**Test Case: Load Testing a Web Application**

**Objective: Simulate multiple users accessing a web application concurrently.**

Test Steps: Set up a Thread Group to define the number of virtual users, ramp-up period, and loop count.

Add an HTTP Request sampler to simulate the web application's requests (e.g., login, product search).

Configure the necessary parameters (URL, method, headers, body) for each request.

Add listeners (e.g., Aggregate Report, Summary Report) to monitor response times, error rates, and other metrics.

Run the test and analyze the results to identify any performance bottlenecks or issues.

**Test Case: API Performance Testing**

**Objective: Measure the response time and throughput of an API under various loads.**

Test Steps: Create a Thread Group with the desired number of concurrent users.

Add an HTTP Request sampler for the API endpoint you want to test.

Configure the necessary parameters (URL, method, headers, body) for the API request.

Add timers (e.g., Constant Throughput Timer) to control the desired load on the API.

Include listeners (e.g., Aggregate Report, Response Times Over Time) to capture performance metrics.

Run the test and analyze the results to identify any performance issues, such as high response times or errors.

**Test Case: Distributed Load Testing**

**Objective: Distribute the load across multiple JMeter instances for increased testing capacity.**

Test Steps: Set up a master-slave configuration in JMeter, where the master coordinates the test and the slaves execute the load.

Configure the Thread Group, HTTP Request samplers, and other elements as required.

Start the JMeter master and connect the slave instances.

Specify the number of users and other test parameters in the master instance.

Run the test, and JMeter will distribute the load across the slave instances.

Monitor the test execution and analyze the aggregated results from all the instances.