

Verizon

DEVOPS FRAMEWORK

DevOps Framework and Execution steps

Subramanian, Raja
[Pick the date]

Tools and Add-ons

Please find below the different tools involved in our Devops framework.

JIRA

JIRA is used as a Test case management and Issue tracking tool.



There are three different add-ons in JIRA used by our framework.

ZEPHYR – This add on enable JIRA to act as Test Case Management tool. Without this add on JIRA is just a issue tracking tool.

ZAPI – This add on is used to access test management related components in JIRA through API calls. ZAPI stands for Zephyr Application Program Interface.

Bob Swift – It's nothing but an add-on that allows us to access JIRA through CLI(Command Line Interface).

JENKINS

Jenkins is our Continuous Integration tool.



There are a number of useful plugins that can be used with Jenkins. Please find few plugins that are mainly used in our framework.

JIRA CLI Client – This is a client software for Bob Swift CLI add on in JIRA. We place this software inside the workspace of our Jenkins job to interact with JIRA. This can be downloaded for the following location <https://bobswift.atlassian.net/wiki/display/ACLI/Downloads>.

Git Plugin – This plugin is used to checkout the automation code from stash to Jenkins wokspace.

TestNG Plugin – This plugin publishes the testng results into a graphical format in Jenkins.

Email Notification – Used to send execution status and reports of the Jenkins build through email.

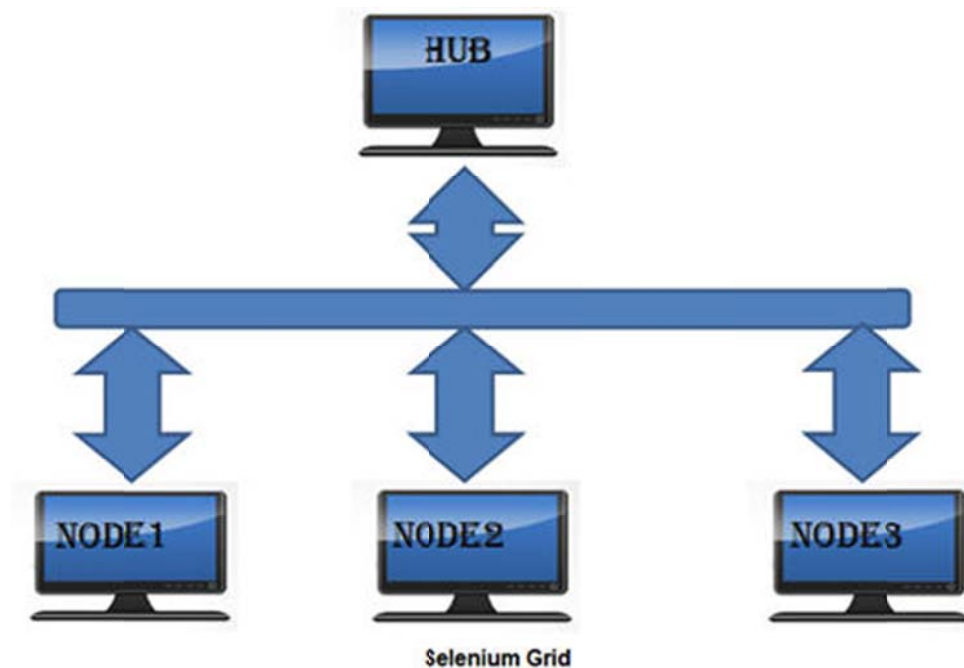
STASH

Atlassian Stash is used as our repository to store Selenium automation code.



SELENIUM GRID

Selenium Grid is a server that allows tests to use web browser instances running on remote machines. Selenium Grid allows running tests in parallel on multiple machines, and to manage different browser versions and browser configurations centrally.

