**OVERVIEW**

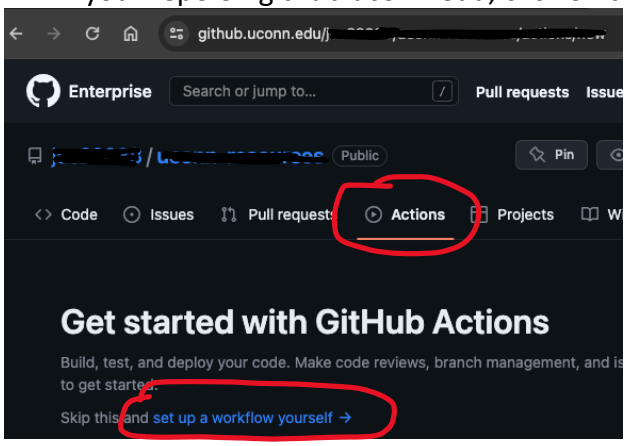
This tutorial will help you create a CICD Pipeline from the UConn instance of GitHub and deploy code to the AWS Learner Lab environment. There are two ways to do this. (1) Use your SSH key for the Learner Lab, or (2) Use the AWS CLI and your CLI credentials for the Learner Lab.

**YOUR CODE**

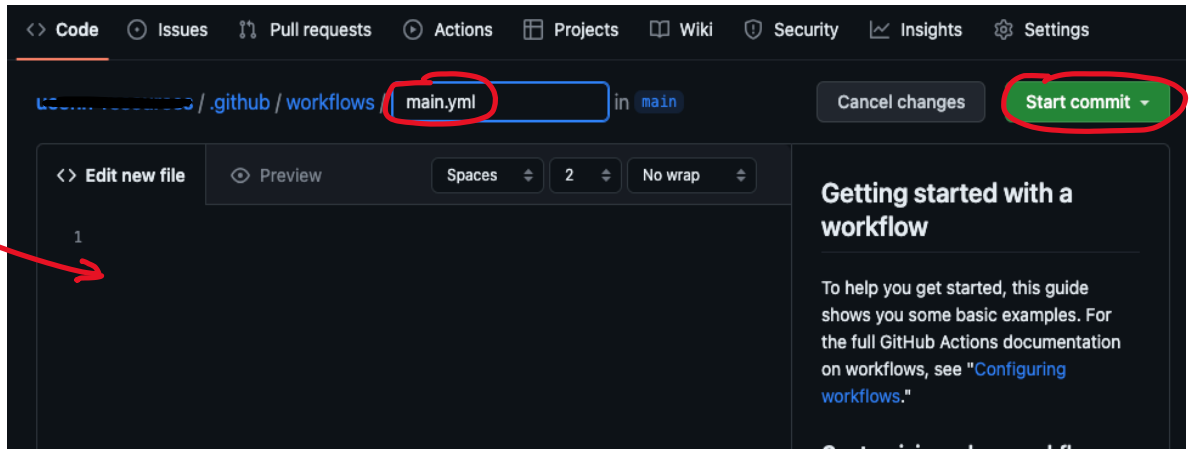
Create some code in your repo. (e.g. a simple python file for lambda or some files to push to EC2.)

**Create a Github Action workflow**

- In your repo on github.uconn.edu, click on the "Actions" tab.




- Change "main.yml" to whatever file name you desire for your new GitHub Action. (Shown below.)
- Create the yaml file from scratch and use this code as a template for your Action. Copy paste that code into the editing section shown below. (Edit the On-Push-Branches target to correspond to your branch. E.g. "hw02".)



