**Practical Exercise 31 - Creating and Using Reusable Workflows**

**Exercise Description**

**In this practical exercise, our goal is to get familiar with creating and using reusable workflows.**

Here are the instructions for the exercise:

1. Create a file named 18-1-reusable-workflows.yaml under the .github/workflows folder at the root of your repository.
2. Name the workflow 18 - 1 - Reusable Workflows - Reusable Definition.
3. Add the following triggers with event filters and activity types to your workflow:
   1. workflow\_call: additionally, the workflow\_call trigger should accept a single input and produce two outputs:
      1. The input should be named target-directory, be required, and be of type string.
      2. The first output should be named build-status, have a description of The status of the build process, and have the value set to the build-status output of the deploy job (see below for instructions on defining the job).
      3. The second output should be named url, have a description of The url of the deployed version, and have the value key set to the url output of the deploy job (see below for instructions on defining the job).
4. Add a single job named deploy to the workflow.
   1. It should run on ubuntu-latest.
   2. It should define two outputs:
      1. build-status, taking its value from an analogous output of the build step.
      2. url, taking its value from an analogous output of the deploy step.
   3. It should contain three steps:
      1. The first step, named Checkout repo, should checkout the code using the appropriate third-party action.
      2. The second step, named Build, should have an id of build, print the message "Building using directory <retrieve the value of the target-directory input here>", and set the output build-status, with a value of success.
      3. The second step, named Deploy, should have an id of deploy, print the message "Deploying build artifacts", and set the output url, with a value of https://www.google.com.
5. Now create a new workflow file named 18-2-reusable-workflow.yaml under the .github/workflows folder at the root of your repository.
6. Name the workflow 18 - 2 - Reusable Workflows.
7. Add the following triggers with event filters and activity types to your workflow:
   1. workflow\_dispatch
8. Add a first job named deploy to the workflow.
   1. It should use the reusable workflow defined above. For that, simply use the uses key at the job level and pass the path of the workflow: ./.github/workflows/18-1-reusable-workflows.yaml
      1. Pass any value as a target-directory argument to the reusable workflow.
9. Add a second job named print-outputs to the workflow.
   1. It should run on ubuntu-latest.
   2. It should have a dependency to the deploy job.
   3. It should contain a single step, named Print outputs, which prints the following two messages on the screen:
      1. "Build status: <retrieve the build-status output from the deploy job>"
      2. "URL: <retrieve the url output from the deploy job>"
10. Commit the changes and push the code. Trigger the workflow manually from the UI and take a few moments to inspect the result of the workflow run.