GIT

To prevent error: “SSL certificate problem: unable to get local issuer certificate”:

$ git config --global http.sslVerify false

$ git init

$ echo "# SnowflakeLearning" >> README.md

$ git add README.md

$ git commit -m "first commit"

$ git branch -M main

$ git remote add origin https://github.com/Lana367Canada/SnowflakeLearning.git

$ git push -u origin main

# Create Action Secrets

Action Secrets in GitHub are used to securely store values/variables which will be used in your CI/CD pipelines. In this step we will create secrets for each of the parameters used by schemachange.

From the repository, click on the Settings tab near the top of the page. From the Settings page, click on the Secrets tab in the left hand navigation. The Actions secrets should be selected. For each secret listed below click on New repository secret near the top right and enter the name given below along with the appropriate value (adjusting as appropriate).

|  |  |
| --- | --- |
| Secret name | Secret value |

SK\_ACCOUNT = go18021.ca-central-1.aws

SK\_USERNAME = LANA367CANADA

SK\_PASSWORD = SLI123!!!

SK\_ROLE = ACCOUNTADMIN

SK\_WAREHOUSE = COMPUTE\_WH

SK\_DATABASE = LAB3

# Create an Actions Workflow

Action Workflows represent automated pipelines, which inludes both build and release pipelines. They are defined as YAML files and stored in your repository in a directory called .github/workflows. In this step we will create a deployment workflow which will run schemachange and deploy changes to our Snowflake database.

* From the repository, click on the Actions tab near the top middle of the page.
* Click on the set up a workflow yourself -> link (if you already have a workflow defined click on the new workflow button and then the set up a workflow yourself -> link)
* On the new workflow page
  + Name the workflow snowflake-devops-demo.yml
  + In the Edit new file box, replace the contents with the the following:

name: snowflake-devops-demo

*# Controls when the action will run.*

on:

push:

branches:

- main

paths:

- 'migrations/\*\*'

*# Allows you to run this workflow manually from the Actions tab*

workflow\_dispatch:

jobs:

deploy-snowflake-changes-job:

runs-on: ubuntu-latest

steps:

*# Checks-out your repository under $GITHUB\_WORKSPACE, so your job can access it*

- name: Checkout repository

uses: actions/checkout@v2

- name: Use Python 3.8.x

uses: actions/setup-python@v2.2.1

with:

python-version: 3.8.x

- name: Run schemachange

env:

SF\_ACCOUNT: ${{ secrets.SF\_ACCOUNT }}

SF\_USERNAME: ${{ secrets.SF\_USERNAME }}

SF\_ROLE: ${{ secrets.SF\_ROLE }}

SF\_WAREHOUSE: ${{ secrets.SF\_WAREHOUSE }}

SF\_DATABASE: ${{ secrets.SF\_DATABASE }}

SNOWFLAKE\_PASSWORD: ${{ secrets.SF\_PASSWORD }}

run: |

echo "GITHUB\_WORKSPACE: $GITHUB\_WORKSPACE"

python --version

echo "Step 1: Installing schemachange"

pip install schemachange

echo "Step 2: Running schemachange"

schemachange -f $GITHUB\_WORKSPACE/migrations -a $SF\_ACCOUNT -u $SF\_USERNAME -r $SF\_ROLE -w $SF\_WAREHOUSE -d $SF\_DATABASE -c $SF\_DATABASE.SCHEMACHANGE.CHANGE\_HISTORY --create-change-history-table

Finally, click on the green Start commit button near the top right of the page and then click on the green Commit new file in the pop up window (you can leave the default comments and commit settings). You'll now be taken to the workflow folder in your repository.

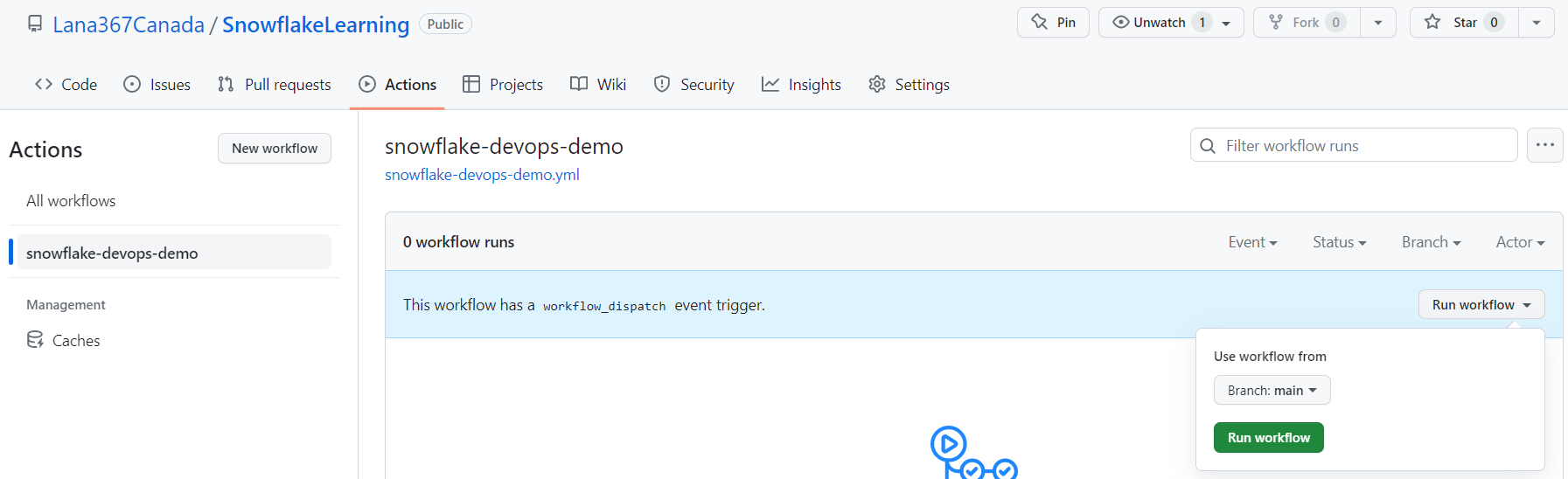
A few things to point out from the YAML pipeline definition:

* The on: definition configures the pipeline to automatically run when a change is committed anywhere in the migrations folder on the main branch of the repository. So any change committed outside of that folder or in a different branch will not automatically trigger the workflow to run.
* Please note that if you are re-using an existing GitHub repository it might retain the old master branch naming. If so, please update the YAML above (see the on: section).
* We're using the default GitHub-hosted Linux agent to execute the pipeline.
* The env section of the Run schemachange step allows us to set environment variables which will be available to the Bash script. In particular, this allows us to securely pass secret values (like the Snowflake password) to applications/scripts running in the workflow like schemachange.

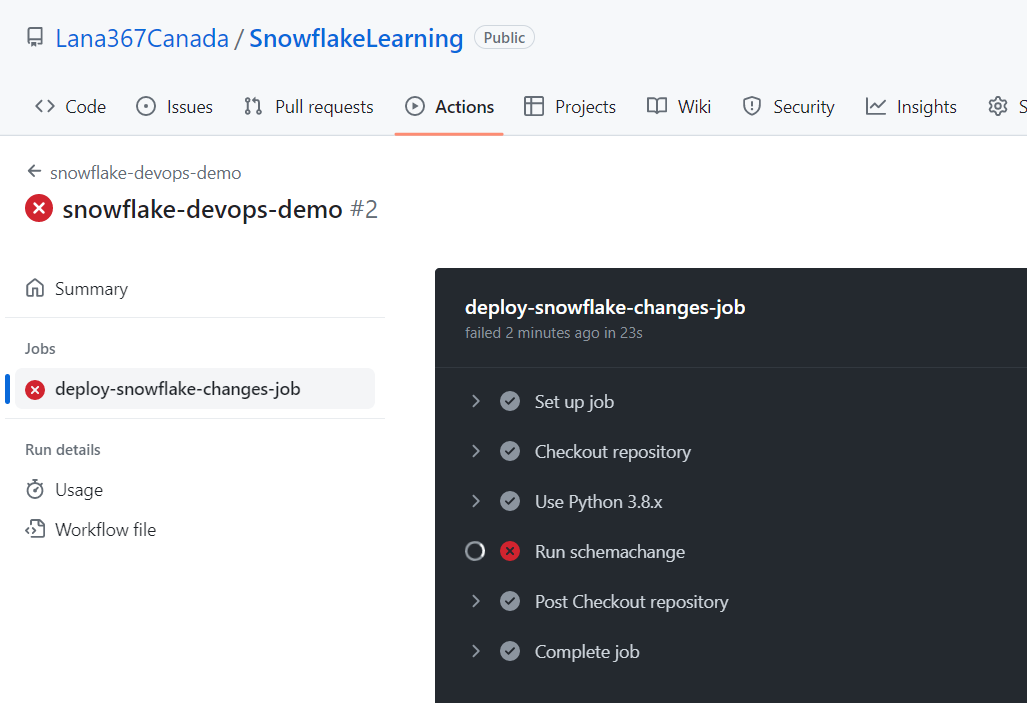
# Manually Run the Actions Workflow

In this step we will manually run the new Actions workflow for the first time. This will deploy the first database migration script we created in step 4.

* From the repository, click on the Actions tab near the top middle of the page
* In the left navigation bar click on the name of the workflow snowflake-devops-demo
* Click on the Run workflow button and then on the green Run workflow button in the pop up window (leaving the default branch selected)



To view the details of a run, click on the name of specific run (you may have to refresh the Actions page for it to show up in the list). From the run overview page click on the deploy-snowflake-changes-job job and then browse through the output from the various steps. In particular you might want to review the output from the Run schemachange step.



# Confirm Changes Deployed to Snowflake

Now that your first database migration has been deployed to Snowflake, log into your Snowflake account and confirm.

## Database Objects

You should now see a few new objects in your DEMO\_DB database:

* A new schema DEMO and table HELLO\_WORLD (created by the first migration script from step 4)
* A new schema SCHEMACHANGE and table CHAGE\_HISTORY (created by schemachange to track deployed changes)

Take a look at the contents of the CHANGE\_HISTORY table to see where/how schemachange keeps track of state. See the [schemachange README](https://github.com/Snowflake-Labs/schemachange" \t "_blank) for more details.

## Query History

From your Snowflake account click on the History tab at the top of the window. From there review the queries that were executed by schemachange. In particular, look at the Query Tag column to see which queries were issued by schemachange. It even tracks which migration script was responsible for which queries.