VAQA Readme

Table of Contents

1. What is VAQA?

VAQA is test automation harness written in python.
Main goal of VAQA is to simplify task of running a test or set of tests multiple times
with different combinations of configuration parameters.

2. How to check out VAQA?

```
# cvs login
# cvs co tools/harness/vaqa
```

3. PREREQUISITES

python 2.4+

Note: Some test case requires python paramiko and psutil packages.

4. VAQA Structure

VAQA

- **test-cases**
  - lib
    - <baselib.sh>
    - <baselib.py>
    - <testlist>
  - [test-dir-1]
    - <testlist>
    - <test-case-1>
    - <test-case-1.conf>
    - <test-case-2>
  - [test-dir-2]

.

.

- [test-dir-n]

- **test-results**
  - <test-result-log>
  - <test-error-log>
  - backup
- **config**
  - GLOBAL.conf
- **Engine**
  - lib
    - agentlib.py
    - vmssc_consts.py
    - os_consts.py
  - bin
    - run
    - const.py

test-cases

- It's a test case dir.

- VAQA searches this directory for test cases to run.

- User can add new test cases in this directory.

test-cases/lib

- regularly used library routines are defines in **baselib.sh/baselib.py** files contained in this dir.

- User test cases can source/import these script to get access to common function like add_gp, del_gp, create_fs, mount_fs etc.

- It's basically used when 'SINGLE_RUN' config. parameter is set to 1

testlist

- This is a file which defines the order in which test cases need to be run.

- testlist can be created in base 'test-cases' dir or any other directory within test-cases dir.

Note: See the 'Running VAQA test cases' section to get better idea on how 'testlist' is used.

test-cases/test-dir-1/test-case-1.conf

- We can define test case specific local configuration parameters in test-case.conf file. i.e. If our test case is 'test1.sh' then we may define local configuration parameters for test1.sh in 'test1.sh.conf' file.

- Local test specific configuration file has higher priority than GLOBAL configuaration file. Meaning, If a parameter is defined in both local as well as GLOBAL conf. file with different value then GLOBAL value is overwritten by local config. value.

config/GLOBAL.conf

- The configuration parameters defines in 'GLOBAL.conf' file has global scope. i.e. The config. parameters defined in this file are accessible to all test cases in VAQA/test-cases

Engine/lib/agentlib.py

- This is main VAQA harness library file used to run test case.

Engine/lib/vmssc_consts.py

- All vmssc consts used by agentlib.py are defined in this file.

Engine/lib/os_consts.py
- All OS constants used while creating/mounting/umounting FS etc. are
defined in this file.

Engine/bin/run
- This is a python executable used to execute VAQA test cases.

Any test case run using VAQA, generally has three phases:-

#	Test Run Cycle	comments
1.	Init Phase	create FS
		mount FS
		add GP
2.	Run Phase	execute test case
3.	Cleanup Phase	delete GP
		umount FS

In cases, where we just wanted to execute a test case (skipping #1 and #3)
set SINGLE_RUN=1 in local test config. file.
i.e. If we want to run test1.sh script without #1 and #3 then test1.sh.conf would look like,
test1.sh.conf
[FIXED]
SINGLE_RUN=1

test-results
- Test case results will be generated in test-results dir.
- Two result log files are generated for each test cases. one for STDOUT and
other for STDERR
- When we run a test case, previous log files from test-results are moved to
test-results/backup dir.

5. General flow of test case execution.

1. Read the configuration file for test, if there is a test-specific
config file then it will be merged with GLOBAL configuration [GLOBAL.conf]
file. Here common GLOBAL configuration parameters will be overwritten
by local configuration parameters. Local configuration has precedence over GLOBAL.conf parameters.
i.e. mergedconf = GLOBAL configuration + LOCAL configuration
If test specific local configuration file does not exists then it will
use GLOBAL configuration file.
i.e. mergedconf = GLOBAL configuration

Example:

If test specific local configuration has POLICIES=test_aes256 and GLOBAL.conf file has

POLICIES = encrypt_all then test will be executed with POLICIES set to "test_aes256"

There are some config. parameters which are used in all test cases like MNTPT,
FSTYPE etc. If VAQA does not find these parameters defined in local as
well as GLOBAL.conf then DEFAULT values defined in Engine/bin/const.py are

used.