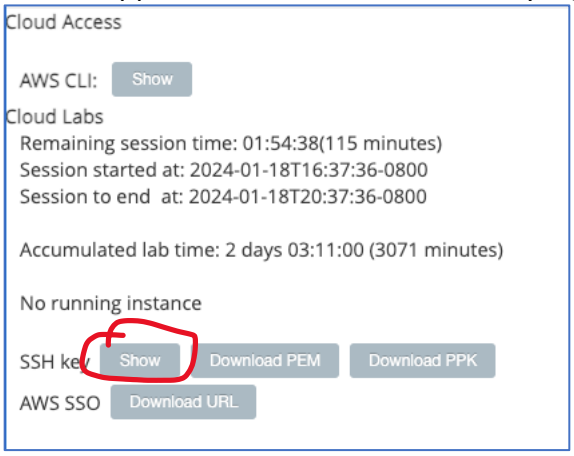

- For example of the SSH method, use this yaml file. For an example of the AWS CLi approach, use this yaml file.

- Press the "Start commit" button when you are finished and complete the rest of the prompts until the file is committed.

**Create Github Action Secrets**

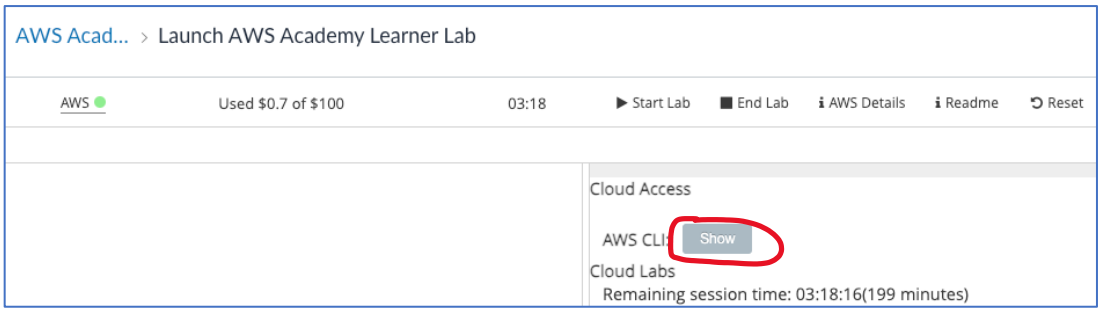
The GitHub Action pipeline needs access to AWS in order to deploy to AWS. We never want to put access credentials into our GitHub repos. Instead, we create what is called a "secret". The pipeline then reads the secret when it executes. Once the secret is created, the secret value cannot be read by anyone. But you can update the key's value when you create a new Learner Lab session. (In the regular AWS accounts, the id & secret would remain valid until explicitly revoked in the AWS account. So updating the secret key values every time we have a new Learner Lab session is an annoying step only necessary when using the Learner Lab. But at least we have a way to do CICD with the Learner Lab accounts!)

- First, harvest your AWS Learner Lab credentials from your current Learner Lab session. (Note: if the green circle is not present next to the "AWS" link on the far left-hand side of window shown below, then click "start lab" to start a new Learner Lab session. You get a 4-hour session. But if you click "Start Lab" during an active session, your current session is extended and all your access tokens/ids/secrets remain valid!)
- To use the SSH approach, click on "Show SSH Key". (Shown below.)

