Load Testing for RBS Maestro team pipelines

Project report submitted in partial fulfillment of the requirement for the degree of Bachelor of Technology

COMPUTER SCIENCE AND ENGINEERING

By

Ankita Sood (181393)

Under the supervision of

Hamsa Maravanthe Rama
Application Engineer

to

Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology Waknaghat, Solan-173234, Himachal Pradesh

CERTIFICATE

I hereby declare that the work presented in this report entitled **Load Testing for RBS Maestro team pipelines** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering/Information Technology** submitted in the department of Computer Science & Engineering and Information Technology**,** Jaypee University of Information Technology Waknaghat is an authentic record of my own work carried out over a period from Feb 22 to June 2022 under the supervision of Hamsa Maravanthe Rama (Application Engineer)

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Ankita Sood (181393)

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

Ekta

Ekta Gandotra
Assistant Professor(S.G)
Computer Science & Engineering and Information Technology
Jaypee University of Information Technology

ACKNOWLEDGEMENT

Firstly, I express my heartiest thanks and gratefulness to almighty God for His divine blessing makes it possible to start the project work successfully.

I am really grateful and wish our profound indebtedness to Supervisor Hamsa Maravanthe Rama (Application Engineer). Deep Knowledge & keen interest of my supervisor in the subject helped a lot in order to carry out this project. Her endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts and correcting them at all stages have made it possible to complete this project.
I would like to express our heartiest gratitude to Hamsa Maravanthe Rama (Application Engineer), for her kind help to finish this project.

We would also generously welcome each one of those individuals who have helped me straightforwardly or in a roundabout way in making this project a win. In this unique situation, We might want to thank the various staff individuals, both educating and non-instructing, which have developed their convenient help and facilitated my undertaking. Finally, we must acknowledge with due respect the constant support and patients of our parents.

Ankita Sood (181393)

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

ABBREVIATION	**WORD**
PT	PERFORMANCE TESTING
ST	SOFTWARE TESTING
FIG	FIGURE
LT	LOAD TESTING
DESCR	DESCRIPTION
CSE	COMPUTER SCIENCE AND ENGINEERING

ABSTRACT

PT is a process of testing software for the different things like testing its speed, response time, throughput, stability, reliability, and scalability under a particular workload. . In simple words testing response and stability of application under load is PT. The major reason why one needs to do performance testing is to avoid any kind of problem which may appear in the future. The main idea of PT is to identify and remove the performance bottlenecks in the application.

ST is important because if there are any kind of errors in the software, they can be identified early and can be solved before the delivery of the software product. Properly tested software product ensures reliability, security, and high performance which further results in time-saving, cost-effectiveness, and customer satisfaction. So before deployment of any application, software testing is done.

Chapter 1

INTRODUCTION

1.1 INTRODUCTION

PT is nothing but testing the stability and response time of an application by applying load on it. This type of testing ensures whether application or software is working well under peak conditions or not .PT is done to provide information about the application regarding stability, speed, and scalability. Not only this , it also helps to know what areas of it need improvement before launching it in the market.

Performance Testing Attributes are-

- Speed
- Scalability
- Stability
- Reliability

Speed – How fast the application works

Scalability – Maximum user load the application can handle

Stability- To check whether application remains stable even after load

Reliability – Whether its okay to launch the application in market

Black Box Testing :

It is a software testing procedure in which we test the software applications without having the knowledge of the internal code structure , internal paths or implementation techniques.It is primarily focused on input and output of the applications and is also known as behavioral testing.

Input
Black Box
Output

Fig 1.1 Represents input/output flow

There are few types of black box testing :

Functional Testing

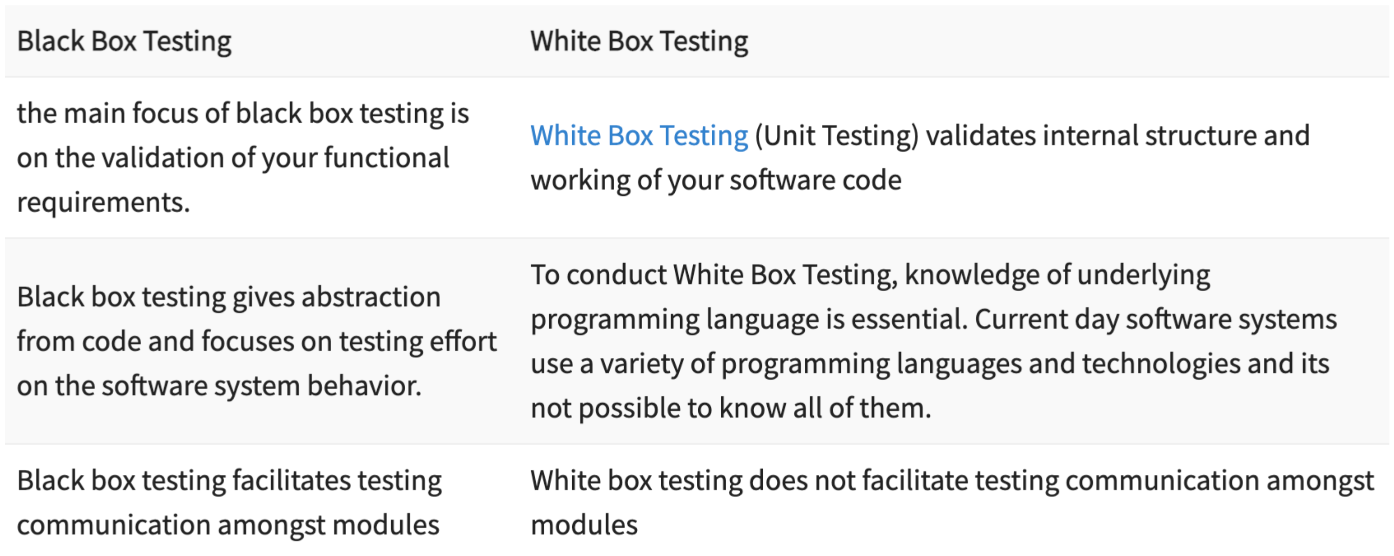
This Type of Black box testing is related to the functional requirements of the software.It is usually done by software testers

Non Functional Testing

It refers to non functional requirements such as speed , scalability and usability.

Regression Testing

This type of testing is done after code fixes to check the new cose has not affected the existing code.



Black Box Testing	White Box Testing
the main focus of black box testing is on the validation of your functional requirements.	White Box Testing (Unit Testing) validates internal structure and working of your software code
Black box testing gives abstraction from code and focuses on testing effort on the software system behavior.	To conduct White Box Testing, knowledge of underlying programming language is essential. Current day software systems use a variety of programming languages and technologies and its not possible to know all of them.
Black box testing facilitates testing communication amongst modules	White box testing does not facilitate testing communication amongst modules

Table 1.1- Difference between black and white box testing

The major objective of performance testing is to eliminate performance congestion .

Different types of performance testing include

- Load testing
- Stress testing
- Endurance testing
- Spike testing
- Volume testing

Load Testing :

Load testing is a kind of non functional performance testing in which performance of an application is test under a specific load. It aims to measure how the application will behave if multiple users starts using the application simultaneously.Load testing enables you to measure your website/applications quality of the service or in other words performance that is based on the actual customer interaction and behavior.Load testing is usually conducted in a test environment identical to the production environment before the software is permitted to go live.

Stress Testing:

It involves testing an application under extreme load.The main idea of stress testing to find the break point for the application.It involves testing the application beyond normal operational capacity in order to observe any of the results :

1) To determine the safeusage limits of the browser/application
2) To confirm that the app/browser meeting the intended specifications
3) To determine the modes of failure

Endurance Testing:

It measures how long the application can bear a given load.It is a non-functional kind of testing where a software is tested with high load that is extended for a significant amount of time to evaluate the behavior of the given application under sustained use.The main purpose of this testing is to ensure that the application is able to handle the extended load and there is no lag of the response time or any performance issues.

Spike Testing :

It measures how the application will behave in when a sudden spike of load is given.The goal of the spike testing is to determine the behavior of the application when it faces extreme variations in traffic.Spike testing not only determines the maximum load an application can bear but also can determine applications recovery time between the spike.Even the spike itself mean the sudden increase or decrease.

Volume Testing:

Under this kind of testing large volume of data is populated in the databases to check the behaviour of application.It is also known as flood testing.It helps to analyze the system performance after increasing the volume.Some of the advantages include:

1)Assures that the system is capable of handling real world.

2)Early identification of bottlenecks

3)A lot of money can be saved , as this money will be used in maintenance of the application

