

Performance testing process:



1) Identify performance scenarios:

Firstly, we will identify the performance scenarios based on these below factors:

Most commonly scenarios: It means that we can find the performance scenarios based on the scenarios, which commonly used like in the **Gmail application**; we will perform **login, inbox, send items, and compose a mail and logout**.

Most critical scenarios: Critical scenarios mean regularly used and important for the business-like in Gmail application **login, compose, inbox, and logout**.

Huge data transaction: If we have huge data means that n-number of the users using the application at the same time.

Once we identify the performance scenarios, we will move to the next step.

2) Plan and design performance test script

In this step, we will install the tools in the Test Engineer Machine and access the test server and then we write some script according to the test scenarios and run the tool.

Once we are done with writing the script, we will go to the next step.

3) Configure the test environment & distribute the load

After writing the test scripts, we will arrange the testing environment before the execution. And also, manage the tools, other resources and distribute the load according to the "Usage Pattern" or mention the duration and stability.

4) Execute test scripts

Once we are done with distributing the load, we will execute, validate, and monitor the test scripts.

5) Result

After executing the test scripts, we will get the test result. And check that the result meeting the goal in the given response time or not, and the response time could be maximum, average, and minimum.

If the response is not meeting the required time response, then we will go for the negative flow where will perform the below steps:

6) Analysis result

First, we will analyze the test result whether it meets with the response time or not.

7) Identify the Bottleneck

After that, we will identify the bottleneck (bug or performance issue). And the bottleneck could occur because of these aspects like the problem in code, hardware issue (hard disk, RAM Processor), network issues, and the software issue (operating system). And after finding the bottleneck, we will perform tuning (fix or adjustment) to resolve this bottleneck.

8) Re-run test

Once we fix the bottlenecks, re-run the test scripts and checks the result whether it meets the required goal or not.