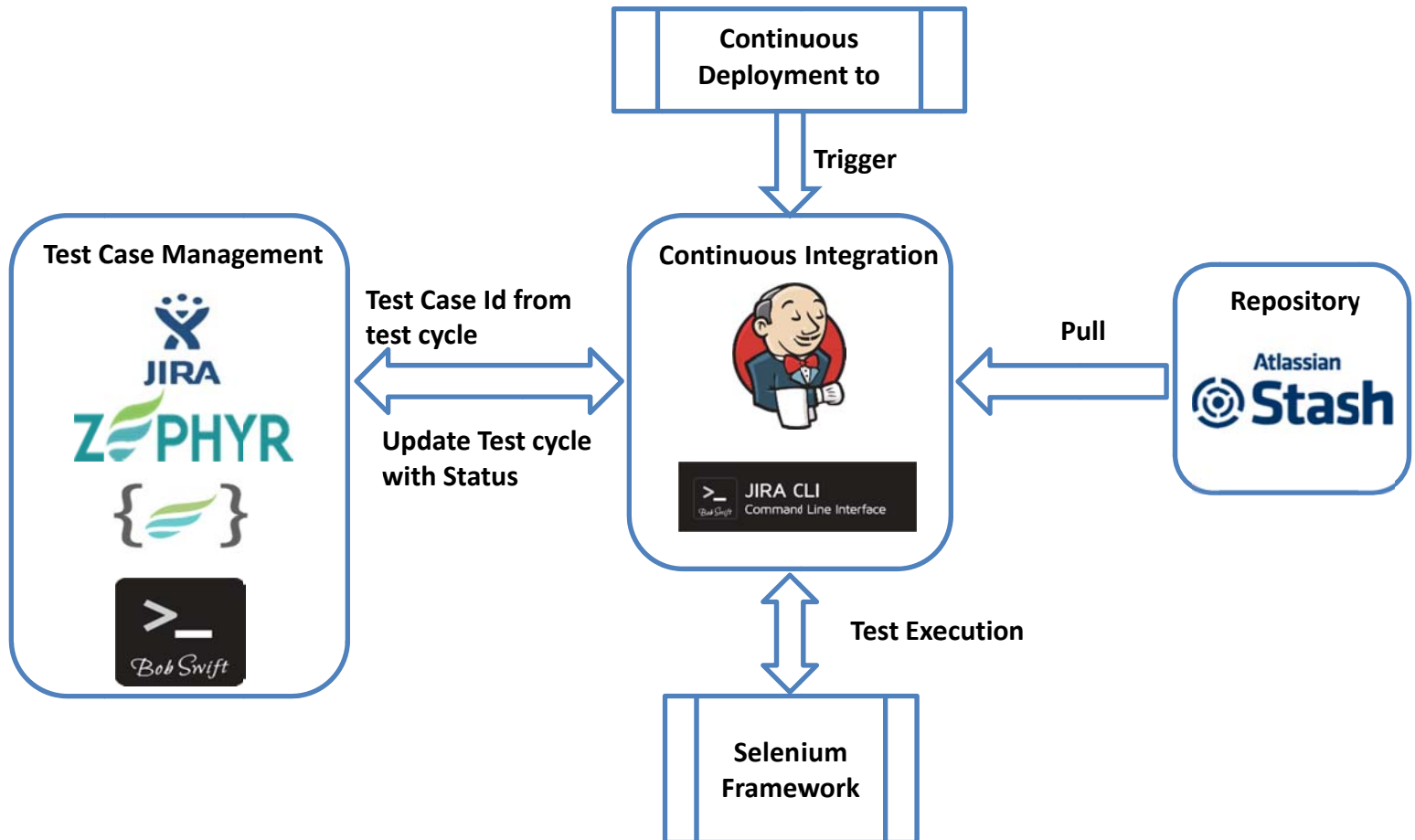


HUB - The hub is the central point that will receive all the test request and distribute them to the right nodes. The hub has a list of servers that provide access to browser instances

NODE – Nodes are a bunch of remote machines with different platforms and browser versions connected to the hub.

DevOps Workflow

The workflow involving all the above tools is depicted below.



CAO DevOps Framework Execution Steps

Please find below the manual steps to execute the CAO DevOps framework for QA Automation.

