**QA Deployment Stage**

**Micro Side Changes:-**

* For dev and qa deployment we are using same deployment class. To use the same class for two stages we have made changes for saving the data into tables for same jobId.
* To pass the stage specific attributes, we created one new column in stage\_class\_name\_entity table. We are mapping the attributes present in that table with the request body.
* We have made changes in the pipeline creation to add new stage.
* We made the pipeline creation fully dynamic.
* As there are two deployment stages, to build the particular deployment stage we have added one new column in deploy\_job entity, where we are storing stage name. based on the stage name we will trigger the particular build.
* Once the build gets trigger we are storing the status of the build in the deploy\_job entity based on stage name.

**UI Side Changes :-**

* First we created new profile, which includes the QA stage. So there are total 6 stages in that profile.
* We linked that profile to our application.
* As to pass stage specific information for QA stage we have added new form.
* To add the new form, we are using same deploy-form component, we need to select the environment and based on the environment value we will get specific fields.
* First will have to fill the deploy dev form then when we reach at the deploy QA form. In deploy QA form we will get all the information which we are providing in deploy dev form, then we can make changes according to QA stage.
* Once we click on the submit button the request body will get generated based on the selected stages.

**Database side changes :-**

* we have made deployment db scripts dynamic in order to use the same scripts for both deployment stages.
* From the database we have removed the stageParameter for deployment stages, now we are passing those parameters as attributes in the script.
* As we are going to use different agent node for different deployment stage, we have changed the script accordingly so that it can accept the node value provided by user from the UI form.
* We checkout the git code at the start of the pipeline and then after at the deployment stage we are changing the node. Once we directed to new node we also need our git checkout workspace data on that node. For that we are using stash unstash. Before going on the different node we are stashing the workspace data from current node and after changing the node we are unstashing the source data.