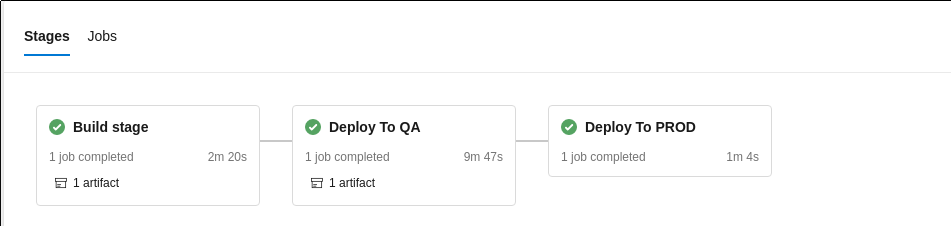
Blood Pressure Application Pipeline

Anthony Staunton

X0004139

Design:



There are 3 sections to my pipeline that will execute whenever new code is added to the master branch in the code repository.

To add code to the repository a user must create a new feature branch, that will only exist for a short period of time (long enough to trach the development of a bug or specific feature). A developer then creates a pull request and requests that the branch be merged to the master branch.

1. Build Stage

* The code is checked out of the repository in to a Linux Virtual machine.
* Java 11 JDK is installed
* Using Gradle the application is built and testing is run.
  + The code is compiled, unit tests using Junit are then run
  + Some integration tests are run using Spring MockMVC
  + Finally some BDD testing is done using Spock Framework
  + The code is analysed with Jacoco Gradle Plugin, which is used in conjunction with SonarCloud to check code quality and coverage
  + Whitesource Bolt is run to verify the licences will not cause an issue.

The application Jar and the tests results are then uploaded if tests pass and code coverage is high enough (determined by SonarClouds’ inbuilt metrics). If a failure occurs then the pipeline will not proceed.

2. QA

* End to End tests are run using Geb, which integrates with Gradle and Spock to provide Selenium testing using Chrome as a headless browser. Running the test suite on a normal computer will also run Chrome and Firefox in full mode.
* k6 Load tests are run at this stage
* OWASP ZAP tests suite is run at this stage

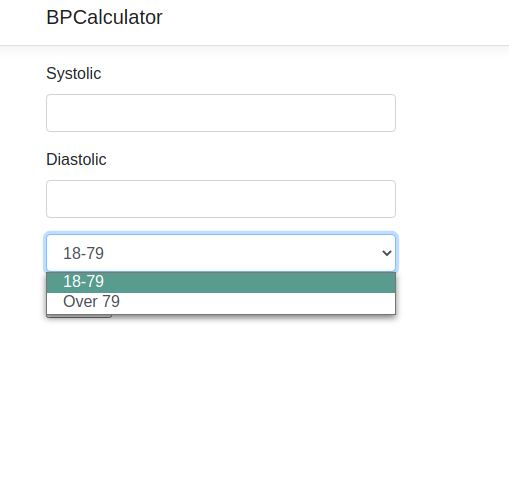
Due to issues with Azure Web services, where deploying Jar or War appears to have been depreciated, in this stage Spring boot and all the accompanying tests are run from a single Bash script task. This was required to ensure the application ran in the VM to allow the tests to complete. We also do not want to stop the build if a failure occurs in testing at this stage, as it would be more beneficial to identify breaking changes in the UI, security issues and / or load issues within the code instead of stopping the pipeline.

3. Production

This would deploy the application to production after some verification was done manually by a tester. As discussed previously I encountered several issues with deploying Spring Boot applications as Jar or War to Azure, therefore this cannot be seen to be running (although the deployment tasks states that the application was deployed successfully when trying to access it an error is seen in the web browser.

New Feature:

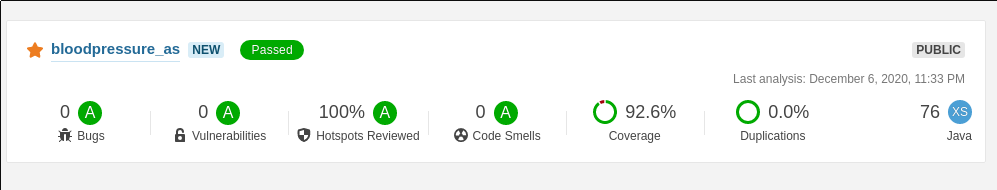
Recent studies have found that adults over the age of 80 have higher blood pressure, as a normal, than those that are younger. The guidelines for measuring blood pressure were updated to state that for those >79 the High Blood Pressure cut-off is 150 / 90, when compared to 140 / 90 for younger adults. This was captured with a new drop-down in the UI and corresponding tests added.

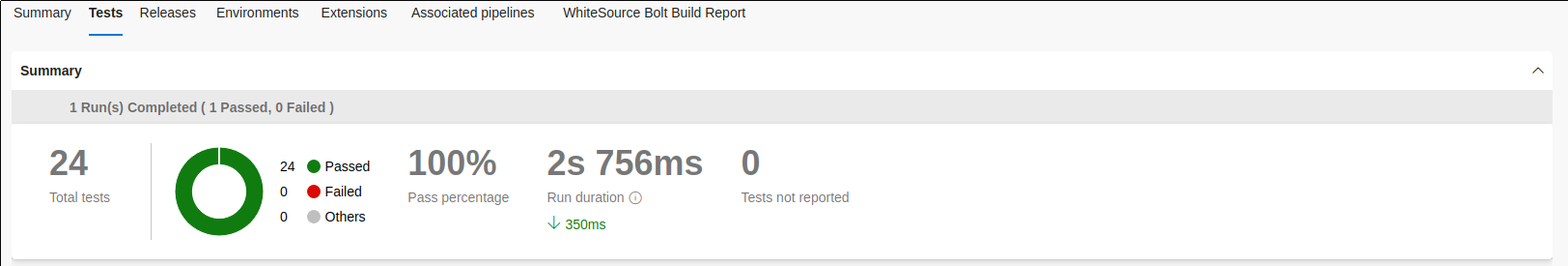


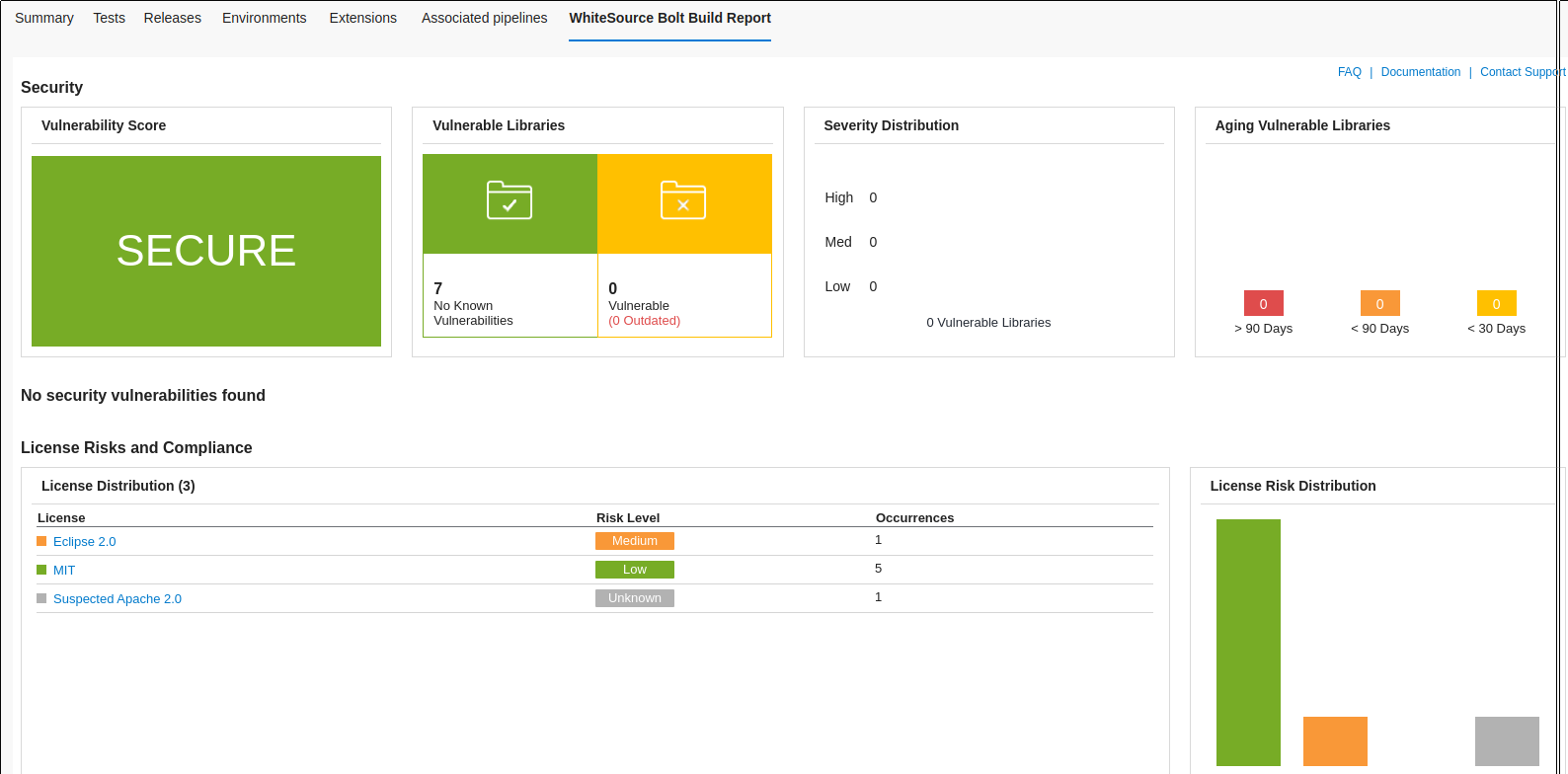
Appendix

**Screenshots**

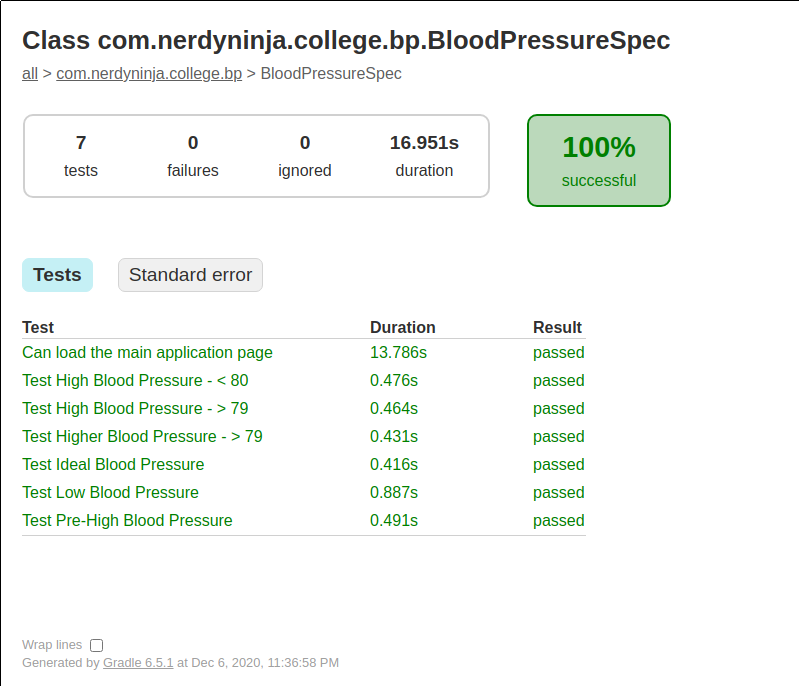
**Build Stage**







E2E



**Pipeline code:**

trigger:

- master

variables:

# Azure Resource Manager connection created during pipeline creation

azureSubscription: 'Azure for Students(a69cdbd2-7c95-4d73-ad84-9cc6e47951a5)'

# Web app name

webAppName: 'bpas'

# Environment name

environmentName: 'bpas'

# Agent VM image name

vmImageName: 'ubuntu-latest'

stages:

- stage: Build

displayName: Build stage

jobs:

- job: GradlePackageAndPublishArtifacts

displayName: Gradle Package and Publish Artifacts

pool:

vmImage: $(vmImageName)

steps:

- task: JavaToolInstaller@0

inputs:

versionSpec: '11'

jdkArchitectureOption: 'x64'

jdkSourceOption: 'PreInstalled'

- task: SonarCloudPrepare@1

inputs:

SonarCloud: 'bp\_app\_sonar'

organization: 'x00004139'

scannerMode: 'Other'

extraProperties: |

sonar.projectKey=X00004139\_bloodpressure\_as

sonar.projectName=bloodpressure\_as

- task: Gradle@2

inputs:

gradleWrapperFile: 'gradlew'

tasks: 'build'

publishJUnitResults: true

testResultsFiles: 'build/reports/\*\*/\*.xml'

javaHomeOption: 'JDKVersion'

jdkVersionOption: '1.11'

gradleOptions: '-Xmx3072m'

sonarQubeRunAnalysis: true

sqGradlePluginVersionChoice: 'build'

- task: SonarCloudPublish@1

inputs:

pollingTimeoutSec: '300'

- task: WhiteSource Bolt@20

- task: CopyFiles@2

inputs:

SourceFolder: 'build/libs'

Contents: '\*\*.?(war|jar)'

TargetFolder: '$(build.artifactstagingdirectory)'

OverWrite: true

- task: PublishBuildArtifacts@1

inputs:

PathtoPublish: '$(Build.ArtifactStagingDirectory)'

ArtifactName: 'drop'

- stage: QA

displayName: Deploy To QA

dependsOn: Build

condition: succeeded()

jobs:

- deployment: DeployLinuxWebApp

displayName: Deploy Linux Web App

environment: $(environmentName)

pool:

vmImage: $(vmImageName)

strategy:

runOnce:

deploy:

steps:

- task: JavaToolInstaller@0

displayName: Install Java 11

inputs:

versionSpec: '11'

jdkArchitectureOption: 'x64'

jdkSourceOption: 'PreInstalled'

- checkout: self

clean: true

- script: |

sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv-keys 379CE192D401AB61

echo "deb https://dl.bintray.com/loadimpact/deb stable main" | sudo tee -a /etc/apt/sources.list

sudo apt-get update

sudo apt-get install k6

displayName: Install k6

- script: docker pull owasp/zap2docker-weekly

displayName: pull docker image for owasp zap

- task: AzureWebApp@1

inputs:

azureSubscription: '$(azureSubscription)'

appType: 'webAppLinux'

appName: '$(webAppName)'

deployToSlotOrASE: true

resourceGroupName: 'bpas-rg'

slotName: 'production'

package: '$(Pipeline.Workspace)/drop/\*.?(war|jar)'

runtimeStack: 'JAVA|11-java11'

startUpCommand: 'java -jar $(Pipeline.Workspace)/drop/\*.jar'

appSettings: '-Port 8080'

- task: Bash@3

displayName: Run E2E Tests, Load Tests and ZAP test suite

inputs:

targetType: 'inline'

script: |

mkdir -p $(Pipeline.Workspace)/reports/e2e

java -jar $(Pipeline.Workspace)/drop/\*.jar &

sleep 10

pushd $(Pipeline.Workspace)/s/e2e

./gradlew clean chromeHeadlessTest

sleep 10

cp -r $(Pipeline.Workspace)/s/e2e/build/reports/tests/\* $(Pipeline.Workspace)/reports/e2e/

popd

sleep 5

pwd

sleep 5

mkdir -p $(Pipeline.Workspace)/reports/k6tests

k6 run k6tests/perf.js | tee k6tests/k6report.txt

sleep 5

cp -r $(Pipeline.Workspace)/s/k6tests/\* $(Pipeline.Workspace)/reports/k6tests

# ZAP

mkdir -p $(Pipeline.Workspace)/reports/zap

docker run -v $(Pipeline.Workspace)/reports/zap:/zap/wrk/:rw --user root -t owasp/zap2docker-weekly zap-baseline.py -t http://$(ip addr show dev eth0 | grep -i inet | grep -iv inet6 | awk '{print $2}' | awk -F'/' '{print $1}'):8080 -I -r testreport.html

sleep 5

# Kill the java process

kill -9 $(ps -ef | grep -i blood | grep -iv grep | awk -F' ' '{print $2}')

wait

- task: ArchiveFiles@2

inputs:

rootFolderOrFile: '$(Pipeline.Workspace)/reports'

includeRootFolder: false

archiveType: 'zip'

archiveFile: '$(Pipeline.Workspace)/reports.zip'

replaceExistingArchive: true

- task: PublishPipelineArtifact@1

inputs:

targetPath: '$(Pipeline.Workspace)/reports'

artifact: 'reportsArchive'

publishLocation: 'pipeline'

- stage: Prod

displayName: Deploy To PROD

dependsOn: QA

condition: succeeded()

jobs:

- deployment: DeployLinuxWebApp

displayName: Deploy Linux Web App

environment: $(environmentName)

pool:

vmImage: $(vmImageName)

strategy:

runOnce:

deploy:

steps:

- task: JavaToolInstaller@0

displayName: Install Java 11

inputs:

versionSpec: '11'

jdkArchitectureOption: 'x64'

jdkSourceOption: 'PreInstalled'

- task: AzureWebApp@1

inputs:

azureSubscription: '$(azureSubscription)'

appType: 'webAppLinux'

appName: '$(webAppName)'

deployToSlotOrASE: true

resourceGroupName: 'bpas-rg'

slotName: 'production'

package: '$(Pipeline.Workspace)/drop/\*.?(war|jar)'

runtimeStack: 'JAVA|11-java11'

startUpCommand: 'java -jar $(Pipeline.Workspace)/drop/\*.jar'

appSettings: '-Port 8080'

- task: Bash@3

inputs:

targetType: 'inline'

script: |

#Add in a small wait to ensure the application is running smooth

sleep 10

**Tests:**

Junit

package com.nerdyninja.college.bp;  
  
import org.junit.jupiter.api.BeforeEach;  
import org.junit.jupiter.api.Test;  
import static org.junit.jupiter.api.Assertions.*assertEquals*;  
  
import javax.validation.ConstraintViolation;  
import javax.validation.Validation;  
import javax.validation.Validator;  
import javax.validation.ValidatorFactory;  
import java.util.Set;  
  
  
  
class BloodPressureDataTest {  
 private static BloodPressureData *bloodPressureData*;  
 private static Validator *validator*;  
  
 @BeforeEach  
 public void setData(){  
 ValidatorFactory factory = Validation.*buildDefaultValidatorFactory*();  
 *validator* = factory.getValidator();  
 *bloodPressureData* = new BloodPressureData();  
 }  
  
  
 @Test  
 void testSetSystolic40() {  
 *bloodPressureData*.setSystolic(40L);  
 *bloodPressureData*.setDiastolic(50L);  
 *bloodPressureData*.setAgerange(1);  
 Set<ConstraintViolation<BloodPressureData>> constraintViolations = *validator*.validate( *bloodPressureData* );  
  
 *assertEquals*(1, constraintViolations.size() );  
 *assertEquals*("Systolic must be greater than or equal to 70mmHG", constraintViolations.iterator().next().getMessage());  
 }  
  
 @Test  
 void testSetSystolic200() {  
 *bloodPressureData*.setSystolic(200L);  
 *bloodPressureData*.setDiastolic(50L);  
 Set<ConstraintViolation<BloodPressureData>> constraintViolations = *validator*.validate( *bloodPressureData* );  
  
 *assertEquals*( 1, constraintViolations.size() );  
 *assertEquals*( "Systolic must be less than or equal to 190mmHG", constraintViolations.iterator().next().getMessage());  
 }  
  
 @Test  
 void testSetSystolic70() {  
 *bloodPressureData*.setSystolic(70L);  
 *bloodPressureData*.setDiastolic(50L);  
 Set<ConstraintViolation<BloodPressureData>> constraintViolations = *validator*.validate( *bloodPressureData* );  
  
 *assertEquals*( 0, constraintViolations.size() );  
 *assertEquals*( 70, *bloodPressureData*.getSystolic());  
 }  
  
 @Test  
 void testSetDiastolic20() {  
 *bloodPressureData*.setDiastolic(20L);  
 *bloodPressureData*.setSystolic(70L);  
 Set<ConstraintViolation<BloodPressureData>> constraintViolations = *validator*.validate( *bloodPressureData* );  
  
 *assertEquals*(1, constraintViolations.size() );  
 *assertEquals*("Diastolic must be greater than or equal to 40mmHG", constraintViolations.iterator().next().getMessage());  
 }  
  
 @Test  
 void testSetDiastolic110() {  
 *bloodPressureData*.setDiastolic(110L);  
 *bloodPressureData*.setSystolic(70L);  
 Set<ConstraintViolation<BloodPressureData>> constraintViolations = *validator*.validate( *bloodPressureData* );  
  
 *assertEquals*(1, constraintViolations.size() );  
 *assertEquals*("Diastolic must be less than or equal to 100mmHG", constraintViolations.iterator().next().getMessage());  
 }  
  
 @Test  
 void testSetDiastolic70() {  
 *bloodPressureData*.setDiastolic(70L);  
 *bloodPressureData*.setSystolic(70L);  
 Set<ConstraintViolation<BloodPressureData>> constraintViolations =  
 *validator*.validate( *bloodPressureData* );  
  
 *assertEquals*( 0, constraintViolations.size() );  
 *assertEquals*( 70, *bloodPressureData*.getDiastolic());  
 }  
}

MockMVC:

package com.nerdyninja.college.bp;  
  
import static org.springframework.test.web.servlet.request.MockMvcRequestBuilders.*post*;  
import static org.springframework.test.web.servlet.result.MockMvcResultMatchers.*model*;  
  
import org.junit.jupiter.api.Test;  
import org.junit.jupiter.api.extension.ExtendWith;  
import org.springframework.test.context.junit.jupiter.SpringExtension;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.boot.test.autoconfigure.web.servlet.AutoConfigureMockMvc;  
import org.springframework.boot.test.context.SpringBootTest;  
import org.springframework.test.web.servlet.MockMvc;  
import org.springframework.test.web.servlet.request.MockHttpServletRequestBuilder;  
  
@ExtendWith(SpringExtension.class)  
@SpringBootTest  
@AutoConfigureMockMvc  
class ApplicationMockMvcTests {  
  
 @Autowired  
 private MockMvc mockMvc;  
  
