ISSN: 2277-9655
Impact Factor: 3.785

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

AUTOMATION OF REST ENDPOINT OF CLUSTERED STORAGE SYSTEM

Anuradha Shenoy*
Computer Network Engineering Department of ISE, BMSCE

DOI: 10.5281/zenodo.51004

ABSTRACT

Planning, designing, developing, testing and deployment are the different phases in software development life cycle. Manually testing the storage systems is a very tedious job. There is also high chance of missing the validation of parameters. In this paper, automation testing of REST endpoint of storage systems is described which will not only reduce the effort involved in testing but also provide an efficient way of validating all the parameters.

KEYWORD: Clustered Storage, REST architecture, Automation, Storage resources.

INTRODUCTION

Storage systems can be classified into scale-out and scale-in storage solutions. Clustered storage systems are scale-out storage solution. These systems provide non-disruptive, on demand flexibility and operational efficiency. These systems don't have down time or maintenance time as nodes can be added and removed even when the system is up on running. Due to which they have very high operational efficiency. Clustered storage systems is made up of resources like aggregate, storage virtual machine, volumes etc.

Clustered storage system can be managed by using management APIs through remote management service. But this requires strong programming knowledge. REST APIs to manage storage system are easy to use. Every resource is uniquely addressable using a uniform and minimal set of commands. Every storage system resource has its equivalent Uniform Resource Identifier (URI). Using this identifier GET, POST, PUT, DELETE operations can be performed. Manually testing the provisioning is time consuming.

An automation framework is introduced to reduce the time involved in testing. This automation framework will do the REST call, wait for the operation to complete, validate the test results, log the results, executes the test case according to their dependencies, creates the pre-requisite objects and deletes the object based on parent-child relationship.

The paper is organized into following sections. Section II discusses the related work in automation testing. Section III and Section IV provides the details of existing system and its limitations. Section V shows the proposed system and its working. Section VI shares the test result achieved by using automation. Section VII and Section VIII speaks about the conclusion and future work in this field.

RELATED WORK

The automation testing of REST endpoint over manual testing is motivated by some of the extensive research and results achieved through it. **Software Testing Research: Achievements, Challenges, Dreams by Antonia Bertolina[1] provided information about the achivements possible through automation testing over manual testing.** Building Scaled Test Virtual Infrastructure Integrated with NetApp Storage by **Vishal Kulkarni, Naresh Garuda and Sreedhar Reddy provides information how to test huge environment of storage system through automation testing[2].**

ISSN: 2277-9655
Impact Factor: 3.785

EXISTING SYSTEM

In existing system, tester manually tests each storage resource for create, read, update and delete operation. In addition to this he might also need to test it against a combination of properties associated with that resource. On successful completion of the operation he has to manually validate the properties set by him against the REST server output and in the storage. This way testing there is a high chance of manual tester missing some of the combinations of parameters and also missing the validation.

Below are the steps involved in manual testing

1. Identify and create all pre-requisite resources
2. Make a note of all the unique identifier of those resources
3. Create a table with the combination of different properties associated with that resource.
4. Test each combination
5. Delete all pre-requisite object
6. Repeat from step 1 on getting new build

LIMITATIONS OF EXISTING SYSTEM

Each time a new build is released, tester has to repeat all the steps involved in manual testing. Also testing has to be done for all the existing features in order to ensure no new regression issues are introduced. This will involve considerable amount of time. Automation framework will involve one time development activity. After that it can be re-used to test existing as well as new features.

PROPOSED SYSTEM

In the proposed system automation framework is developed for testing provisioning of storage resources. Every storage resource is either a physical object or a logical object. Each object comes either under the scope of a cluster or in the scope of a storage virtual machine. If it comes under the scope of storage virtual machine then as a pre-requisite for that resource storage virtual machine is to be created. Similarly other logical objects have different pre-requisites. Hence as a first step all the pre-requisites are identified.

Certain test cases depend on other test cases to be executed first. For example delete test case depends on create and update to happen successfully, resume operation of a test case can be performed only after suspend operation on that resource is successful. This relation between the test cases is to be identified. Automation framework is designed to take this dependency relation between the test cases.

