**Load Test Documentation for 2 Users with 4 Scenarios**

This document outlines the load testing scenarios for two users. Each user has two load test scenarios, which results in a total of four distinct load tests. These tests are created using **k6**, a powerful open-source load testing tool. The primary goal is to simulate realistic user behavior and evaluate the system's performance under different conditions.

**Test Overview**

* **Load Testing Tool**: k6
* **Total Load Test Scenarios**: 4 (2 scenarios per person)
* **Test URL**: https://jsonplaceholder.typicode.com/posts (This is an authentic, publicly available API used for testing)
* **Primary Objective**: To simulate various load patterns and measure the system's ability to handle different numbers of virtual users (VUs) and varying traffic loads.

**Person 1: Load Test Scenarios**

**Scenario 1: Steady Load for 1 Minute**

This scenario simulates a steady load where 10 virtual users (VUs) continuously perform HTTP GET requests for 1 minute to fetch posts from the provided API.

* **Test File**: person1-load-test1.js
* **Virtual Users (VUs)**: 10
* **Test Duration**: 1 minute
* **Request Type**: HTTP GET request to https://jsonplaceholder.typicode.com/posts

**Script**:

// person1-load-test1.js

import http from 'k6/http';

export let options = {

vus: 10, // 10 virtual users

duration: '1m', // Test duration is 1 minute

};

export default function () {

// Simulate a GET request to fetch posts from the API

http.get('https://jsonplaceholder.typicode.com/posts');

};

* **Objective**: To observe how the system performs under a constant, steady load of 10 virtual users.

**Scenario 2: Ramp-Up of Virtual Users (10 to 30)**

This scenario simulates a gradual increase in virtual users from 10 to 30 over 90 seconds. This helps test how the system handles increasing traffic over time.

* **Test File**: person1-load-test2.js
* **Virtual Users (VUs)**: Ramp up from 10 to 30 VUs
* **Test Duration**: 90 seconds (with gradual ramp-up of users)
* **Request Type**: HTTP GET request to https://jsonplaceholder.typicode.com/posts

**Script**:

// person1-load-test2.js

import http from 'k6/http';

export let options = {

scenarios: {

ramp\_up: {

executor: 'ramping-vus',

stages: [

{ duration: '30s', target: 10 }, // Ramp up to 10 users in 30 seconds

{ duration: '30s', target: 30 }, // Ramp up to 30 users in 30 seconds

{ duration: '30s', target: 0 }, // Ramp down to 0 users in 30 seconds

],

},

},

};

export default function () {

// Simulate a GET request to fetch posts from the API

http.get('https://jsonplaceholder.typicode.com/posts');

};

* **Objective**: To evaluate how the system adapts to increasing user load and whether it can handle the gradual surge of traffic.

**Person 2: Load Test Scenarios**

**Scenario 1: Steady Load for 2 Minutes**

This scenario involves simulating a steady load of 20 virtual users over a 2-minute period to assess the system's performance under continuous traffic.

* **Test File**: person2-load-test1.js
* **Virtual Users (VUs)**: 20
* **Test Duration**: 2 minutes
* **Request Type**: HTTP GET request to https://jsonplaceholder.typicode.com/posts

**Script**:

// person2-load-test1.js

import http from 'k6/http';

export let options = {

vus: 20, // 20 virtual users

duration: '2m', // Test duration is 2 minutes

};

export default function () {

// Simulate a GET request to fetch posts from the API

http.get('https://jsonplaceholder.typicode.com/posts');

};

* **Objective**: To observe the system's performance under a steady and consistent load of 20 virtual users for an extended period of 2 minutes.

**Scenario 2: Burst Load (50 to 100 VUs)**

In this scenario, the virtual users rapidly increase from 50 to 100 over 20 seconds, simulating a burst of traffic to evaluate how the system handles sudden spikes.

* **Test File**: person2-load-test2.js
* **Virtual Users (VUs)**: Ramp up from 50 to 100 VUs
* **Test Duration**: 20 seconds (with rapid ramp-up of users)
* **Request Type**: HTTP GET request to https://jsonplaceholder.typicode.com/posts

**Script**:

// person2-load-test2.js

import http from 'k6/http';

export let options = {

scenarios: {

burst\_load: {

executor: 'ramping-vus',

stages: [

{ duration: '10s', target: 50 }, // Ramp up to 50 users in 10 seconds

{ duration: '10s', target: 100 }, // Ramp up to 100 users in 10 seconds

{ duration: '10s', target: 0 }, // Ramp down to 0 users in 10 seconds

],

},

},

};

export default function () {

// Simulate a GET request to fetch posts from the API

http.get('https://jsonplaceholder.typicode.com/posts');

};

* **Objective**: To test how the system reacts to a sharp increase in traffic and its ability to handle burst loads.