IMPORTANT: This GitHub Enterprise instance has been upgraded to v2.12.7. Release notes.

📖 AMSPreSales-Demos / **Trainings**

DevOps Filling In Octane Analysis (Ex 2)

Edit   New Page

Dave Flynn edited this page 20 days ago · 15 revisions

Goal

Create a Job to Run Tests that Fail and Link to the Job in Ex 1

Steps you will need to perform to accomplish this task

1. Edit the "Corndog" job to add more tests for execution
2. Run the "Corndog" pipeline and check the pipeline and determine if the test failed
3. Propagate a LeanFT failure back to Jenkins and add Custom Build Report to see the LeanFT report quickly and easily
4. Edit the "Corndog" job to add "alwaysThrowsExceptionButCaughtPayPal" test for execution
5. Edit the "Corndog" job to add "alwaysThrowsExceptionPayPal" test for execution
6. Edit the "Corndog" job to add "UnstableTest" class for execution

1. Edit the "Corndog" job to add more tests for execution

- Edit the build step using the following

Pages 16

General ✏️

- Lynda.com
- Nimbus

AMS Boot Camp 2018

- Pre-work
  - Nimbus & Containers
  - DevOps
  - PPM & Enterprise Agile
  - StormRunner Functional
- Class Work
  - Nimbus and Containerization
  - PPM & Enterprise Agile
  - Borland and Serena Portfolios
  - DevOps

```
    clean test
    —Dtest=LoginTests,StableTests,PayPalTest#alwaysPassesPayPal+alwaysFailsPayPal
```

- To get a clearer understanding of the pattern used above to select individual test classes and methods for execution, refer to the Maven Surfire Plugin

2. Run the "Corndog" pipeline and check the pipeline and determine if the test failed

Clone this wiki locally

Why do we not see any failed tests?

Where can you go to see the LeanFT test results? (hint: look in the job's workspace under RunResults)

3. Propagate a LeanFT failure back to Jenkins and add Custom Build Report to see the LeanFT report quickly and easily

- Uncomment the line in the net.demo.paypal.PapPalTest file for the alwaysFailsPaypal test

```
@Test
public void alwaysFailsPayPal() throws GeneralLeanFtException {
    new LogMessages().logMethod(new Object(){});
    Verify.areEqual( expected: 9, actual: 10);

    //This will force the error as a Junit so it will propagate up to Octane
    //if (Reporter.getStatus() == Status.Failed){
    //   Assert.fail();
    //}
}
```

```
if (Reporter.getStatus() == Status.Failed){
    Assert.fail();
}
```

Commit and push the code.

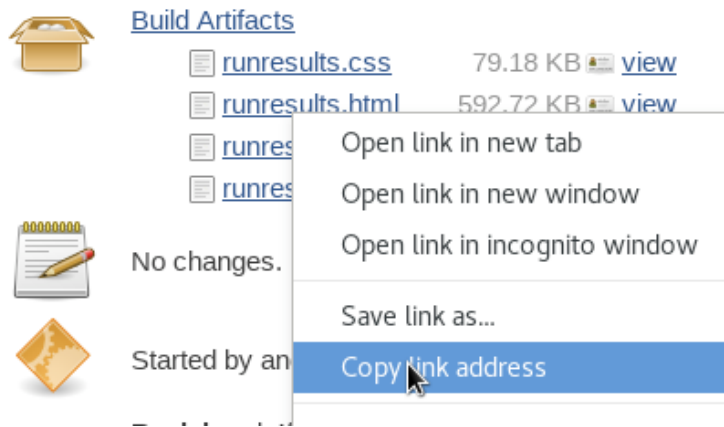- You must make sure to Archive the RunResults/runresults.html file so you can refer back to after multiple builds. So in the "Corndog" job, archive the folder (at least the runresults.html file). This would be a Post-build Action, and select "Archive the Artifacts". Order of the post build actions is not important here

Post-build Actions

Archive the artifacts
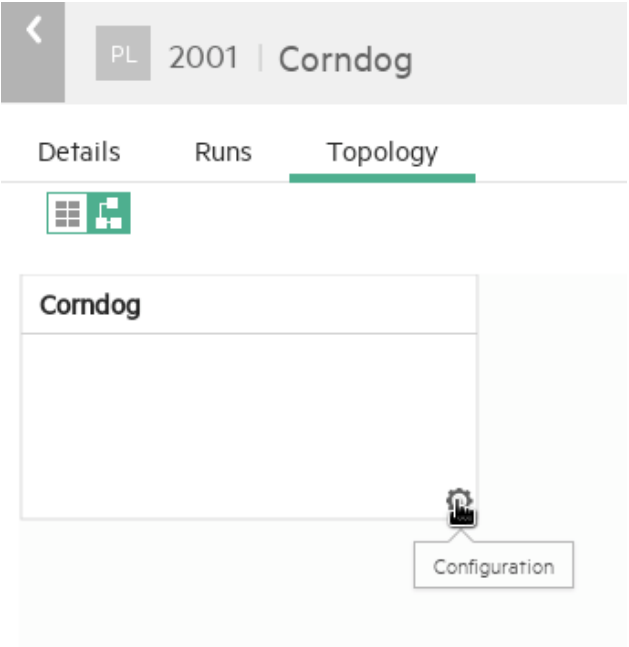
Files to archive   `**//RunResults/*`