 @Test  
 void checkBloodPressureWithDiastolicMissingNameThenFailure() throws Exception {  
 MockHttpServletRequestBuilder createPerson = *post*("/")  
 .param("systolic", "90").param("agerange","0");  
  
 mockMvc.perform(createPerson)  
 .andExpect(*model*().hasErrors());  
 }  
  
 @Test  
 void checkBloodPressureWithSystolicMissingNameThenFailure() throws Exception {  
 MockHttpServletRequestBuilder createPerson = *post*("/")  
 .param("diastolic", "50").param("agerange","0");  
  
 mockMvc.perform(createPerson)  
 .andExpect(*model*().hasErrors());  
 }  
  
 @Test  
 void checkBloodPressure() throws Exception {  
 MockHttpServletRequestBuilder createPerson = *post*("/")  
 .param("systolic", "90")  
 .param("diastolic", "50")  
 .param("agerange","0");  
  
 mockMvc.perform(createPerson)  
 .andExpect(*model*().hasNoErrors());  
 }  
}

BDD (Spock):

package com.nerdyninja.college.bp  
  
import org.springframework.boot.test.context.SpringBootTest  
import spock.lang.\*  
@SpringBootTest  
class BloodPressureCategoryTestSpec extends Specification {  
  
 BloodPressureData testData  
  
 def "Testing the different category types"(){  
 given: "We create a new BloodPressure Object"  
 testData = new BloodPressureData()  
  
 when: "We set the Systolic and Diastolic Correctly"  
 testData.setSystolic(systolic)  
 testData.setDiastolic(diastolic)  
 testData.setAgerange(agerange)  
 testData.getCategoryType()  
  
 then:  
 testData.category == result  
  
 where:  
 systolic || diastolic || agerange || result  
 60L || 30L || 0 || null  
 60L || 50L || 0 || null  
 80L || 50L || 0 || "Low Blood Pressure"  
 80L || 70L || 0 || "Ideal Blood Pressure"  
 80L || 82L || 0 || "Pre-High Blood Pressure"  
 80L || 92L || 0 || "High Blood Pressure"  
 80L || 100L || 0 || null  
 80L || 30L || 0 || null  
 100L || 70L || 0 || "Ideal Blood Pressure"  
 130L || 70l || 0 || "Pre-High Blood Pressure"  
 145L || 70L || 0 || "High Blood Pressure"  
 145L || 70L || 1 || "Pre-High Blood Pressure"  
 155L || 70L || 1 || "High Blood Pressure"  
 190L || 70L || 0 || null  
 }  
}

E2E (Geb):

package com.nerdyninja.college.bp  
  
import geb.spock.GebSpec  
  
class BloodPressureSpec extends GebSpec {  
  
 def "Can load the main application page"() {  
 when:  
 to BPFrontPage  
  
 then:  
 at BPFrontPage  
 }  
  
 def "Test Low Blood Pressure"(){  
 given:  
 go()  
  
 when:  
 at BPFrontPage  
  
 systolic.value("75")  
 diastolic.value("55")  
 submitButton.click()  
  
 then:  
 assert result.text() == "Low Blood Pressure"  
 }  
  
 def "Test Ideal Blood Pressure"(){  
 given:  
 go()  
  
 when:  
 at BPFrontPage  
  
 systolic.value("80")  
 diastolic.value("70")  
 submitButton.click()  
  
 then:  
 assert result.text() == "Ideal Blood Pressure"  
 }  
  
 def "Test Pre-High Blood Pressure"(){  
 given:  
 go()  
  
 when:  
 at BPFrontPage  
  
 systolic.value("130")  
 diastolic.value("70")  
 age\_select.value("0")  
 submitButton.click()  
  
 then:  
 assert result.text() == "Pre-High Blood Pressure"  
 }  
  
 def "Test High Blood Pressure - < 80"(){  
 given:  
 go()  
  
 when:  
 at BPFrontPage  
  
 systolic.value("145")  
 diastolic.value("70")  
 age\_select.value("0")  
 submitButton.click()  
  
 then:  
 assert result.text() == "High Blood Pressure"  
 }  
  
 def "Test Higher Blood Pressure - > 79"(){  
 given:  
 go()  
  
 when:  
 at BPFrontPage  
  
 systolic.value("145")  
 diastolic.value("70")  
 age\_select.value("1")  
 submitButton.click()  
  
 then:  
 assert result.text() == "Pre-High Blood Pressure"  
 }  
  
 def "Test High Blood Pressure - > 79"(){  
 given:  
 go()  
  
 when:  
 at BPFrontPage  
  
 systolic.value("150")  
 diastolic.value("70")  
 age\_select.value("1")  
 submitButton.click()  
  
 then:  
 assert result.text() == "High Blood Pressure"  
 }  
}

K6 Input:

import { check, sleep } from "k6";  
import http from "k6/http";  
  
// Export an options object to configure how k6 will behave during test execution.  
//  
// See https://docs.k6.io/docs/options  
//  
export let ***options*** = {  
  
 stages: [  
 { duration: "1m", target: 20 }, // 1 new vu every 3 seconds  
 { duration: "1m", target: 20 },  
 { duration: "1m", target: 0 }, // 1 less vu every 3 seconds  
 { duration: "1m", target: 50 },  
 { duration: "1m", target: 20 },  
 { duration: "1m", target: 0 }  
 ],  
  
 thresholds: {  
 "http\_req\_duration": ["p(95) < 100"]  
 },  
  
 discardResponseBodies: false,  
  
 ext: {  
 loadimpact: {  
 // Specify the distribution across load zones  
 //  
 // See https://docs.k6.io/docs/cloud-execution#section-cloud-execution-options  
 //  
 distribution: {  
 loadZoneLabel1: { loadZone: "amazon:ie:dublin", percent: 100 },  
 }  
 }  
 }  
};  
  
*/\*\**  
 *\* Get a random integer between `min` and `max`.*  
 *\**   
 *\** ***@param*** *{number} min - min number*  
 *\** ***@param*** *{number} max - max number*  
 *\** ***@return*** *{number} a random integer*  
 *\*/*  
  
function getRandomInt(min, max) {  
 return Math.floor(Math.random() \* (max - min + 1) + min);  
}  
  
export default function() {  
  
 let res = http.get("http://gc-bmicalculator-qa.azurewebsites.net/bmi", {"responseType": "text"});  
  
 check(res, {  
 "is status 200": (r) => r.status === 200  
 });  
  
 res = res.submitForm({  
 fields: { systolic: getRandomInt(70, 190).toString(), diastolic: getRandomInt(40, 100).toString()}  
 });  
  
 check(res, {  
 "is status 200": (r) => r.status === 200  
 });  
  
 // "think" for 3 seconds  
 sleep(3);  
}

K6 Output:

/\ |‾‾| /‾‾/ /‾‾/

/\ / \ | |/ / / /

/ \/ \ | ( / ‾‾\

/ \ | |\ \ | (‾) |

/ \_\_\_\_\_\_\_\_\_\_ \ |\_\_| \\_\_\ \\_\_\_\_\_/ .io

execution: local

script: k6tests/perf.js

output: -

scenarios: (100.00%) 1 scenario, 50 max VUs, 6m30s max duration (incl. graceful stop):

\* default: Up to 50 looping VUs for 6m0s over 6 stages (gracefulRampDown: 30s, gracefulStop: 30s)