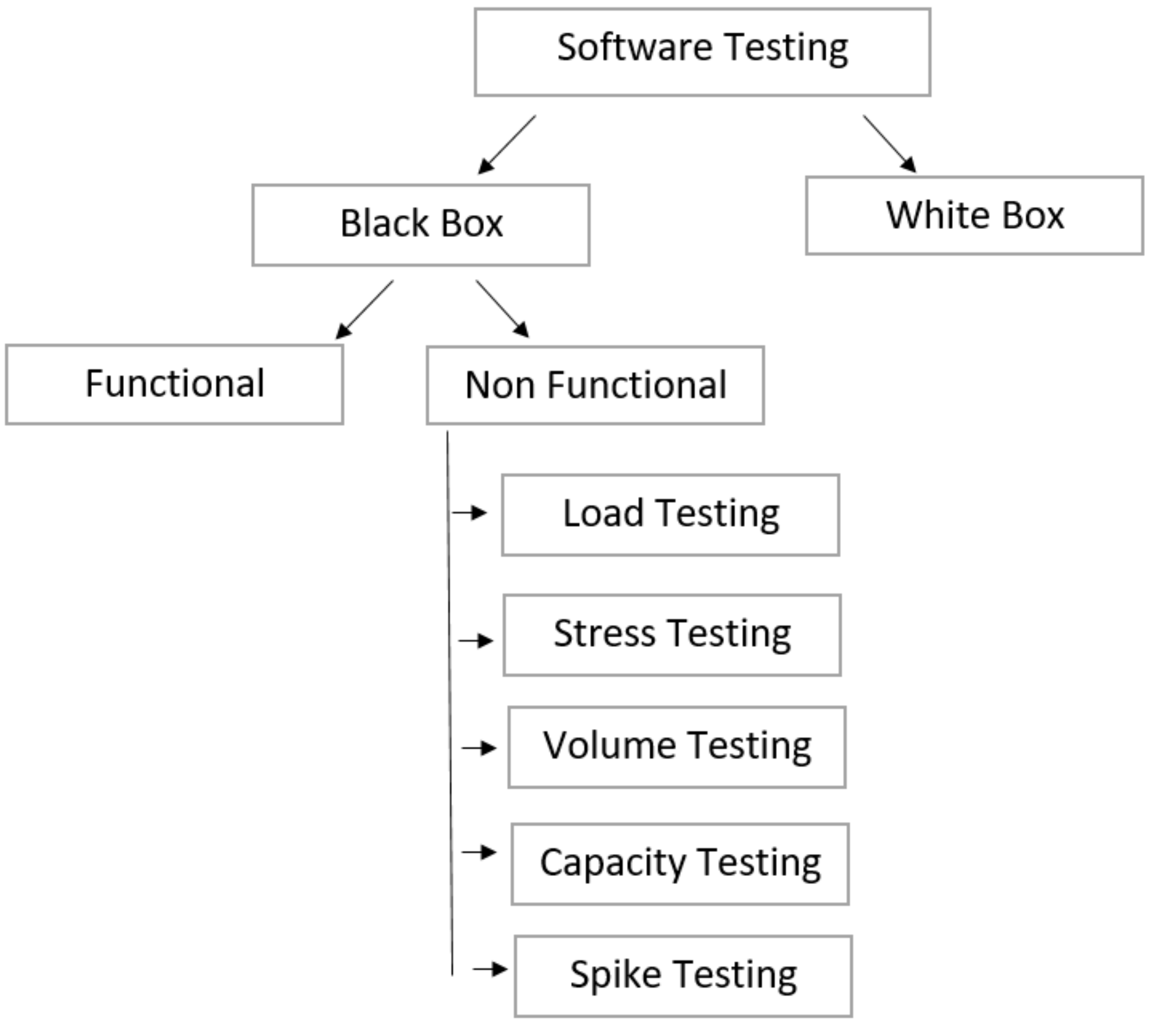


Fig 1.2: Types of performance testing

Performance testing process-

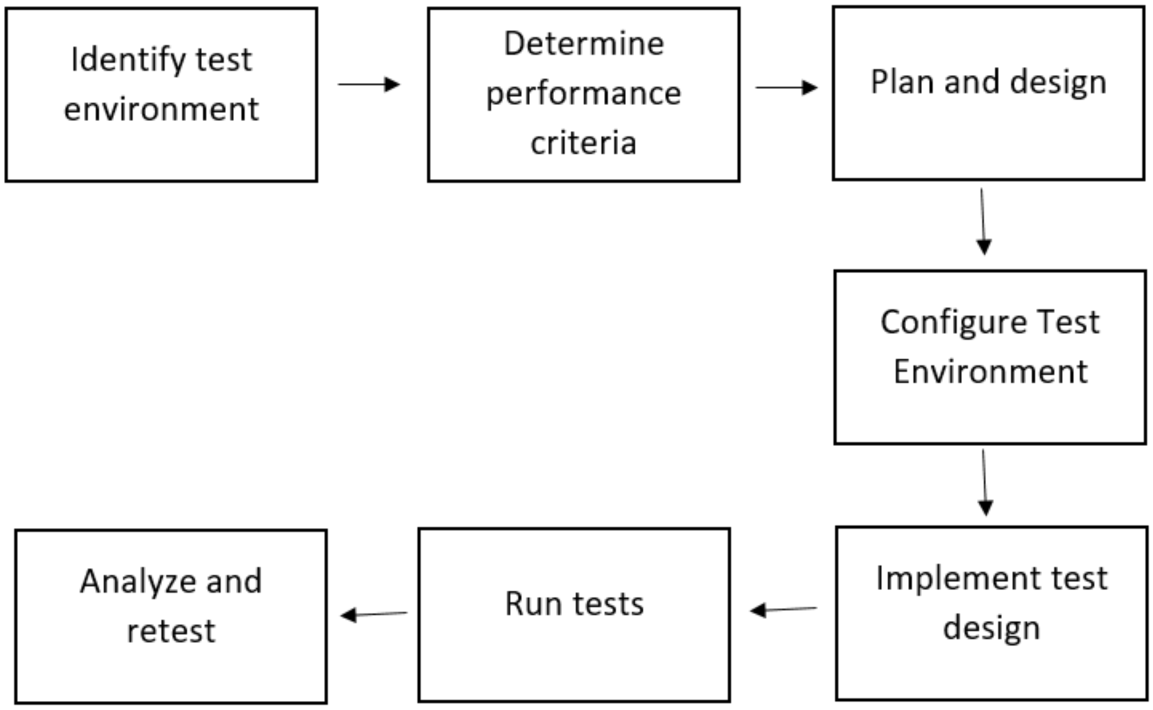

Identify test environment
Determine performance criteria
Plan and design
Configure Test Environment
Implement test design
Run tests
Analyze and retest

Fig 1.3: Process of performance testing

Identify The Test Environment:

First of all we will need to determine the physical test environment , production environment and testing tools that are available with us .Understanding the details of the software , it will then eventually help the testers to create a lot more efficient use cases and also help them identify any problem they might encounter during performance Testing.

Determine the Performance Acceptance Criteria :

This will include the goal and constraints for the throughput, and response time and resource allocation.Testers should be empowered to set performance ctriteria outside of these goals.

Plan and Design :

In this step we need to determine the variety of usage among the end users and identify the key scenarios in order to cover a wide variety of test cases.It is necessary to simulate variety of end users so that data can be gathered .

Configuring Test Environment:

Prepare the test environment and all the tools that are required before execution.

Implement Test Design :

Create performance test according to the test design.

Run The test :

Execute and monitor all the tests.

Analyze and Retest:

Consolidate and analyze the test results.Then fine tune and restest to see if there is an improvement or decrease in performance.

1.2 PROBLEM STATEMENT

Before deploying any kind of software in market it is of utmost importance to check whether the application or software you are deploying is in position to handle huge load ie large number of clients using the same application /software . Will the application perform well in such peak situations ? Will it be able to match the speed ? The answer to these questions are very important otherwise a slow running application will lose potential users

Why load testing is an important part of the Software Development lifecycle?

- It simulates real user scenarios.
- It evaluates how the performance of an application can be affected by normal and peak loads.
- It helps save money by identifying bottlenecks and defects on time.

1.3 OBJECTIVES

Objectives of this project are:

- To allow more user to access the application at the same time.
- To determine what how the application behave in peak condition.
- To determine the count of users who can access the application at a single time.
- To increase the operating capacity of a software application

1.4 METHODOLOGY

Figure 1.3 represents the load testing process .Following are the steps which one needs to follow while performing the load testing .

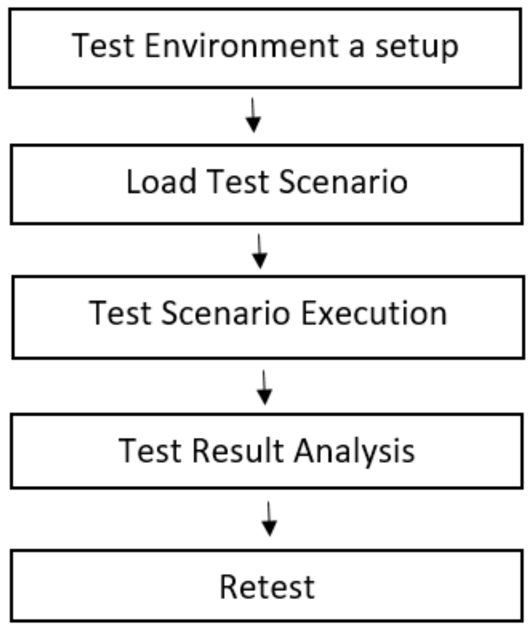


Fig 1.4: Process of load testing

Explanation of the process of load testing in detail-

1. **Test Environment Setup-**
   The very first step is to create a dedicated test environment setup in order to perform the load testing .The setup ensures whether the load testing happens in a proper way.Setting up an environment is a very important task as this is the environment which you will be using for running the load testing

2. **Load Test Scenario-**

   In this step we determine the test cases which we need to run again and again for example the transactions which we need to perform again and again also set the initializer in the same step only which could act as a constructor.In this step we also decide for the terminator for eg the cleanup that is required after the load testing is done.

   In this step we also decide the parameters which we will set up in the approval step for eg the transaction per second , how to distribute between the transactions all these things are decided in this step.

3. **Test Scenario Execution-**

   Load test scenarios that were generated in the previous step are known executed in this step. Different measurements and methods are gathered to collect the information.

4. **Test Result Analysis-**

   Results of the testing performed is analysed. Different recommendations are proposed based on the result gained. We gain insight about the performance of the application through a dashboard where we get information like cpu utilization , memory used at what transaction the application will start to fail etc.

5. **Re-test-**

   If any test case defined is failed then the test is performed again in order to get the result in correct way.

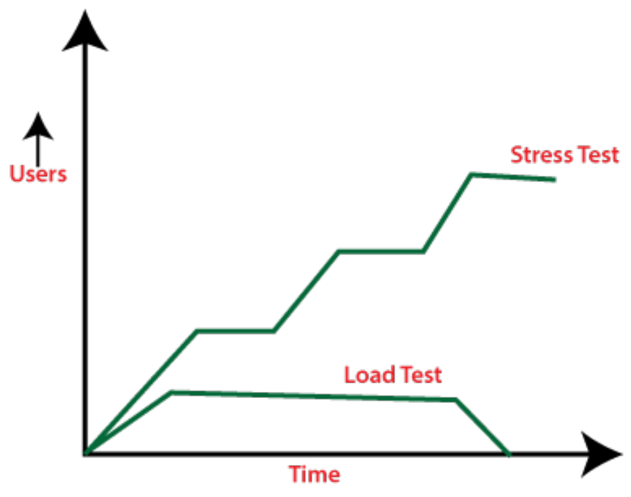


Fig 1.5: Stress Testing vs Load Testing

The above figure represents the difference between the stress and load testing , with load testing on one hand an engineer can detect the bottleneck , what is the reason behind bottleneck before deploying the application to production whereas in case of stress testing an engineer can test the capacity of the application at time of huge users interacting with the website.

Types of load testing tools –

- Load testing tool (Open source)
- Manual Load testing tool
- Load testing tool (Enterprise)
- In house developed Load testing tools

Most commonly used load testing tool used in industry are-

- LoadNinja
- Apache JMeter
- NeoLoad
- HP Performance Tester
- WebLoad
- LoadView

Pipeline :A pipeline for any software engineering team can be defined as a set of automated processes that allows Developers and DevOps professional to efficiently and reliably compile, build, and deploy their code to their production compute platforms.
There are no such rules stating how should a pipeline should look or behave or the tools it must utilise. A pipeline consists of a chain of elements like processes, threads , functions which are arranged in such a way so that output of each element is the input of the next .Any kind of information can flow through these pipelines it can be a stream of records , bytes or bits or can depend upon the its usage.

Pipeline is generally linear and one directional though sometime the term is applied to more general flow.

