Environment

Spawn a Ubuntu 16.04 *virtual machine* - you can use readily available VMware/VirtualBox/Parallels VMs for this purpose. You can use Vagrant.

Exercise 1:

Programming

NOTE: *Although below Python is mentioned, any of the following programming languages are accepted: Python, Ruby, Golang, C/C++, Java.*

Create a simple python script `getweather.py` with the following specs

- Retrieves weather data from https://openweathermap.org/api
- It uses the `current weather data` API: https://openweathermap.org/current
- It uses pyown: https://pypi.python.org/pypi/pyowm
- The python script uses *no arguments*, only the following environment variables:
  - `OPENWEATHER_API_KEY`
  - `CITY_NAME`
- It outputs to stdout

Example of acceptable results:

```
$ export
declare -x OPENWEATHER_API_KEY="xxxxxxxxxxxx"
declare -x CITY_NAME="Honolulu"
$ python getweather.py
source=openweathermap, city="Honolulu", description="few clouds", temp=70.2,
humidity=75
```

Ansible

All steps below must be done using Ansible:

- Install the `Docker service`
- Enable container logging to Docker host's syslog file [1]

NOTE: Settings for privilege escalation and modularisation are acceptable (`become: yes`, `ansible-roles`, etc)

Example of expected result:

```
$ docker
The program 'docker' is currently not installed. You can install it by typing:
```

```
sudo apt install docker.io
$ ansible-playbook -i "localhost," -c local site.yml
...
...
PLAY RECAP *********
localhost                  : ok=9    changed=1    unreachable=0    failed=0
$ docker -v
Docker version 17.12.0-ce, build c97c6d6
$ docker info | grep 'Logging Driver'
Logging Driver: syslog
```

Docker

- Build a docker image (`Dockerfile`) configured to run as executable
- The docker container should pack the `getweather.py` script

Example of expected result:

```
$ docker run --rm -e OPENWEATHER_API_KEY="xxxxxxxxxxxx" -e CITY_NAME="Honolulu"
weather:dev
$ grep openweathermap /var/log/syslog
Nov 30 11:50:07 ubuntu-vm ae9395e86676[1621]: source=openweathermap,
city="Honolulu", description="few clouds", temp=70.2, humidity=75
```

Exercise 2:

Programming

Build a program (in any language) for repetitive network scans displaying differences between subsequent scans.

- scan can be executed either using external tools or using dedicated libraries of selected programming language
- target of the scan must be parametrized as CLI argument
- target can be single IP address as well as network range

Example of expected result:

Initial scan:

```
$ ./scanner 10.1.1.1
*Target - 10.1.1.1: Full scan results:*
Host: 10.1.1.1    Ports: 22/open/tcp////
Host: 10.1.1.1    Ports: 25/open/tcp////
```

Repetitive scan with no changes on target host:

```
$ ./scanner 10.1.1.1
*Target - 10.1.1.1: No new records found in the last scan.*
```

Repetitive scan with changes on target host:

```
$ ./scanner 10.1.1.1
*Target - 10.1.1.1: Full scan results:*
Host: 10.1.1.1    Ports: 22/open/tcp////
Host: 10.1.1.1    Ports: 25/open/tcp////
Host: 10.1.1.1    Ports: 80/open/tcp////
```

Exercise 3:

Syslog configuration

Configure rsyslog service with the following settings:

- logging of default log files from /var/log/*
- logging of custom log files

Ansible

Configuration must be executed using Ansible utilizing concept of Ansible roles. Ansible role should accept the following parameters:

- logging only default log files
- logging custom files
- selecting external log server to send logs to

Example of expected result:

- proper contents of /etc/rsyslog.d/ folder
- logs properly delivered to external syslog server

Evaluation

Candidate selects the amount of exercises to elaborate. In case not full scope of the exercise is delivered, we ask candidate to mention gaps and applied workarounds.

Results - Source Code & Readme-s

All material used for this exercise should be uploaded to Github and the repository shared.

References

[1] https://docs.docker.com/engine/admin/logging/overview/#configure-the-default-logging-driver