running (0m00.5s), 01/50 VUs, 0 complete and 0 interrupted iterations

default [ 0% ] 01/50 VUs 0m00.5s/6m00.0s

running (0m01.5s), 01/50 VUs, 0 complete and 0 interrupted iterations

default [ 0% ] 01/50 VUs 0m01.5s/6m00.0s

running (0m02.5s), 01/50 VUs, 0 complete and 0 interrupted iterations

default [ 1% ] 01/50 VUs 0m02.5s/6m00.0s

running (0m03.5s), 02/50 VUs, 1 complete and 0 interrupted iterations

default [ 1% ] 02/50 VUs 0m03.5s/6m00.0s

running (0m04.5s), 02/50 VUs, 1 complete and 0 interrupted iterations

default [ 1% ] 02/50 VUs 0m04.5s/6m00.0s

running (0m05.5s), 02/50 VUs, 1 complete and 0 interrupted iterations

default [ 2% ] 02/50 VUs 0m05.5s/6m00.0s

running (0m06.5s), 03/50 VUs, 3 complete and 0 interrupted iterations

default [ 2% ] 03/50 VUs 0m06.5s/6m00.0s

running (0m07.5s), 03/50 VUs, 3 complete and 0 interrupted iterations

default [ 2% ] 03/50 VUs 0m07.5s/6m00.0s

running (0m08.5s), 03/50 VUs, 3 complete and 0 interrupted iterations

default [ 2% ] 03/50 VUs 0m08.5s/6m00.0s

running (0m09.5s), 03/50 VUs, 6 complete and 0 interrupted iterations

default [ 3% ] 03/50 VUs 0m09.5s/6m00.0s

running (0m10.5s), 04/50 VUs, 6 complete and 0 interrupted iterations

default [ 3% ] 04/50 VUs 0m10.5s/6m00.0s

running (0m11.5s), 04/50 VUs, 6 complete and 0 interrupted iterations

default [ 3% ] 04/50 VUs 0m11.5s/6m00.0s

running (0m12.5s), 04/50 VUs, 9 complete and 0 interrupted iterations

default [ 3% ] 04/50 VUs 0m12.5s/6m00.0s

running (0m13.5s), 05/50 VUs, 10 complete and 0 interrupted iterations

default [ 4% ] 05/50 VUs 0m13.5s/6m00.0s

running (0m14.5s), 05/50 VUs, 10 complete and 0 interrupted iterations

default [ 4% ] 05/50 VUs 0m14.5s/6m00.0s

running (0m15.5s), 05/50 VUs, 13 complete and 0 interrupted iterations

default [ 4% ] 05/50 VUs 0m15.5s/6m00.0s

running (0m16.5s), 06/50 VUs, 15 complete and 0 interrupted iterations

default [ 5% ] 06/50 VUs 0m16.5s/6m00.0s

running (0m17.5s), 06/50 VUs, 15 complete and 0 interrupted iterations

default [ 5% ] 06/50 VUs 0m17.5s/6m00.0s

running (0m18.5s), 06/50 VUs, 18 complete and 0 interrupted iterations

default [ 5% ] 06/50 VUs 0m18.5s/6m00.0s

running (0m19.5s), 07/50 VUs, 21 complete and 0 interrupted iterations

default [ 5% ] 07/50 VUs 0m19.5s/6m00.0s

running (0m20.5s), 07/50 VUs, 21 complete and 0 interrupted iterations

default [ 6% ] 07/50 VUs 0m20.5s/6m00.0s

running (0m21.5s), 07/50 VUs, 24 complete and 0 interrupted iterations

default [ 6% ] 07/50 VUs 0m21.5s/6m00.0s

running (0m22.5s), 08/50 VUs, 28 complete and 0 interrupted iterations

default [ 6% ] 08/50 VUs 0m22.5s/6m00.0s

running (0m23.5s), 08/50 VUs, 28 complete and 0 interrupted iterations

default [ 7% ] 08/50 VUs 0m23.5s/6m00.0s

running (0m24.5s), 08/50 VUs, 31 complete and 0 interrupted iterations

default [ 7% ] 08/50 VUs 0m24.5s/6m00.0s

running (0m25.5s), 09/50 VUs, 36 complete and 0 interrupted iterations

default [ 7% ] 09/50 VUs 0m25.5s/6m00.0s

running (0m26.5s), 09/50 VUs, 36 complete and 0 interrupted iterations

default [ 7% ] 09/50 VUs 0m26.5s/6m00.0s

running (0m27.5s), 09/50 VUs, 39 complete and 0 interrupted iterations

default [ 8% ] 09/50 VUs 0m27.5s/6m00.0s

running (0m28.5s), 10/50 VUs, 45 complete and 0 interrupted iterations

default [ 8% ] 10/50 VUs 0m28.5s/6m00.0s

running (0m29.5s), 10/50 VUs, 45 complete and 0 interrupted iterations

default [ 8% ] 10/50 VUs 0m29.5s/6m00.0s

running (0m30.5s), 10/50 VUs, 47 complete and 0 interrupted iterations

default [ 8% ] 10/50 VUs 0m30.5s/6m00.0s

running (0m31.5s), 10/50 VUs, 55 complete and 0 interrupted iterations

default [ 9% ] 10/50 VUs 0m31.5s/6m00.0s

running (0m32.5s), 11/50 VUs, 55 complete and 0 interrupted iterations

default [ 9% ] 11/50 VUs 0m32.5s/6m00.0s

running (0m33.5s), 11/50 VUs, 57 complete and 0 interrupted iterations

default [ 9% ] 11/50 VUs 0m33.5s/6m00.0s

running (0m34.5s), 11/50 VUs, 65 complete and 0 interrupted iterations

default [ 10% ] 11/50 VUs 0m34.5s/6m00.0s

running (0m35.5s), 12/50 VUs, 66 complete and 0 interrupted iterations

default [ 10% ] 12/50 VUs 0m35.5s/6m00.0s

running (0m36.5s), 12/50 VUs, 66 complete and 0 interrupted iterations

default [ 10% ] 12/50 VUs 0m36.5s/6m00.0s

running (0m37.5s), 12/50 VUs, 75 complete and 0 interrupted iterations

default [ 10% ] 12/50 VUs 0m37.5s/6m00.0s

running (0m38.5s), 13/50 VUs, 78 complete and 0 interrupted iterations

default [ 11% ] 13/50 VUs 0m38.5s/6m00.0s

running (0m39.5s), 13/50 VUs, 78 complete and 0 interrupted iterations

default [ 11% ] 13/50 VUs 0m39.5s/6m00.0s

running (0m40.5s), 13/50 VUs, 87 complete and 0 interrupted iterations

default [ 11% ] 13/50 VUs 0m40.5s/6m00.0s

running (0m41.5s), 14/50 VUs, 91 complete and 0 interrupted iterations

default [ 12% ] 14/50 VUs 0m41.5s/6m00.0s

running (0m42.5s), 14/50 VUs, 91 complete and 0 interrupted iterations

default [ 12% ] 14/50 VUs 0m42.5s/6m00.0s

running (0m43.5s), 14/50 VUs, 100 complete and 0 interrupted iterations

default [ 12% ] 14/50 VUs 0m43.5s/6m00.0s

running (0m44.5s), 15/50 VUs, 105 complete and 0 interrupted iterations

default [ 12% ] 15/50 VUs 0m44.5s/6m00.0s

running (0m45.5s), 15/50 VUs, 105 complete and 0 interrupted iterations

default [ 13% ] 15/50 VUs 0m45.5s/6m00.0s

running (0m46.5s), 15/50 VUs, 114 complete and 0 interrupted iterations

default [ 13% ] 15/50 VUs 0m46.5s/6m00.0s

running (0m47.5s), 16/50 VUs, 120 complete and 0 interrupted iterations

default [ 13% ] 16/50 VUs 0m47.5s/6m00.0s

running (0m48.5s), 16/50 VUs, 120 complete and 0 interrupted iterations

default [ 13% ] 16/50 VUs 0m48.5s/6m00.0s

running (0m49.5s), 16/50 VUs, 129 complete and 0 interrupted iterations

default [ 14% ] 16/50 VUs 0m49.5s/6m00.0s

running (0m50.5s), 16/50 VUs, 136 complete and 0 interrupted iterations

default [ 14% ] 16/50 VUs 0m50.5s/6m00.0s

running (0m51.5s), 17/50 VUs, 136 complete and 0 interrupted iterations

default [ 14% ] 17/50 VUs 0m51.5s/6m00.0s

running (0m52.5s), 17/50 VUs, 145 complete and 0 interrupted iterations

default [ 15% ] 17/50 VUs 0m52.5s/6m00.0s

running (0m53.5s), 17/50 VUs, 152 complete and 0 interrupted iterations

default [ 15% ] 17/50 VUs 0m53.5s/6m00.0s

running (0m54.5s), 18/50 VUs, 153 complete and 0 interrupted iterations

default [ 15% ] 18/50 VUs 0m54.5s/6m00.0s

running (0m55.5s), 18/50 VUs, 162 complete and 0 interrupted iterations

default [ 15% ] 18/50 VUs 0m55.5s/6m00.0s

running (0m56.5s), 18/50 VUs, 169 complete and 0 interrupted iterations

default [ 16% ] 18/50 VUs 0m56.5s/6m00.0s

running (0m57.5s), 19/50 VUs, 171 complete and 0 interrupted iterations

default [ 16% ] 19/50 VUs 0m57.5s/6m00.0s

running (0m58.5s), 19/50 VUs, 175 complete and 0 interrupted iterations

default [ 16% ] 19/50 VUs 0m58.5s/6m00.0s

running (0m59.5s), 19/50 VUs, 187 complete and 0 interrupted iterations

default [ 17% ] 19/50 VUs 0m59.5s/6m00.0s

running (1m00.5s), 20/50 VUs, 190 complete and 0 interrupted iterations

default [ 17% ] 20/50 VUs 1m00.5s/6m00.0s

running (1m01.5s), 20/50 VUs, 194 complete and 0 interrupted iterations

default [ 17% ] 20/50 VUs 1m01.5s/6m00.0s

running (1m02.5s), 20/50 VUs, 206 complete and 0 interrupted iterations

default [ 17% ] 20/50 VUs 1m02.5s/6m00.0s

running (1m03.5s), 20/50 VUs, 210 complete and 0 interrupted iterations

default [ 18% ] 20/50 VUs 1m03.5s/6m00.0s

running (1m04.5s), 20/50 VUs, 211 complete and 0 interrupted iterations

default [ 18% ] 20/50 VUs 1m04.5s/6m00.0s

running (1m05.5s), 20/50 VUs, 225 complete and 0 interrupted iterations

default [ 18% ] 20/50 VUs 1m05.5s/6m00.0s

running (1m06.5s), 20/50 VUs, 230 complete and 0 interrupted iterations

default [ 18% ] 20/50 VUs 1m06.5s/6m00.0s

running (1m07.5s), 20/50 VUs, 231 complete and 0 interrupted iterations

default [ 19% ] 20/50 VUs 1m07.5s/6m00.0s

running (1m08.5s), 20/50 VUs, 245 complete and 0 interrupted iterations

default [ 19% ] 20/50 VUs 1m08.5s/6m00.0s

running (1m09.5s), 20/50 VUs, 250 complete and 0 interrupted iterations

default [ 19% ] 20/50 VUs 1m09.5s/6m00.0s

running (1m10.5s), 20/50 VUs, 251 complete and 0 interrupted iterations

default [ 20% ] 20/50 VUs 1m10.5s/6m00.0s

running (1m11.5s), 20/50 VUs, 265 complete and 0 interrupted iterations

default [ 20% ] 20/50 VUs 1m11.5s/6m00.0s

running (1m12.5s), 20/50 VUs, 270 complete and 0 interrupted iterations

default [ 20% ] 20/50 VUs 1m12.5s/6m00.0s

running (1m13.5s), 20/50 VUs, 271 complete and 0 interrupted iterations

default [ 20% ] 20/50 VUs 1m13.5s/6m00.0s

running (1m14.5s), 20/50 VUs, 285 complete and 0 interrupted iterations

default [ 21% ] 20/50 VUs 1m14.5s/6m00.0s

running (1m15.5s), 20/50 VUs, 290 complete and 0 interrupted iterations