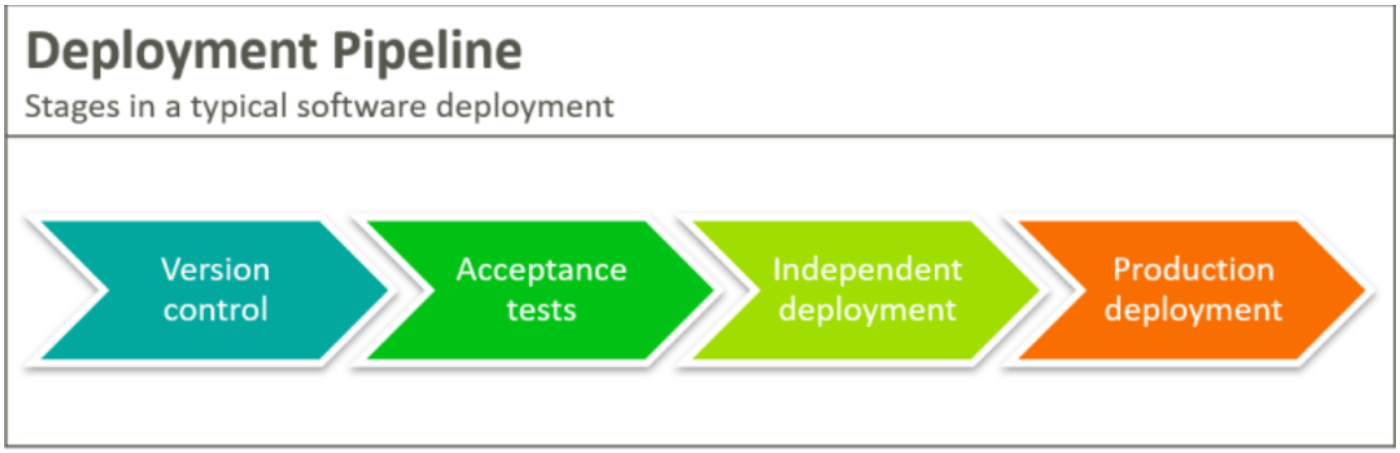


Fig 1.6: Stages in software deployment

Version control- Software developers after writing a code commit their changes into source control.When they commit the code changes to the source control the first stage of deployment pipeline begins which triggers –

- Code compilation
- Unit tests
- Code analysis
- Installer creation

In version set the dry-run build should build successfully otherwise the flow would not flow for the next steps in pipeline.

In version set step various other approval steps can be added for eg code review verification which would fail if the cr is not raised and one push the code .

Acceptance tests – It is a process of running some test over the code that has been pushed to check against some predefined acceptance rule set by the organisation.

Independent Deployment- Independent deployment is the process of deploying the tested and compiled artefacts into the development environments.

Production Deployment – This step is very similar to previous step it should deliver the code to live production server.

CHAPTER 2

TOOLS AND TECHNOLOGY USED

2.1. TOOLS

Here in the organisation we have various tools for performing the load testing .But we majorly focus on two tools for doing different type of testing not only limited to load testing.
These two tools names are-

- TPS Generator
- Hydra

TPS Generator is the one which is fully build and is applied for different types of testing whereas on the other hand we have Hydra which still is in beta stage and changes are still made to it depending upon the requirements

So for our load testing purpose we are using TPS Genarator right now and hydra will be used in future.
Hydra is build from learning the mistakes of TPS Generator.The major advantage of using hydra over TPS Generator is that we don't have to write any code .It will automate the various processes itself.

In our pipelines which are a set of automated processes we write and create code in java language and also for the tps generator we need to make changes into our java code to make it run.

A brief description about the tools that we use here-

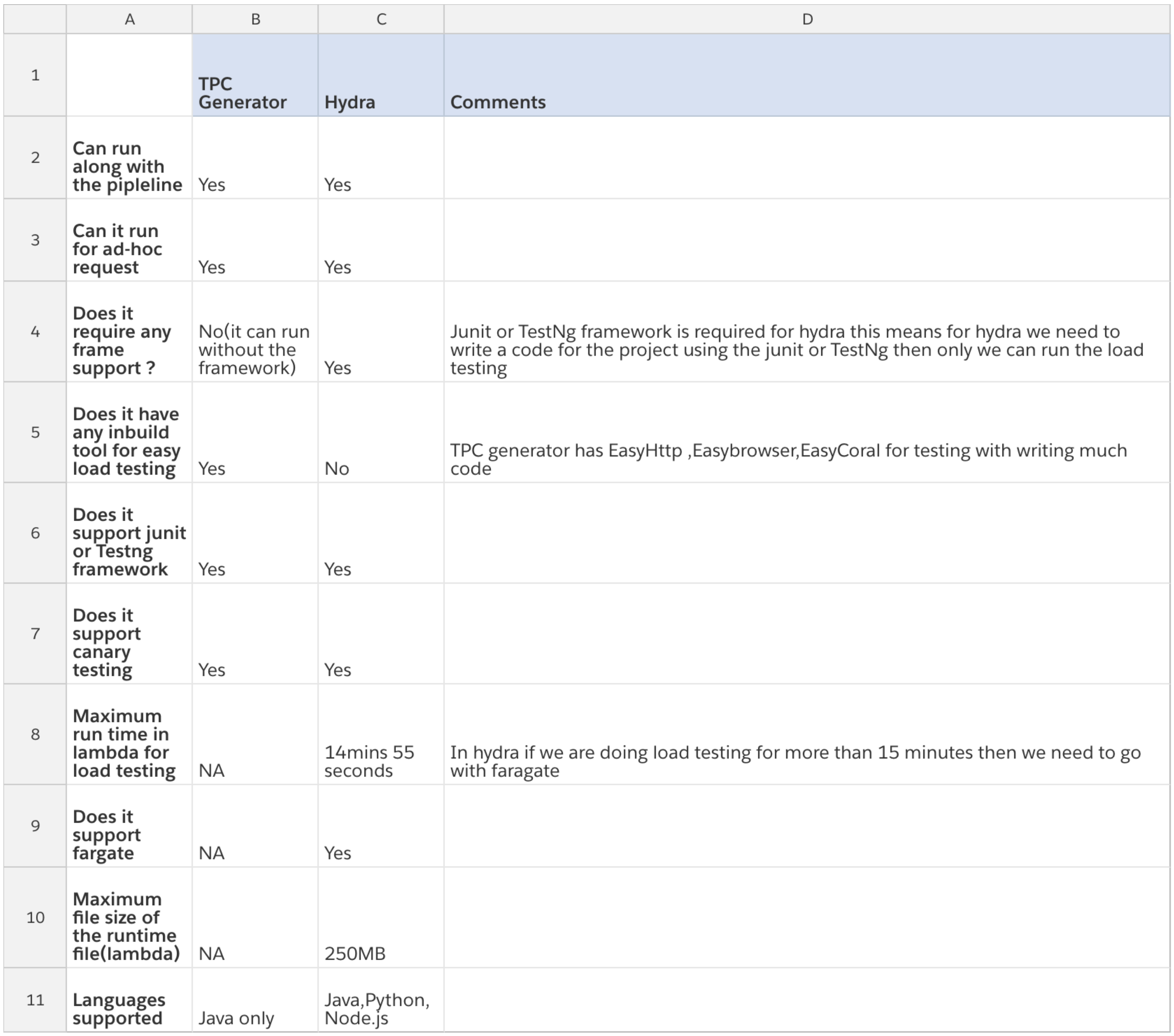
TPS Generator is the Builder Tools-supported, funded, Amazon-internal load generator tool, used widely across Amazon. It runs about 150,000 times per week as an approval step in Pipelines, for one-off adhoc load tests, or for large-scale gamedays. TPS Generator generates synthetic traffic against a service (or to replay prod traffic), to understand how the service scales, to validate that it can withstand peak traffic, and to ensure it degrades gradually at throughput higher than expected.

Hydra Test Platform ("Hydra") is a serverless test orchestration platform for Native AWS services. It automatically configures your test run infrastructure, orchestrates your test runs and reports your test results. If you have a Native AWS service and some tests that you want run as an approval step in Pipelines, Hydra is the perfect solution for your team.

Onboard to TPS generator –
For performing the load testing using the amazon internal tool tps generator one will need the following –

- A laptop/desktop
- Eclipse or any other ide
- An endpoint you want to load test
- Java code to hit your endpoint (Integration code)
- A material set or IAM role for providing the credentials

Points one need to take into consideration while choosing between the TPS generator or hydra for load testing.



	A	B	C	D
1		**TPC Generator**	**Hydra**	**Comments**
2	**Can run along with the pipleline**	Yes	Yes	
3	**Can it run for ad-hoc request**	Yes	Yes	
4	**Does it require any frame support ?**	No(it can run without the framework)	Yes	Junit or TestNg framework is required for hydra this means for hydra we need to write a code for the project using the junit or TestNg then only we can run the load testing
5	**Does it have any inbuild tool for easy load testing**	Yes	No	TPC generator has EasyHttp ,Easybrowser,EasyCoral for testing with writing much code
6	**Does it support junit or Testng framework**	Yes	Yes	
7	**Does it support canary testing**	Yes	Yes	
8	**Maximum run time in lambda for load testing**	NA	14mins 55 seconds	In hydra if we are doing load testing for more than 15 minutes then we need to go with faragate
9	**Does it support fargate**	NA	Yes	
10	**Maximum file size of the runtime file(lambda)**	NA	250MB	
11	**Languages supported**	Java only	Java,Python, Node.js	

Table 2.1 Difference between TPS Generator and Hydra

CHAPTER 3

SYSTEM DEVELOPMENT

3.1 HOW LOAD TESTING IS ACHIEVED ?

3.1.1 ENVIRONMENT SETUP

The very first step for any application/software to develop is to create a proper or right kind of environment .Before writing any code we need to know how we can setup a pipeline - Pipelines is a continuous deployment tool you can use to model, visualize, and automate the steps required to release your software. It provides both a web interface and API to give you the ability to quickly design and configure the different stages of your release process.
So in order to create a pipeline and start jumping to write a code need to follow steps given in the amazon documentation.

Once the pipeline is set the developer is now ready to write the code.So here in this step we determine which software will be used for writing the code , what packages do we require , which language one should use for creating the software. So here at the organisation most the applications are built using the java language and the text editor used is IntelliJ. We also need to make sure we also create a cloud environment for making code changes eventually from local to cloud.

After checking for the necessary packages installed the software developer job is to write code and create the application or software .Once this process is done and build is successful the changes are committed and can be seen in the pipelines.

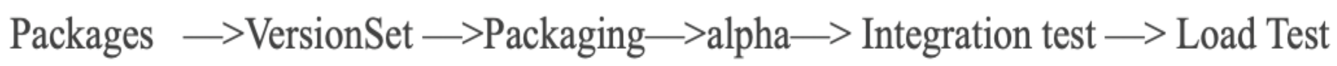

Packages —>VersionSet —>Packaging—>alpha—> Integration test —> Load Test

Fig 3.1 Given figure represent the general flow of pipline

3.1.2 CHANGES IN THE CODE

The very next step after creating a pipeline is to write a code which can eventually help us in doing the load testing.In our case we used integration test code which we can use for load testing .So the very initial step was to download the code from the package where integration test code is written so that some modification can be done on that integration test code to make it as a load test code efficient.
In order to download the code from a package these commands are required –

Using the terminal open the folder where you want to have that code.

```
brazil ws --create --name <your workspace name> --root <workspace path>
```

Figure 3.2 Represents how to download the code

After creating the root directory the next step is to importing the version set of the project.

```
brazil ws use --versionset <version set name>
```

Figure 3.3 Represents how to import the version set

The last step required in the step for downloading the code you need to change is to import the package you want to use .

```
brazil ws use --package <package name> –branch <branch name>
```

Figure 3.4 Represents how to import the package

Once you follow these three simple command one can now open the code in their local setup.

