

[Link to 2024 ACAMP Wiki](#)

**Advance CAMP Fri. Dec 13, 2024**

**Room - V**

**Session Title: GitOps Topics (follow-up from Monday's tutorial)**

CONVENERS: Bruce Timberlake (Univ of Michigan), Matthew Economou (RDCT)

MAIN SCRIBE(S): Bruce Timberlake (Univ of Michigan),

ADDITIONAL CONTRIBUTORS:

# of ATTENDEES: 17

#### **DISCUSSION:**

Topics of most interest for Michigan: Dynamic AWS credentials, managing secrets in AWS and GitHub; when to use each. If possible, touch on dev environment topic, reproducible builds, and anything container-related.

<https://github.com/ResearchDataCom/get-good-with-gitops>

See *GitHub Releases* Section to find the Presentation slides (PDF, PPX)

See the [README.md in the repo](#) for help setting up the environment(s) at GitHub and AWS to run the tutorial yourself.

Manny (UT) - Interested in testing topics, esp automating tests

Nadia (Internet2) - Dynamic credentials

#### **Dynamic Credentials: GitHub OIDC Provider**

In the workshop, we used [ACT](#) to install GitHub runner “locally” - a testing infrastructure that you can run locally that allows debugging, etc. on a local workstation, then push+run on GitHub Actions.

Easiest way to handle credentials is to generate AWS id + secret credentials and put it into a .secrets file that is used by GitHub Actions. However, you have hardcoded credentials with all the problems that creates.

Better: Create a GitHub as an OIDC Provider (OP). And then create a IAM role that is allowed to log in as that OIDC provider. GitHub Actions workflow has its own GitHub provided login token, it has a auth. token that will work with GitHub in the background. It can also be configured to log into AWS and assume a role with the appropriate rights to set up AWS.

GitHub Actions has a token, which can authenticate itself with GitHub APIs. The Job may also use this token to log into the GitHub OIDC provider, which then can assume an AWS role- which gives it the ability to do things.

<https://github.com/ResearchDataCom/get-good-with-gitops/blob/main/ZZ-dynamic-credentials/github-actions.yaml>

- slides have not been created yet

AWS --> IAM --> Identity Providers

Best practice: Give the user specific roles needed to set up Deployments. However, this is hard and often people give the 'robot' Administrative access (bad).

Best practice: Keep your AWS account ID secret - as an attacker can make use of this id to make attacks to figure out your infrastructure. Note: S3 paths often include account ids.

## **Test-driven development**

Automate resources on your development computer/local workstation and create test databases (in this use case) and run the test on your framework.

<https://github.com/irtnog/lethbridge/blob/main/.github/workflows/ci.yml> (see test-matrix section)

Some key points

- You can use GitHub actions matrix strategy to create the test for multiple python versions. Or perhaps different OS Distributions. You can extend this into multiple-dimensions and create coverage for multiple criteria.

Q from audience: What is **pytest**

**pytest** is a program to allow us to mock up research and run tests in a structured manner. One writes functions that do individual tests: unit test, integration test, etc.

**pytest** arranges the tests for me. You don't need to write the CLI, you just write the test code.

Can create test fixtures, 'mock' endpoints such as a database, or even AWS to allow for pytest to test against the mock endpoint.

```
import re
import pytest
from moto import mock_aws

@pytest.mark.usefixtures("_mock_scoreboard")
@mock_aws
def test_html_form(socket_disabled):
    from simple_scoreboard import lambda_handler

    response = lambda_handler({}, {})
    assert re.search("(?i)<form[ >]", response["body"])
# link to code
```

Example: I have software that uses the AWS Lambda function. But I don't want to use the real Lambda because it has real data and can do real things. Instead you can create a mock service that represents this.

Python package [moto](#) helps with this test mock-ups.

See folders starting with 22- and 23-