From MARUFJAN Classes

What is jenkins?

- Continuous integration tool.
- It is used to schedule and automated builds, deployments etc. it is used for by developers and testers.

What kinds processes can be automated using jenkins?

- running unit tests
- building the application
- running automated UI tests
- deploying to different environments

How jenkins is used in your company for testing?

- In my project, we use jenkins for scheduling and running out automated smoke tests.
- our smoke tests run every day.

What tests you have on jenkins?

- In my project we have our smoke and regression tests for jenkins.

What kind of tests can be done in jenkins?

- So, jenkins can run any automated test.
- For example, we can have unit tests, smoke tests, integration tests, regression tests, sanity.

What layers of tests can be tested using jenkins?

- Testing different layers of the application is done with our test code.
- Jenkins does not care if we are testing UI or database or api. it only kicks of the tests and sends reports.
- So, if my automated test is a UI test, it means jenkins is running UI tests. or if my automated test is an API tests it means jenkins is running API tests.

Who sets up Jenkins?

- In my company we have DevOps/Operations Support team who is responsible for maintaining jenkins. those installed and configured jenkins.
- For deployments etc. they work with developers to create jobs for building and deploying the applications.
- For creating smoke tests, they work with automation engineers.

Who sets up smoke tests in Jenkins?

- As an automation engineer, I work with the devops/operations supports to create my smoke tests.
- For running my tests i need certain plugins installed on Jenkins, I also need java, maven configured on jenkins, I also need browsers installed on the server where the tests will run.
- In my project, only devops/operations support team members have right to do the configurations above. that is why I have to work with them.
- Once the configuration of the Jenkins is completed, I can create and run the smoke test job.

How your smoke tests are configured on jenkins?

- Our jenkins job is configured to get my automated tests from GitHub and run every day using maven.
- in configuration, first I created a new job and gave name "smoke tests". then I selected option git in the Source Code Management section and entered the path to my framework on GitHub. and also entered the git credentials.
- next configuration is about build triggers where I chose option Build periodically and enter the time how often to run tests.

In the Build section, I choose option invoke top level maven command since my automation framework is created using maven.

- in the section I enter the maven command (without the mvn part). so fi normal command to run my test using terminal is 'mvn test', in jenkins I only enter 'test'. here I also enter the tag i want to run using command line.

  test -Dcucumber.options="--tags @smoke"
- if I want to run regression tests, command is:

  test -Dcucumber.options="--tags @regression"

In the Post-build Actions, I do couple configuration,

- I add Cucumber reports option. This is available because we have installed cucumber html reports plugin to our jenkins. Cucumber reports plugin will generate html report for every build. Every time we run smoke tests on jenkins, we get a new report. and all reports for all build are saved.
- I add email option as the next Post-build Actions. Here I configure it so that everyone in my agile team is notified about the test results.

Who gets the email reports for tests?

- Everyone in my agile team is notified about the test results.

"Have you worked on DevOps pipeline?"

- I have built smoke tests on jenkins which part of the devops pipeline are. devops pipeline is built and managed by DevOps/Operations Support. But our smoke test are part of the pipeline.
- So, I participated by creating and configuring the smoke tests on jenkins.

What is the difference between integration test and end to end test?

- **Google often suggests a 70/20/10 split: 70% unit-tests, 20% integration tests, and 10% end-to-end tests.**
- **Unit Tests**
  - Tests the smallest unit of functionality, typically a method/function (e.g. given a class with a particular state, calling x method on the class should cause y to happen). Unit tests should be focused on one particular feature
- **Integration Tests**
  - Integration tests build on unit tests by combining the units of code and testing that the resulting combination functions correctly. This can be either the innards of one system or combining multiple systems together to do something useful. It is white box testing approach . Developer develop single unit one by one and integrate it with each other and test it .
- **Functional Tests**
  - Functional tests check a particular feature for correctness by comparing the results for a given input against the specification. Functional tests don't concern themselves with intermediate results or side-effects, just the result (they don't care that after doing x, object y has state z).
- **Acceptance Testing**
  - This is the last test that is conducted before the software is handed over to the client. It is carried out to ensure that the developed software meets all the customer requirements. There are two types of acceptance testing - one that is carried out by the members of the development team, known as internal acceptance testing (Alpha testing), and the other that is carried out by the customer or end user known as (Beta testing)
- **End to End testing**
  - is where you go from point a to z and you touch various different points on the way there. It could be for a single system like the process of sending an email or it could be used where multiple systems are involved for example a student registering for a test, taking the test and then finally receiving his/her scores.
  - Its real-world system testing. Application is tested with all integrated hardware, database, network and other interfaces.