Jenkins, JIRA and Stash URI's:

<https://onejenkins.ebiz.verizon.com/test>

<http://onejira-test.verizon.com/>

<https://onestash-test.verizon.com/>

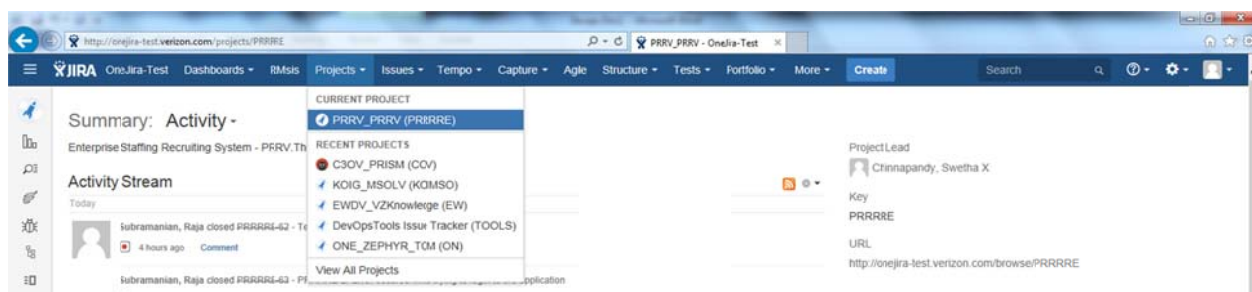
Pre Requisites:

1. Test cases to be created in JIRA for the specific project.
2. Selenium test scripts are to be updated in Stash for the specific project.
3. Selenium hub url should be provided in property file of the selenium script.

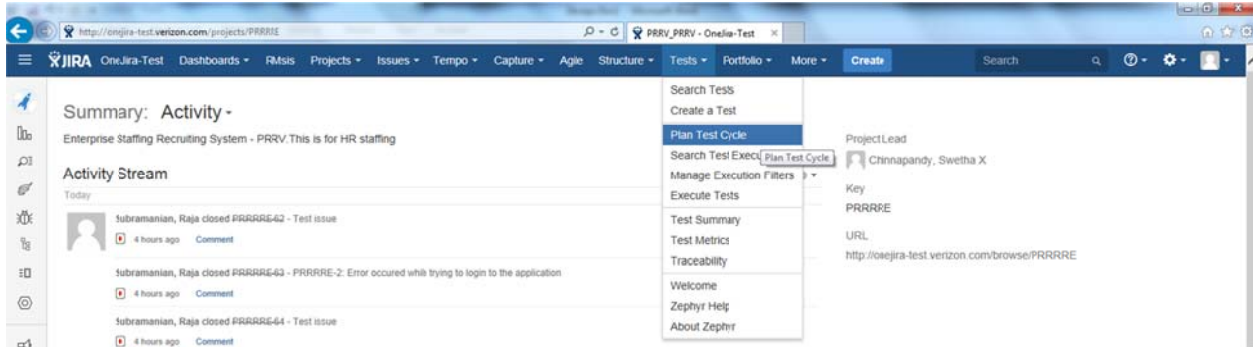
Step 1:

First step is to create test cycle in JIRA.

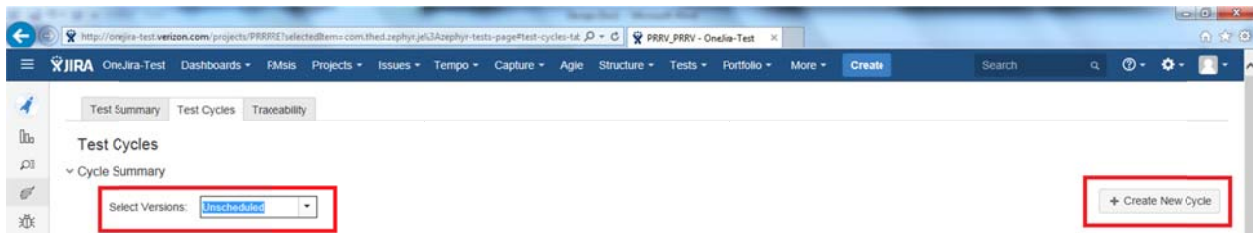
- Login to onejira-test and navigate to the Specific project.