2. Initialize the test execution environment as per computed merged configuration parameters.

3. Run the test case with that environment.

4. Clean up the setup environment on exit.

5. Result for test cases are logged in the 'test-results' directory.

6. Frequently used configuration parameters.

BUILD_VERSION

#	Confiuration Parameter	Example value	Comment
1	SINGLE_RUN	0 OR 1	SINGLE_RUN=1 Run test case without init and cleanup phase. i.e. create FS, mount FS, add GP, delete GP and umount FS steps will be skipped. SINGLE_RUN=0 This is default behavior.
2.	MNTPT	/gp	When SECFS_DEVICE parameter is definied then FS will be mounted on this directory. It is also used as parent directory of GP's. i.e. Guard points will be created under MNTPT like GP0=/gp/gp0 MNTPT=/gp
3	FSTYPE	ext3, ext4, vxfs etc.	Which FS to create on SECFS_DEVICE. FSTYPE=ext4
4	SECFS_DEVICE	/dev/sdb1	Which device to use to create FS. SECFS_DEVICE=/dev/sdb1
5	SECVM_DEVICE	/dev/sdc	Which device to use as RAW device. SECVM_DEVICE_PATH will be created which will map to /dev/secvm/$SECVM_DEVICE SECVM_DEVICE=/dev/sdc

6	NFS_SHARE	10.3.57.90:/mnt/vol1/vol1/vol1/10.3.57.22	It defines NFS 3 share path. NFS_SHARE=10.3.57.90:/mnt/vol1/vol1/vol1/10.3.57.22
7	NFS4_SHARE	10.3.57.91:/10.3.57.30	It defines NFS v4 share path. NFS4_SHARE=10.3.57.91:/10.3.57.30
8	NO_ENCRYPTION	0 OR 1	It is another useful configuration parameter which can be used to run test case skipping add GP and delete GP steps. i.e. It's similar to SINGLE_RUN=1, but here init and cleanup phases also comes into picture, skipping add GP and delete GP part resp. Another diff. between SINGLE_RUN and NO_ENCRYPTION flag is, NO_ENCRYPTION=1 uses ITERABLES section in config. files. i.e. If there are two values assigned to a key in ITERABLES section then test will execute twice skipping add GP and delete GP in both cases. NO_ENCRYPTION default value is 0.
9.	POLICIES	encrypt_all, test_aes256 etc.	It defines which encryption policy to use to guard GP. POLICIES=encrypt_all If test case need to run multiple times with different values of POLICIES then write it under ITERABLES section with comma separated policy values.
10	BUILD_SERVER	10.3.10.45:/nfs-builds/nightly/agent	build server location used by buildValidation test cases to pick new build for installation. It is usually written in GLOBAL.conf under BUILD section. BUILD_SERVER=10.3.10.45:/nfs-builds/nightly/agent

11	BUILD_MNTPT	/agentbuild, /mnt etc.	It is local directory path where BUILD_SERVER is mounted. This parameter is used in buildValidation test cases. It is usually written in GLOBAL.conf under BUILD section. BUILD_MNTPT=/agentbuild
12	BUILD_GREP	vor_5.2.3_dse_rel_5_2_3_branch_20, vor_5.2.2_dse_rel_5_2_2_branch_20 etc.	It's a pattern which used by buildValidation test cases and is used to select build branch to be used for installation. It is usually written in GLOBAL.conf under BUILD section, to install build from 5.2.3_dse branch use, BUILD_GREP=vor_5.2.3_dse_rel_5_2_3_branch_20
13	BUILD_VERSION	5.2.3, 5.2.2, 6.0.0 etc	It is used by buildValidation test cases and is used to select build version to be installed. It is usually written in GLOBAL.conf under BUILD section, to install build from 5.2.3_dse branch use, BUILD_VERSION=5.2.3
14	BUILD_NAME	vor_5.2.3_dse_rel_5_2_3_branch_2015.01.05.09.06.44 etc.	It is used by buildValidation test cases and is used to install mentioned build. It is usually written in GLOBAL.conf under BUILD section. BUILD_NAME=vor_5.2.3_dse_rel_5_2_3_branch_2015.01.05.09.06.44
15	AGENT_TYPE	fs, key or db	It is used by buildValidation test cases and used to decide type of build to be installed. i.e. fs build, key build or db build. Default is set to 'fs' build. It is usually written in GLOBAL.conf under BUILD section, to install build from 5.2.3_dse branch use, AGENT_TYPE=fs