The values set in the request body of create and update operation needs to be validated both in storage and REST server output. In every test case after the successful completion of operation, validation method is invoked to validate both the output from REST server and storage. If there is any mismatch then the test case is to be failed even though the operation is successful. Validating the error scenarios is also important. Automation includes the expected error code and error message and the REST server output is validated against it.

Each test case requires different amount of time to get executed. This also depends on the kind of operation to be performed. Sufficient wait period is included in the test case for the operation to get complete before validating the REST server output. If the test case doesn't get executed within the specified time, test case is marked as failed.

In order to marshal the test output, for every resource respective "Plain Old Java Object" (POJO) file is written. This file should contain all the properties of that resource. It should also contain other attributes associated with them like mandatory, optional, read only etc. For every property the getter and setter functions is written if it is not just read only parameter. As this needs to be generated for every resource, auto generation script file is written for this. This script file takes the JSON file for the resource as input and generates POJO. Rest server output is serialized to this POJO class file and then validation method is invoked.

The following diagram depicts the flow of a test case

ISSN: 2277-9655
Impact Factor: 3.785

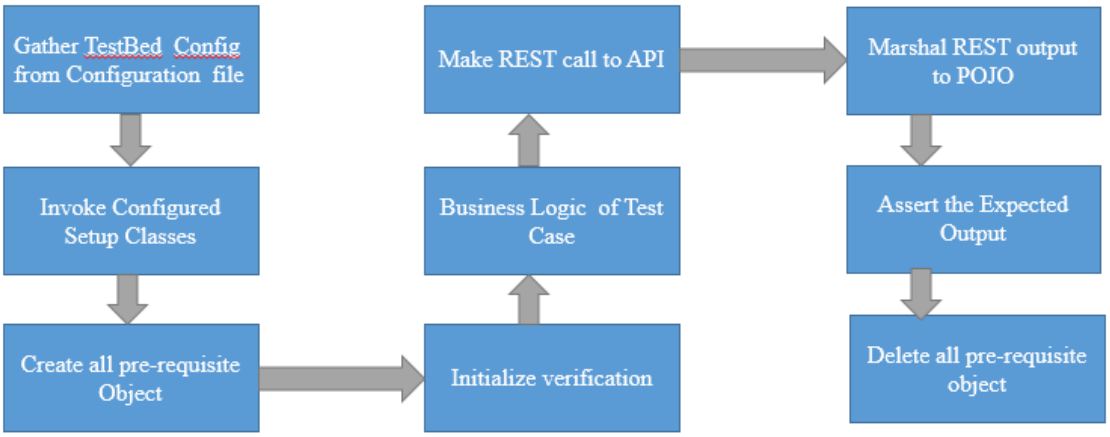

Fig 1: Flow of test case

Resources will be in parent-child relationship. For example storage virtual machine is the parent resource for volume resource, LUN is the child resource for volume resource etc. Deleting the pre-requisite object will involve deleting all the child resources first. Parent resource can be deleted only after all the child resource is deleted. This relationship is identified and a common delete method is written that will ensure the deletion of the child resources first and then the parent.

TEST RESULT

With automation testing it took 1.5 hours to test around 30 different storage resources. Manual testing required a 3 to 4 days to complete it. Also this test suite can be used in future releases to test for regression issues.

CONCLUSION

Automating the storage test cases has many advantages over manual testing. Major drawback of manual testing is the time and human effort involved. Manual testing is also prone to error. In manual process user has to manually create all pre-requisite objects, execute the test case and delete them. He has to wait for the test case to complete execution before going ahead with next one. With automation, test cases can be executed in parallel and validation can be done efficiently. Thus automating the testing of provisioning of storage resources has become an efficient and less time consuming operation over manual process.

FUTUTRE DEVELOPMENT

Dependency relationship exists among test cases. Delete operation always depends on create and update test case. If the intermediate test case fails i.e. if update fails then the delete test case never gets executed. As a result unused resource gets created on storage. This scenario is not handled in current automation framework. Also very good GUI for this automation will provide a very user friendly environment for executing test cases and analyzing the logs.

REFERENCES

[1] **Software Testing Research: Achievements, Challenges, Dreams by Antonia Bertolina in** FOSE '07 2007 Future of Software Engineering
[2] Building Scaled Test Virtual Infrastructure Integrated with NetApp Storage by **Vishal Kulkarni, Naresh Garuda and Sreedhar Reddy**
[3] **Software testing primer by Nick Jenkins**