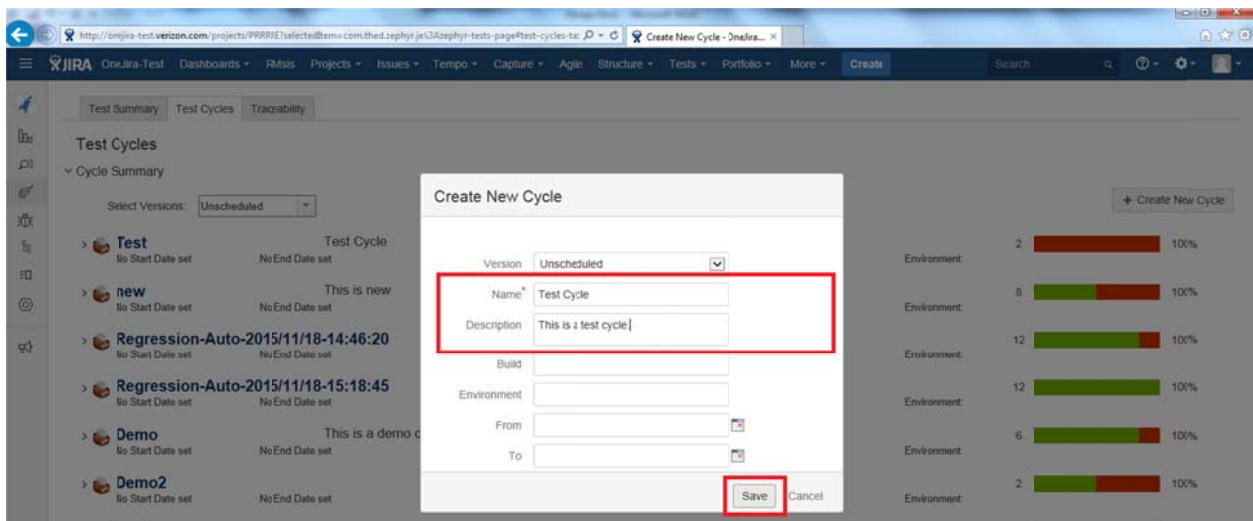
- Click on “Tests” tab and select “Plan Test Cycle”.



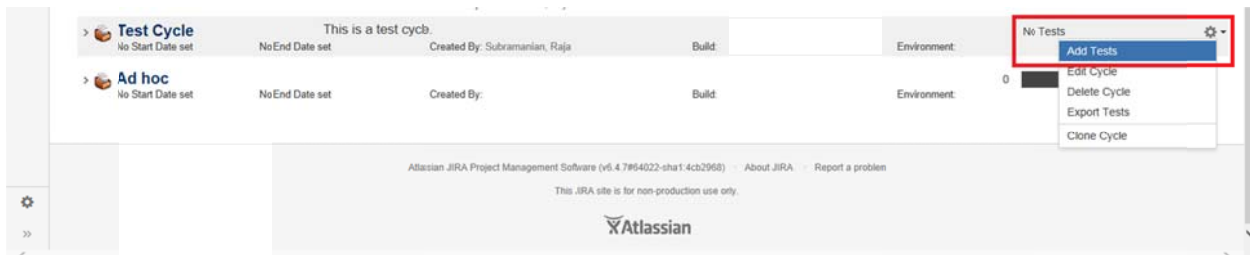
- In Test Cycles page choose “Select Versions” as “Unscheduled” from the drop down and click “Create New Cycle”.



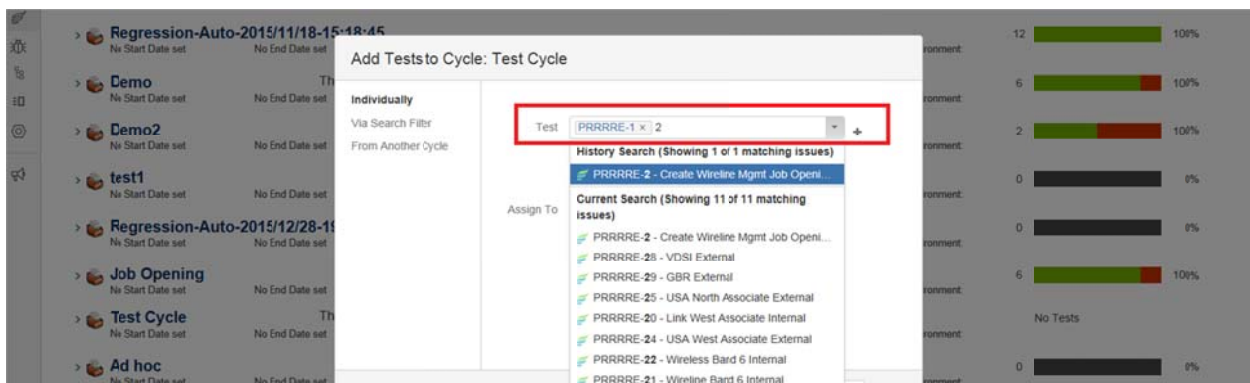
- Provide Name, description and click save.