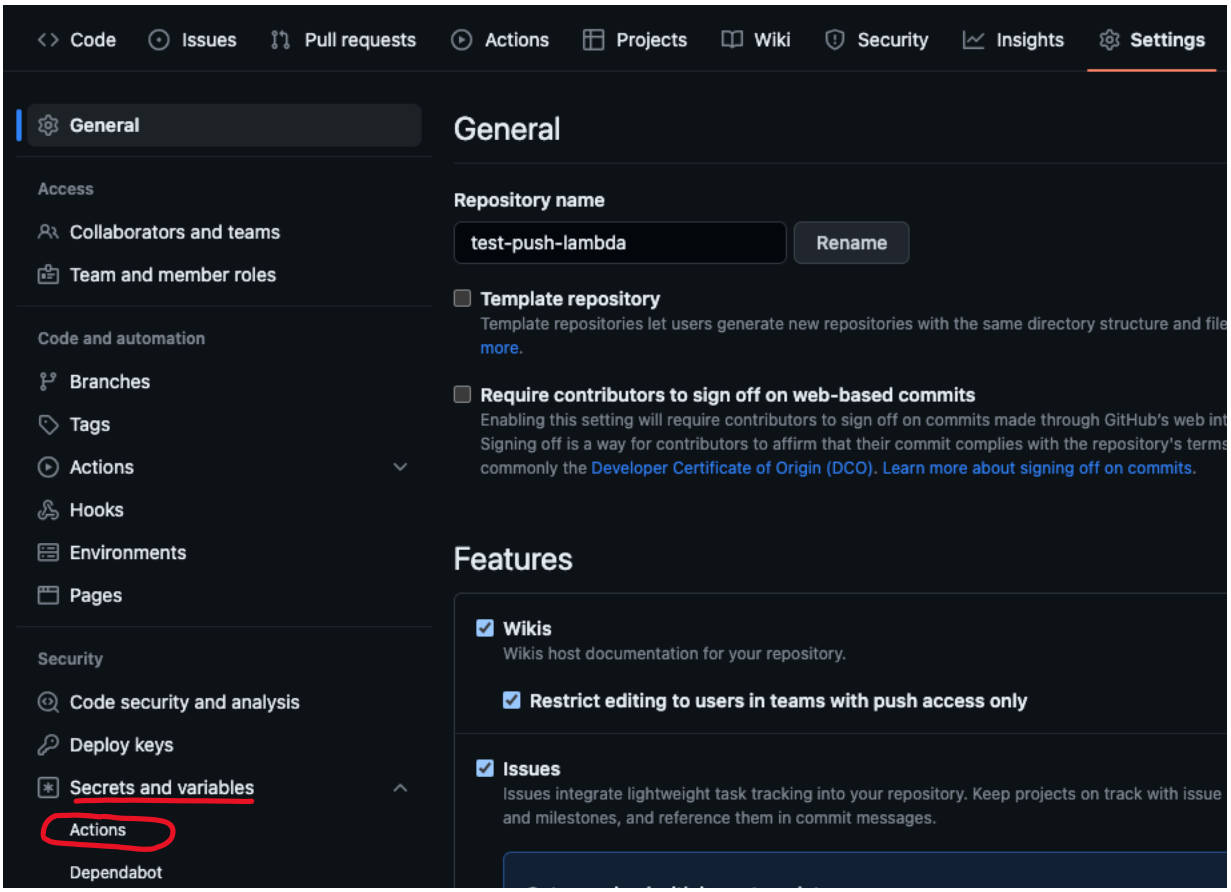




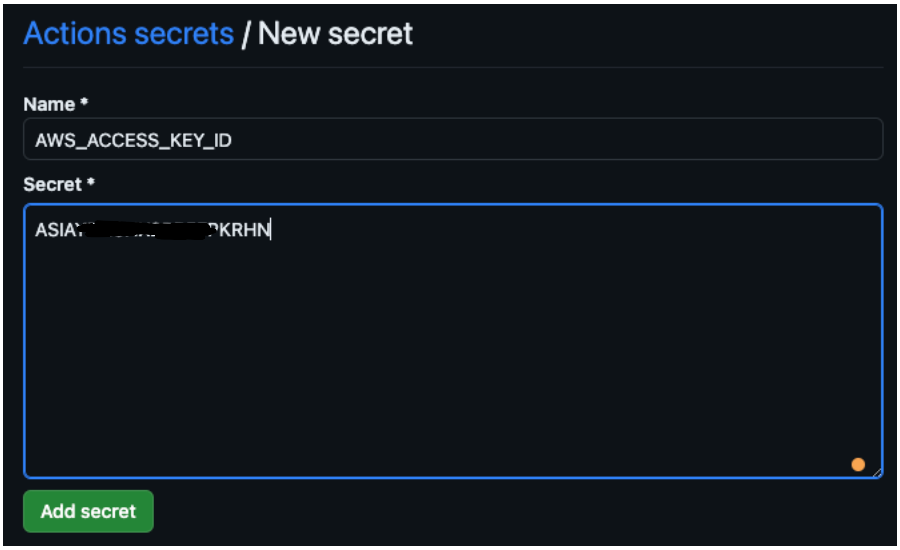
  - Copy the entire contents of the text box (including the ---- BEGIN RSA PRIVATE KEY ---- portion) into the EC2_SSH_KEY secret value.
  - Enter the public DNS for your EC2 instance into the REMOTE_HOST key value.
  - Enter "ec2-user" into the USERNAME key value.
  - Enter the location of your target folder (e.g. "/var/html/www") into the TARGET_DIR key value.
- To use the AWS CLI approach, click on "Show" AWS CLI.