default [ 21% ] 20/50 VUs 1m15.5s/6m00.0s

running (1m16.5s), 20/50 VUs, 291 complete and 0 interrupted iterations

default [ 21% ] 20/50 VUs 1m16.5s/6m00.0s

running (1m17.5s), 20/50 VUs, 305 complete and 0 interrupted iterations

default [ 22% ] 20/50 VUs 1m17.5s/6m00.0s

running (1m18.5s), 20/50 VUs, 310 complete and 0 interrupted iterations

default [ 22% ] 20/50 VUs 1m18.5s/6m00.0s

running (1m19.5s), 20/50 VUs, 311 complete and 0 interrupted iterations

default [ 22% ] 20/50 VUs 1m19.5s/6m00.0s

running (1m20.5s), 20/50 VUs, 324 complete and 0 interrupted iterations

default [ 22% ] 20/50 VUs 1m20.5s/6m00.0s

running (1m21.5s), 20/50 VUs, 330 complete and 0 interrupted iterations

default [ 23% ] 20/50 VUs 1m21.5s/6m00.0s

running (1m22.5s), 20/50 VUs, 331 complete and 0 interrupted iterations

default [ 23% ] 20/50 VUs 1m22.5s/6m00.0s

running (1m23.5s), 20/50 VUs, 344 complete and 0 interrupted iterations

default [ 23% ] 20/50 VUs 1m23.5s/6m00.0s

running (1m24.5s), 20/50 VUs, 350 complete and 0 interrupted iterations

default [ 23% ] 20/50 VUs 1m24.5s/6m00.0s

running (1m25.5s), 20/50 VUs, 351 complete and 0 interrupted iterations

default [ 24% ] 20/50 VUs 1m25.5s/6m00.0s

running (1m26.5s), 20/50 VUs, 364 complete and 0 interrupted iterations

default [ 24% ] 20/50 VUs 1m26.5s/6m00.0s

running (1m27.5s), 20/50 VUs, 370 complete and 0 interrupted iterations

default [ 24% ] 20/50 VUs 1m27.5s/6m00.0s

running (1m28.5s), 20/50 VUs, 371 complete and 0 interrupted iterations

default [ 25% ] 20/50 VUs 1m28.5s/6m00.0s

running (1m29.5s), 20/50 VUs, 384 complete and 0 interrupted iterations

default [ 25% ] 20/50 VUs 1m29.5s/6m00.0s

running (1m30.5s), 20/50 VUs, 390 complete and 0 interrupted iterations

default [ 25% ] 20/50 VUs 1m30.5s/6m00.0s

running (1m31.5s), 20/50 VUs, 391 complete and 0 interrupted iterations

default [ 25% ] 20/50 VUs 1m31.5s/6m00.0s

running (1m32.5s), 20/50 VUs, 403 complete and 0 interrupted iterations

default [ 26% ] 20/50 VUs 1m32.5s/6m00.0s

running (1m33.5s), 20/50 VUs, 410 complete and 0 interrupted iterations

default [ 26% ] 20/50 VUs 1m33.5s/6m00.0s

running (1m34.5s), 20/50 VUs, 411 complete and 0 interrupted iterations

default [ 26% ] 20/50 VUs 1m34.5s/6m00.0s

running (1m35.5s), 20/50 VUs, 423 complete and 0 interrupted iterations

default [ 27% ] 20/50 VUs 1m35.5s/6m00.0s

running (1m36.5s), 20/50 VUs, 430 complete and 0 interrupted iterations

default [ 27% ] 20/50 VUs 1m36.5s/6m00.0s

running (1m37.5s), 20/50 VUs, 431 complete and 0 interrupted iterations

default [ 27% ] 20/50 VUs 1m37.5s/6m00.0s

running (1m38.5s), 20/50 VUs, 443 complete and 0 interrupted iterations

default [ 27% ] 20/50 VUs 1m38.5s/6m00.0s

running (1m39.5s), 20/50 VUs, 448 complete and 0 interrupted iterations

default [ 28% ] 20/50 VUs 1m39.5s/6m00.0s

running (1m40.5s), 20/50 VUs, 451 complete and 0 interrupted iterations

default [ 28% ] 20/50 VUs 1m40.5s/6m00.0s

running (1m41.5s), 20/50 VUs, 463 complete and 0 interrupted iterations

default [ 28% ] 20/50 VUs 1m41.5s/6m00.0s

running (1m42.5s), 20/50 VUs, 467 complete and 0 interrupted iterations

default [ 28% ] 20/50 VUs 1m42.5s/6m00.0s

running (1m43.5s), 20/50 VUs, 471 complete and 0 interrupted iterations

default [ 29% ] 20/50 VUs 1m43.5s/6m00.0s

running (1m44.5s), 20/50 VUs, 483 complete and 0 interrupted iterations

default [ 29% ] 20/50 VUs 1m44.5s/6m00.0s

running (1m45.5s), 20/50 VUs, 487 complete and 0 interrupted iterations

default [ 29% ] 20/50 VUs 1m45.5s/6m00.0s

running (1m46.5s), 20/50 VUs, 491 complete and 0 interrupted iterations

default [ 30% ] 20/50 VUs 1m46.5s/6m00.0s

running (1m47.5s), 20/50 VUs, 503 complete and 0 interrupted iterations

default [ 30% ] 20/50 VUs 1m47.5s/6m00.0s

running (1m48.5s), 20/50 VUs, 507 complete and 0 interrupted iterations

default [ 30% ] 20/50 VUs 1m48.5s/6m00.0s

running (1m49.5s), 20/50 VUs, 511 complete and 0 interrupted iterations

default [ 30% ] 20/50 VUs 1m49.5s/6m00.0s

running (1m50.5s), 20/50 VUs, 523 complete and 0 interrupted iterations

default [ 31% ] 20/50 VUs 1m50.5s/6m00.0s

running (1m51.5s), 20/50 VUs, 527 complete and 0 interrupted iterations

default [ 31% ] 20/50 VUs 1m51.5s/6m00.0s

running (1m52.5s), 20/50 VUs, 531 complete and 0 interrupted iterations

default [ 31% ] 20/50 VUs 1m52.5s/6m00.0s

running (1m53.5s), 20/50 VUs, 541 complete and 0 interrupted iterations

default [ 32% ] 20/50 VUs 1m53.5s/6m00.0s

running (1m54.5s), 20/50 VUs, 547 complete and 0 interrupted iterations

default [ 32% ] 20/50 VUs 1m54.5s/6m00.0s

running (1m55.5s), 20/50 VUs, 550 complete and 0 interrupted iterations

default [ 32% ] 20/50 VUs 1m55.5s/6m00.0s

running (1m56.5s), 20/50 VUs, 561 complete and 0 interrupted iterations

default [ 32% ] 20/50 VUs 1m56.5s/6m00.0s

running (1m57.5s), 20/50 VUs, 567 complete and 0 interrupted iterations

default [ 33% ] 20/50 VUs 1m57.5s/6m00.0s

running (1m58.5s), 20/50 VUs, 570 complete and 0 interrupted iterations

default [ 33% ] 20/50 VUs 1m58.5s/6m00.0s

running (1m59.5s), 20/50 VUs, 581 complete and 0 interrupted iterations

default [ 33% ] 20/50 VUs 1m59.5s/6m00.0s

running (2m00.5s), 20/50 VUs, 587 complete and 0 interrupted iterations

default [ 33% ] 20/50 VUs 2m00.5s/6m00.0s

running (2m01.5s), 20/50 VUs, 590 complete and 0 interrupted iterations

default [ 34% ] 20/50 VUs 2m01.5s/6m00.0s

running (2m02.5s), 20/50 VUs, 601 complete and 0 interrupted iterations

default [ 34% ] 20/50 VUs 2m02.5s/6m00.0s

running (2m03.5s), 20/50 VUs, 607 complete and 0 interrupted iterations

default [ 34% ] 20/50 VUs 2m03.5s/6m00.0s

running (2m04.5s), 19/50 VUs, 610 complete and 0 interrupted iterations

default [ 35% ] 19/50 VUs 2m04.5s/6m00.0s

running (2m05.5s), 19/50 VUs, 621 complete and 0 interrupted iterations

default [ 35% ] 19/50 VUs 2m05.5s/6m00.0s

running (2m06.5s), 19/50 VUs, 627 complete and 0 interrupted iterations

default [ 35% ] 19/50 VUs 2m06.5s/6m00.0s

running (2m07.5s), 18/50 VUs, 629 complete and 0 interrupted iterations

default [ 35% ] 18/50 VUs 2m07.5s/6m00.0s

running (2m08.5s), 18/50 VUs, 637 complete and 0 interrupted iterations

default [ 36% ] 18/50 VUs 2m08.5s/6m00.0s

running (2m09.5s), 18/50 VUs, 646 complete and 0 interrupted iterations

default [ 36% ] 18/50 VUs 2m09.5s/6m00.0s

running (2m10.5s), 17/50 VUs, 647 complete and 0 interrupted iterations

default [ 36% ] 17/50 VUs 2m10.5s/6m00.0s

running (2m11.5s), 17/50 VUs, 655 complete and 0 interrupted iterations

default [ 37% ] 17/50 VUs 2m11.5s/6m00.0s

running (2m12.5s), 16/50 VUs, 664 complete and 0 interrupted iterations

default [ 37% ] 16/50 VUs 2m12.5s/6m00.0s

running (2m13.5s), 16/50 VUs, 664 complete and 0 interrupted iterations

default [ 37% ] 16/50 VUs 2m13.5s/6m00.0s

running (2m14.5s), 16/50 VUs, 672 complete and 0 interrupted iterations

default [ 37% ] 16/50 VUs 2m14.5s/6m00.0s

running (2m15.5s), 16/50 VUs, 680 complete and 0 interrupted iterations

default [ 38% ] 16/50 VUs 2m15.5s/6m00.0s

running (2m16.5s), 16/50 VUs, 680 complete and 0 interrupted iterations

default [ 38% ] 16/50 VUs 2m16.5s/6m00.0s

running (2m17.5s), 16/50 VUs, 684 complete and 0 interrupted iterations

default [ 38% ] 16/50 VUs 2m17.5s/6m00.0s

running (2m18.5s), 15/50 VUs, 696 complete and 0 interrupted iterations

default [ 38% ] 15/50 VUs 2m18.5s/6m00.0s

running (2m19.5s), 15/50 VUs, 696 complete and 0 interrupted iterations

default [ 39% ] 15/50 VUs 2m19.5s/6m00.0s

running (2m20.5s), 15/50 VUs, 700 complete and 0 interrupted iterations

default [ 39% ] 15/50 VUs 2m20.5s/6m00.0s

running (2m21.5s), 14/50 VUs, 711 complete and 0 interrupted iterations

default [ 39% ] 14/50 VUs 2m21.5s/6m00.0s

running (2m22.5s), 14/50 VUs, 711 complete and 0 interrupted iterations

default [ 40% ] 14/50 VUs 2m22.5s/6m00.0s

running (2m23.5s), 14/50 VUs, 715 complete and 0 interrupted iterations

default [ 40% ] 14/50 VUs 2m23.5s/6m00.0s

running (2m24.5s), 13/50 VUs, 725 complete and 0 interrupted iterations

default [ 40% ] 13/50 VUs 2m24.5s/6m00.0s

running (2m25.5s), 13/50 VUs, 725 complete and 0 interrupted iterations

default [ 40% ] 13/50 VUs 2m25.5s/6m00.0s

running (2m26.5s), 13/50 VUs, 729 complete and 0 interrupted iterations

default [ 41% ] 13/50 VUs 2m26.5s/6m00.0s

running (2m27.5s), 12/50 VUs, 738 complete and 0 interrupted iterations

default [ 41% ] 12/50 VUs 2m27.5s/6m00.0s

running (2m28.5s), 12/50 VUs, 738 complete and 0 interrupted iterations

default [ 41% ] 12/50 VUs 2m28.5s/6m00.0s

running (2m29.5s), 12/50 VUs, 742 complete and 0 interrupted iterations

default [ 42% ] 12/50 VUs 2m29.5s/6m00.0s