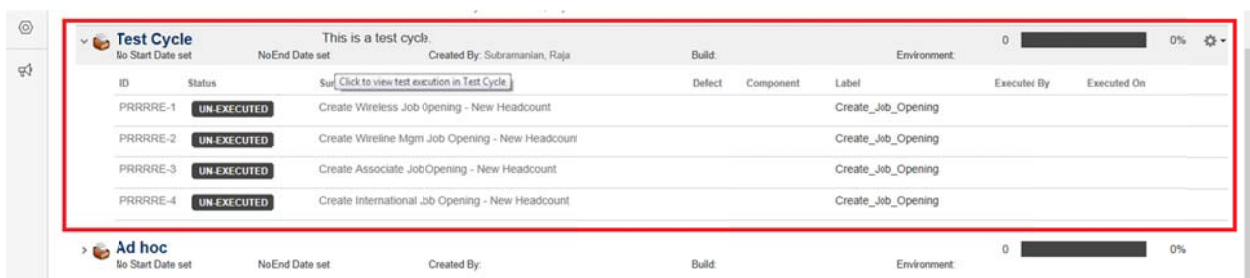
- After the test cycle is created click on settings button near the test case name and select “Add Tests”



- On the Add tests pop up type the test case id or description to add test case to the cycle.



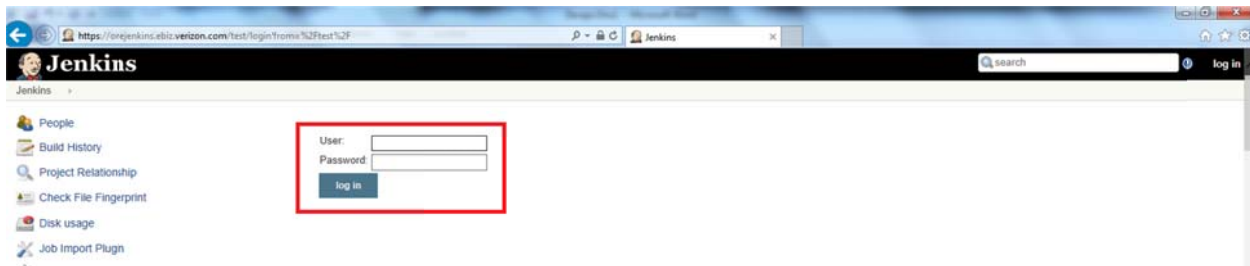
- After adding test cases to the test cycle it will look like below.



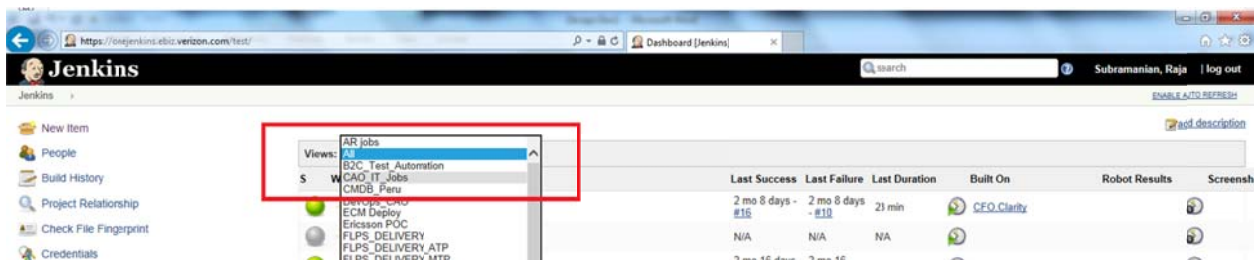
Step 2:

The next step is to trigger the Jenkins build Job.