16	username	voradmin etc	It defines DSM username. It's used by VAQA to generate and export vmssc.conf file on the fly. It is usually written in GLOBAL.conf file under SEC_SERVER section. username = voradmin
17	password	Ssl12345# etc.	It defines DSM login password. It's used by VAQA to generate and export vmssc.conf file on the fly. It is usually written in GLOBAL.conf file under SEC_SERVER section. password = Ssl12345#
18	domain	doamin1 etc.	It defines DSM domain name. It's used by VAQA to generate and export vmssc.conf file on the fly. It is usually written in GLOBAL.conf file under SEC_SERVER section. domain = domain1
19	server	asrm-ss5.i.vormetric.com etc.	It defines DSM hostname. It's used by VAQA to generate and export vmssc.conf file on the fly. It is usually written in GLOBAL.conf file under SEC_SERVER section.

Note:

There are some other configuration parameters like N_GUARDS, LOCAL_FLAG, SECFSD_STATUS_LAG, SHARE_RESULT, AUTO_MNTPT etc.

VAQA internally create some configuration parameters like HOST (FQDN of host), ROOT_DIR, SECVM_DEVICE_PATH (secvm device mapping path) etc.

7. How to run VAQA test cases?

1. Go to VAQA_ROOT/Engine/bin

2. ./run test/test1.sh
=> Above command executes test1.sh script from VAQA/test-cases/test/ dir.

3. How to use 'testlist' file to run test1.sh file with some build (say X)?

```
VAQA/test-cases:-
    - test
        - test1.sh
```

```
            - test2.sh
        - buildValidation
            - install.sh
            - register.sh
            - uninstall.sh
```

A. Create a 'testlist' file in VAQA/test-cases/test dir.

B. Add below lines in 'testlist' file
vim VAQA/test-cases/test/testlist
../buildValidation/install.sh
../buildValidation/register.sh
test1.sh
../buildValidation/uninstall.sh

C. Go to VAQA/Engine/bin directory and run,
./run test

D. As 'testlist' file is present in test-cases/test dir, VAQA executes only files mentioned in testlist file in specified order.

8. What are different sections of configuration files?

- Configuration files can basically have four types of sections viz. FIXED, ITERABLES, SEC_SERVER and BUILD

- FIXED:
  - All config parameters with FIXED value are defined under this section.
- ITERABLES:
  - Config parameters with different values are defined under this section.
  - If we set two values for a configuration parameter then the test case will run twice unless SINGLE_RUN is set.
- SEC_SERVER:
  - DSM related configuration parameters are defined in this section.
- BUILD :
  - Build related configuration parameters are defined in this section.

Example: Global.conf/local conf. file structure:

```
[FIXED]
MNTPT = /gp
FSTYPE=ext4
USER_VAR1 = ABC
USER_VAR2 = XYZ

[ITERABLES]
POLICIES=AES128,encrypt_all
USER_VAR3=AB, AC, BC


[SEC_SERVER]
username=voradmin
password=ssl12345
domain=domain1
server=asrm-ss5.i.vormetric.com
```

Note: User can add any number of config parameters in test specific local configuration files.
All config. parameters in turn will be exported to test scripts as environment variables.
There are some config parameters which are exported to test case and not required to mention in any configuration file.

Example : HOST, ROOT_DIR etc are automatically exported and can be used in user test cases.
HOST: FQDN of local system.
ROOT_DIR : Absolute path to VAQA root dir.

If we run a test case with above mentioned config. parameters set,
then the test case will run for 6 times with different combinations of parameters.

Run 1. FIXED + POLICIES=AES128 + USER_VAR=AB
Run 2. FIXED + POLICIES=AES128 + USER_VAR=AC
Run 3. FIXED + POLICIES=AES128 + USER_VAR=BC
Run 4. FIXED + POLICIES=encrypt_all + USER_VAR=AB
Run 5. FIXED + POLICIES=encrypt_all + USER_VAR=AC
Run 6. FIXED + POLICIES=encrypt_all + USER_VAR=BC