[1. Overview 1](#_Toc52795249)

[2. Deployment Diagram 1](#_Toc52795250)

[2.1 Detailed Deployment 2](#_Toc52795251)

[3. Data Flow 1 3](#_Toc52795252)

[4. Data Flow 2 3](#_Toc52795253)

[5. Data Flow 3 3](#_Toc52795254)

[6. Yaksha Logical Component Diagram 5](#_Toc52795255)

[7. Yaksha Technical Component Functions Glossary 6](#_Toc52795256)

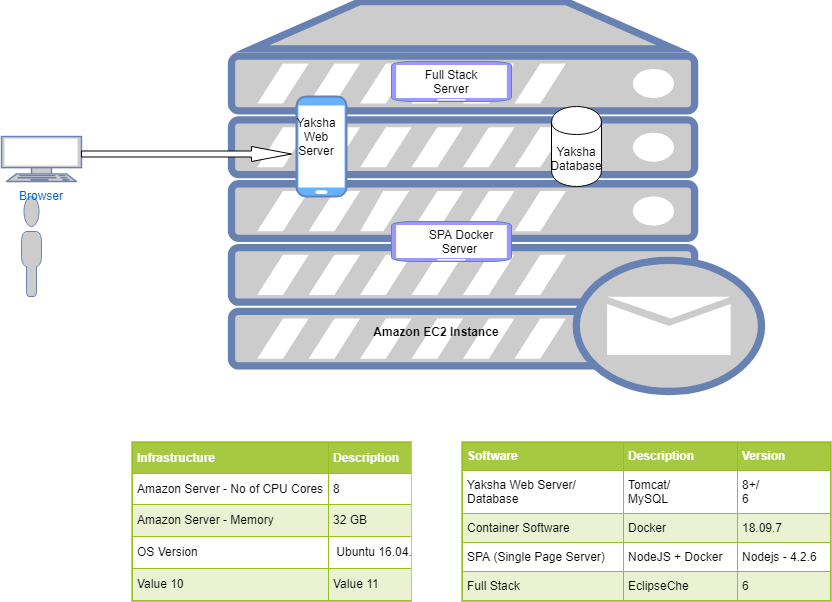
## Overview

Yaksha is a fully blown Assessment platform that tests learners on following type of tests

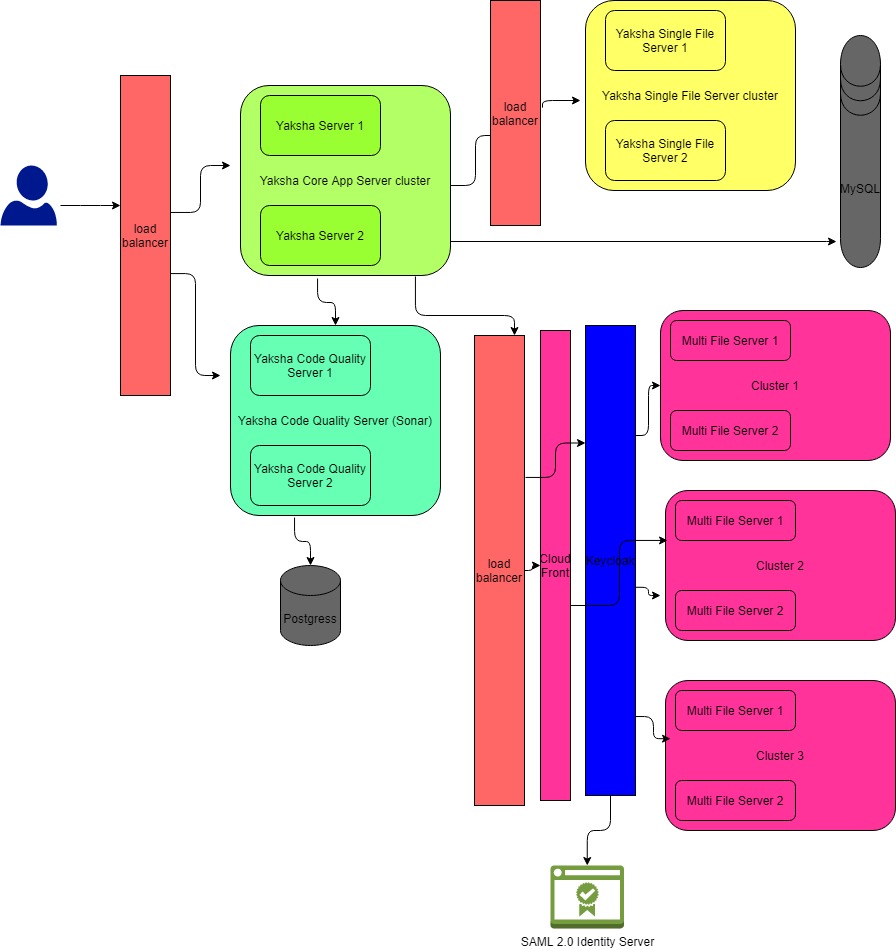
1. McQ
2. SPA – Single Page Applications
3. Full Stack assignments

This document is a summary of deployment specifics for Yaksha Assessment platform as it is now and listing of all data flows. Each data flow is distinguished by a set of common software component involved irrespective of function – There might be multiple features that might be using the same component flow.

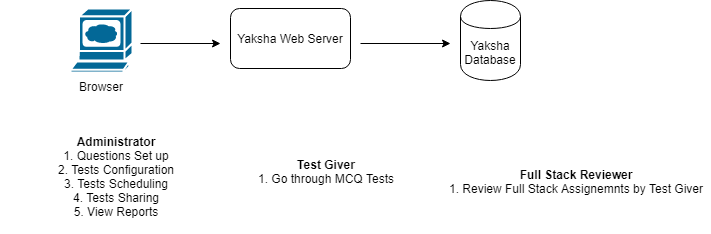
## Deployment Diagram



### 2.1 Detailed Deployment

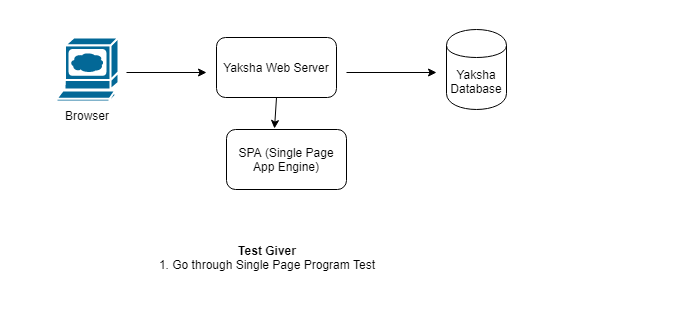


## Data Flow 1



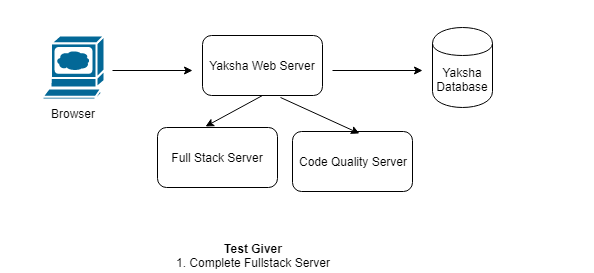
Actors for this flow – Test Admin / Test Giver / Reviewer

## Data Flow 2



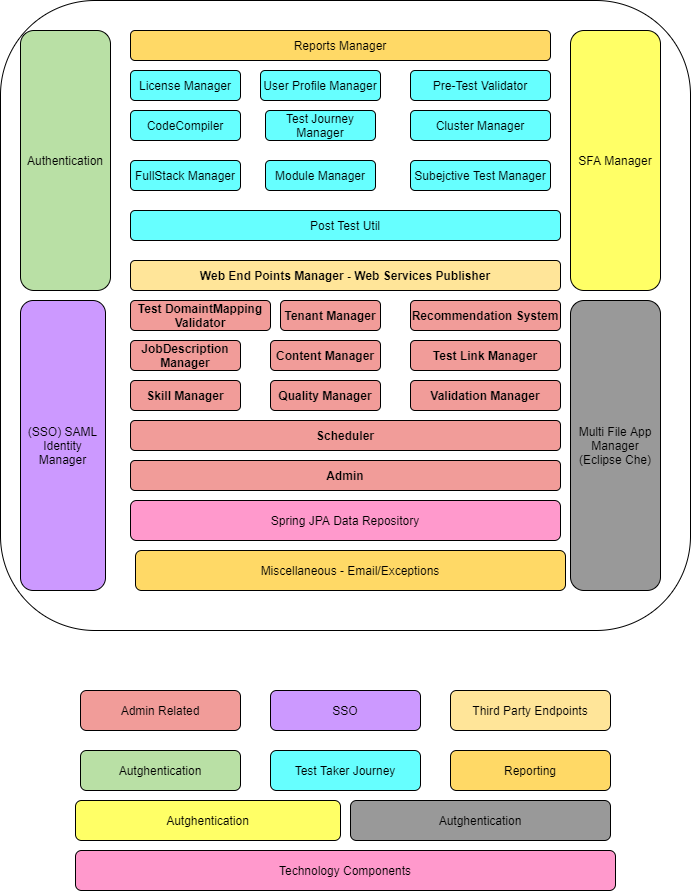
SPA – Single Page Application Test : Yaksha has a capability to ask test givers to code a problem statement in one of many programming languages that Yaksha supports. Single page programs are basically, as name suggests, 1 file isolated programs – e.g: Write a Fibonicci series, Arm strong number or so. Single page programs does not depend on third party libraries nor do they require more than 1 files to code a solution.

## Data Flow 3



1. Test Givers are also asked to complete a full stack test based on Java/.NET/JavaScript/Python/etc languages. Full stack assignments are different vis-à-vis SPA assignments in a way wherein the expectation from test givers is to develop an application spanning front/middle and data tiers. Test givers are allowed full liberty to use any third party library(s) as they deem fit for the purpose of the assignment.
2. Post the submit of the assignment there are 2 types of reviews that happen
   1. Automated Code Quality Check through Sonar Code Quality Server
   2. Manual review on functional and programming/architectural parameters by a manual reviewer (See Data flow 1)

## Yaksha Logical Component Diagram



## Yaksha Technical Component Functions Glossary

|  |  |  |
| --- | --- | --- |
| **Component Type** | **Component** | **Description** |
| Admin | Tenant Domain / Mapping Validator | Configure Tests only for test takers belonging to a specific domain (@www.com) |
| Tenant Manager | Create Tenants on the fly |
| Recommendation System | Configure the mapping between qualifiers and recommendations |
| Job Description Manager | Create Job Descriptions as per set of qualifiers with different weights |
| Content Manager | Create Questions, Tests with necessarily configurations |
| Test Link Manager | Create Test links with time stamps |
| Skill Manager | Create Skills to map with start and end dates |
| Quality Manager | Verify the Question Excel data before actual upload |
| Validation Manager | Set up business validations for tests - no of attempts/time/etc |
| Test Taker Related | License Manager | Fetch the tests/modules as per a set of licenses a Tenant Admin might have procured |
| User Profile Manager | Generates the learner profile |
| Pre Test Validator | Before actually commencing the test, validates the conditions - no of attempts/ domain check / date validity/etc |
| Code Compiler | Does the compilation and execution of test spa programs by users |
| Test Journey Manager | Manages the entire test giver journey from start to finish |
| Cluster Manager | Check the Eclipse Che cluster for availability before creating the workspace there |
| Full stack Manager | Manages the full stack test flow |
| Module Manager | Displays modules (set of tests related to a subject) to the learner in his dashboard |
| Subjective Test Manager | Manages the Subjective Test work flow |
| Post Test Util Manager | Post the test completion - does p[ost facto activities like sending mails, calling IRIS queues, etc |
| SFA | SFA Manager | Manages SFA work flow - run all test cases, compiles and show results in test screen |
| MFA | Multi File App Manager | Interacts with Eclipse Che for workspace creation, code quality generation & running automated test cases |
| Basic Authentication | Authentication |  |
| SAML Authentication SSO | SAML Identity Manager | SSO |
| Third Party Integrator | Web End points Manager | Web Service Endpoints (Documented) as shared with IRIS |
| Basic Util | Miscellaneous | Sending Emails, Managing Error screens etc |