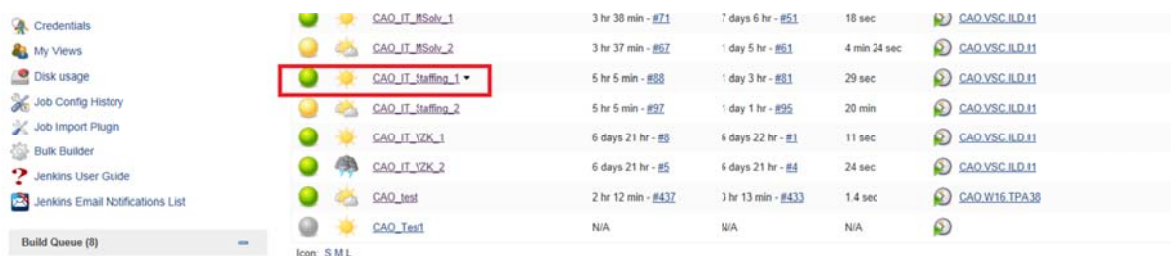
- Login to onejenkins-test url.



- Modify the “Views” dropdown from All to CAO_IT_Jobs.



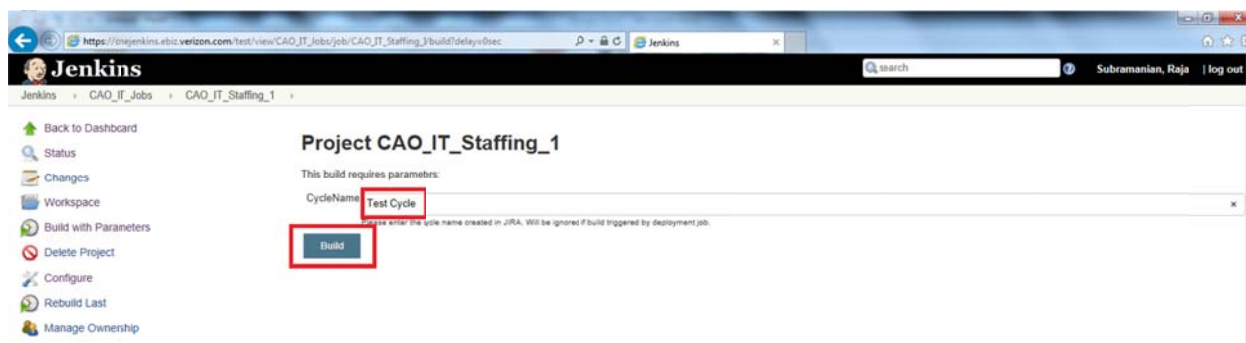
- Select the appropriate job related to the project. I have created test cycle is staffing so I am going to select staffing job [CAO_IT_Staffing_1](#).



- Click on “Build with Parameters” from the job’s dropdown.



- Provide the test cycle name created in step 1 inside the text box and click on build.



This completes the manual trigger of the CAO test automation framework. Please refer to the next section for the background process flow of this framework.

Note: As the selenium automation scripts are pulled from Stash repo, we need to make sure that the selenium hub url is provided inside the appropriate property file.

Ex: For staffing the hub url should be modified in
prrv_repo/VzStaffingApp/data/ObjectRepository.properties file.