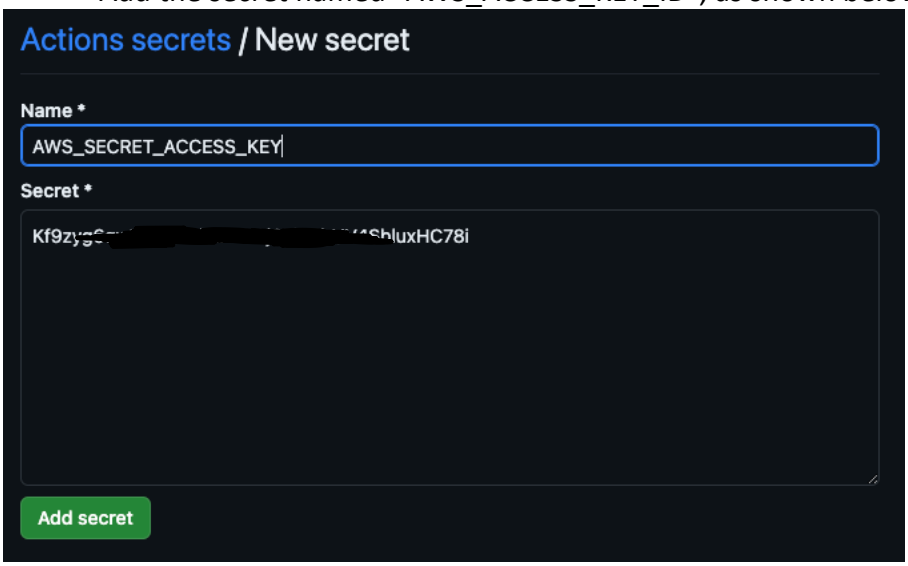

    - In the steps below, you are going to use the value of "aws_access_key_id" for AWS_ACCESS_KEY_ID secret, "aws_secret_access_key" for AWS_SECRET_ACCESS_KEY secret, and "aws_session_token" for AWS_SESSION_TOKEN secret.
- To create these GitHub Actions secrets for either approach, click on the "Actions" menu, as shown below.