Write Product-Specific Code with Annotations for TPS generator –

If you decide TPSGenerator is for you, the first thing you need to do is implement a Java class with special annotations that will tell us how to talk to your service. Since we're a generic platform, you provide us with the product-specific knowledge and we do the rest.

Things we need to take care in mind while writing the code –

- Decide what Brazil package your load gen will live in.
- Add a dependency to our interface package to your Brazil package
- Write code to interact with your product and annotate it with the annotations
- Run your load gen

Example code with the annotations -

```
@TPSGenerator( "MyTeamLoadGenerator" )

public class myTeamLoadGenerator {

// Have code in here which you want to run again and again

// You can add in here anything ,any functionality which u think your application
require

// Keep in mind your transactions may be called from potentially hundreds of threads.

  }
```

Figure 3.5 Represents the name of load generator

```
 @TPSGeneratorInitialize
  public void initialize( String[] args ) throws Exception {

 // The initializer , the setup required by your code

  }
```

Figure 3.6 Represents the annotation for setup method

```
@TPSGeneratorTerminate
  public void terminate() {

   // this code for the cleanup part

  }
```

Figure 3.7 Represents the annotation for cleanup method

We need to select some functions in the code as initializers which can act as a constructor .The setup which is required for rest of the functions to implement that function we can select as a initializer.
We need to write something like this -

```
@TPSGeneratorInitialize
```

We need to select some functions in the code for multiple transactions the processes which will run multiple times. This can be those functionality in the code which are required by the application again and again
We need to write something like this -

```
@TPSGeneratorTransaction( "TransactionName" )
```

We also have annotation for the final cleanup method , for eg the database which we created needs to be cleaned after the load test are run
We need to write something like this -

```
@TPSGeneratorTerminate
```

Once the suitable changes are made into the code the process does not end here.There are other things which need to be done.

CREATION OF CHILD WORKFLOW

Other than making changes in the code we also need to create different child environment using apollo.There are various steps which one need to follow for the process.

Since the load test require an environment on which it can run so we need to create TOD test on demand child workflow which will provide the environment setup for the load testing to run.

For creating the apollo one click child environment the steps one need to follow are –

Visit - https://apollo.amazon.com/homepage.html

Click on Create one click child environment for creating the child environment .

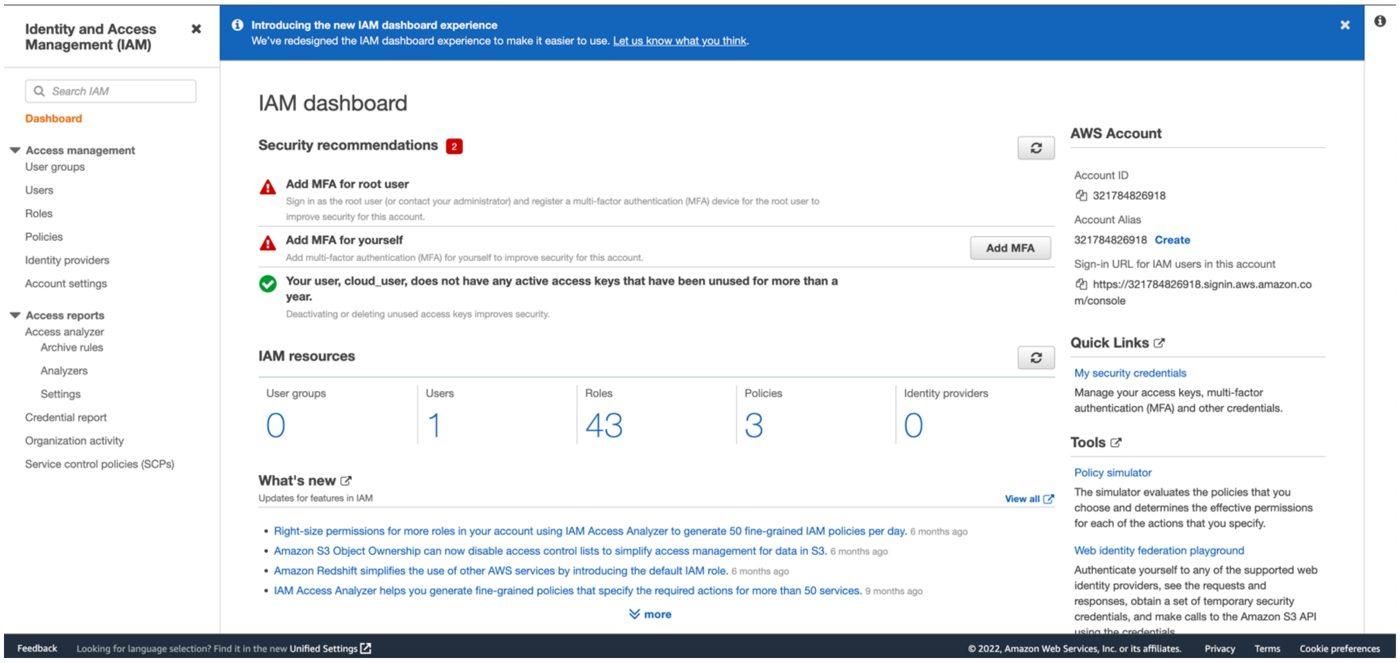
This will provide the environment for running the load test code.

CREATION OF CUSTOM TOD WORK ENVIRONMENT

There are some authentication issues when one works with the tps generator so in order to resolve this issue we need to make a custom TOD worker environment and use credentials provided by that worker environment

After creation of custom tod worker environment one need to create an IAM role using AWS (Amazon web service) in which we will add custom permission which will allow the custom tod worker to access the credentials and run the code.

Steps for creation of aws iam role -



Identity and Access Management (IAM)

Introducing the new IAM dashboard experience
We've redesigned the IAM dashboard experience to make it easier to use. Let us know what you think.

IAM dashboard

Security recommendations 2

Add MFA for root user
Sign in as the root user (or contact your administrator) and register a multi-factor authentication (MFA) device for the root user to improve security for this account.

Add MFA for yourself
Add multi-factor authentication (MFA) for yourself to improve security for this account.

Your user, cloud_user, does not have any active access keys that have been unused for more than a year.
Deactivating or deleting unused access keys improves security.

IAM resources

User groups	Users	Roles	Policies	Identity providers
0	1	43	3	0

What's new
Updates for features in IAM

- Right-size permissions for more roles in your account using IAM Access Analyzer to generate 50 fine-grained IAM policies per day. 6 months ago
- Amazon S3 Object Ownership can now disable access control lists to simplify access management for data in S3. 6 months ago
- Amazon Redshift simplifies the use of other AWS services by introducing the default IAM role. 6 months ago
- IAM Access Analyzer helps you generate fine-grained policies that specify the required actions for more than 50 services. 9 months ago

AWS Account

Account ID
321784826918

Account Alias
321784826918 Create

Sign-in URL for IAM users in this account
https://321784826918.signin.aws.amazon.com/console

Quick Links

My security credentials
Manage your access keys, multi-factor authentication (MFA) and other credentials.

Tools

Policy simulator
The simulator evaluates the policies that you choose and determines the effective permissions for each of the actions that you specify.

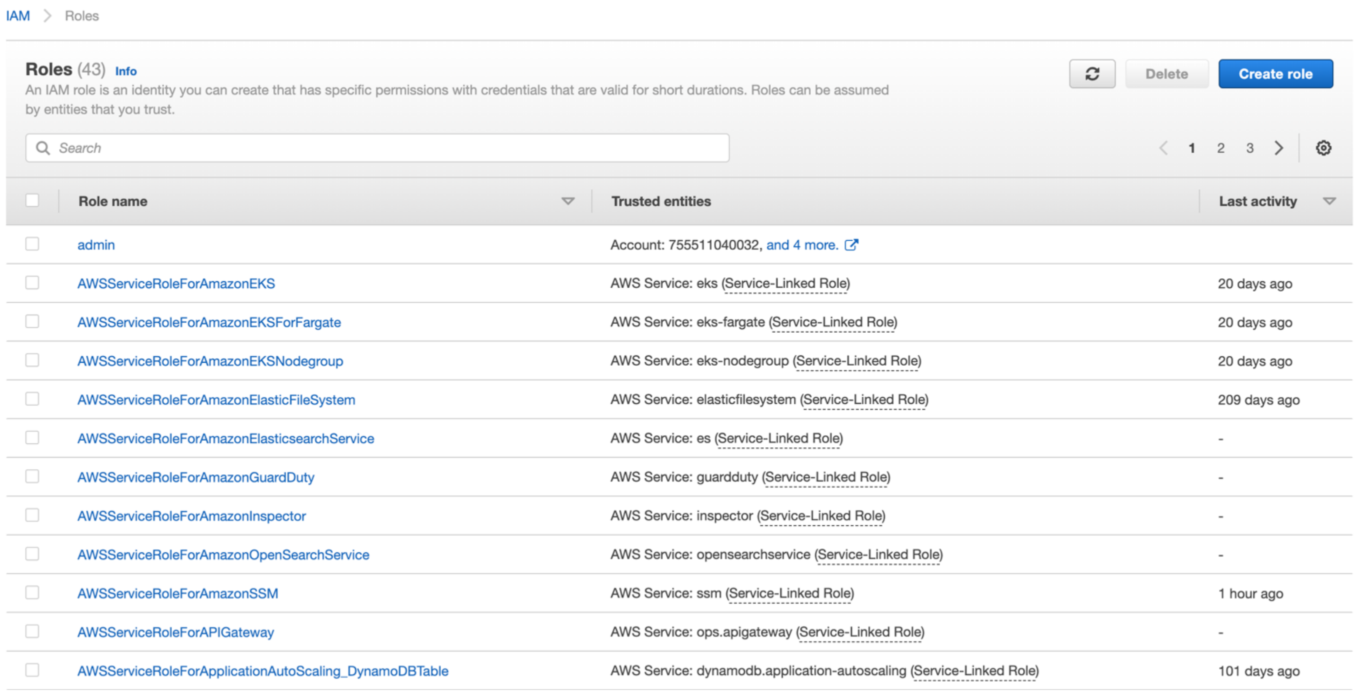
Web identity federation playground
Authenticate yourself to any of the supported web identity providers, see the requests and responses, obtain a set of temporary security credentials, and make calls to the Amazon S3 API using the credentials.

Looking for language selection? Find it in the new Unified Settings

© 2022, Amazon Web Services, Inc. or its affiliates.

