Automated Testing via XTesting

Xtesting framework has been developed in OPNFV which deals with dockerization of test suites and can thus be integrated in different CI systems. For more about XTesting and how to start with it please check it out here.

Our goal here is to demonstrate a work flow that deploy the OSC RIC platform and afterwards does a health check, which gives a pretty good idea on how it can be applied to other OSC projects. Then on top of the deployed RIC platform the kpimon-go xApp will be onboarded as a setup for further testing. Hopefully this can be used as a CI/CD pipeline in OSC to serve in the RICPLT/RICAPP areas and can also help automated test and integration.

For this workflow to work one needs to prepare 2 instances (physical server or VM) with fresh **Ubuntu 20.04**. One is to be used as the XTesting control node, and the other one to be the system under test, on which OSC release F RIC platform is to be deployed (including the initial Kubernetes setup, and all the helm charts deployed). Then a XTesting test case on RIC Platform health check will be run from the control node against the SUT prepared.

1. On the XTesting control node, some dependencies are to be installed which include:
    a. Openssl-server if it doesn't come with the instance provisioned
    b. Docker
    c. XTesting dependencies (ansible etc.)
    d. All above are incorporated into a shell script called install-dependencies for your convenience
2. On the SUT node:
    a. Openssl-server
    b. Install authorized key for remote ssh access (especially for root as it needs to copy over the .kube/config file). Otherwise one may have to configure user/password credentials to be used in the workflow.

It's a 4-step process in the workflow:

1. Deploy a kubernetes cluster via a customized kubeadm container, that takes all necessary configuration parameters (such as IP, username /password, etc.):
    - git clone https://github.com/pekwatch746/kubeadm.git
    - Update the sample_env file to point to the IP address of the SUT node and other user/password settings etc. (ANSIBLE_HOST_IP= SUT's IP). With an instance created on the POWDER testbed, it may look like:

        ANSIBLE_HOST_IP=155.98.36.9

        ANSIBLE_USERNAME=osc_int

        ANSIBLE_PASSWORD=osc_int

        PROJECT_ROOT=/kubeadm/config

    - Add a private key to the inventory/sample folder if a private key is used for ssh access.
    - Build docker image: sudo docker build -t kubeadm .
    - Run once it's built successfully: sudo docker run -v ~/.kube:/kubeadm/config kubeadm // this will copy over the Kubernetes config file to the home directory
2. Build a 2nd richelm container to complete the remaining steps for RIC platform deployment:
    - git clone https://github.com/pekwatch746/richelm.git
    - Build the container: sudo ./**build**.sh static
    - Run with the Kubernetes configuration: sudo docker run -ti --rm -w /apps -v ~/.kube:/root/.kube -t richelmlegacy:1.19.16
3. Run a health check test case against the SUT that's supposed to have the RIC platform successfully deployed. Below a simple robot script for health check:

    ```
    ---
    - name: Shell module example
    hosts: 127.0.0.1
    tasks:

    - name: Check system information
    shell:
    "curl -v http://155.98.36.9:32080/appmgr/ric/v1/health/ready 2>&1"
    register: curl_info

    - debug:
    msg: "{{curl_info.stdout_lines}}"
    ```

4. Onboard/install the kpimon-go xApp from the XTesting host using ansible:
    - git clone "https://gerrit.o-ran-sc.org/r/it/test"
    - cd test/XTesting/xapp
    - scp over the deploy.sh from the folder to the SUT node to the /root directory, and add a line of the IP address of the SUT node to the hosts.yaml file
    - run: ansible-playbook -b -v -i hosts.yaml --become --private-key $KEYFILE cluster.yaml

To make things easier I also put together a Shell script that combines the above steps so you can just run it in one shot:

git clone "https://gerrit.o-ran-sc.org/r/it/test"

cd test/XTesting/XTesting-demo && sudo ./demo.sh target-ip private-key-file-path OPTIONAL-working-directory

Happy XTesting and hope your test automation can be as easy as above 1-2-3.

Finally I would like to thank Sridhar Rao, the Linux Foundation TPM to OSC, who offered huge help and support to this demonstration. He's also an expert to XTesting so feel free to contact James Li and Sridhar Raoif you have any question.