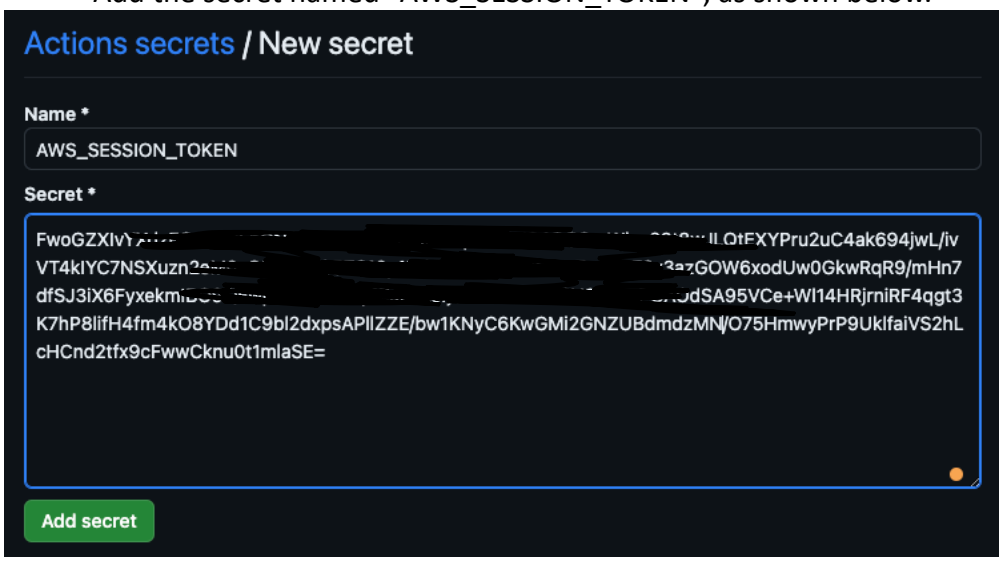

- On the subsequent screen, click the "New repository secret" button.
- Add the secret named "AWS_ACCESS_KEY_ID", as shown below.




- Add the secret named "AWS_ACCESS_KEY_ID", as shown below.



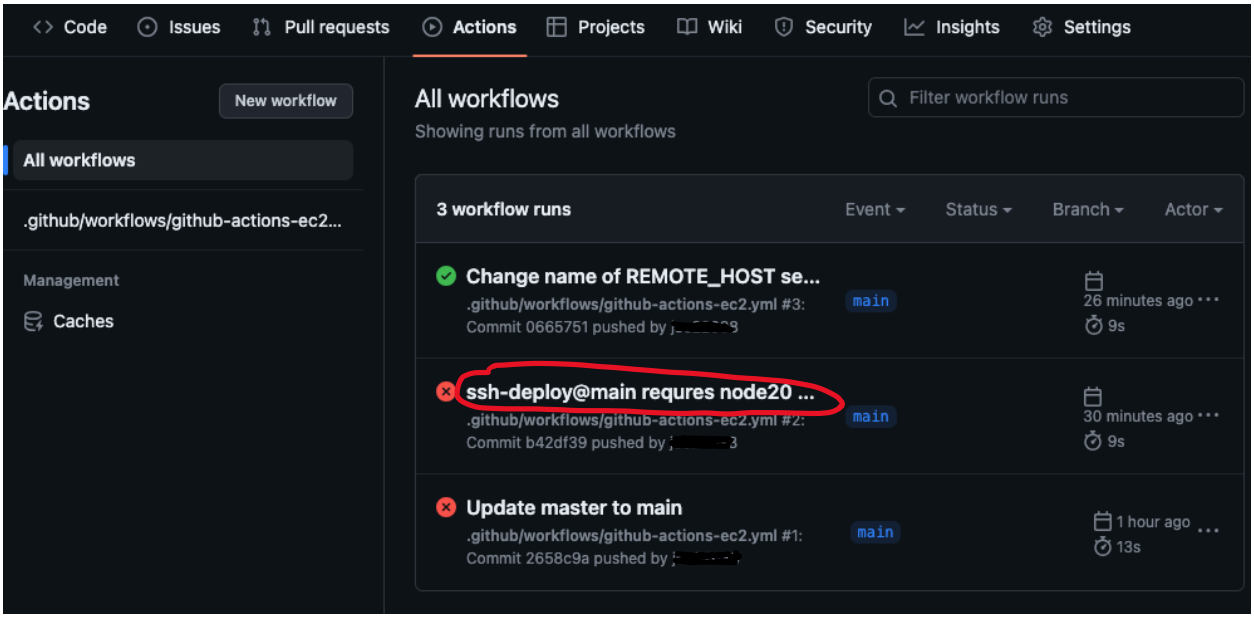

- Add the secret named "AWS_SESSION_TOKEN", as shown below.



