**POSTMAN integration with TeamCity:**

1. Pre-requisite:
2. TeamCity, Node JS and NPM should be installed.
3. Newman should be installed (**IF NOT installed** Running following command on Command prompt npm install -g newman).
4. Open Teamcity and create New Project and provide name.
5. Click on Build Configuration and provide name.
6. Click on Build Steps and click on Add Build Step and select Command Line.
7. In Command Script provide following to execute Newman run “Collection URL”
8. Press Save button.
9. Press Run to trigger build.
10. Also, Create a job and trigger build on schedule time daily.

**Reference**: <https://www.youtube.com/watch?v=QhncdC_Q58c>

**JMeter Setup on Local machine:**

**Steps to Install JMeter:**

1. Install Java (using <https://www.oracle.com/technetwork/java/javase/downloads/index.html>)
2. Download JMeter (using <http://jmeter.apache.org/download_jmeter.cgi>)
3. Installation of JMeter is simply unzip the zip/tar file into the directory where you want JMeter to be installed. There is no tedious installation screen to deal with! Simply unzip and you are done!
4. Launch JMeter on click jmeter.batch. File will be available on following location.

\apache-jmeter-5.1.1\apache-jmeter-5.1.1\bin\jmeter.bat

**Reference**: <https://www.guru99.com/guide-to-install-jmeter.html>

**JMeter integration with TeamCity:**

1. Pre-requisite:
2. Download the latest version of the plugin from <https://teamcity.jetbrains.com/repository/download/TeamCityPluginsByJetBrains_JMeterPlugin_Build/.lastSuccessful/jmeter.zip>
3. Open TeamCity and navigate to *Administration > Plugins List* and click **Upload plugin zip.**
4. Find Jmeter.zip just downloaded earlier and save and enable it.
5. Open Teamcity and create New Project and provide name.
6. Click on Build Configuration and provide name.
7. Click on Build Steps and click on Add Build Step
8. In **Runner Type**, select *Command Line*.
9. In **Run**, select *Executable with Parameters*.
10. In **Command executable**, give path to the JMeter startup script, e.g. **C:\jmeter\bin\jmeter.bat**
11. In **Command parameters**, give standard JMeter arguments like you run the test from the command line, e.g. **-n -t c:\jmeter\test.jmx -l test.jtl**
12. Press Save button.
13. Press Run to trigger build.
14. Also, Create a job and trigger build on schedule time daily.

**References**:

* <https://www.blazemeter.com/blog/how-run-jmeter-tests-teamcity-continuous-integration/>
* <https://www.jarchitect.com/Doc_CI_TeamCity>

**Performance Testing Scenarios (Normal Load):**

1. One user (CIS login) making 100 requests to about 30 – 40 endpoints

* Virtual Users = 10
* Ramp Up Period = 100 Sec
* Iteration = 1

**Note:** The ramp-up period tells JMeter how long to take to "ramp-up" to the full number of threads chosen. If 10 threads are used, and the ramp-up period is 100 seconds, then JMeter will take 100 seconds to get all 10 threads up and running. Each thread will start 10 (100/10) seconds after the previous thread was begun.

1. One user (CIS login) making 100 requests to about 30 – 40 endpoints

* Virtual Users = 100
* Ramp Up Period =150 Sec
* Iteration = 1

1. One user (CIS login) making 100 requests to about 30 – 40 endpoints

* Virtual Users = 100
* Ramp Up Period = 200 Sec
* Iteration = 1

1. One user (CIS login) making 200 requests to about 30 – 40 endpoints

* Virtual Users = 100
* Ramp Up Period = 1 Sec
* Iteration = 1

1. One user (CIS login) making 200 requests to about 30 – 40 endpoints

* Virtual Users = 70
* Ramp Up Period = 5 Sec
* Iteration = 1

1. One user (CIS login) making 200 requests to about 30 – 40 endpoints

* Virtual Users = 200
* Ramp Up Period = 50 Sec
* Iteration = 1

**Performance Testing Scenarios (Realistic Performance test- pacing):**

1. One user (CIS login) making 100 requests to about 30 – 40 endpoints

* Virtual Users = 100
* Think time up to = 10 Sec
* Starts 10 thread
* Add more 10 threads on every 10 seconds
* Keep load till 5 minutes
* Then, Stops 10 thread on every 10 Seconds.

1. One user (CIS login) making 200 requests to about 30 – 40 endpoints

* Virtual Users = 200
* Think time up to = 10 Sec
* Starts 5 thread
* Add more 10 threads on every 10 seconds
* Keep load till 10 minutes
* Then, Stops 10 thread on every 10 Seconds.

**Performance Testing Scenarios (Stress Testing/Concurrency Testing):**

1. One user (CIS login) making 300 requests to about 30 – 40 endpoints

* Virtual Users = 300
* Ramp Up Period = 100 Sec
* Iteration = Forever

1. One user (CIS login) making More than 300 requests to about 30 – 40 endpoints

* Virtual Users = 400 to extend (To Find out reliability point of API’s)
* Ramp Up Period = 100 Sec
* Iteration = Forever

**Note**: On Every Performance Test: CPU Utilization and Memory consumption of Server will also be captured.