',
   patchLevel = <unset>,
   build = '18778458',
   localeVersion = 'INTL',
   localeBuild = '000',
   osType = 'linux-x64',
   productLineId = 'vpx',
   apiType = 'VirtualCenter',
   apiVersion = '7.0.3.0',
   instanceUuid = '4b3f82d3-f63b-4790-a149-cc72add7a9bb',
   licenseProductName = 'VMware VirtualCenter Server',
   licenseProductVersion = '7.0'
}
>>> print(aboutInfo.fullName)
VMware vCenter Server 7.0.3 build-18778458
>>> 
```

Updated Sep 27, 2023

Your Python Program

Constraints

- You are running python3
- You use either the pyvmomi library or the REST API (note: not a lot of advice on this one but it might be fun).

Requirements

Deliverable 3. Provide a link to a short video hosted on googledrive or panopto that clearly demonstrates that you've met the following requirements.
Ensure you share this video with your instructor. Your video must be captured at 1080p or better and must have voice. Talk through the demonstration of deliverables 1 and 2 and your python program. Provide a very brief walkthrough of the python code you wrote in the execution of this milestone.

- Requirement 1: Read your vmware username and vcenter hostname from a file.

- Requirement 2: Provide data from your current pyvmomi session. This must include DOMAIN/username, vcenter server and your source IP address.
  Update: Pull the domain/username and source ip from the connection information and vcenter_server can be the variable you read from your configuration file. See the following video for debugging advice.

- Requirement 3: Create a search function that filters all vms in vcenter by name. If no filter is added, return all VMs

- Requirement 4: For each VM returned, provide the following meta-data
  - VM Name
  - Power State
  - Number of CPUs
  - Memory in GB
  - IP Address (don't forget to add vmware tools to pf and any other "on" system that does not have an IP address returned.

Updated Sep 27, 2023

Deliverable 4: Tech Journaling - Technical (Demo)
Briefly demonstrate your documentation and code for this milestone. The demo provided allows us to easily navigate your journal's technical components. The journal is easy to read, there are screenshots where appropriate and formatted commands when it would be useful to copy paste. You link to repository content rather than formatting lengthy commands in markdown. Content that is suitable to multiple courses should be elevated and linked to rather than hidden in your course specific wiki.

Deliverable 5: Tech Journaling - Reflection (Demo)
You discuss the task in general terms and you highlight where you went off course, where you burned time and any breakthroughs you had.