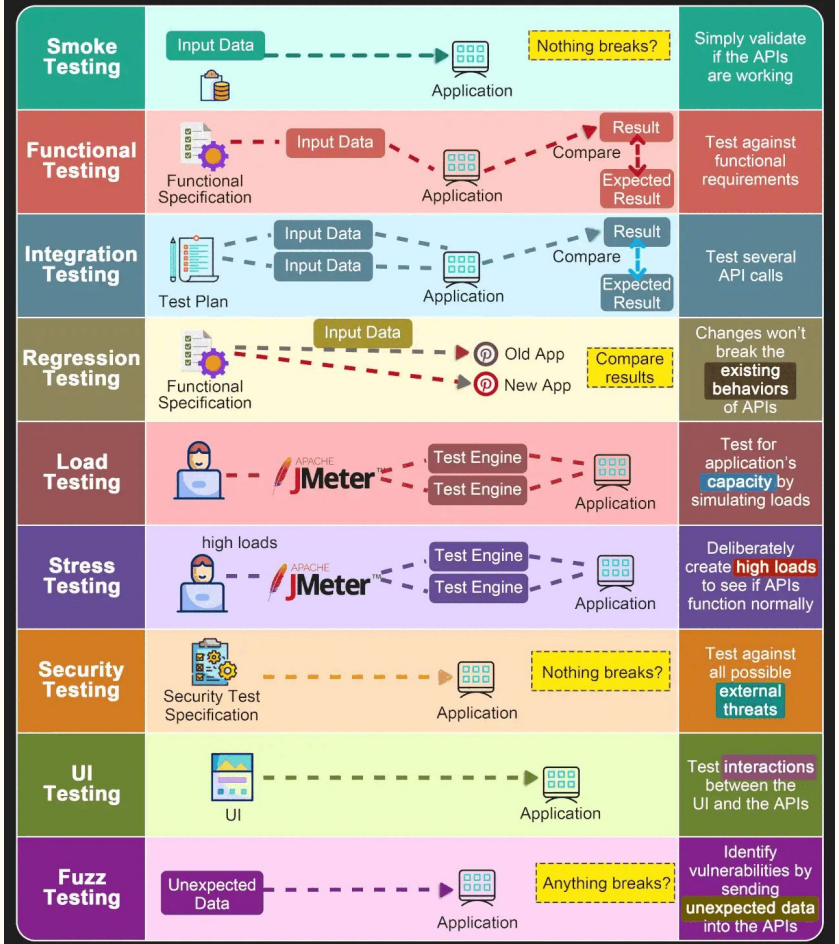
Introduction to Performance Testing

Understanding the Importance and Fundamentals

9 Types API Testing

blog.bytebytego.com



Introduction to Performance Testing

- Definition of load testing
- Importance in software engineering
- Overview of topics to be covered

Definition

Performance testing is crucial for ensuring the reliability, scalability, and performance of software applications. It helps identify performance bottlenecks, assess system capacity, and optimize resource allocation.

Ensures application stability

Load testing helps uncover potential issues such as slow response times, crashes, and degraded performance under heavy load. By identifying and addressing these issues early in the development process, we can ensure the stability of the application in production

Identifies performance bottlenecks:

Load testing allows us to pinpoint areas of the application that are prone to performance degradation under load. This could be due to inefficient algorithms, database queries, or network latency. By identifying and addressing these bottles

Enhances user experience

A well-performing application leads to a better user experience. By conducting load testing, we can ensure that the application can handle the expected number of users and provide a smooth and responsive user experience even during peak usage periods

Metric Of Performance Testing

Average Response Time

Measures the average time taken to respond to user requests, indicating application speed.

Error Rate

Percentage of errors in requests, reflecting application reliability. Higher rates suggest efficiency issues.

Throughput

Indicates bandwidth used during load tests, measured in kilobytes per second, showing data flow between user and server.

Requests Per Second

Counts the number of requests to the server per second, including images, documents, and web pages.

Concurrent Users

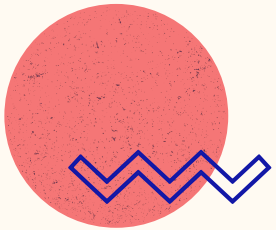
Tracks the number of users actively on the application at any given time, highlighting peak usage periods.

Peak Response Time

Tracks the number of users actively on the application at any given time, highlighting peak usage periods.

Performance Testing Techniques

Load Testing

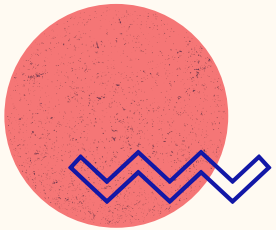


assess how the system performs under a typical load for your system or application. Typical load might be a regular day in production or an average timeframe in your daily traffic. This test also might be called a day-in-life test or volume test

When to run?

- Assess the performance of your system under a typical load.
- Identify early degradation signs during the ramp-up or full load periods.
- Assure that the system still meets the performance standards after system changes (code and infrastructure).