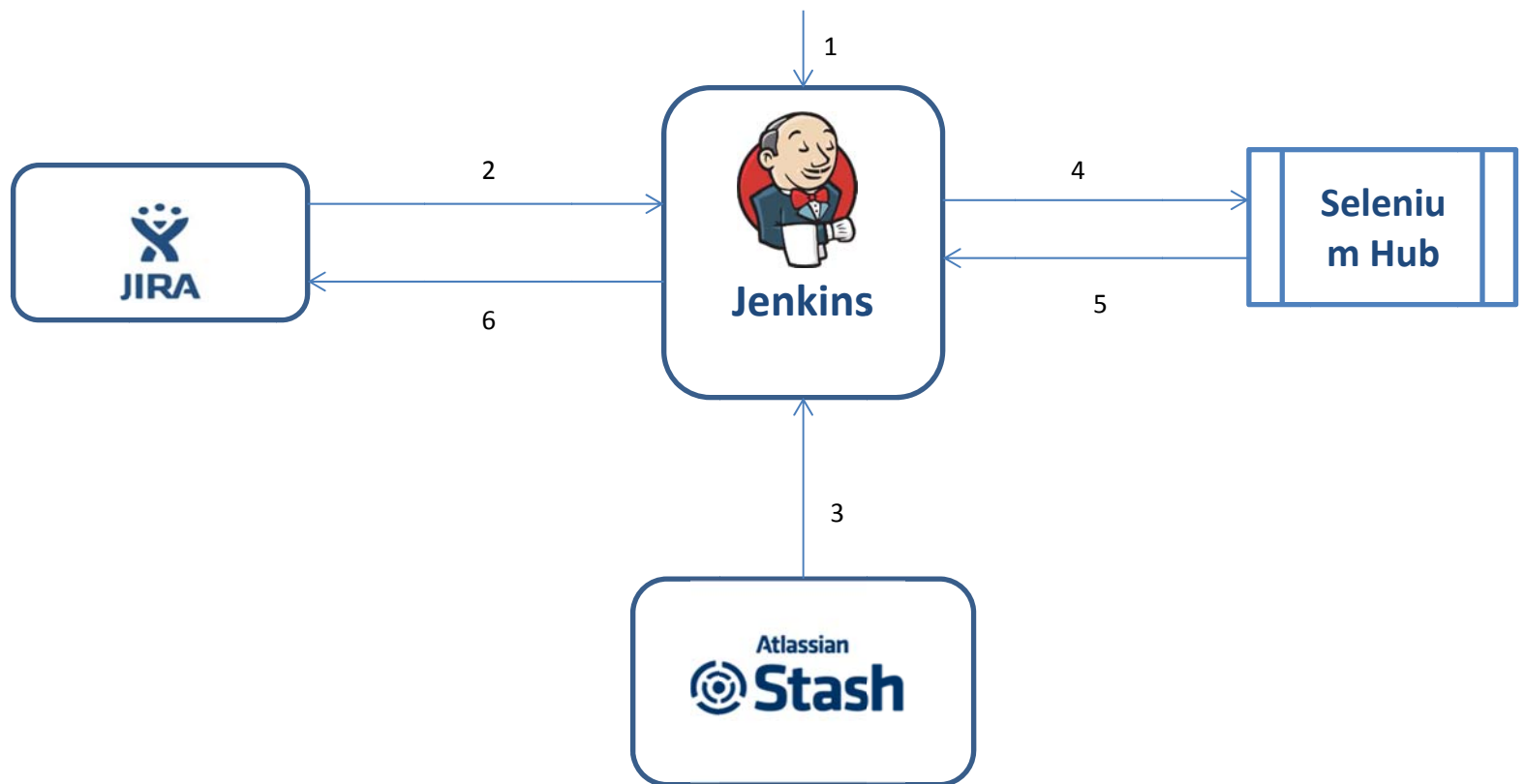
SELENIUM_HUB_URL_DEVOPS = http://114.9.152.53:4444/wd/hub

Similarly for Msolv the properties file is msolv-repo/src/config_file/dataFile.properties

SELENIUM_HUB_URL = http://114.9.183.98:4444/wd/hub

DevOps Framework Process Flow

This section explains the process flow of what happens after the Jenkins build is triggered.



As depicted above the process flow starts from Jenkins.

1. Jenkins job for the corresponding project is triggered which cycle name as parameter. This invokes a python script `xml_generator.py`.
Note : If no cycle ID is provided then the Python script will create a new test cycle for the corresponding project in JIRA.
2. This python script will communicate with JIRA through API calls and find's out the test cases added to the test cycle provided. Then a testng xml file is generated with the test cases available in the test cycle.

Note : If no cycle ID then the python script will add all the test case's in that project to newly created cycle id and generate the testng xml file.

3. After the generation of xml file the Jenkins build job will automatically trigger the next Build job. This job will first clone the Selenium Automation code for the specific project from Stash to Jenkins Workspace.
4. Using the testng xml file generated in the previous job the automation scripts will be executed on a Selenium Hub environment from Jenkins.
5. After the execution of the selenium scripts the result xml file is parsed by a python script `update_result.py`.
6. This script will update the execution status of all the test cases back to JIRA test cycle and will create issues in JIRA for the failed test cases.

Finally an e-mail will be triggered with consolidated report of the execution .

There are two Jenkins job for each project(Ex: CAO_IT_Staffing_1 and CAO_IT_Staffing_2). Job 1 is responsible for steps 1 & 2 in process flow. Job 2 is responsible for steps 3 to 6 in process flow.