- There are various ways to get the link to the runresults.html file, I find the easiest is to run the job and then go grab the link from the "Build Artifacts" similar to the image below.
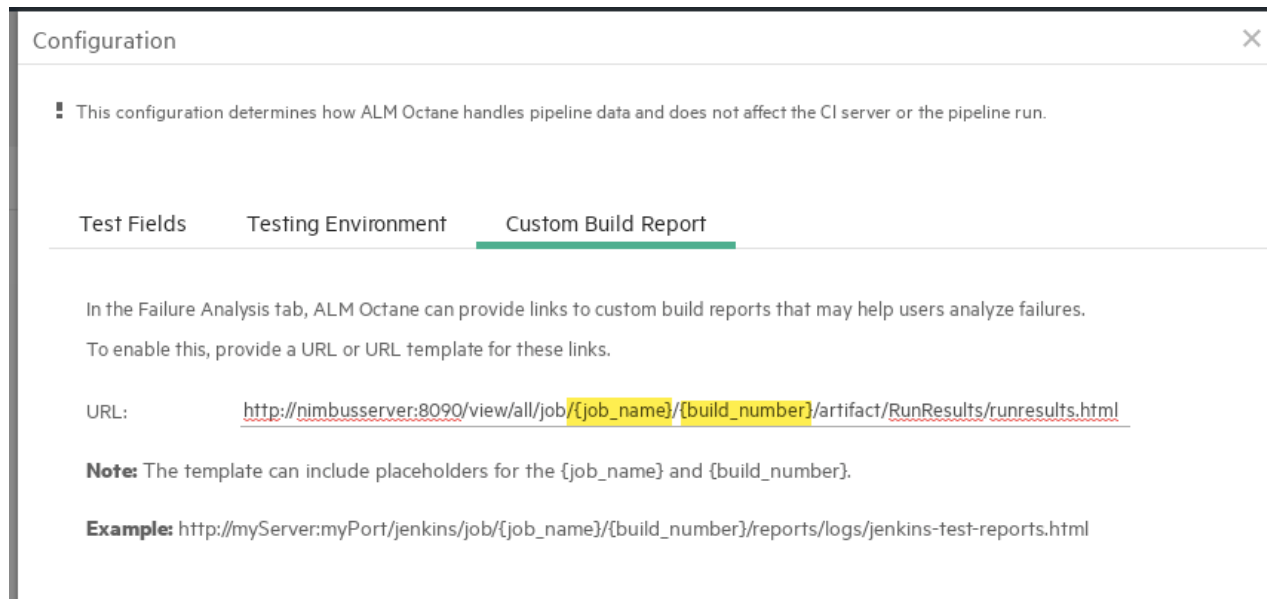

- In Octane, click on the pipeline ID, got to Topology, select the "Corndog" and then the Configure gear.

- Paste the URL you got from Jenkins for the report for the build. You will notice the two highlight arguments. You need to at least do the '{build_number}'. The custom report can be created to reference files and reports which are generated by assets other than LeanFT.
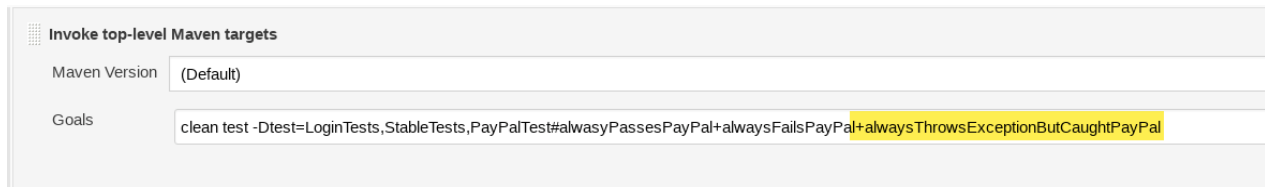
Configuration   ✕

❗ This configuration determines how ALM Octane handles pipeline data and does not affect the CI server or the pipeline run.

Test Fields   Testing Environment   **Custom Build Report**

In the Failure Analysis tab, ALM Octane can provide links to custom build reports that may help users analyze failures.
To enable this, provide a URL or URL template for these links.

URL: http://nimbusserver:8090/view/all/job/{job_name}/{build_number}/artifact/RunResults/runresults.html

Note: The template can include placeholders for the {job_name} and {build_number}.

Example: http://myServer:myPort/jenkins/job/{job_name}/{build_number}/reports/logs/jenkins-test-reports.html

- Now run the "Corndog" job and find the report.

4. Edit the "Corndog" job to add "alwaysThrowsExceptionButCaughtPayPal" test for execution

- Edit the Corndog Job and add the following so it runs the "alwaysThrowsExceptionButCaughtPayPal" test

```
clean test
–Dtest=LoginTests,StableTests,PayPalTest#alwaysPassesPayPal+alwaysFailsPayPal+al
waysThrowsExceptionButCaughtPayPal
```

Invoke top-level Maven targets

Maven Version	(Default)
Goals	clean test -Dtest=LoginTests,StableTests,PayPalTest#alwasyPassesPayPal+alwaysFailsPayPal+alwaysThrowsExceptionButCaughtPayPal

---

- **Do you see any failures?**
- **Why do you think not seeing a failure is correct?**
- **Take a look at the Octane pipeline Topology now. Do you know why it looks the way it does?**

5. Edit the Corndog Job and add the following so it runs all the tests in the "PayPalTest" class

- Edit the Corndog Job and add the following so it runs all the tests in the "PayPalTest" class

  ```
  clean test —Dtest=LoginTests,StableTests,PayPalTest
  ```

- Run the pipeline from Octane or from Jenkins

6. Edit the "Corndog" job to add "UnstableTest" class for execution

- Edit the Corndog Job and add the following so it runs all the tests in the "UnstableTest" class

  ```
  clean test —Dtest=LoginTests,StableTests,PayPalTest,UnstableTest
  ```

- Run the pipeline from Octane or the "Corndog" from Jenkins

Continue on to Exercise 3 👉

+ Add a custom footer