VUs/Throughput : Average Production or Expected Load
Duration : Mid (5-60 minutes)



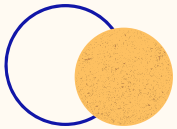
Stress Testing

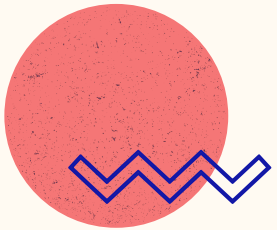
Stress tests help you discover how the system functions with the load at peak traffic. Stress testing might also be called rush-hour testing, surge testing, breakpoint testing or scale testing

VUs/Throughput : Increases until break
Duration : As long as necessary

When to run?

Stress tests verify the stability and reliability of the system under conditions of heavy use. Systems may receive higher than usual workloads on unusual moments such as process deadlines, paydays, rush hours, ends of the workweek, and many other behaviors that might cause frequent higher-than-average traffic.





Spike Testing

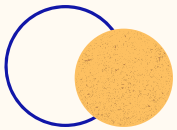
A spike test verifies whether the system survives and performs under sudden and massive rushes of utilization.

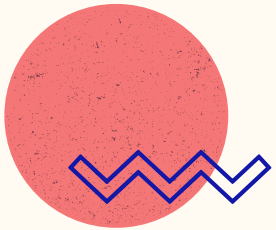
VUs/Throughput : Very High
Duration : Short (a few minutes)

When to run?

This test must be executed when the system expects to receive a sudden rush of activity.

When the system expects this type of behavior, the spike test helps identify how the system will behave and if it will survive the sudden rush of load. The load is considerably above the average and might focus on a different set of processes than the other test types.

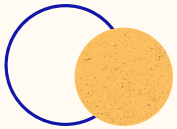




Endurance Test

Endurance/soak test are a variation of the average-load test. The main difference is the test duration. In a soak test, the peak load is usually an average load, but the peak load duration extends several hours or even days

VUs/Throughput : Average
Duration : Long



When to run?

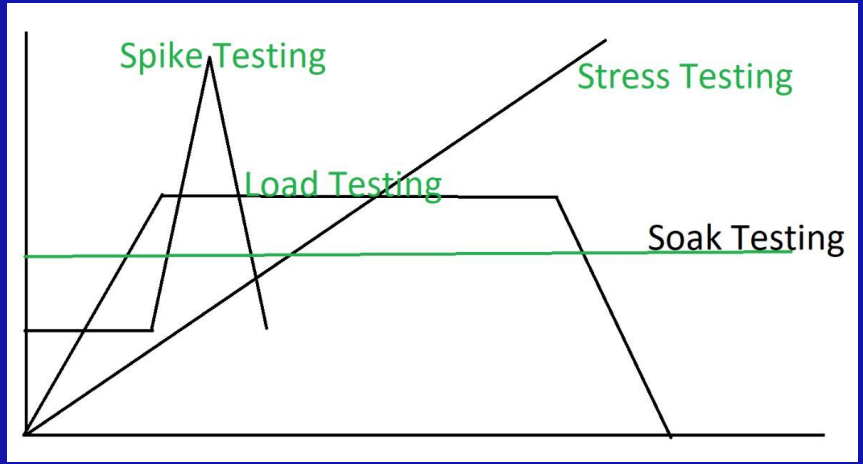
Most systems must stay turned on and keep working for days, weeks, and months without intervention. This test verifies the system stability and reliability over extended periods of use.

This test type checks for common performance defects that show only after extended use. Those problems include response time degradation, memory or other resource leaks, data saturation, and storage depletion.

Demonstration K6

- Install K6 https://k6.io/docs/get-started/installation/
- Read the Documentation
- Create sample test for load, stress, spike and soak testing

Performance Testing Comparison Rresult



Common Load Testing Tools

- Jmeter https://jmeter.apache.org/
- Gatling https://gatling.io/
- K6 https://k6.io/
- Artillery https://www.artillery.io/
- Vegeta https://github.com/tsenart/vegeta
- Locus https://locust.io/
- etc...

Demonstration Vegeta

- Install vegeta https://github.com/tsenart/vegeta?tab=readme-ov-file#install
- Run some sample cli

  Ex :

```
echo "GET http://httpbin.org/get" | vegeta attack -duration=5s -rate=5 | vegeta report --type=text
cd && echo "GET http://httpbin.org/get" | vegeta attack -duration=30s -rate=10 -output=results-veg-httpbin-get.bin && cat
results-veg-httpbin-get.bin | vegeta plot --title="HTTP Bin GET 10 rps for 30 seconds" > httpbin-get-10rps-30seconds.html
```

Demonstration Artillery

- Install Artillery https://www.artillery.io/docs/get-started/get-artillery
- Create sample YML config
- Run the yml config

Sources

- https://blog.bytebytego.com/
- https://jmeter.apache.org/
- https://www.artillery.io/
- https://testguild.com/load-testing-tools/
- https://google.com/
- https://chat.openai.com/
- https://dev.to/
- https://medium.com/
- https://reddit.com/
- https://giphy.com/
- All internet stuff...

Thanks!