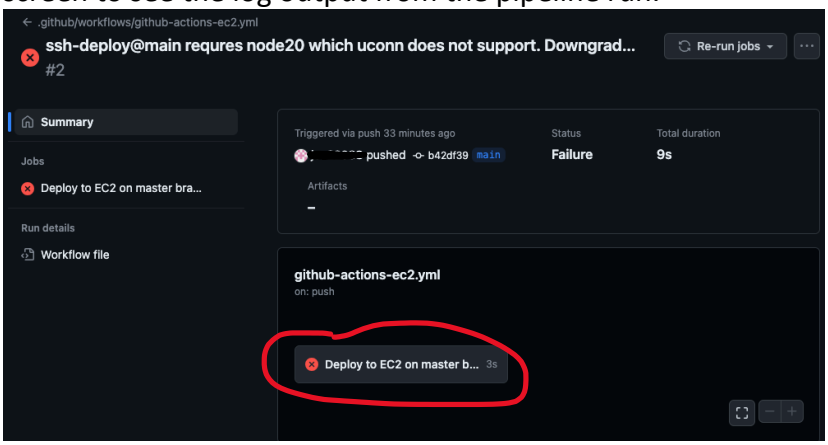

Note: If you wish to try the SSH approach, just use the instructions above and create secrets for EC2_SSH_KEY, REMOTE_HOST, USERNAME, and TARGET_DIR.

**Running your Github Action workflow**

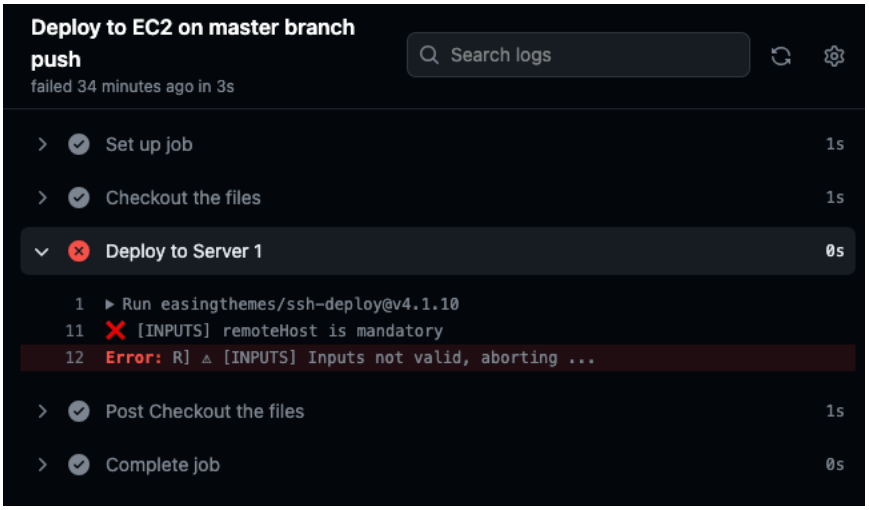
If you used one of the templates listed above, your Action pipeline will run whenever code is committed/merge/pushed to the indicated branch. You can view the status of your job on the Actions tab. Green check marks show successful pipeline runs. Red X's indicate errors.




Click on the text next to the check mark or X, and then click on the text again in the subsequent screen to see the log output from the pipeline run.




The following output shows the error encountered by this example pipeline. Use this information to debug your yaml file.




**Verifying Your Deploy to the remote Learner Lab Server**

- SSH into your EC2 server
- Navigate (cd) to the local folder that you specified in the yaml file

```
TARGET: ${{ secrets.TARGET_DIR }}
```

- Use the "ls -al" command to view the contents of folder and the timestamps of the files.
- Whether or not your new code is automatically hot-loaded to the running application server depends on how you have setup your application and/or EC2 instance or lambda functions. You may have to stop and restart your application manually.

References

- Useful GitHub Actions templates
- https://lightrains.com/blogs/deploy-aws-ec2-using-github-actions/