Figure 3.8 Represents the AWS IAM role service

- Login into the AWS Account and search for IAM service in search bar
- Click on roles in the left sidebar

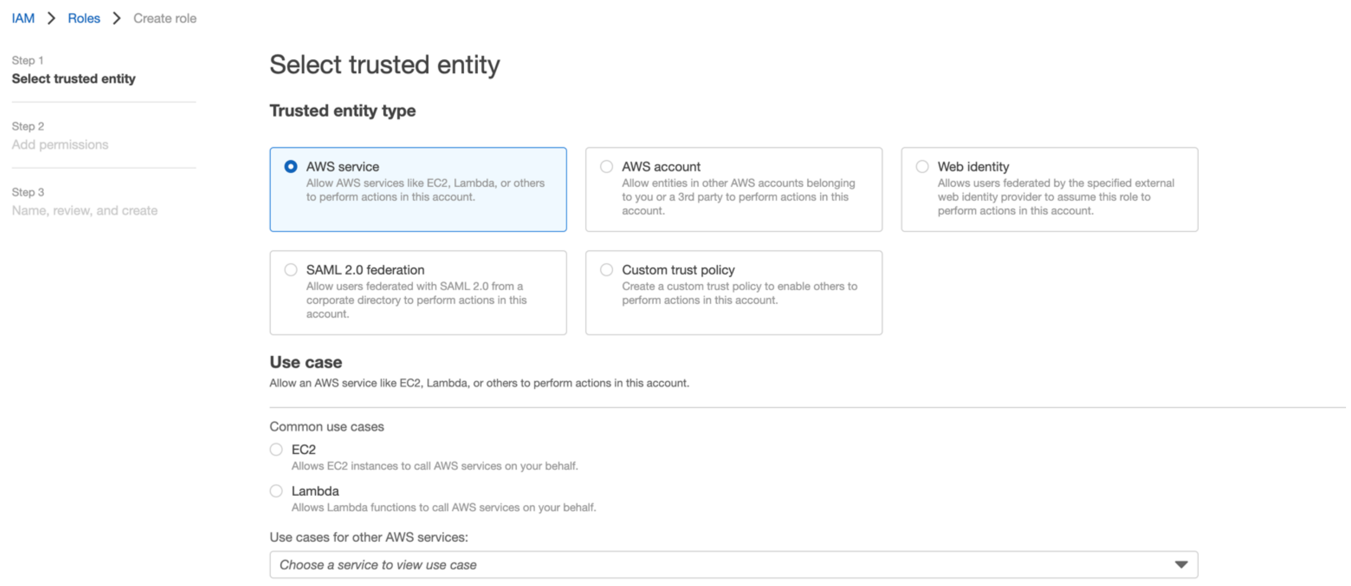


Roles (43) Info

An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.

Role name	Trusted entities	Last activity
admin	Account: 755511040032, and 4 more.	
AWSServiceRoleForAmazonEKS	AWS Service: eks (Service-Linked Role)	20 days ago
AWSServiceRoleForAmazonEKSForFargate	AWS Service: eks-fargate (Service-Linked Role)	20 days ago
AWSServiceRoleForAmazonEKSNodegroup	AWS Service: eks-nodegroup (Service-Linked Role)	20 days ago
AWSServiceRoleForAmazonElasticFileSystem	AWS Service: elasticfilesystem (Service-Linked Role)	209 days ago
AWSServiceRoleForAmazonElasticsearchService	AWS Service: es (Service-Linked Role)	-
AWSServiceRoleForAmazonGuardDuty	AWS Service: guardduty (Service-Linked Role)	-
AWSServiceRoleForAmazonInspector	AWS Service: inspector (Service-Linked Role)	-
AWSServiceRoleForAmazonOpenSearchService	AWS Service: opensearchservice (Service-Linked Role)	-
AWSServiceRoleForAmazonSSM	AWS Service: ssm (Service-Linked Role)	1 hour ago
AWSServiceRoleForAPIGateway	AWS Service: ops.apigateway (Service-Linked Role)	-
AWSServiceRoleForApplicationAutoScaling_DynamoDBTable	AWS Service: dynamodb.application-autoscaling (Service-Linked Role)	101 days ago

- Click on create role


IAM > Roles > Create role

Step 1
Select trusted entity

Step 2
Add permissions

Step 3
Name, review, and create

Select trusted entity

Trusted entity type

(●) AWS service
Allow AWS services like EC2, Lambda, or others to perform actions in this account.

( ) AWS account
Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.

( ) Web identity
Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.

( ) SAML 2.0 federation
Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.

( ) Custom trust policy
Create a custom trust policy to enable others to perform actions in this account.

Use case
Allow an AWS service like EC2, Lambda, or others to perform actions in this account.

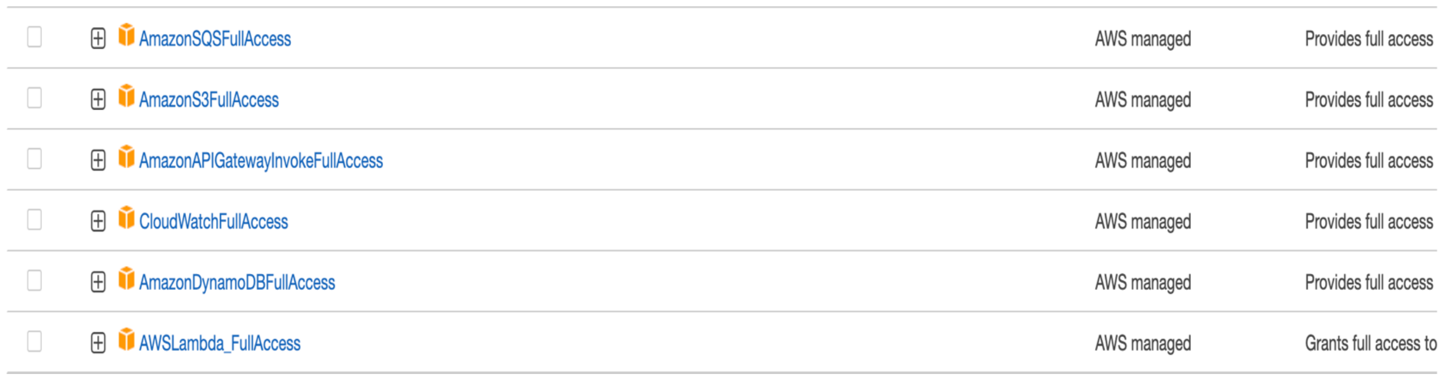
Common use cases
( ) EC2
Allows EC2 instances to call AWS services on your behalf.
( ) Lambda
Allows Lambda functions to call AWS services on your behalf.

Use cases for other AWS services:
Choose a service to view use case

Figure 3.9 Represents the selecting of trusted entity

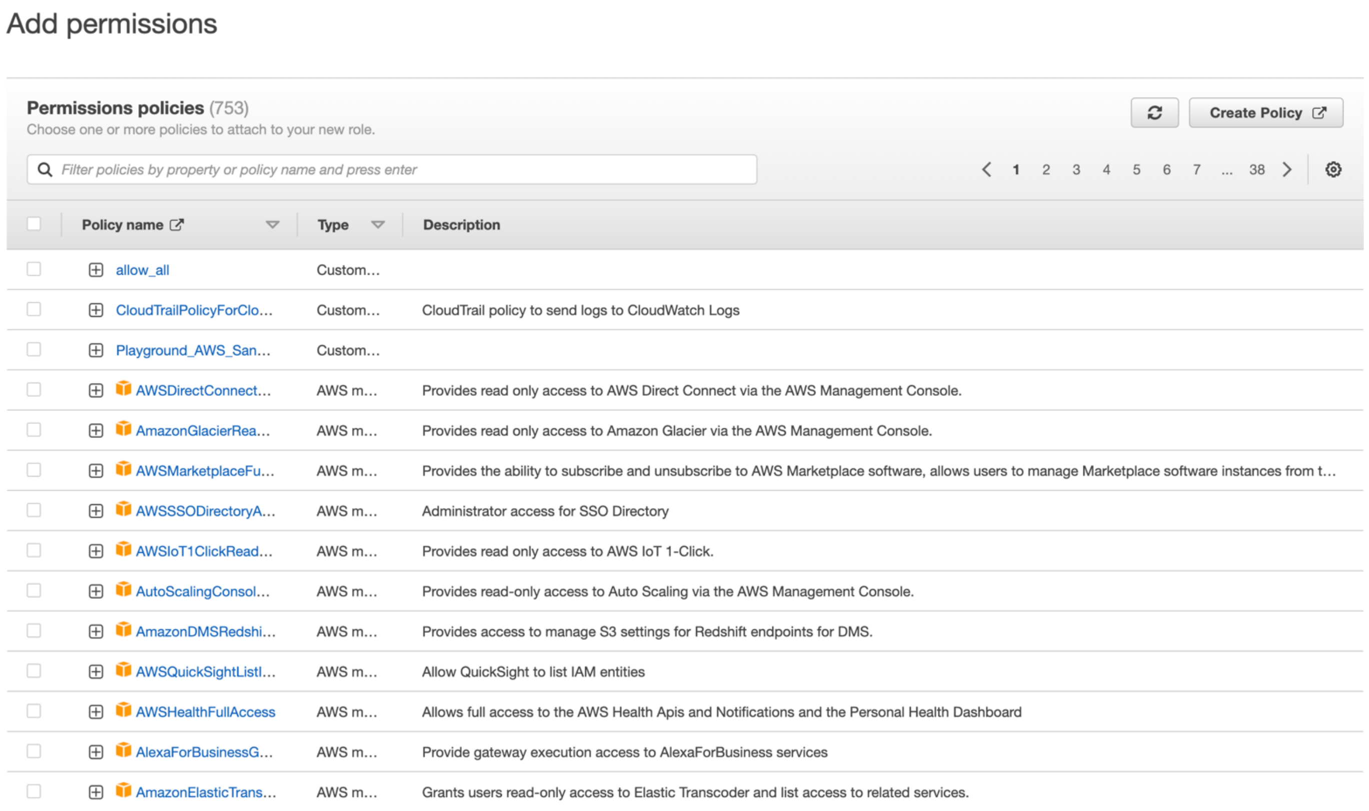
After selecting the trusted entity click the next and add the permission policy based on the service you are interacting in your service or application.

