EVERY DEVOPS - COLLABORATION PLATFORM

Contact: sales@testonneed.com

Follow us: LinkedIn and Twitter

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
0.1	Vignesh A	First Draft version	

Review & Approval

Requirements Document Approval History

Approving Party	Version Approved	Signature	Date

Requirements Document Review History

Reviewer	Version Reviewed	Signature	Date

Table of Contents

1.Introduction

When there is an individual /organization (testoper) in need, they approach customers. But all the times, the requirement cannot be fulfilled by any one customer or sometimes individual (testoper) not know which customer to reach out for a specific requirement. In such hard times, customer/testoper should avoid competition and compromise and instead cooperate to achieve better results. But misunderstandings and collaboration without trust can prevent partnerships from reaching their potential.

Hence here is a devops based solution for Testoper and Customer collaboration, in which Customers with a diverse/same can come forward and be a part of consortium. When Testoper approaches a Customers with some need (say books for project testing), that customer can not help for this need or it can not be fulfilled by that customer alone, then that customer will view Testoper rating in the network. All other customers whoever can contribute with Testoper will update the rating previous over the blockchain network. This network will give the holistic view of the customer and it gives current status of that testoper. This way a need is catered to in an efficient manner. The problems of over conflict or mistaken for a need is reduced. The platform brings in trust, accountability and transparency of operations between customer and testopers.

2.Overview

when the development team is sitting side-by-side cracking out code quickly, TestOnNeed offers on-need QA testing, automation, and TestOps solution. We enable entrepreneurs to discover defects that affect your customer experience before deploying in production. We better the speed, scale, coverage, and quality of your web, mobile, API and desktop applications.

In order to help customer we creating DevOps solution

To make Zero-touch automation of source code management, build management, code quality and test management, infrastructure management, deployment and configuration management and Setting up Continuous Development, Continuous Testing, Continuous Integration, Continuous Delivery pipeline with Jenkins

- Monitoring 24/7 to detect any anomalies such as CPU, memory, and application crash.
- If any abnormalities detected, inform via email or SMS
- Automatically recover from failures without manual intervention

- Advice, test and Implement elastic scaling up, scale down, scale in and scale out for applications

3.Solution Overview

- Setting up source code management using GitHub
- Build management using Maven or Griddle
- Code quality and test management to ensure the error-free deployment of application
- Deploying application, database and relevant system software such as NodeJS and Nginx in AWS
- Reducing manual configuration of the application using Ansible and Python

  Set of python files along conf setup file

  createConfigFile.pyc
  createDependFolder.pyc
  createServerFile.pyc

4.Source Code management

Set up source code access along project path

git: Yes *// Enter your Prefer [Yes,No]*

If your wish to use git - mention git as Yes
Once you confirmed with git then set up access for git

git_id_rsa: *-----END RSA PRIVATE KEY-----"*
git_id_rsa_pub: "ssh-rsa key"

5.Infrastructure Management 7

As for Cloud Server should be AWS

cloud_type: *// cloud to be deployed - eg:[AWS]*

Provide Access and secret key allow to access AWS as automation servers

access_key: *// Enter AWS access key - eg:[AXMTSA]*
secret_key: *// Enter AWS secret key - eg:[63854V231HccXFP92]*

Provide Which region instance need to create

region: *// Enter region - eg:[ca-central-1]*

Mention is Backup required

instance_backup_preference *// Yes or No - eg:[Yes]*

Mention where backup need to store , how frequent and timing

instance_backup *// Backup Place- eg:[github]*
instance_backup_path: *// Path - eg:[https://github/projectData]*
instance_backup_time: *// 24:00 or 12:00 - eg:[8:00pm or 20:00]*
instance_backup_frequency: *// Days / Week / Month - eg:[Week]*