running (2m30.5s), 11/50 VUs, 750 complete and 0 interrupted iterations

default [ 42% ] 11/50 VUs 2m30.5s/6m00.0s

running (2m31.5s), 11/50 VUs, 750 complete and 0 interrupted iterations

default [ 42% ] 11/50 VUs 2m31.5s/6m00.0s

running (2m32.5s), 11/50 VUs, 754 complete and 0 interrupted iterations

default [ 42% ] 11/50 VUs 2m32.5s/6m00.0s

running (2m33.5s), 10/50 VUs, 761 complete and 0 interrupted iterations

default [ 43% ] 10/50 VUs 2m33.5s/6m00.0s

running (2m34.5s), 10/50 VUs, 761 complete and 0 interrupted iterations

default [ 43% ] 10/50 VUs 2m34.5s/6m00.0s

running (2m35.5s), 10/50 VUs, 763 complete and 0 interrupted iterations

default [ 43% ] 10/50 VUs 2m35.5s/6m00.0s

running (2m36.5s), 09/50 VUs, 771 complete and 0 interrupted iterations

default [ 43% ] 09/50 VUs 2m36.5s/6m00.0s

running (2m37.5s), 09/50 VUs, 771 complete and 0 interrupted iterations

default [ 44% ] 09/50 VUs 2m37.5s/6m00.0s

running (2m38.5s), 09/50 VUs, 773 complete and 0 interrupted iterations

default [ 44% ] 09/50 VUs 2m38.5s/6m00.0s

running (2m39.5s), 08/50 VUs, 780 complete and 0 interrupted iterations

default [ 44% ] 08/50 VUs 2m39.5s/6m00.0s

running (2m40.5s), 08/50 VUs, 780 complete and 0 interrupted iterations

default [ 45% ] 08/50 VUs 2m40.5s/6m00.0s

running (2m41.5s), 08/50 VUs, 782 complete and 0 interrupted iterations

default [ 45% ] 08/50 VUs 2m41.5s/6m00.0s

running (2m42.5s), 06/50 VUs, 788 complete and 0 interrupted iterations

default [ 45% ] 06/50 VUs 2m42.5s/6m00.0s

running (2m43.5s), 06/50 VUs, 788 complete and 0 interrupted iterations

default [ 45% ] 06/50 VUs 2m43.5s/6m00.0s

running (2m44.5s), 06/50 VUs, 790 complete and 0 interrupted iterations

default [ 46% ] 06/50 VUs 2m44.5s/6m00.0s

running (2m45.5s), 06/50 VUs, 794 complete and 0 interrupted iterations

default [ 46% ] 06/50 VUs 2m45.5s/6m00.0s

running (2m46.5s), 06/50 VUs, 794 complete and 0 interrupted iterations

default [ 46% ] 06/50 VUs 2m46.5s/6m00.0s

running (2m47.5s), 06/50 VUs, 796 complete and 0 interrupted iterations

default [ 47% ] 06/50 VUs 2m47.5s/6m00.0s

running (2m48.5s), 04/50 VUs, 800 complete and 0 interrupted iterations

default [ 47% ] 04/50 VUs 2m48.5s/6m00.0s

running (2m49.5s), 04/50 VUs, 800 complete and 0 interrupted iterations

default [ 47% ] 04/50 VUs 2m49.5s/6m00.0s

running (2m50.5s), 04/50 VUs, 802 complete and 0 interrupted iterations

default [ 47% ] 04/50 VUs 2m50.5s/6m00.0s

running (2m51.5s), 04/50 VUs, 804 complete and 0 interrupted iterations

default [ 48% ] 04/50 VUs 2m51.5s/6m00.0s

running (2m52.5s), 04/50 VUs, 804 complete and 0 interrupted iterations

default [ 48% ] 04/50 VUs 2m52.5s/6m00.0s

running (2m53.5s), 03/50 VUs, 806 complete and 0 interrupted iterations

default [ 48% ] 03/50 VUs 2m53.5s/6m00.0s

running (2m54.5s), 02/50 VUs, 808 complete and 0 interrupted iterations

default [ 48% ] 02/50 VUs 2m54.5s/6m00.0s

running (2m55.5s), 02/50 VUs, 808 complete and 0 interrupted iterations

default [ 49% ] 02/50 VUs 2m55.5s/6m00.0s

running (2m56.5s), 02/50 VUs, 809 complete and 0 interrupted iterations

default [ 49% ] 02/50 VUs 2m56.5s/6m00.0s

running (2m57.5s), 02/50 VUs, 810 complete and 0 interrupted iterations

default [ 49% ] 02/50 VUs 2m57.5s/6m00.0s

running (2m58.5s), 02/50 VUs, 810 complete and 0 interrupted iterations

default [ 50% ] 02/50 VUs 2m58.5s/6m00.0s

running (2m59.5s), 01/50 VUs, 811 complete and 0 interrupted iterations

default [ 50% ] 01/50 VUs 2m59.5s/6m00.0s

running (3m00.5s), 01/50 VUs, 812 complete and 0 interrupted iterations

default [ 50% ] 01/50 VUs 3m00.5s/6m00.0s

running (3m01.5s), 01/50 VUs, 812 complete and 0 interrupted iterations

default [ 50% ] 01/50 VUs 3m01.5s/6m00.0s

running (3m02.5s), 02/50 VUs, 812 complete and 0 interrupted iterations

default [ 51% ] 02/50 VUs 3m02.5s/6m00.0s

running (3m03.5s), 02/50 VUs, 813 complete and 0 interrupted iterations

default [ 51% ] 02/50 VUs 3m03.5s/6m00.0s

running (3m04.5s), 03/50 VUs, 813 complete and 0 interrupted iterations

default [ 51% ] 03/50 VUs 3m04.5s/6m00.0s

running (3m05.5s), 04/50 VUs, 814 complete and 0 interrupted iterations

default [ 52% ] 04/50 VUs 3m05.5s/6m00.0s

running (3m06.5s), 05/50 VUs, 815 complete and 0 interrupted iterations

default [ 52% ] 05/50 VUs 3m06.5s/6m00.0s

running (3m07.5s), 06/50 VUs, 816 complete and 0 interrupted iterations

default [ 52% ] 06/50 VUs 3m07.5s/6m00.0s

running (3m08.5s), 07/50 VUs, 818 complete and 0 interrupted iterations

default [ 52% ] 07/50 VUs 3m08.5s/6m00.0s

running (3m09.5s), 07/50 VUs, 820 complete and 0 interrupted iterations

default [ 53% ] 07/50 VUs 3m09.5s/6m00.0s

running (3m10.5s), 08/50 VUs, 822 complete and 0 interrupted iterations

default [ 53% ] 08/50 VUs 3m10.5s/6m00.0s

running (3m11.5s), 09/50 VUs, 824 complete and 0 interrupted iterations

default [ 53% ] 09/50 VUs 3m11.5s/6m00.0s

running (3m12.5s), 10/50 VUs, 827 complete and 0 interrupted iterations

default [ 53% ] 10/50 VUs 3m12.5s/6m00.0s

running (3m13.5s), 11/50 VUs, 830 complete and 0 interrupted iterations

default [ 54% ] 11/50 VUs 3m13.5s/6m00.0s

running (3m14.5s), 12/50 VUs, 833 complete and 0 interrupted iterations

default [ 54% ] 12/50 VUs 3m14.5s/6m00.0s

running (3m15.5s), 12/50 VUs, 837 complete and 0 interrupted iterations

default [ 54% ] 12/50 VUs 3m15.5s/6m00.0s

running (3m16.5s), 13/50 VUs, 841 complete and 0 interrupted iterations

default [ 55% ] 13/50 VUs 3m16.5s/6m00.0s

running (3m17.5s), 14/50 VUs, 844 complete and 0 interrupted iterations

default [ 55% ] 14/50 VUs 3m17.5s/6m00.0s

running (3m18.5s), 15/50 VUs, 849 complete and 0 interrupted iterations

default [ 55% ] 15/50 VUs 3m18.5s/6m00.0s

running (3m19.5s), 16/50 VUs, 854 complete and 0 interrupted iterations

default [ 55% ] 16/50 VUs 3m19.5s/6m00.0s

running (3m20.5s), 17/50 VUs, 858 complete and 0 interrupted iterations

default [ 56% ] 17/50 VUs 3m20.5s/6m00.0s

running (3m21.5s), 17/50 VUs, 864 complete and 0 interrupted iterations

default [ 56% ] 17/50 VUs 3m21.5s/6m00.0s

running (3m22.5s), 18/50 VUs, 870 complete and 0 interrupted iterations

default [ 56% ] 18/50 VUs 3m22.5s/6m00.0s

running (3m23.5s), 19/50 VUs, 874 complete and 0 interrupted iterations

default [ 57% ] 19/50 VUs 3m23.5s/6m00.0s

running (3m24.5s), 20/50 VUs, 881 complete and 0 interrupted iterations

default [ 57% ] 20/50 VUs 3m24.5s/6m00.0s

running (3m25.5s), 21/50 VUs, 888 complete and 0 interrupted iterations

default [ 57% ] 21/50 VUs 3m25.5s/6m00.0s

running (3m26.5s), 22/50 VUs, 893 complete and 0 interrupted iterations

default [ 57% ] 22/50 VUs 3m26.5s/6m00.0s

running (3m27.5s), 22/50 VUs, 901 complete and 0 interrupted iterations

default [ 58% ] 22/50 VUs 3m27.5s/6m00.0s

running (3m28.5s), 23/50 VUs, 909 complete and 0 interrupted iterations

default [ 58% ] 23/50 VUs 3m28.5s/6m00.0s

running (3m29.5s), 24/50 VUs, 915 complete and 0 interrupted iterations

default [ 58% ] 24/50 VUs 3m29.5s/6m00.0s

running (3m30.5s), 25/50 VUs, 923 complete and 0 interrupted iterations

default [ 58% ] 25/50 VUs 3m30.5s/6m00.0s

running (3m31.5s), 26/50 VUs, 932 complete and 0 interrupted iterations

default [ 59% ] 26/50 VUs 3m31.5s/6m00.0s

running (3m32.5s), 27/50 VUs, 938 complete and 0 interrupted iterations

default [ 59% ] 27/50 VUs 3m32.5s/6m00.0s

running (3m33.5s), 27/50 VUs, 948 complete and 0 interrupted iterations

default [ 59% ] 27/50 VUs 3m33.5s/6m00.0s

running (3m34.5s), 28/50 VUs, 958 complete and 0 interrupted iterations

default [ 60% ] 28/50 VUs 3m34.5s/6m00.0s

running (3m35.5s), 29/50 VUs, 965 complete and 0 interrupted iterations

default [ 60% ] 29/50 VUs 3m35.5s/6m00.0s

running (3m36.5s), 30/50 VUs, 975 complete and 0 interrupted iterations

default [ 60% ] 30/50 VUs 3m36.5s/6m00.0s

running (3m37.5s), 31/50 VUs, 986 complete and 0 interrupted iterations

default [ 60% ] 31/50 VUs 3m37.5s/6m00.0s

running (3m38.5s), 32/50 VUs, 994 complete and 0 interrupted iterations

default [ 61% ] 32/50 VUs 3m38.5s/6m00.0s

running (3m39.5s), 32/50 VUs, 1005 complete and 0 interrupted iterations

default [ 61% ] 32/50 VUs 3m39.5s/6m00.0s

running (3m40.5s), 33/50 VUs, 1017 complete and 0 interrupted iterations

default [ 61% ] 33/50 VUs 3m40.5s/6m00.0s

running (3m41.5s), 34/50 VUs, 1024 complete and 0 interrupted iterations

default [ 62% ] 34/50 VUs 3m41.5s/6m00.0s

running (3m42.5s), 35/50 VUs, 1037 complete and 0 interrupted iterations

default [ 62% ] 35/50 VUs 3m42.5s/6m00.0s

running (3m43.5s), 36/50 VUs, 1049 complete and 0 interrupted iterations

default [ 62% ] 36/50 VUs 3m43.5s/6m00.0s

running (3m44.5s), 37/50 VUs, 1058 complete and 0 interrupted iterations