Most common trust policy which one can add are –



AmazonSQSFullAccess	AWS managed	Provides full access
AmazonS3FullAccess	AWS managed	Provides full access
AmazonAPIGatewayInvokeFullAccess	AWS managed	Provides full access
CloudWatchFullAccess	AWS managed	Provides full access
AmazonDynamoDBFullAccess	AWS managed	Provides full access
AWSLambda_FullAccess	AWS managed	Grants full access to

Figure 3.10 Represents the most common permission policy



Add permissions

Permissions policies (753)

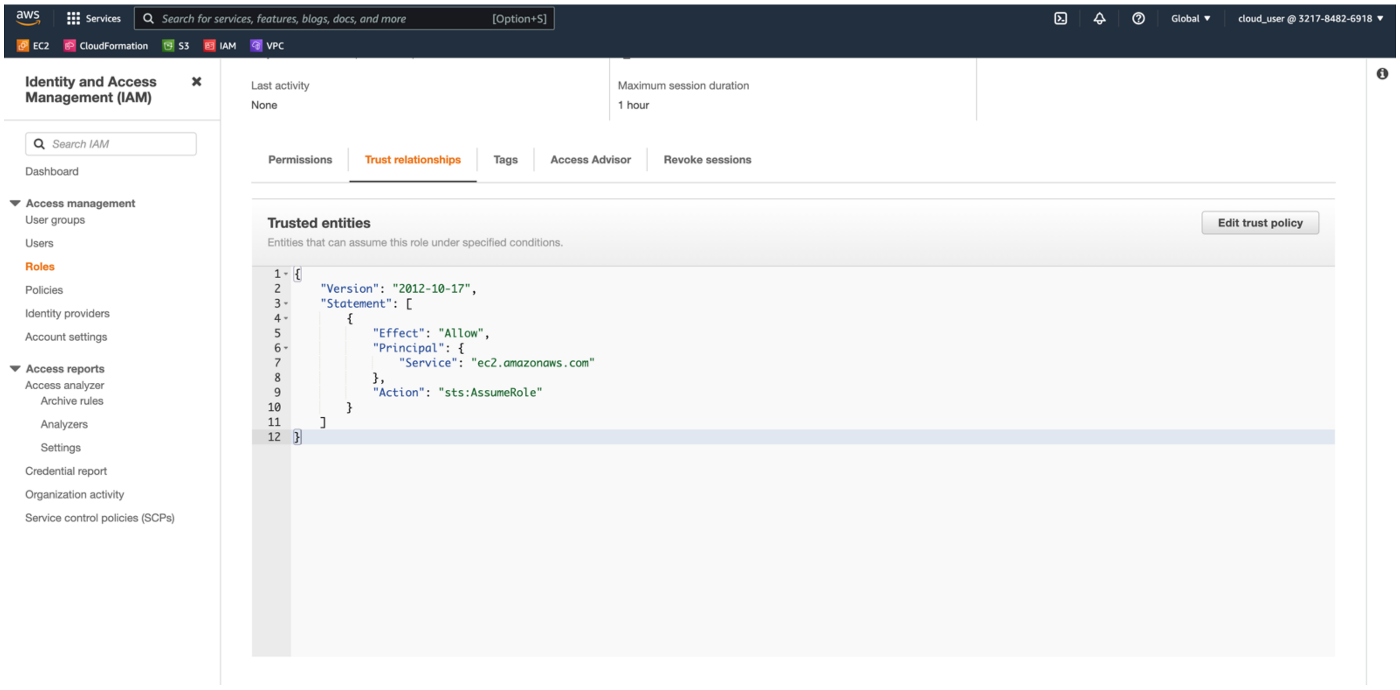
Choose one or more policies to attach to your new role.

	Policy name	Type	Description
	allow_all	Custom...	
	CloudTrailPolicyForClo...	Custom...	CloudTrail policy to send logs to CloudWatch Logs
	Playground_AWS_San...	Custom...	
	AWSDirectConnect...	AWS m...	Provides read only access to AWS Direct Connect via the AWS Management Console.
	AmazonGlacierRea...	AWS m...	Provides read only access to Amazon Glacier via the AWS Management Console.
	AWSMarketplaceFu...	AWS m...	Provides the ability to subscribe and unsubscribe to AWS Marketplace software, allows users to manage Marketplace software instances from t...
	AWSSSODirectoryA...	AWS m...	Administrator access for SSO Directory
	AWSIoT1ClickRead...	AWS m...	Provides read only access to AWS IoT 1-Click.
	AutoScalingConsol...	AWS m...	Provides read-only access to Auto Scaling via the AWS Management Console.
	AmazonDMSRedshi...	AWS m...	Provides access to manage S3 settings for Redshift endpoints for DMS.
	AWSQuickSightListI...	AWS m...	Allow QuickSight to list IAM entities
	AWSHealthFullAccess	AWS m...	Allows full access to the AWS Health Apis and Notifications and the Personal Health Dashboard
	AlexaForBusinessG...	AWS m...	Provide gateway execution access to AlexaForBusiness services
	AmazonElasticTrans...	AWS m...	Grants users read-only access to Elastic Transcoder and list access to related services.

Figure 3.11 Represents the permission policy

Search for the permission policy which you want to add into the role which totally depend upon the service you are doing load testing for.

Once the role is created we need to change the trust policy and make sure we add the tod worker we created in the trust policy



Last activity: None

Maximum session duration: 1 hour

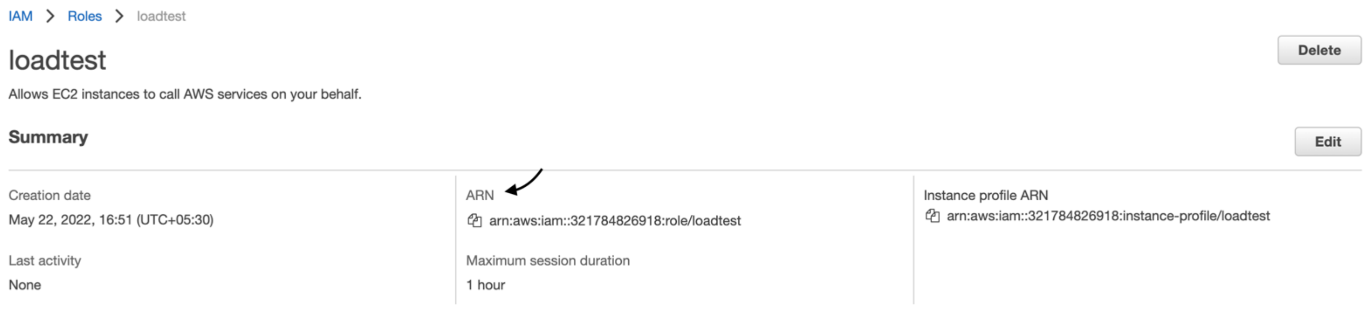
Trusted entities

Entities that can assume this role under specified conditions.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "ec2.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

Figure 3.12 Represents the Trust relationship

After the role is created we need to copy the role arn which we will be using in the load test approval step.



loadtest

Allows EC2 instances to call AWS services on your behalf.

Creation date: May 22, 2022, 16:51 (UTC+05:30)

Last activity: None

ARN: arn:aws:iam::321784826918:role/loadtest

Maximum session duration: 1 hour

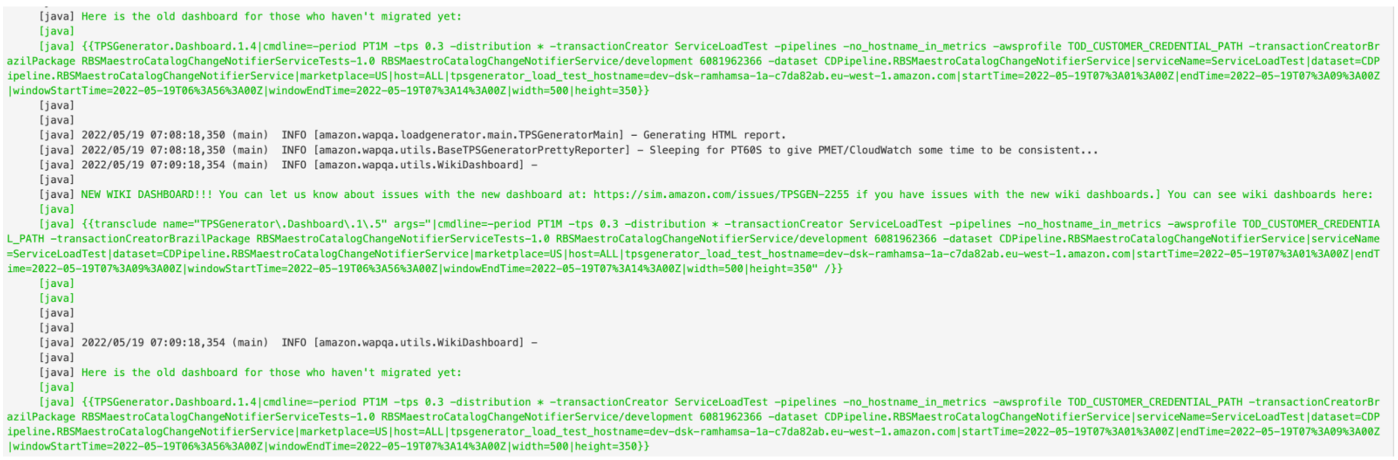
Instance profile ARN: arn:aws:iam::321784826918:instance-profile/loadtest

Figure 3.13 Represents the Role ARN

Once all the configurations have been done the last or final step is to add a load test approval step in the pipeline .
Where information like –

- Step Name
- Your load generator name
- TPS or Concurrent Connections
- Your load generator package OPTIONAL
- Period to run this test for OPTIONAL
- TodWorker Environment Stage OPTIONAL
- Distribution of transactions OPTIONAL
- Approval Heuristic to use OPTIONAL
- Additional command line args OPTIONAL
- Platform OPTIONAL
- Role ARN OPTIONAL
- Lock Other Environment Stages OPTIONAL
- Wait For Other Steps OPTIONAL*This step waits for 0 other steps to finish before it runs*

Once you fill out the values for the above value in the load test approval step.Now one can run the load test file to check the output of the load test



```
     [java] Here is the old dashboard for those who haven't migrated yet:
     [java]
     [java] {{TPSGenerator.Dashboard.1.4|cmdline=-period PT1M -tps 0.3 -distribution * -transactionCreator ServiceLoadTest -pipelines -no_hostname_in_metrics -awsprofile TOD_CUSTOMER_CREDENTIAL_PATH -transactionCreatorBr
azilPackage RBSMaestroCatalogChangeNotifierServiceTests-1.0 RBSMaestroCatalogChangeNotifierService/development 6081962366 -dataset CDPipeline.RBSMaestroCatalogChangeNotifierService|serviceName=ServiceLoadTest|dataset=CDP
ipeline.RBSMaestroCatalogChangeNotifierService|marketplace=US|host=ALL|tpsgenerator_load_test_hostname=dev-dsk-ramhamsa-1a-c7da82ab.eu-west-1.amazon.com|startTime=2022-05-19T07%3A01%3A00Z|endTime=2022-05-19T07%3A09%3A00Z
|windowStartTime=2022-05-19T06%3A56%3A00Z|windowEndTime=2022-05-19T07%3A14%3A00Z|width=500|height=350}}
     [java]
     [java]
     [java] 2022/05/19 07:08:18,350 (main)  INFO [amazon.wapqa.loadgenerator.main.TPSGeneratorMain] - Generating HTML report.
     [java] 2022/05/19 07:08:18,350 (main)  INFO [amazon.wapqa.utils.BaseTPSGeneratorPrettyReporter] - Sleeping for PT60S to give PMET/CloudWatch some time to be consistent...
     [java] 2022/05/19 07:09:18,354 (main)  INFO [amazon.wapqa.utils.WikiDashboard] -
     [java]
     [java] NEW WIKI DASHBOARD!!! You can let us know about issues with the new dashboard at: https://sim.amazon.com/issues/TPSGEN-2255 if you have issues with the new wiki dashboards.] You can see wiki dashboards here:
     [java]
     [java] {{transclude name="TPSGenerator\.Dashboard\.1\.5" args="|cmdline=-period PT1M -tps 0.3 -distribution * -transactionCreator ServiceLoadTest -pipelines -no_hostname_in_metrics -awsprofile TOD_CUSTOMER_CREDENTIA
L_PATH -transactionCreatorBrazilPackage RBSMaestroCatalogChangeNotifierServiceTests-1.0 RBSMaestroCatalogChangeNotifierService/development 6081962366 -dataset CDPipeline.RBSMaestroCatalogChangeNotifierService|serviceName
=ServiceLoadTest|dataset=CDPipeline.RBSMaestroCatalogChangeNotifierService|marketplace=US|host=ALL|tpsgenerator_load_test_hostname=dev-dsk-ramhamsa-1a-c7da82ab.eu-west-1.amazon.com|startTime=2022-05-19T07%3A01%3A00Z|endT
ime=2022-05-19T07%3A09%3A00Z|windowStartTime=2022-05-19T06%3A56%3A00Z|windowEndTime=2022-05-19T07%3A14%3A00Z|width=500|height=350" /}}
     [java]
     [java]
     [java]
     [java]
     [java] 2022/05/19 07:09:18,354 (main)  INFO [amazon.wapqa.utils.WikiDashboard] -
     [java]
     [java] Here is the old dashboard for those who haven't migrated yet:
     [java]
     [java] {{TPSGenerator.Dashboard.1.4|cmdline=-period PT1M -tps 0.3 -distribution * -transactionCreator ServiceLoadTest -pipelines -no_hostname_in_metrics -awsprofile TOD_CUSTOMER_CREDENTIAL_PATH -transactionCreatorBr
azilPackage RBSMaestroCatalogChangeNotifierServiceTests-1.0 RBSMaestroCatalogChangeNotifierService/development 6081962366 -dataset CDPipeline.RBSMaestroCatalogChangeNotifierService|serviceName=ServiceLoadTest|dataset=CDP
ipeline.RBSMaestroCatalogChangeNotifierService|marketplace=US|host=ALL|tpsgenerator_load_test_hostname=dev-dsk-ramhamsa-1a-c7da82ab.eu-west-1.amazon.com|startTime=2022-05-19T07%3A01%3A00Z|endTime=2022-05-19T07%3A09%3A00Z
|windowStartTime=2022-05-19T06%3A56%3A00Z|windowEndTime=2022-05-19T07%3A14%3A00Z|width=500|height=350}}
```

Figure 3.14 Represents the link for dashboard

Once you run the load test in the pipeline and check the output you will get the link to the dashboard for example as shown in the above image.One can open the link by opening amazon wiki pasting the link and clicking preview button.

CHAPTER 4

RESULTS

Some of the output from the dashboard –

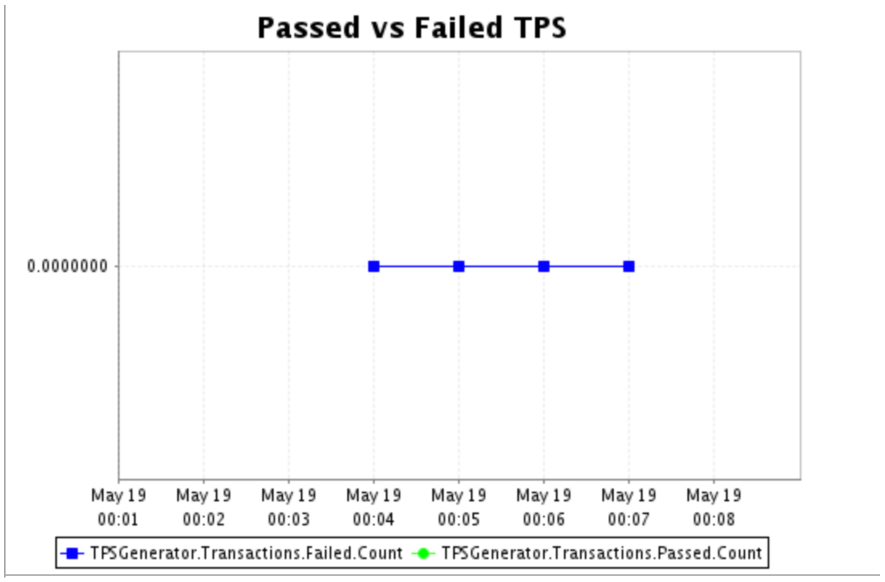


Figure 3.15 Represents the graph for passed/failed TPS

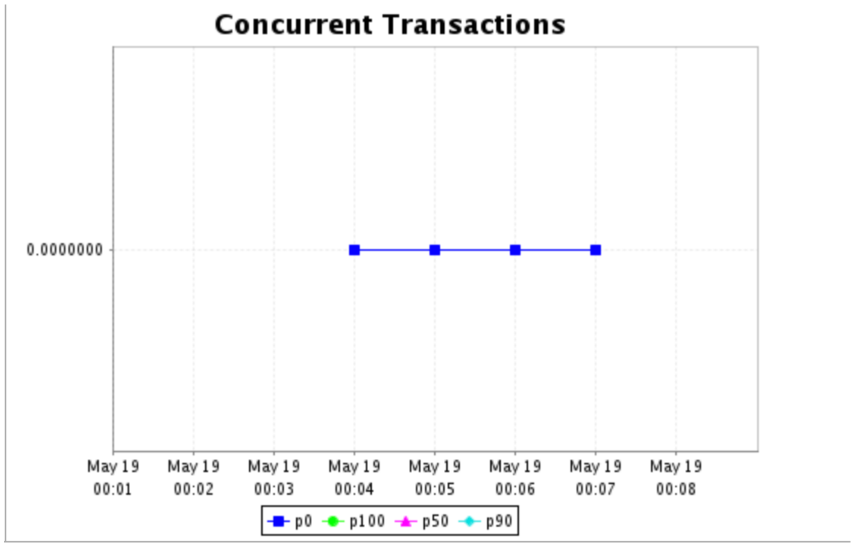


Figure 3.16 Represents the graph concurrent transaction

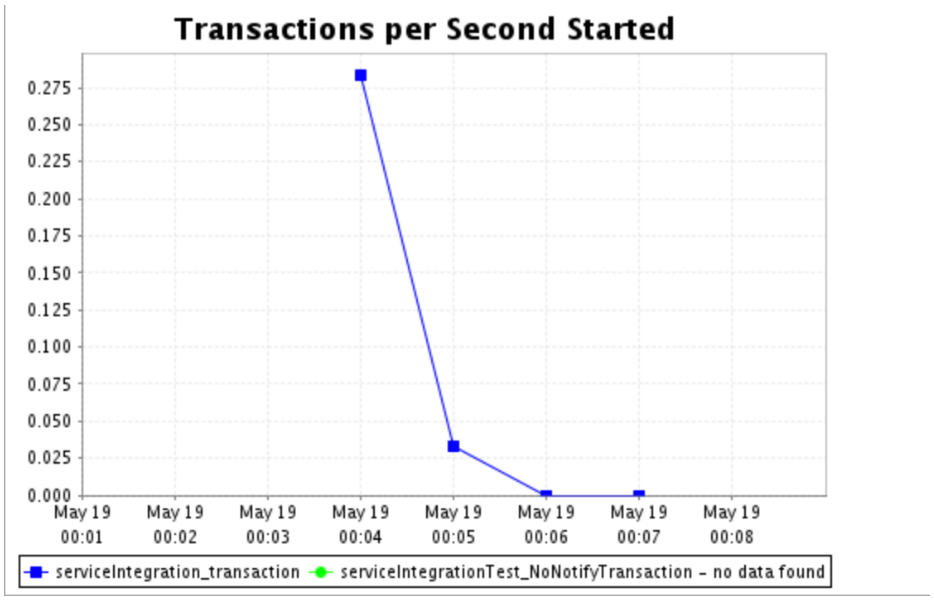


Figure 3.17 Represents the graph for Transaction per second started

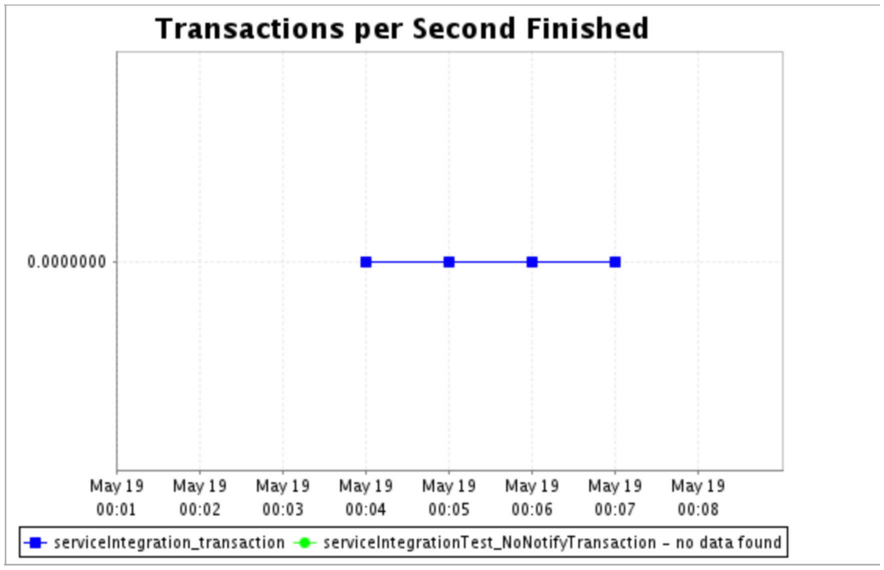


Figure 3.18 Represents the graph for Transaction per second finished

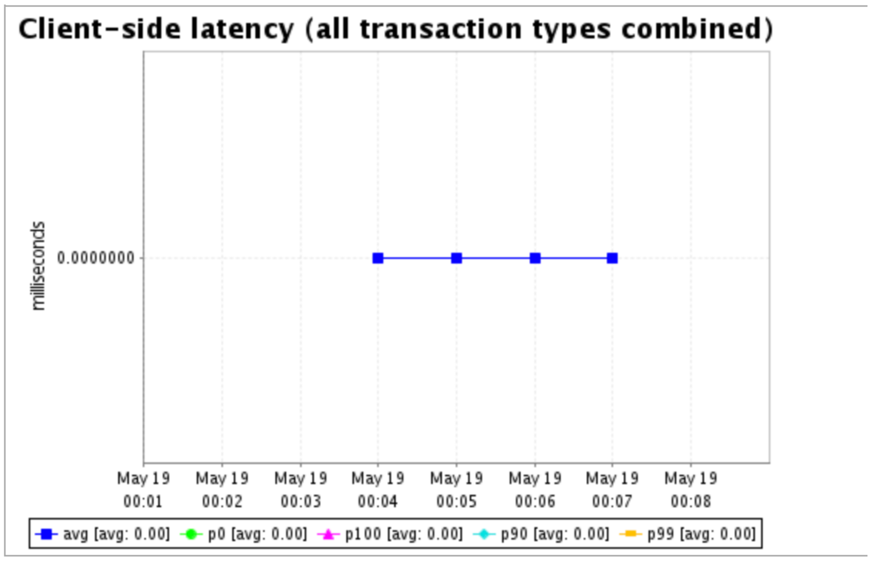


Figure 3.19 Represents the graph for client side latency

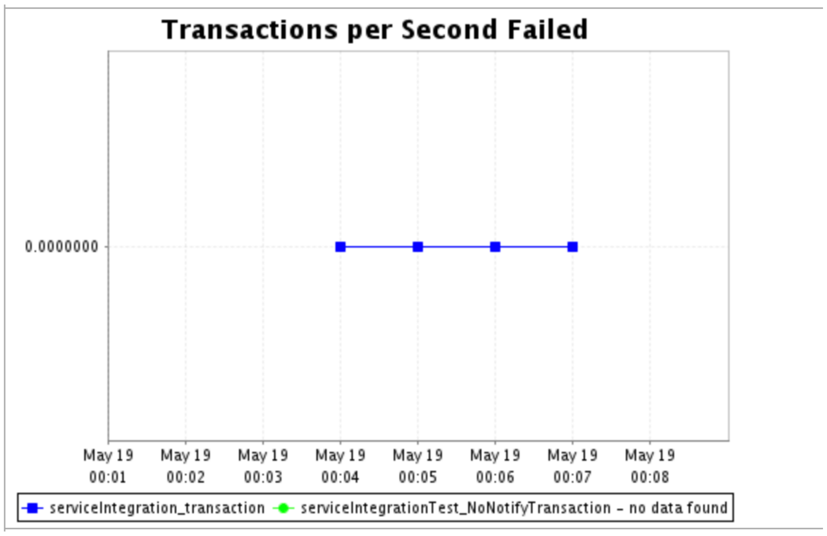


Figure 3.20 Represents the graph for transaction per second failed

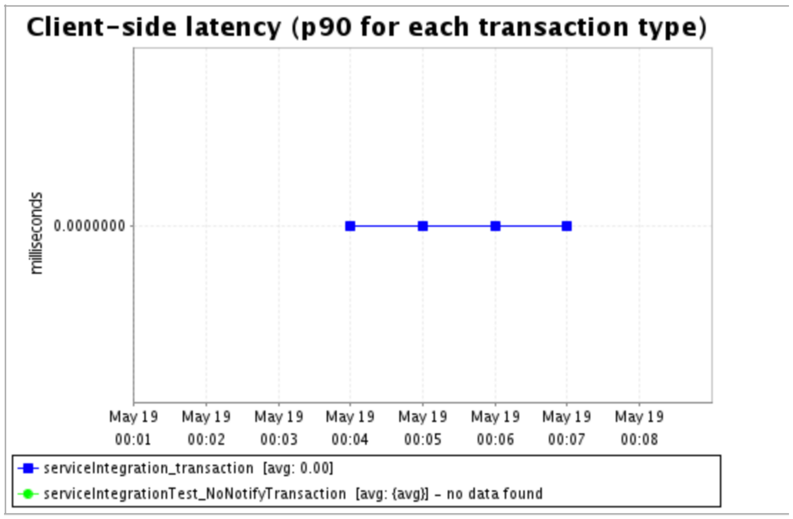


Figure 3.21 Represents the graph for client side latency

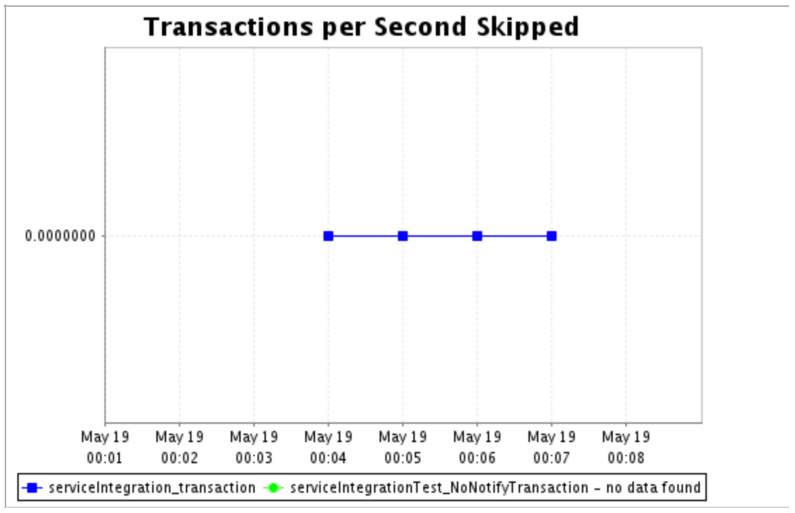


Figure 3.22 Represents the graph for transaction per second skipped

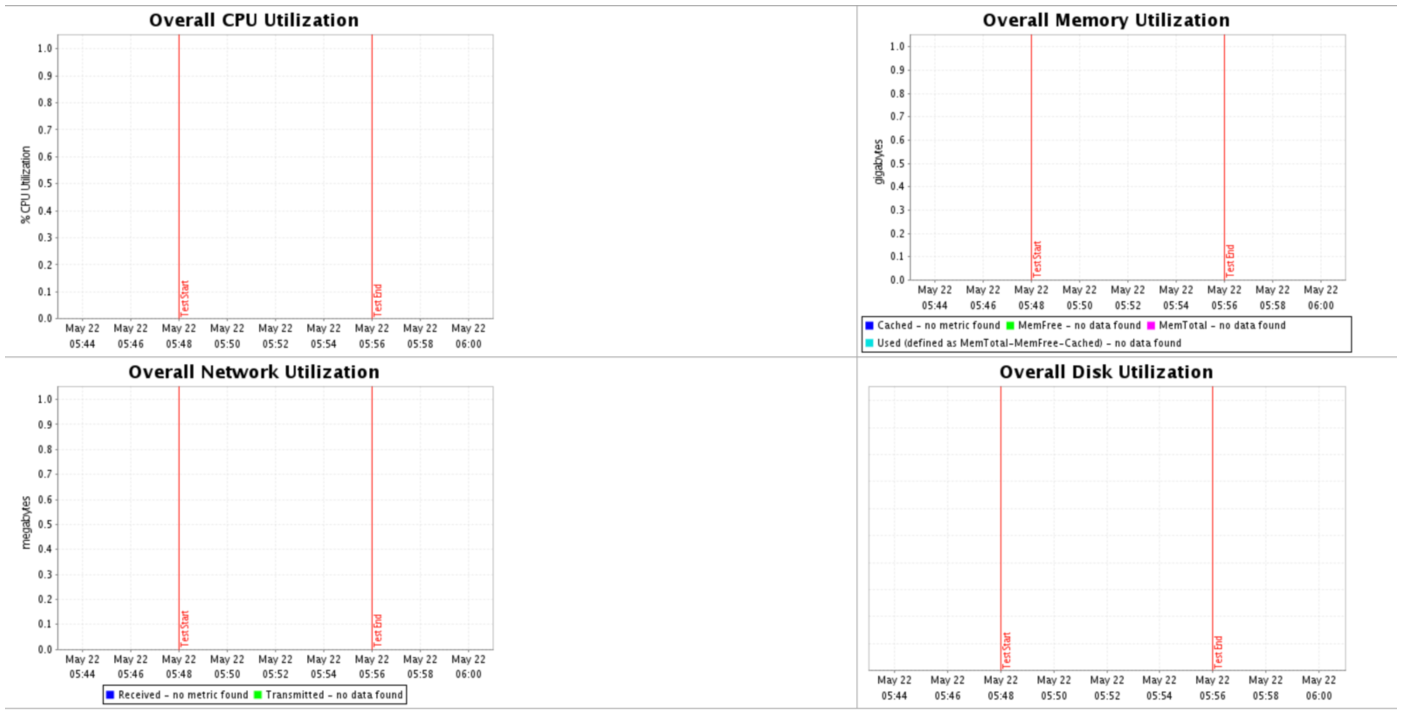


Figure 3.23 Represents the graph cpu ,memory, disk and network utilization

CHAPTER 5

5 . CONCLUSIONS

5.1 CONCLUSION

So from the project of load testing we can clearly understand why it is important for any organisation to perform the different kind of testing before launching it in the market.Why the trust of a customer is important and what is the need of load testing.

The main points one need to highlight are-

- The testing is important since it discovers defects/bugs before the delivery to the client, which guarantees the quality of the software.
- It makes the software more reliable and easy to use.
- Thoroughly tested software ensures reliable and high-performance software operation.

Why load testing is important ?

- Real Users Simulations
- Performance under load
- Estimating scalability of application
- Low downtime
- Code change affect product
- Don’t annoy users
- Saves money

Load testing is an important subset of PT that plays a significant role in the present-day technology scenario. Performance testing measures the ability of a web application to handle large volumes and patterns of traffic effectively before going live.

Measuring the Quality and Performance of Service of a Website – Load testing allows company to measure QOS i.e. quality of service

Identifying Important Application Performance Parameters-Load testing can be used to successfully identify the critical parameters of an application. These criteria include an application's response time per transaction, system/database component performance under varying load levels, software design difficulties, and network delays between client request and server answer.

Also highlighted are issues with software settings, such as web server, application server, and database server, as well as hardware flaws, such as CPU maximisation, memory limitation, and network bottlenecks.

Determining an application's maximum operating capacity An organisation can efficiently determine the maximum operating capacity of its application with the help of load testing. Load testing examines how the system behaves under normal load conditions as well as anticipated peak load conditions, allowing testers to identify the elements that cause degradation or hold-ups. As a result, load testing is also known as reliability testing, concurrency testing, or volume testing, and it assists testers in determining the capability of the existing infrastructure to run the application under test.

Understanding the Root Causes of Slow System Performance Load testing can pave the way for high-quality performance testing. The process is centred on evaluating system performance to ensure that there is no system downtime when multiple users access an application at the same time. Load testing tools can help testers learn about the causes of slow system performance. Application server or software, database server, network latency, network congestion, client-side processing, and load balancing between multiple servers are some of the reasons.

Improving an Application's Scalability- Load testing allows testers to identify the maximum number of users a programme can handle. This information will come in handy for testers if new infrastructure is needed to satisfy future demand for scaling the application. Aside from application scalability, testers can undertake special load testing, in addition to typical load testing, during certain events such as Cyber Monday, Black Friday, music festivals or sporting events, and the like - when traffic spikes dramatically owing to flash sales or mass public interest.

Lowering the costs of an application's failure Load testing is a critical testing approach that testers can use to uncover faults in an application before they become a problem. The monetary consequences of an application failure can be significantly reduced if issues are detected early and flaws are prevented. Failures during production, in the absence of load testing, can result in huge expenses for an organisation due to the application's unavailability in high-traffic situations

CHAPTER 6

6. REFERENCES

1. Quip Document Amazon - https://quip-amazon.com/

2. Amazon Wikipedia - https://w.amazon.com/bin/view/Main/

3. Builder hub for documentation - https://builderhub.corp.amazon.com/

4. TOD test on demand - https://builderhub.corp.amazon.com/tools/tod/

5. TPS generator - https://builderhub.corp.amazon.com/tools/tps-generator/