Mercedes Benz R&D India

ADAS - Software Factory & Tools

Solutions Architect Assignment

Section Overview

Section 1 – Jenkins Pipeline Implementation

1. The expectation out of this section for the candidate is to develop one/more Jenkins pipeline scripts to satisfy the requirements mentioned in the question.
2. The Jenkins pipeline script must be pushed to the candidate's personal github repository and link must be shared with the interviewer.
3. If the candidate has access to a free tier account of any Cloud Platform, you can host it on that and share the Jenkins server link so that we can verify from our end.
4. This is a mandatory question to answer to proceed to further discussions
5. Prerequisites: Jenkins instance, Github account, Plugin to make REST calls.

Section 2 – Cloud based Architecture Proposal

1. The expectation out of this section for the candidate is to describe the approach the candidate would take for the given problem in a text/word file.
2. The text/word file must be pushed to the candidate's personal github repository and link must be shared with the interviewer.
3. This is a mandatory question to answer to proceed to further discussions
4. Prerequisites: Github account

Mercedes Benz R&D India

ADAS - Software Factory & Tools

Section 1: Jenkins Pipeline Implementation

Create a Jenkins job using the pipeline script in Declarative format. Below are the details of the Parameter and stages to be configured.

Parameter:

1. **Name:** Static_Check ; **Type:** Checkbox
2. **Name:** QA; **Type:** Checkbox
3. **Name:** Unit_Test ; **Type:** Checkbox
4. **Name:** Success_Email ; **Type:** String
5. **Name:** Failure_Email ; **Type:** String

Stage 1: Name: Git Pull

Steps:

1. Post setup of git in your machine as mentioned in the pre-rqeuisites, use the Git Plugin from Jenkins in the pipeline script to just pull the current branch.

Stage 2: Name: Is the run required?

Steps:

1. Use a Jenkins plugin (the candidate has to find a suitable plugin) which would make a REST call and read the response from the URL: https://calendarific.com/api/v2/holidays?&api_key=d20d05ccb411d9ce3b56b654971e17a29b0aa1ed&country=IN&year=2021

   Explanation of API:

   a. **METHOD**: GET
   b. **Parameters in the URL**:
      i. api_key – This is generated for an account and setup from the interviewer's side. If there is any problem with the API, contact the interviewer immediately and new api_key will be given.
      ii. country – The country for which we are finding holidays
      iii. year – The year for which we are finding holidays
      iv. All above 3 parameters are mandatory
   c. **Response**:
      i. If request is made with all three above parameters, the response will have holidays array inside a response json.

   The values of the parameters in the URL api_key and country can be hardcoded. The year value must be taken from the year attribute of the current date and supplied to the URL

2. Once the request is made with above parameters, the response will be received in which a “holiday” array will contain several json objects where each json object represents a particular holiday. The candidate has to go through the entire array and see if current date is present in the array. If it is present, it means that today is a holiday and we do not have to run this build, which means the forthcoming stages should be skipped. However, if current date is not present in the array, the below mentioned stages should be executed.

Stage 3:

Name: Build

Condition: This stage should run only if current date is not a holiday from Stage 2.

Steps:

1. Parse the given json file (build.json) content and create the text files with the value given in the “name” field in the build.json and write the respective “content” value in the respective txt files.
2. Place all the text file in one common directory with name of the directory being “builds” and zip it.

Stage 4:

Name: Static check

Condition: This stage should run only if current date is not a holiday from Stage 2 and if Static_Check checkbox enabled.

Steps:

1. Create the directory in current stage name and place the respective text file from the zip file created in the build stage.

Stage 5:

Name: QA

Condition: This stage should run only if current date is not a holiday from Stage 2 if QA checkbox enabled and the Static check stage passed.

Steps:

1. Create the directory in stage name and place the respective text file from the zip file created in the build stage.

Stage 6:

Name: Unit test

Condition: This stage should run only if current date is not a holiday from Stage 2, if Unit_Test checkbox is enabled and should run in parallel to Static Check and QA stages as shown in below image.

Steps:

1. Create the directory in stage name and place the respective text file from the zip file created in the build stage.

Stage 7:

Name: Summary

Steps:

1. Print in the console that which stages are executed based on checkboxes enabled and what files copied in that stage.

Postbuild Stage:

On Success:

Send email to the mail id, which is provided in the parameter Success_Email

On Failure:

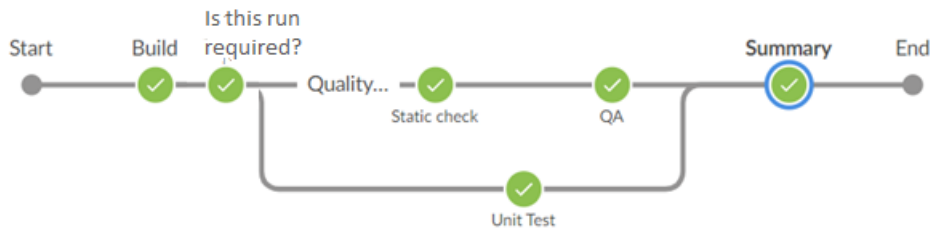
Send email to the mail id, which is provided in the parameter Failure_Email

Note:

1. On execution of the above job configuration, the pipeline should look like below in Blue Ocean.

2. Bonus points if proper comments are put wherever necessary on what optimizations can be done before taking to production.

3. We will run the pipeline script shared from your end, so handle all conditions properly and a readme file containing pre-requisites to run from our end would be required.



Mercedes Benz R&D India

ADAS - Software Factory & Tools

Section 2: Cloud based monitoring architecture

(i) In addition to the continuous integration environments, it is a general practice for enterprise applications to also have a monitoring environment. This monitoring application helps us to keep track of what builds are running, statistics about how many builds have passed for the day/month/year and how many have failed, average runtime of a build, etc.

(ii) Given that the source is Jenkins (the CI environment) and the destination is some sort of data store(which candidate has to propose) and assuming that a data processing layer is also required to process data between the source and destination, kindly propose a solution on how this system could be designed.

(iii) All of the above points to be considered from a cloud based perspective. The candidate has to consider all factors like security, reliability, scalability, performance, etc. Either of AWS and Azure cloud platform can be considered for this.

(iv) The candidate also has to create an IaaC template either using Terraform or Cloud Formation or other equivalent tools and provide proper comments for all the parameters that has been considered. The template must be either in yaml or json format and should adhere to the constraints of the file type.

(v) Please delve into as much detail as possible, for example, name the open source components that would satisfy the condition of data store and data visualization, how would you move the data from source to destination, will the data be processed synchronously or asynchronously, etc. A sketch of the system architecture would be appreciated.

(vi) The sketch could be done using MS paint, PPT or could be drawn in a book and a snapshot of that can be attached. Please note that we are not looking for any fancy diagram. It can be as rough a sketch as possible as long as the architecture works. The expectation is just a description in candidate's own words of the approach they would take.