default [ 62% ] 37/50 VUs 3m44.5s/6m00.0s

running (3m45.5s), 37/50 VUs, 1071 complete and 0 interrupted iterations

default [ 63% ] 37/50 VUs 3m45.5s/6m00.0s

running (3m46.5s), 38/50 VUs, 1085 complete and 0 interrupted iterations

default [ 63% ] 38/50 VUs 3m46.5s/6m00.0s

running (3m47.5s), 39/50 VUs, 1093 complete and 0 interrupted iterations

default [ 63% ] 39/50 VUs 3m47.5s/6m00.0s

running (3m48.5s), 40/50 VUs, 1108 complete and 0 interrupted iterations

default [ 63% ] 40/50 VUs 3m48.5s/6m00.0s

running (3m49.5s), 41/50 VUs, 1123 complete and 0 interrupted iterations

default [ 64% ] 41/50 VUs 3m49.5s/6m00.0s

running (3m50.5s), 42/50 VUs, 1132 complete and 0 interrupted iterations

default [ 64% ] 42/50 VUs 3m50.5s/6m00.0s

running (3m51.5s), 42/50 VUs, 1148 complete and 0 interrupted iterations

default [ 64% ] 42/50 VUs 3m51.5s/6m00.0s

running (3m52.5s), 43/50 VUs, 1163 complete and 0 interrupted iterations

default [ 65% ] 43/50 VUs 3m52.5s/6m00.0s

running (3m53.5s), 44/50 VUs, 1174 complete and 0 interrupted iterations

default [ 65% ] 44/50 VUs 3m53.5s/6m00.0s

running (3m54.5s), 45/50 VUs, 1188 complete and 0 interrupted iterations

default [ 65% ] 45/50 VUs 3m54.5s/6m00.0s

running (3m55.5s), 46/50 VUs, 1206 complete and 0 interrupted iterations

default [ 65% ] 46/50 VUs 3m55.5s/6m00.0s

running (3m56.5s), 47/50 VUs, 1218 complete and 0 interrupted iterations

default [ 66% ] 47/50 VUs 3m56.5s/6m00.0s

running (3m57.5s), 47/50 VUs, 1232 complete and 0 interrupted iterations

default [ 66% ] 47/50 VUs 3m57.5s/6m00.0s

running (3m58.5s), 48/50 VUs, 1251 complete and 0 interrupted iterations

default [ 66% ] 48/50 VUs 3m58.5s/6m00.0s

running (3m59.5s), 49/50 VUs, 1265 complete and 0 interrupted iterations

default [ 67% ] 49/50 VUs 3m59.5s/6m00.0s

running (4m00.5s), 50/50 VUs, 1279 complete and 0 interrupted iterations

default [ 67% ] 50/50 VUs 4m00.5s/6m00.0s

running (4m01.5s), 50/50 VUs, 1299 complete and 0 interrupted iterations

default [ 67% ] 50/50 VUs 4m01.5s/6m00.0s

running (4m02.5s), 50/50 VUs, 1313 complete and 0 interrupted iterations

default [ 67% ] 50/50 VUs 4m02.5s/6m00.0s

running (4m03.5s), 49/50 VUs, 1329 complete and 0 interrupted iterations

default [ 68% ] 49/50 VUs 4m03.5s/6m00.0s

running (4m04.5s), 49/50 VUs, 1349 complete and 0 interrupted iterations

default [ 68% ] 49/50 VUs 4m04.5s/6m00.0s

running (4m05.5s), 48/50 VUs, 1363 complete and 0 interrupted iterations

default [ 68% ] 48/50 VUs 4m05.5s/6m00.0s

running (4m06.5s), 48/50 VUs, 1378 complete and 0 interrupted iterations

default [ 68% ] 48/50 VUs 4m06.5s/6m00.0s

running (4m07.5s), 47/50 VUs, 1397 complete and 0 interrupted iterations

default [ 69% ] 47/50 VUs 4m07.5s/6m00.0s

running (4m08.5s), 47/50 VUs, 1410 complete and 0 interrupted iterations

default [ 69% ] 47/50 VUs 4m08.5s/6m00.0s

running (4m09.5s), 46/50 VUs, 1425 complete and 0 interrupted iterations

default [ 69% ] 46/50 VUs 4m09.5s/6m00.0s

running (4m10.5s), 45/50 VUs, 1443 complete and 0 interrupted iterations

default [ 70% ] 45/50 VUs 4m10.5s/6m00.0s

running (4m11.5s), 45/50 VUs, 1457 complete and 0 interrupted iterations

default [ 70% ] 45/50 VUs 4m11.5s/6m00.0s

running (4m12.5s), 44/50 VUs, 1471 complete and 0 interrupted iterations

default [ 70% ] 44/50 VUs 4m12.5s/6m00.0s

running (4m13.5s), 44/50 VUs, 1485 complete and 0 interrupted iterations

default [ 70% ] 44/50 VUs 4m13.5s/6m00.0s

running (4m14.5s), 44/50 VUs, 1502 complete and 0 interrupted iterations

default [ 71% ] 44/50 VUs 4m14.5s/6m00.0s

running (4m15.5s), 44/50 VUs, 1515 complete and 0 interrupted iterations

default [ 71% ] 44/50 VUs 4m15.5s/6m00.0s

running (4m16.5s), 44/50 VUs, 1529 complete and 0 interrupted iterations

default [ 71% ] 44/50 VUs 4m16.5s/6m00.0s

running (4m17.5s), 43/50 VUs, 1543 complete and 0 interrupted iterations

default [ 72% ] 43/50 VUs 4m17.5s/6m00.0s

running (4m18.5s), 43/50 VUs, 1559 complete and 0 interrupted iterations

default [ 72% ] 43/50 VUs 4m18.5s/6m00.0s

running (4m19.5s), 42/50 VUs, 1573 complete and 0 interrupted iterations

default [ 72% ] 42/50 VUs 4m19.5s/6m00.0s

running (4m20.5s), 42/50 VUs, 1586 complete and 0 interrupted iterations

default [ 72% ] 42/50 VUs 4m20.5s/6m00.0s

running (4m21.5s), 41/50 VUs, 1602 complete and 0 interrupted iterations

default [ 73% ] 41/50 VUs 4m21.5s/6m00.0s

running (4m22.5s), 41/50 VUs, 1615 complete and 0 interrupted iterations

default [ 73% ] 41/50 VUs 4m22.5s/6m00.0s

running (4m23.5s), 40/50 VUs, 1628 complete and 0 interrupted iterations

default [ 73% ] 40/50 VUs 4m23.5s/6m00.0s

running (4m24.5s), 39/50 VUs, 1643 complete and 0 interrupted iterations

default [ 73% ] 39/50 VUs 4m24.5s/6m00.0s

running (4m25.5s), 39/50 VUs, 1656 complete and 0 interrupted iterations

default [ 74% ] 39/50 VUs 4m25.5s/6m00.0s

running (4m26.5s), 38/50 VUs, 1668 complete and 0 interrupted iterations

default [ 74% ] 38/50 VUs 4m26.5s/6m00.0s

running (4m27.5s), 38/50 VUs, 1682 complete and 0 interrupted iterations

default [ 74% ] 38/50 VUs 4m27.5s/6m00.0s

running (4m28.5s), 37/50 VUs, 1693 complete and 0 interrupted iterations

default [ 75% ] 37/50 VUs 4m28.5s/6m00.0s

running (4m29.5s), 37/50 VUs, 1706 complete and 0 interrupted iterations

default [ 75% ] 37/50 VUs 4m29.5s/6m00.0s

running (4m30.5s), 36/50 VUs, 1720 complete and 0 interrupted iterations

default [ 75% ] 36/50 VUs 4m30.5s/6m00.0s

running (4m31.5s), 36/50 VUs, 1730 complete and 0 interrupted iterations

default [ 75% ] 36/50 VUs 4m31.5s/6m00.0s

running (4m32.5s), 35/50 VUs, 1743 complete and 0 interrupted iterations

default [ 76% ] 35/50 VUs 4m32.5s/6m00.0s

running (4m33.5s), 35/50 VUs, 1753 complete and 0 interrupted iterations

default [ 76% ] 35/50 VUs 4m33.5s/6m00.0s

running (4m34.5s), 34/50 VUs, 1766 complete and 0 interrupted iterations

default [ 76% ] 34/50 VUs 4m34.5s/6m00.0s

running (4m35.5s), 33/50 VUs, 1778 complete and 0 interrupted iterations

default [ 77% ] 33/50 VUs 4m35.5s/6m00.0s

running (4m36.5s), 33/50 VUs, 1788 complete and 0 interrupted iterations

default [ 77% ] 33/50 VUs 4m36.5s/6m00.0s

running (4m37.5s), 32/50 VUs, 1799 complete and 0 interrupted iterations

default [ 77% ] 32/50 VUs 4m37.5s/6m00.0s

running (4m38.5s), 32/50 VUs, 1811 complete and 0 interrupted iterations

default [ 77% ] 32/50 VUs 4m38.5s/6m00.0s

running (4m39.5s), 31/50 VUs, 1821 complete and 0 interrupted iterations

default [ 78% ] 31/50 VUs 4m39.5s/6m00.0s

running (4m40.5s), 31/50 VUs, 1831 complete and 0 interrupted iterations

default [ 78% ] 31/50 VUs 4m40.5s/6m00.0s

running (4m41.5s), 30/50 VUs, 1843 complete and 0 interrupted iterations

default [ 78% ] 30/50 VUs 4m41.5s/6m00.0s

running (4m42.5s), 30/50 VUs, 1852 complete and 0 interrupted iterations

default [ 78% ] 30/50 VUs 4m42.5s/6m00.0s

running (4m43.5s), 29/50 VUs, 1862 complete and 0 interrupted iterations

default [ 79% ] 29/50 VUs 4m43.5s/6m00.0s

running (4m44.5s), 28/50 VUs, 1873 complete and 0 interrupted iterations

default [ 79% ] 28/50 VUs 4m44.5s/6m00.0s

running (4m45.5s), 28/50 VUs, 1882 complete and 0 interrupted iterations

default [ 79% ] 28/50 VUs 4m45.5s/6m00.0s

running (4m46.5s), 27/50 VUs, 1891 complete and 0 interrupted iterations

default [ 80% ] 27/50 VUs 4m46.5s/6m00.0s

running (4m47.5s), 27/50 VUs, 1900 complete and 0 interrupted iterations

default [ 80% ] 27/50 VUs 4m47.5s/6m00.0s

running (4m48.5s), 26/50 VUs, 1910 complete and 0 interrupted iterations

default [ 80% ] 26/50 VUs 4m48.5s/6m00.0s

running (4m49.5s), 26/50 VUs, 1918 complete and 0 interrupted iterations

default [ 80% ] 26/50 VUs 4m49.5s/6m00.0s

running (4m50.5s), 26/50 VUs, 1927 complete and 0 interrupted iterations

default [ 81% ] 26/50 VUs 4m50.5s/6m00.0s

running (4m51.5s), 26/50 VUs, 1936 complete and 0 interrupted iterations

default [ 81% ] 26/50 VUs 4m51.5s/6m00.0s

running (4m52.5s), 26/50 VUs, 1944 complete and 0 interrupted iterations

default [ 81% ] 26/50 VUs 4m52.5s/6m00.0s

running (4m53.5s), 25/50 VUs, 1953 complete and 0 interrupted iterations

default [ 82% ] 25/50 VUs 4m53.5s/6m00.0s

running (4m54.5s), 25/50 VUs, 1962 complete and 0 interrupted iterations

default [ 82% ] 25/50 VUs 4m54.5s/6m00.0s

running (4m55.5s), 24/50 VUs, 1970 complete and 0 interrupted iterations

default [ 82% ] 24/50 VUs 4m55.5s/6m00.0s

running (4m56.5s), 23/50 VUs, 1978 complete and 0 interrupted iterations

default [ 82% ] 23/50 VUs 4m56.5s/6m00.0s

running (4m57.5s), 23/50 VUs, 1987 complete and 0 interrupted iterations

default [ 83% ] 23/50 VUs 4m57.5s/6m00.0s

running (4m58.5s), 23/50 VUs, 1992 complete and 0 interrupted iterations

