TMT

Test Management Tool

Introduction to tmt (Test Management Tool)

- tmt is used to run multiple tests in different architectures and operating systems at once.
- tmt is a tool that helps developers manage and run tests.
- tmt uses podman to pull images from different sources like docker,git, quay and even our own storage also.
- The tests are organized, and you can reuse tests across different projects.

tmt installation on ubuntu

1. First, update your package list to ensure you're working with the latest versions:

   ```
   sudo apt update
   ```

2. Install tmt Using pip:

   ```
   sudo apt install python3-pip
   ```

   Now, install tmt:

   ```
   pip3 install tmt
   ```

3. The installed tmt command is located in **/home/vlab/.local/bin**, so add its path in **~/.bashrc** file

   ```
   nano ~/.bashrc
   ```

   Add the following line at the end of the file: **export PATH=$PATH:/home/vlab/.local/bin**

4. Install or upgrade the click library (dependency)

   ```
   pip3 install --user --upgrade click
   ```

5. Verify Installation:

   ```
   tmt --version
   ```

Podman installation on ubuntu

```
$ . /etc/os-release

$ sudo sh -c "echo 'deb http://download.opensuse.org/repositories/devel:/kubic:/libcontainers:/stable/xUbuntu_${VERSION_ID}/ /' > /etc/apt/sources.list.d/devel:kubic:libcontainers:stable.list"

$ wget -nv https://download.opensuse.org/repositories/devel:kubic:libcontainers:stable/xUbuntu_${VERSION_ID}/Release.key -O- | sudo apt-key add -

$ sudo apt update

$ sudo apt -y install podman

$ podman --version
```

Image Creation through podman

```
podman pull --arch riscv64 ubuntu    #pull the target image and specify the architecture

podman run -it --name riscv docker.io/library/ubuntu /bin/bash    #this creates a container for the image with a unique id

podman images                        #lists all the images available and find your image name which you pulled

podman ps -a                         #this lists all the containers available, find your container id

podman start -ai 541154f026ae        #here my container id is 541154f026ae, start the image and install the requirements

podman commit 541154f026ae ubuntu_riscv64    #commit the changes like any installations or test requirements to the image
```

Note:

1. The container id changes for every run and the image id also changes for every commit.
2. You must login into your docker account prior to pulling of the image from podman else you may get authentication failure.

tmt initialization

```
$ tmt init --template mini

$ ls -a

$ cd plans

$ ls

$ gvim example.fmf                    #place the test scripts you want to run, refer to next slide

$ gvim example.py                     #any test script, this file contains a print statement
```

Example.fmf code

```
summary: Run a python script

provision:
 - how: container
   image: ubuntu_riscv64
 - how: container
   image: debian_trixie_riscv64
 - how: container
   image: docker.io/imbearchild/fedora-rv64

execute:
  how: tmt
  script: python3 example.py
```

Note : The image name is the image which we create through podman .

#Provision describes the environment needed for testing. We can use container (via podman) or a local image.

#execute section contains the test scripts

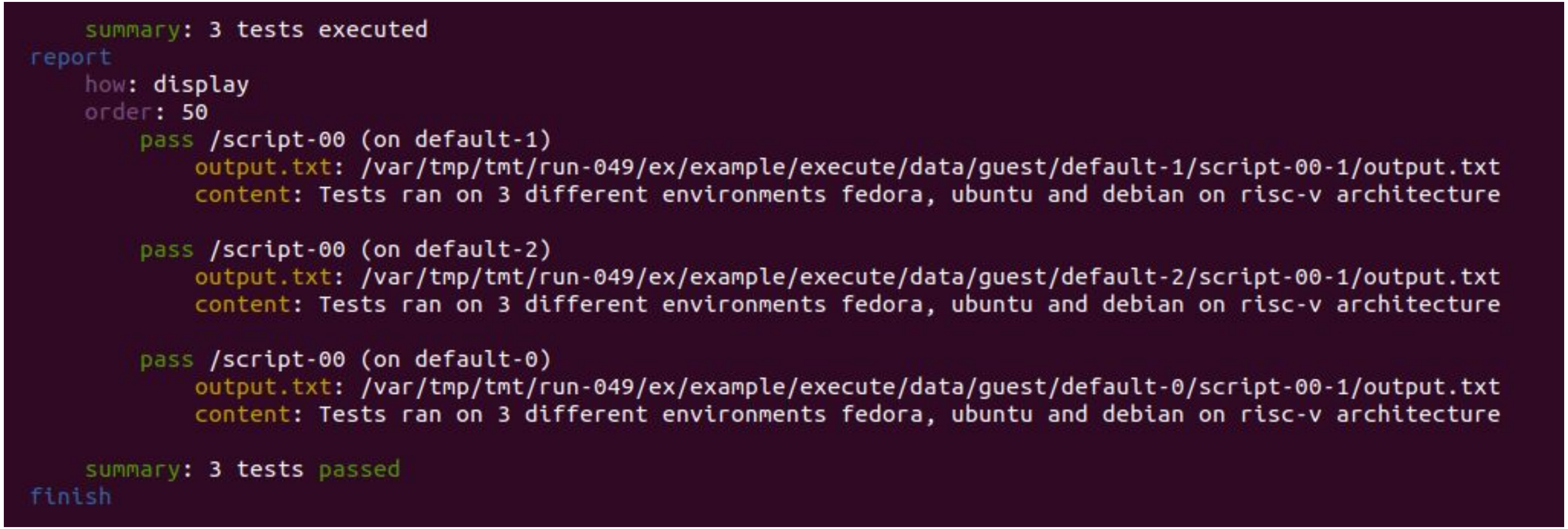
Run the tmt

```
cd ..

tmt run -v              #run the tests

tmt run -vvv            #this gives the complete verbose and output from the tests in the terminal
```

Output from the run



```
    summary: 3 tests executed
report
    how: display
    order: 50
        pass /script-00 (on default-1)
            output.txt: /var/tmp/tmt/run-049/ex/example/execute/data/guest/default-1/script-00-1/output.txt
            content: Tests ran on 3 different environments fedora, ubuntu and debian on risc-v architecture

        pass /script-00 (on default-2)
            output.txt: /var/tmp/tmt/run-049/ex/example/execute/data/guest/default-2/script-00-1/output.txt
            content: Tests ran on 3 different environments fedora, ubuntu and debian on risc-v architecture

        pass /script-00 (on default-0)
            output.txt: /var/tmp/tmt/run-049/ex/example/execute/data/guest/default-0/script-00-1/output.txt
            content: Tests ran on 3 different environments fedora, ubuntu and debian on risc-v architecture

    summary: 3 tests passed
finish
```

THANK YOU