How do you run them on Jenkins?

It is a long story…

Tell me about your framework?

- In my recent Book-IT automation framework, we used Cucumber BDD framework for automated tests. this framework is a very flexible framework. It is a hybrid framework which integrates many different automation testing concepts.
- The framework is built using MAVEN. maven is used to build the framework, manage the dependencies and plugins, run the tests as a maven life cycle.
- It is written using Java language.
- It is a Cucumber BDD framework. Cucumber is a tool used to make automated testing understandable to the non-technical team members. Cucumber serves as a bridge in connecting the automation engineers and the non-technical team members. we use cucumber version 4
- We use junit as a testing tool. Junit is used to kick off cucumber tests and also do assertions.
- We use Selenium WebDriver for automating the browsers. with selenium we can run tests in different browsers.
- Our framework generates step by step HTML reports with screenshots using the maven-cucumber-report plugin.
- Our framework supports data driven testing. Cucumber natively supports data driven testing using scenario outlines. The framework can also do Data driven tests from excel using the apache POI libraries.
- My framework is based on page object model. page object model is when we represent pages from the application in page object classes.
- we use factory model to create a webdriver. our webdriver class can generate a webdriver based on what kind of driver we want. it uses the singleton pattern for the webdriver object
- we use properties file to store the important information about our framework such as URL, browser type, login info etc.
- We use IntelliJ in our company, but I am also quite good at eclipse.

What layer of application does it test?

Using our framework, we test the UI, database and the API of the application.

- **UI** → for testing the UI we use selenium webdriver
- **Database** → for testing the database we use JDBC libraries
- **API** → for testing the API we use REST Assured libraries

Types of tests:

- we do **functional**, **acceptance tests**, **smoke tests**, **regression tests**, **integration tests**.
- We use git for version control tool in my project. (SVN)
- We use Jenkins for scheduling and running our automated smoke and regression tests and emailing the test results. Using jenkins, we can run tests and update the JIRA with the test results using the Xray plugin.
- We use Log4J for logging in our framework.
- organization of code and tests
- we use packages for organizing our java code, we have packages for page objects, utilities, StepDefinition, pojos, runners we use different utilities, we have utilities for WebDriver, browser utilities, excel utility, configuration utility, Database utility, api related utilities, date utility (since we test calendar related application, we do a lot date related stuff).
- features:
  we have many feature files and we use folders and tags to organize them. we have folders for each major component of the application : reservations, map, account information….
  we use the issue number from Jira as one of the tags in the feature file so that we can map it to Jira.
  - features
    account
    reservation
    map

```
Feature: User roles
  # in the given you need to figure out a way to find out if there are any available spots

  @login @BRIT_3521 @smoke
  Scenario Outline: login as <user> type
    Given the user logs in as a light-side <user>
    And there are available spots for scheduling
    When the user hunts for a spot
    Then book button <expected> be displayed

    Examples:
      |user        | expected   |
      |team member | should not |
      |team lead   | should     |
      | teacher    | should     |
```

Parallelization → we run our tests in parallel using the maven-surefire-plugin. we use cucumber 4 which supports parallel testing natively.
Test data → use feature files, excel files to store our test data in the test/resources package

Flow of the execution or how we run tests

1. **running from cukes runner**
   we put tag in the cukes runner and right click and run the file. cukes runner runs all the matching tags.
2. **run from terminal as a maven command**
   we enter command mvn verify or mvn test to kick of the tests. and maven will run the cukes runner files shown in the pom file. cukes runner file runs the features matching the tags it has.

GIT branching

how do you do git branching now?

Currently HOW IT IS DONE IN GROUP PROJECTS:

There is master branch and separate branches for each team member. when someone finishes work, they push to their own branch, then after reviewing it is merged to master.

HOW WE DID IT?

in my project we had master, develop and branch for person. so, if we have 2 automation testers, we will have

master
develop
tester1
tester2

each tester check in to their own branches. then after reviewing it is merged to develop branch. we merge master and develop only once a sprint.

✪ in the interview when they ask about branching, talk about your automation project branching strategy. in you project, your code is separate repo from the application code repo. Automation framework have a smaller code base and fewer people involved. So, we can have less complicated branching policy.