default [ 83% ] 23/50 VUs 4m58.5s/6m00.0s

running (4m59.5s), 22/50 VUs, 2001 complete and 0 interrupted iterations

default [ 83% ] 22/50 VUs 4m59.5s/6m00.0s

running (5m00.5s), 21/50 VUs, 2008 complete and 0 interrupted iterations

default [ 83% ] 21/50 VUs 5m00.5s/6m00.0s

running (5m01.5s), 21/50 VUs, 2015 complete and 0 interrupted iterations

default [ 84% ] 21/50 VUs 5m01.5s/6m00.0s

running (5m02.5s), 20/50 VUs, 2023 complete and 0 interrupted iterations

default [ 84% ] 20/50 VUs 5m02.5s/6m00.0s

running (5m03.5s), 20/50 VUs, 2029 complete and 0 interrupted iterations

default [ 84% ] 20/50 VUs 5m03.5s/6m00.0s

running (5m04.5s), 19/50 VUs, 2036 complete and 0 interrupted iterations

default [ 85% ] 19/50 VUs 5m04.5s/6m00.0s

running (5m05.5s), 19/50 VUs, 2043 complete and 0 interrupted iterations

default [ 85% ] 19/50 VUs 5m05.5s/6m00.0s

running (5m06.5s), 19/50 VUs, 2049 complete and 0 interrupted iterations

default [ 85% ] 19/50 VUs 5m06.5s/6m00.0s

running (5m07.5s), 19/50 VUs, 2055 complete and 0 interrupted iterations

default [ 85% ] 19/50 VUs 5m07.5s/6m00.0s

running (5m08.5s), 19/50 VUs, 2062 complete and 0 interrupted iterations

default [ 86% ] 19/50 VUs 5m08.5s/6m00.0s

running (5m09.5s), 18/50 VUs, 2068 complete and 0 interrupted iterations

default [ 86% ] 18/50 VUs 5m09.5s/6m00.0s

running (5m10.5s), 18/50 VUs, 2074 complete and 0 interrupted iterations

default [ 86% ] 18/50 VUs 5m10.5s/6m00.0s

running (5m11.5s), 17/50 VUs, 2081 complete and 0 interrupted iterations

default [ 87% ] 17/50 VUs 5m11.5s/6m00.0s

running (5m12.5s), 17/50 VUs, 2086 complete and 0 interrupted iterations

default [ 87% ] 17/50 VUs 5m12.5s/6m00.0s

running (5m13.5s), 16/50 VUs, 2092 complete and 0 interrupted iterations

default [ 87% ] 16/50 VUs 5m13.5s/6m00.0s

running (5m14.5s), 16/50 VUs, 2098 complete and 0 interrupted iterations

default [ 87% ] 16/50 VUs 5m14.5s/6m00.0s

running (5m15.5s), 16/50 VUs, 2103 complete and 0 interrupted iterations

default [ 88% ] 16/50 VUs 5m15.5s/6m00.0s

running (5m16.5s), 16/50 VUs, 2108 complete and 0 interrupted iterations

default [ 88% ] 16/50 VUs 5m16.5s/6m00.0s

running (5m17.5s), 15/50 VUs, 2114 complete and 0 interrupted iterations

default [ 88% ] 15/50 VUs 5m17.5s/6m00.0s

running (5m18.5s), 15/50 VUs, 2119 complete and 0 interrupted iterations

default [ 88% ] 15/50 VUs 5m18.5s/6m00.0s

running (5m19.5s), 14/50 VUs, 2124 complete and 0 interrupted iterations

default [ 89% ] 14/50 VUs 5m19.5s/6m00.0s

running (5m20.5s), 14/50 VUs, 2129 complete and 0 interrupted iterations

default [ 89% ] 14/50 VUs 5m20.5s/6m00.0s

running (5m21.5s), 13/50 VUs, 2134 complete and 0 interrupted iterations

default [ 89% ] 13/50 VUs 5m21.5s/6m00.0s

running (5m22.5s), 13/50 VUs, 2136 complete and 0 interrupted iterations

default [ 90% ] 13/50 VUs 5m22.5s/6m00.0s

running (5m23.5s), 13/50 VUs, 2141 complete and 0 interrupted iterations

default [ 90% ] 13/50 VUs 5m23.5s/6m00.0s

running (5m24.5s), 13/50 VUs, 2147 complete and 0 interrupted iterations

default [ 90% ] 13/50 VUs 5m24.5s/6m00.0s

running (5m25.5s), 13/50 VUs, 2149 complete and 0 interrupted iterations

default [ 90% ] 13/50 VUs 5m25.5s/6m00.0s

running (5m26.5s), 12/50 VUs, 2154 complete and 0 interrupted iterations

default [ 91% ] 12/50 VUs 5m26.5s/6m00.0s

running (5m27.5s), 11/50 VUs, 2160 complete and 0 interrupted iterations

default [ 91% ] 11/50 VUs 5m27.5s/6m00.0s

running (5m28.5s), 11/50 VUs, 2162 complete and 0 interrupted iterations

default [ 91% ] 11/50 VUs 5m28.5s/6m00.0s

running (5m29.5s), 11/50 VUs, 2166 complete and 0 interrupted iterations

default [ 92% ] 11/50 VUs 5m29.5s/6m00.0s

running (5m30.5s), 11/50 VUs, 2170 complete and 0 interrupted iterations

default [ 92% ] 11/50 VUs 5m30.5s/6m00.0s

running (5m31.5s), 11/50 VUs, 2173 complete and 0 interrupted iterations

default [ 92% ] 11/50 VUs 5m31.5s/6m00.0s

running (5m32.5s), 11/50 VUs, 2177 complete and 0 interrupted iterations

default [ 92% ] 11/50 VUs 5m32.5s/6m00.0s

running (5m33.5s), 10/50 VUs, 2181 complete and 0 interrupted iterations

default [ 93% ] 10/50 VUs 5m33.5s/6m00.0s

running (5m34.5s), 09/50 VUs, 2184 complete and 0 interrupted iterations

default [ 93% ] 09/50 VUs 5m34.5s/6m00.0s

running (5m35.5s), 09/50 VUs, 2188 complete and 0 interrupted iterations

default [ 93% ] 09/50 VUs 5m35.5s/6m00.0s

running (5m36.5s), 08/50 VUs, 2191 complete and 0 interrupted iterations

default [ 93% ] 08/50 VUs 5m36.5s/6m00.0s

running (5m37.5s), 08/50 VUs, 2193 complete and 0 interrupted iterations

default [ 94% ] 08/50 VUs 5m37.5s/6m00.0s

running (5m38.5s), 08/50 VUs, 2197 complete and 0 interrupted iterations

default [ 94% ] 08/50 VUs 5m38.5s/6m00.0s

running (5m39.5s), 08/50 VUs, 2199 complete and 0 interrupted iterations

default [ 94% ] 08/50 VUs 5m39.5s/6m00.0s

running (5m40.5s), 08/50 VUs, 2201 complete and 0 interrupted iterations

default [ 95% ] 08/50 VUs 5m40.5s/6m00.0s

running (5m41.5s), 07/50 VUs, 2205 complete and 0 interrupted iterations

default [ 95% ] 07/50 VUs 5m41.5s/6m00.0s

running (5m42.5s), 07/50 VUs, 2207 complete and 0 interrupted iterations

default [ 95% ] 07/50 VUs 5m42.5s/6m00.0s

running (5m43.5s), 06/50 VUs, 2209 complete and 0 interrupted iterations

default [ 95% ] 06/50 VUs 5m43.5s/6m00.0s

running (5m44.5s), 06/50 VUs, 2212 complete and 0 interrupted iterations

default [ 96% ] 06/50 VUs 5m44.5s/6m00.0s

running (5m45.5s), 06/50 VUs, 2214 complete and 0 interrupted iterations

default [ 96% ] 06/50 VUs 5m45.5s/6m00.0s

running (5m46.5s), 06/50 VUs, 2215 complete and 0 interrupted iterations

default [ 96% ] 06/50 VUs 5m46.5s/6m00.0s

running (5m47.5s), 06/50 VUs, 2218 complete and 0 interrupted iterations

default [ 97% ] 06/50 VUs 5m47.5s/6m00.0s

running (5m48.5s), 05/50 VUs, 2220 complete and 0 interrupted iterations

default [ 97% ] 05/50 VUs 5m48.5s/6m00.0s

running (5m49.5s), 05/50 VUs, 2221 complete and 0 interrupted iterations

default [ 97% ] 05/50 VUs 5m49.5s/6m00.0s

running (5m50.5s), 04/50 VUs, 2224 complete and 0 interrupted iterations

default [ 97% ] 04/50 VUs 5m50.5s/6m00.0s

running (5m51.5s), 03/50 VUs, 2225 complete and 0 interrupted iterations

default [ 98% ] 03/50 VUs 5m51.5s/6m00.0s

running (5m52.5s), 03/50 VUs, 2226 complete and 0 interrupted iterations

default [ 98% ] 03/50 VUs 5m52.5s/6m00.0s

running (5m53.5s), 03/50 VUs, 2228 complete and 0 interrupted iterations

default [ 98% ] 03/50 VUs 5m53.5s/6m00.0s

running (5m54.5s), 03/50 VUs, 2228 complete and 0 interrupted iterations

default [ 98% ] 03/50 VUs 5m54.5s/6m00.0s

running (5m55.5s), 03/50 VUs, 2229 complete and 0 interrupted iterations

default [ 99% ] 03/50 VUs 5m55.5s/6m00.0s

running (5m56.5s), 02/50 VUs, 2231 complete and 0 interrupted iterations

default [ 99% ] 02/50 VUs 5m56.5s/6m00.0s

running (5m57.5s), 02/50 VUs, 2231 complete and 0 interrupted iterations

default [ 99% ] 02/50 VUs 5m57.5s/6m00.0s

running (5m58.5s), 01/50 VUs, 2232 complete and 0 interrupted iterations

default [ 100% ] 01/50 VUs 5m58.5s/6m00.0s

running (5m59.5s), 01/50 VUs, 2233 complete and 0 interrupted iterations

default [ 100% ] 01/50 VUs 5m59.5s/6m00.0s

running (6m00.5s), 01/50 VUs, 2233 complete and 0 interrupted iterations

default ↓ [ 100% ] 01/50 VUs 6m0s

running (6m01.5s), 01/50 VUs, 2233 complete and 0 interrupted iterations

default ↓ [ 100% ] 01/50 VUs 6m0s

running (6m02.0s), 00/50 VUs, 2234 complete and 0 interrupted iterations

default ✓ [ 100% ] 00/50 VUs 6m0s

✓ is status 200

checks.....................: 100.00% ✓ 4468 ✗ 0

data\_received..............: 43 MB 118 kB/s

data\_sent..................: 1.9 MB 5.2 kB/s

http\_req\_blocked...........: avg=37.48µs min=1.5µs med=4µs max=40.8ms p(90)=6.3µs p(95)=6.9µs

http\_req\_connecting........: avg=23.24µs min=0s med=0s max=7.87ms p(90)=0s p(95)=0s

✓ http\_req\_duration..........: avg=14.83ms min=5.09ms med=10.71ms max=568.58ms p(90)=26.11ms p(95)=34.94ms

http\_req\_receiving.........: avg=1.92ms min=31.7µs med=321.96µs max=204.59ms p(90)=6.33ms p(95)=8.32ms

http\_req\_sending...........: avg=32.32µs min=6.7µs med=30.05µs max=2.9ms p(90)=41.7µs p(95)=45.76µs

http\_req\_tls\_handshaking...: avg=0s min=0s med=0s max=0s p(90)=0s p(95)=0s

http\_req\_waiting...........: avg=12.88ms min=4.75ms med=9ms max=565.7ms p(90)=21.33ms p(95)=28.8ms

http\_reqs..................: 4468 12.343402/s

iteration\_duration.........: avg=3.03s min=3.01s med=3.02s max=3.61s p(90)=3.05s p(95)=3.06s

iterations.................: 2234 6.171701/s

vus........................: 1 min=1 max=50

vus\_max....................: 50 min=50 max=50

ZAP Output:

