DevOpsDreamin'

GitHub Continuous Integration in 30 minutes or less

DevOps Dreamin'

Sponsored by Gearset

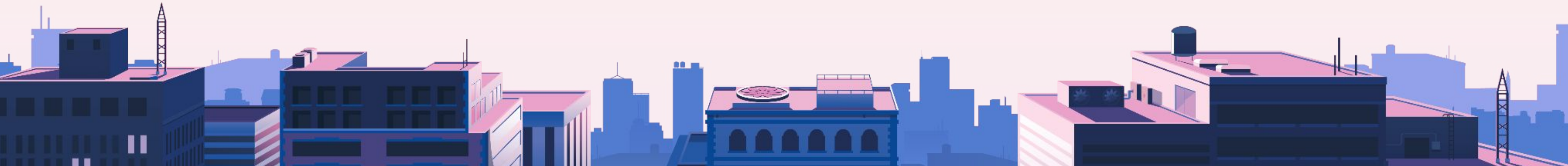
Matt Kaufman

CEO, MK Partners

After working at Salesforce from 2002-2005, Matt Kaufman founded MK Partners in 2006 with a vision to provide cutting-edge Salesforce solutions custom tailored to each customer. As one of the early Salesforce employees, Kaufman has been regarded as a leader on Salesforce for two decades, and has provided training and consulting to Salesforce employees and thousands of Salesforce users. He is the author of “Salesforce for Dummies” and “Learning Apex Programming” as well as several PluralSight courses. Kaufman holds over 25 Salesforce certifications and accreditations and over 600 Trailhead badges. Under Kaufman’s direction, MK Partners has provided services to thousands of companies, government agencies, and nonprofit organizations.

Skip to the end

https://github.com/the1mattkaufman/sfdx-githubci



Continuous Integration is neither continuous nor a database integration!

-Matt Kaufman, DevOps Non-Expert

Some GitHub Terms

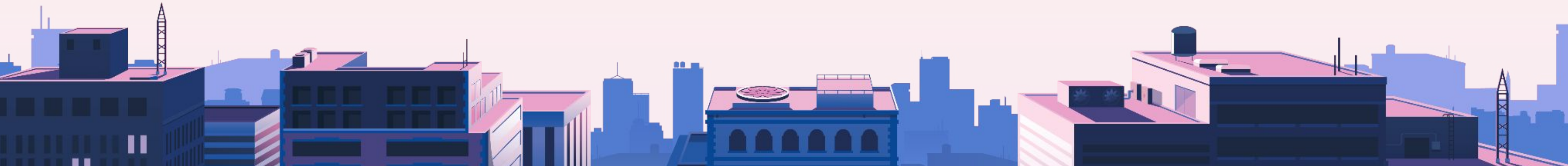
Repository (Repo): Like a project in GitHub, stores all your files

Branch: An isolated copy of all your files, repos have multiple branches

Commit: A change to a branch you're working on

Pull Request: A request to another branch to pull in changes from your branch

GitHub Action: A script that runs automatically when something happens in GitHub



Salesforce Environments

Sandboxes

Come with all production metadata

May come with some/all/none of production data

Must be refreshed sometimes

Can take time to create

Scratch Orgs

Very short lived

Come with no metadata/data

Created very quickly

Require a dev hub org

Production

Long Lived

There can be only one

Mistakes will result in your head being cut off

Success is electrifying

*Yes, Developer and Playground orgs exist, for our purposes they're like long-lived Scratch Orgs

Why bother with Continuous Integration

Code Quality - Prettify / Linting / Source Code Analysis (Apex PMD)

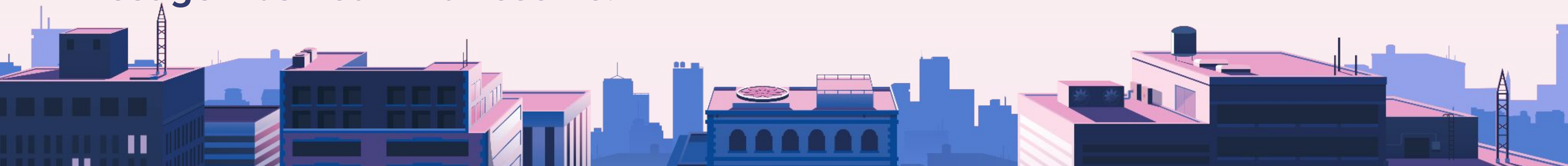
Automation - Testing / Rules / Deployment

Validation - Apex Tests / Jest Tests

Visibility - Status Badges / Code Coverage Reporting

Governance - Track Changes / Proven Process / Version Control

Prestige - It's freakin' awesome!



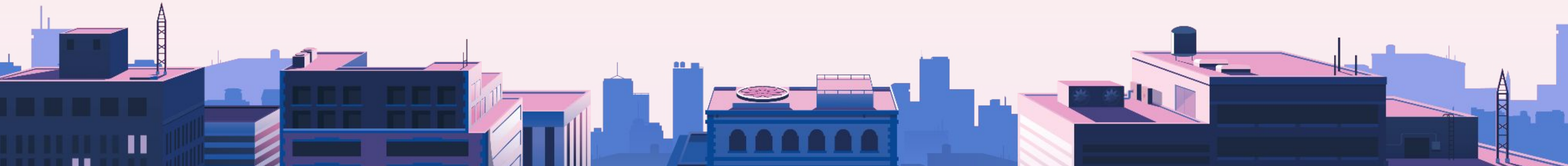
Why avoid Continuous Integration

Speed - A formal process is slower than making changes directly

Control - You want to do things your way, rules be damned

ROI - You never make changes to Salesforce anyway, so why invest in CI

Time - You're busy enough already



DevOps Dreamin'

Sponsored by Gearset

We follow rules and processes in every aspect of our lives. Salesforce Development and Deployment should be no exception.

-Matt Kaufman, Quote Generator

DevOps Dreamin’

Sponsored by Gearset

Development Models

Change Sets

Admin Friendly / Built-In

No Automation / Scheduling

No Destructive Changes

No Source Control

Super frustrating to work with

Package Development

Source Control Based

CI/CD Friendly

Enhancements are built and deployed independently of each other

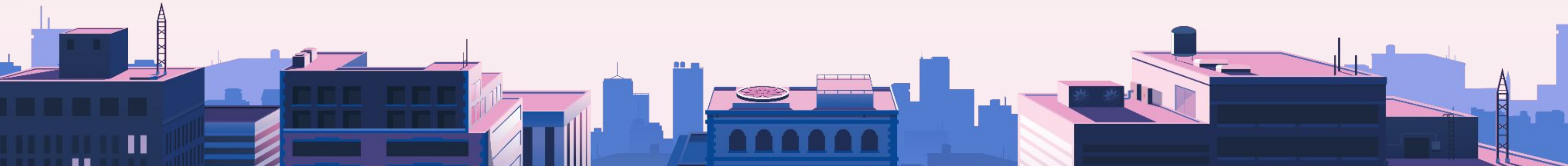
Org Development

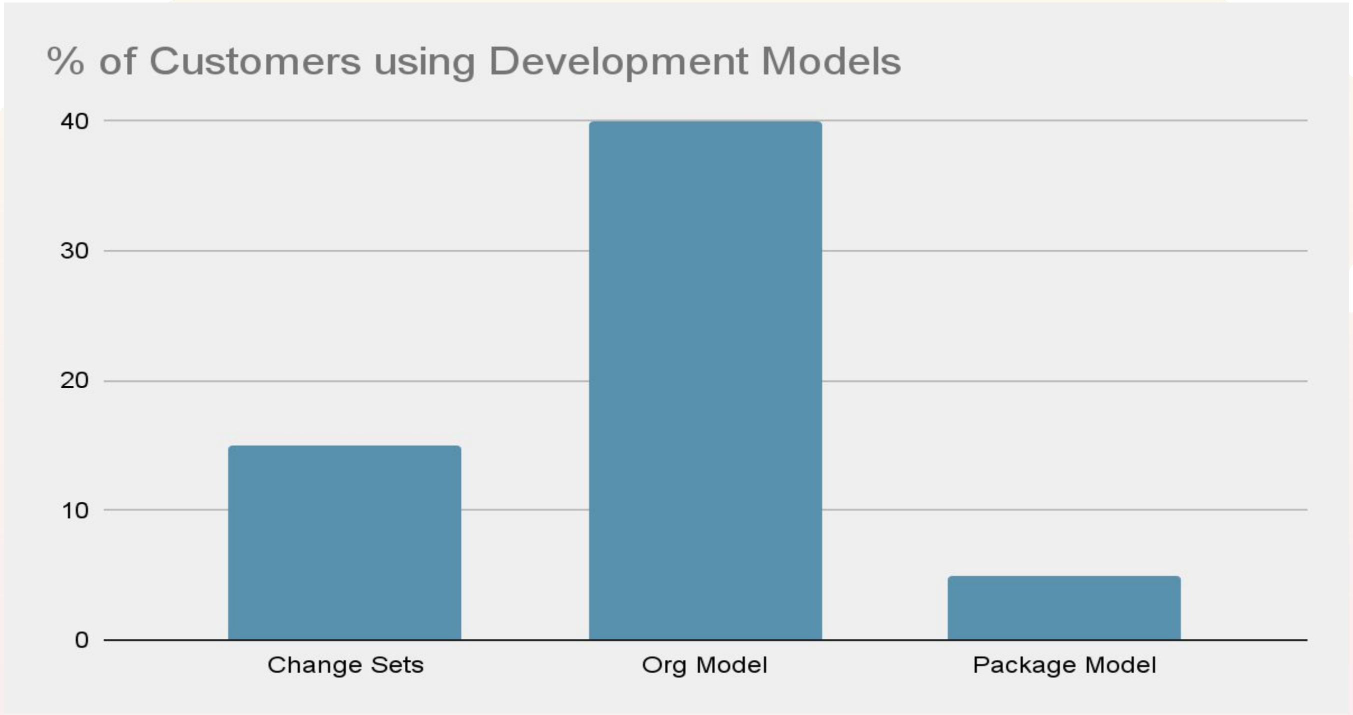
Source Control Based

CI Friendly

Holistic Org-Wide Approach

Big Releases of multiple enhancements at once




% of Customers using Development Models
40
30
20
10
0
Change Sets
Org Model
Package Model

Continuous Integration only works when everyone follows the rules every time.

Human Process Rules!!!

Changes are never made directly to Prod (or Staging or UAT or QA)

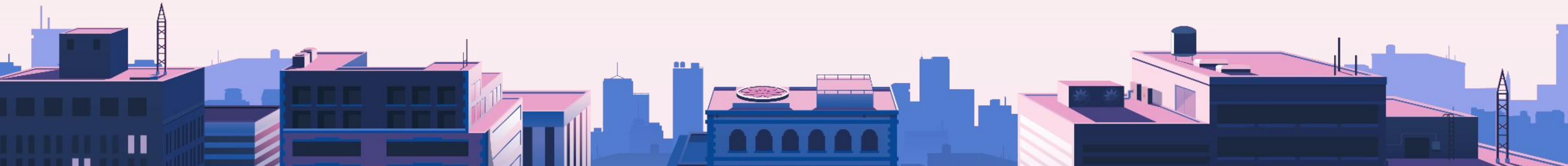
Changes are made in “dev” orgs and committed to source control

Changes are tested and approved before deployed to higher environments

Changes are documented ahead of time with passing criteria, etc

Changes are never made directly to Prod (or Staging or UAT or QA)

*Bold = Absolutely required



DevOps Dreamin'

Automated Process Rules

.github/workflows/*.yml

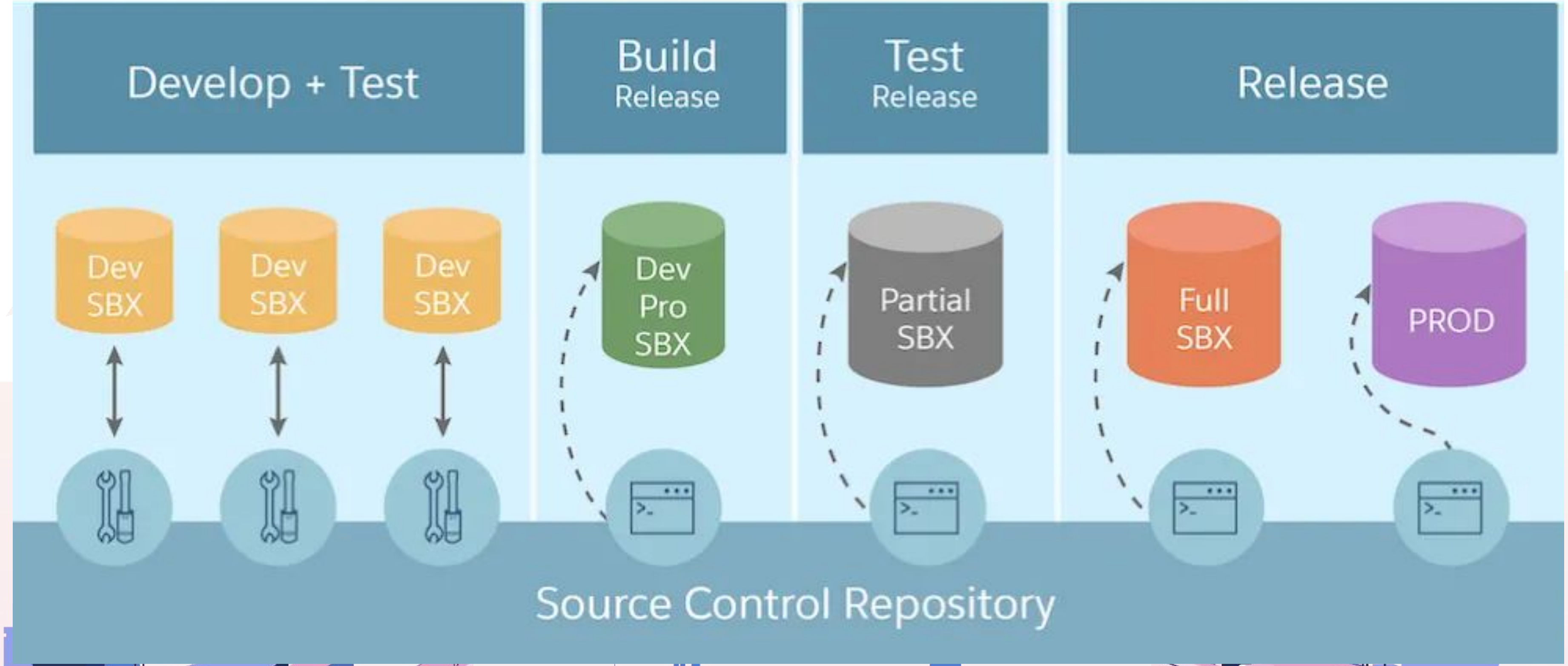
```
name: push to uat
run
on:
  push:
    branches:
      - uat


jobs:
  formatting-and-linting:
    steps:
```

DevOps Dreamin’

Branch Protection Rules

- Require Pull Requests
- Require Approvals
- Require Status Checks
- Restrict who can do what
- And more


Develop + Test
Build Release
Test Release
Release
Dev SBX
Dev SBX
Dev SBX
Dev Pro SBX
Partial SBX
Full SBX
PROD
Source Control Repository

High Level Steps to setup GitHub CI for Salesforce

Salesforce

Lock down permissions

Create sandboxes

Train your users

sfdx

Auth into your orgs

Retrieve auth urls

Github

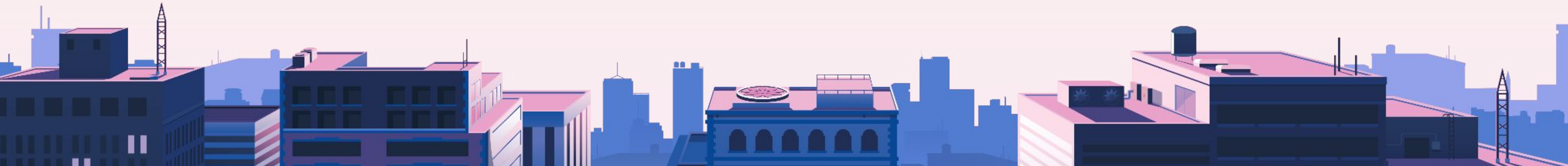
Create Repo

Create Branches

Create .yml files

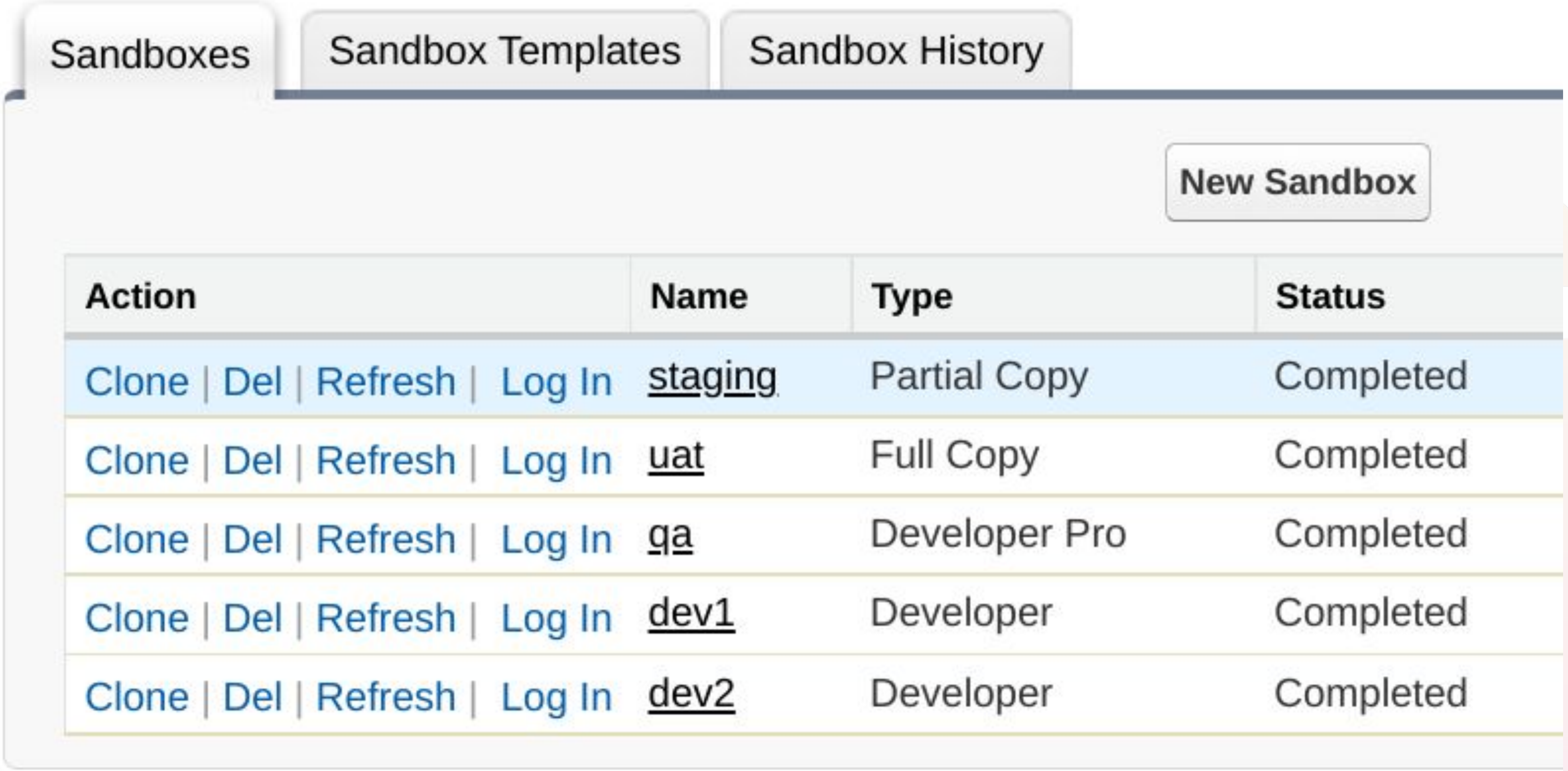
Create Branch Rules

Store auth urls in Secrets



DevOps Dreamin'

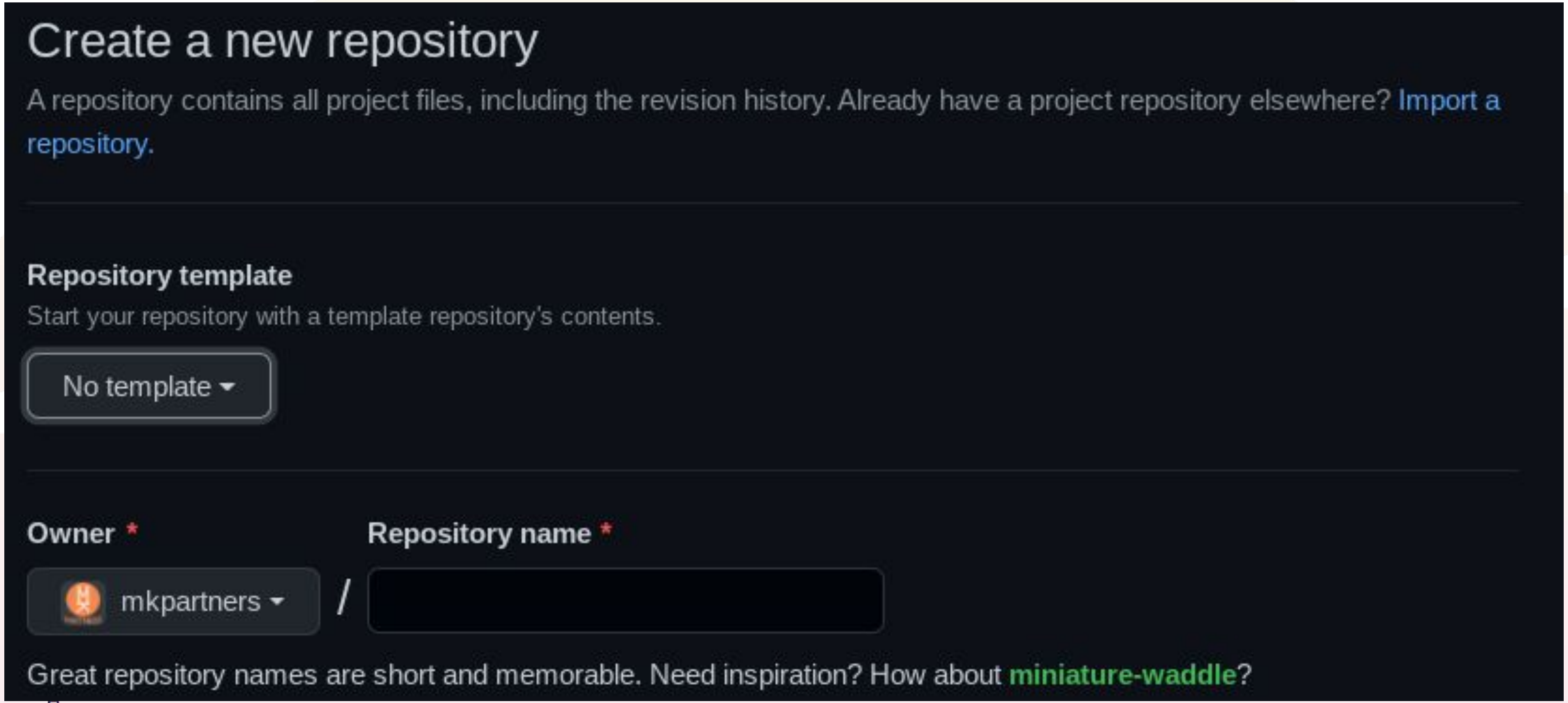
Sponsored by Gearset



Action	Name	Type	Status
Clone | Del | Refresh | Log In	staging	Partial Copy	Completed
Clone | Del | Refresh | Log In	uat	Full Copy	Completed
Clone | Del | Refresh | Log In	qa	Developer Pro	Completed
Clone | Del | Refresh | Log In	dev1	Developer	Completed
Clone | Del | Refresh | Log In	dev2	Developer	Completed

DevOps Dreamin'

Sponsored by Gearset


Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Repository template

Start your repository with a template repository's contents.

Owner *

/

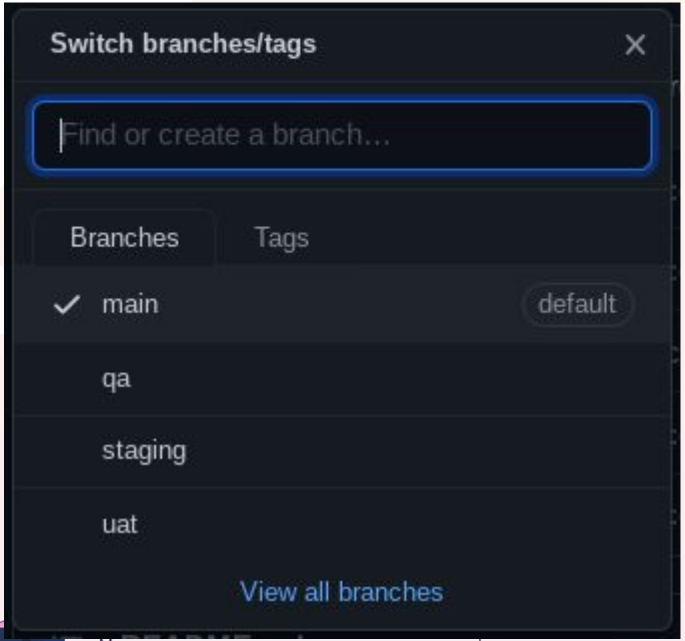
Repository name *

Great repository names are short and memorable. Need inspiration? How about miniature-waddle?

DevOps Dreamin'

Sponsored by Gearset

Branches



main: Production

staging: This Release

uat: User Testing

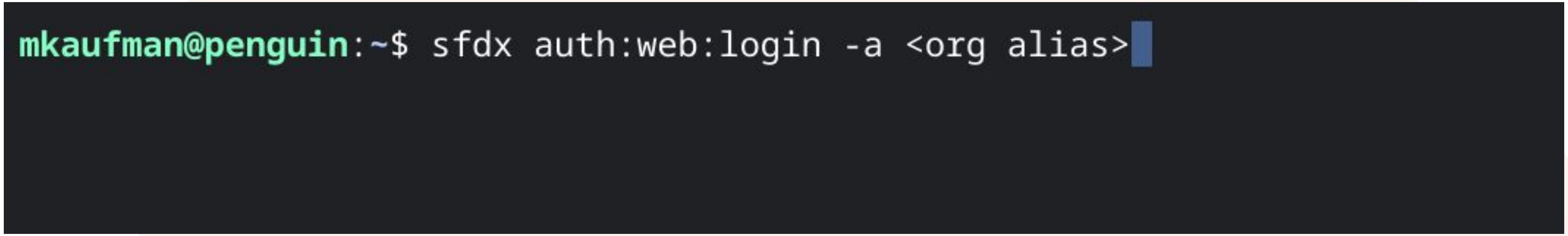
qa: Developer Testing

<devx>: feature/fix specific

DevOps Dreamin'

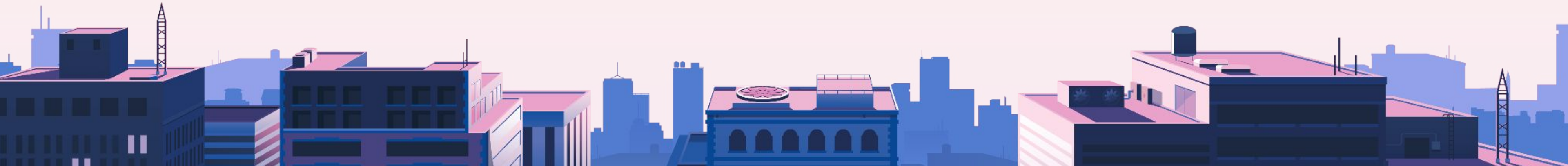
Sponsored by Gearset

Easily Authenticate into your orgs



```
mkaufman@penguin:~$ sfdx auth:web:login -a <org alias>
```

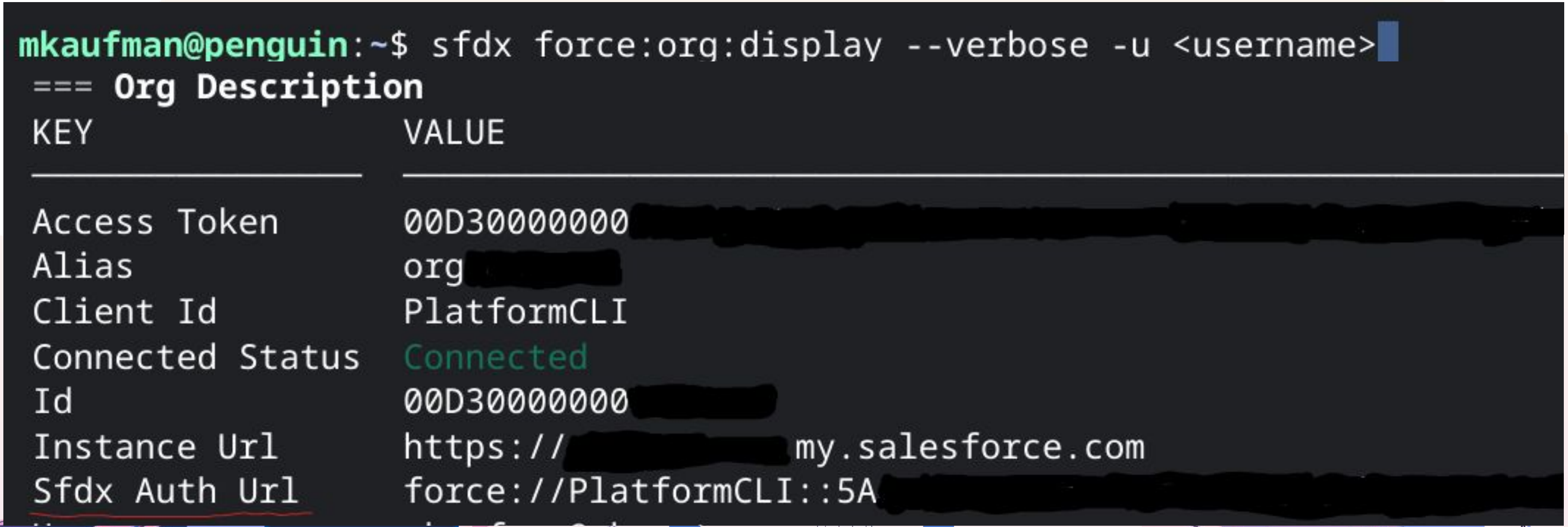
Note: You will need to redo this for sandboxes after they are refreshed



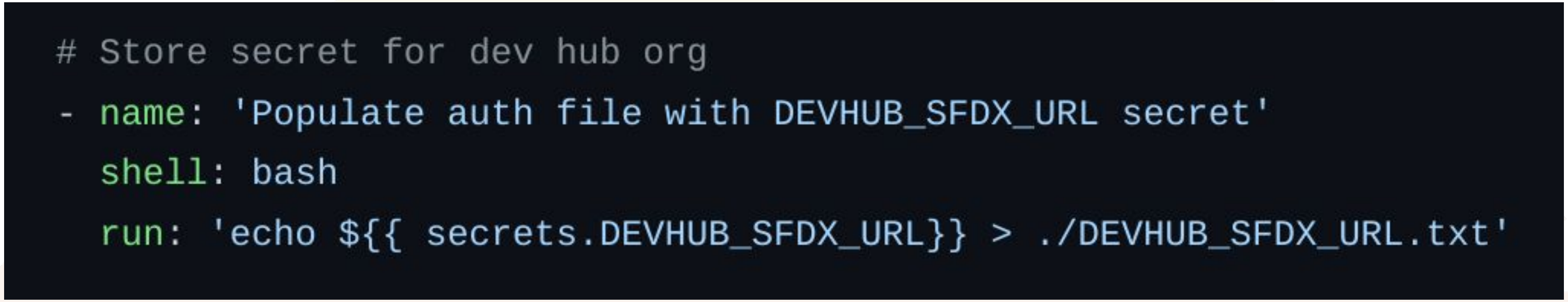
DevOps Dreamin'

Sponsored by Gearset

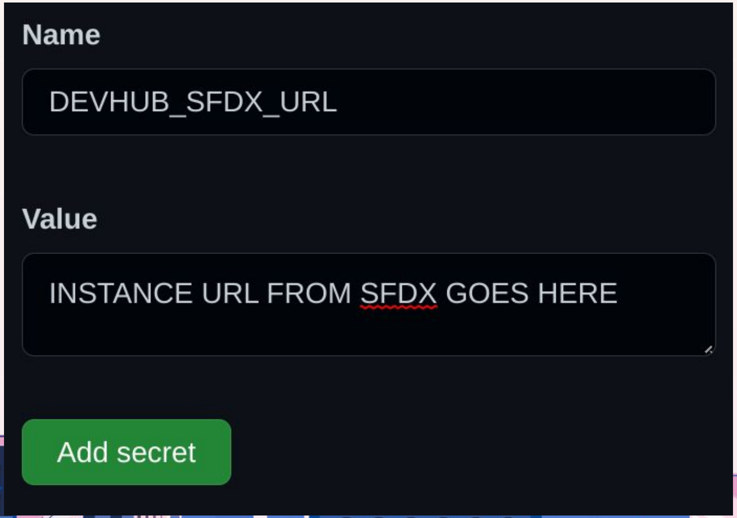
Super Easy way for Github to authenticate



```
mkaufman@penguin:~$ sfdx force:org:display --verbose -u <username>
 === Org Description
 KEY                VALUE
 ─────────────────  ──────────────────────────────────────────
 Access Token       00D30000000[redacted]
 Alias              org[redacted]
 Client Id          PlatformCLI
 Connected Status   Connected
 Id                 00D30000000[redacted]
 Instance Url       https://[redacted]my.salesforce.com
 Sfdx Auth Url      force://PlatformCLI::5A[redacted]
```



```yaml
# Store secret for dev hub org
- name: 'Populate auth file with DEVHUB_SFDX_URL secret'
  shell: bash
  run: 'echo ${{ secrets.DEVHUB_SFDX_URL}} > ./DEVHUB_SFDX_URL.txt'
```



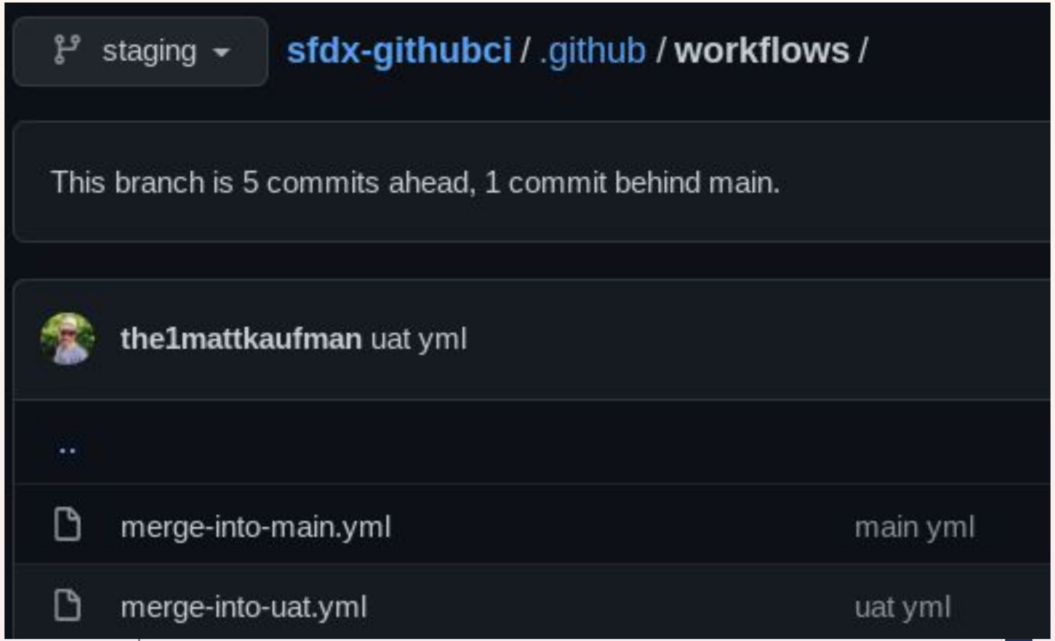
Name

DEVHUB_SFDX_URL

Value

INSTANCE URL FROM SFDX GOES HERE

.yml Files are stored in .github/workflows/


staging

sfdx-githubci / .github / workflows /

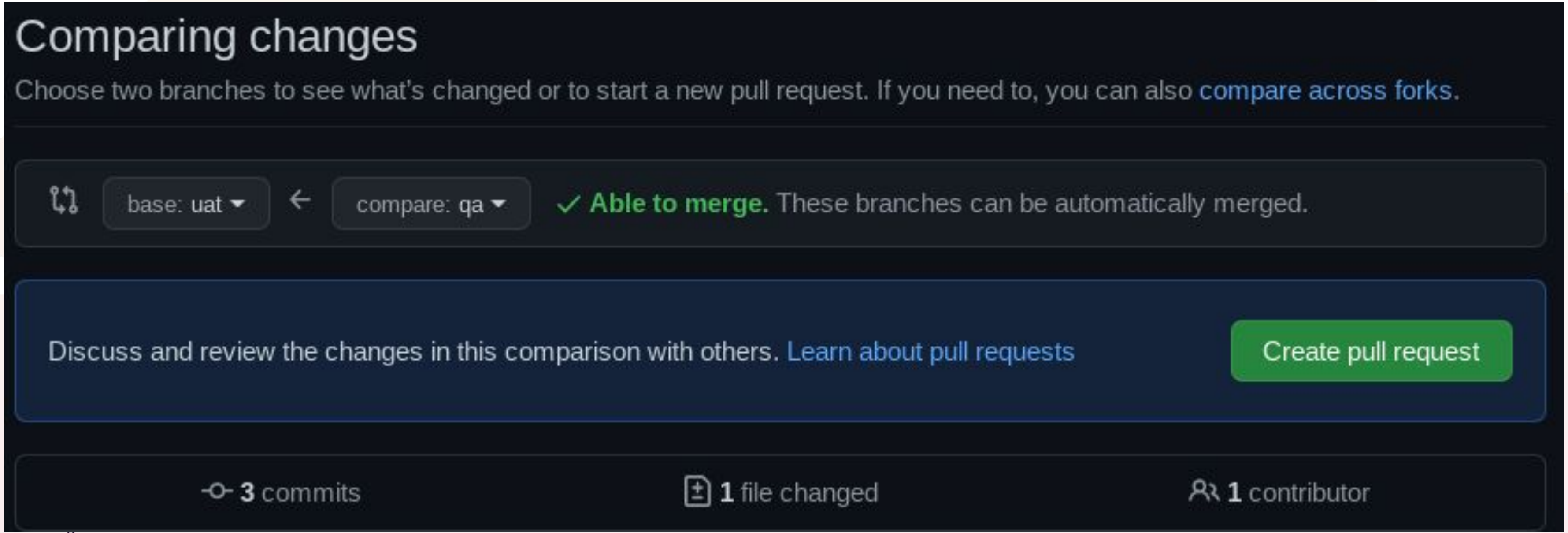
This branch is 5 commits ahead, 1 commit behind main.

the1mattkaufman uat yml

..

merge-into-main.yml	main yml
merge-into-uat.yml	uat yml

Creating a Pull Request



Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also compare across forks.

base: uat ← compare: qa ✓ Able to merge. These branches can be automatically merged.

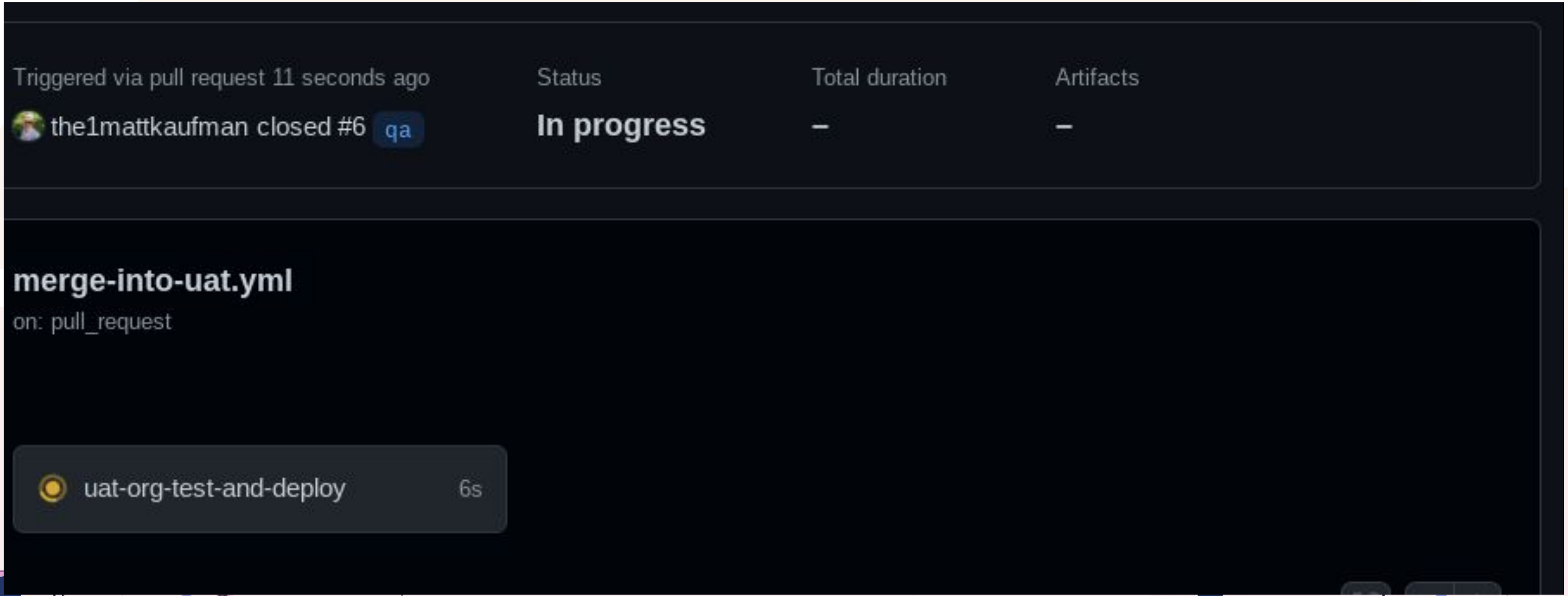
Discuss and review the changes in this comparison with others. Learn about pull requests

3 commits · 1 file changed · 1 contributor

DevOps Dreamin'

Sponsored by Gearset

GitHub Actions Run Automatically



Triggered via pull request 11 seconds ago

the1mattkaufman closed #6 qa

Status

In progress

Total duration

–

Artifacts

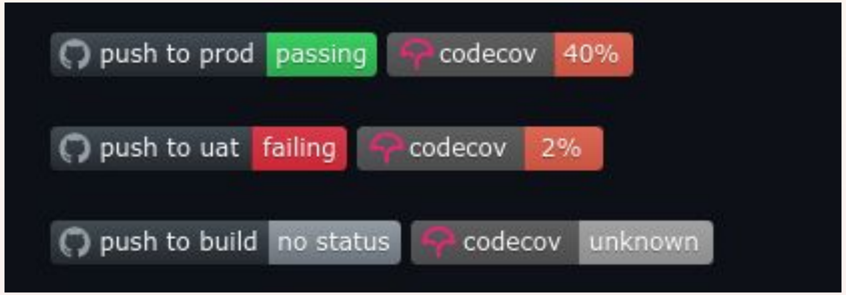
–

merge-into-uat.yml

on: pull_request

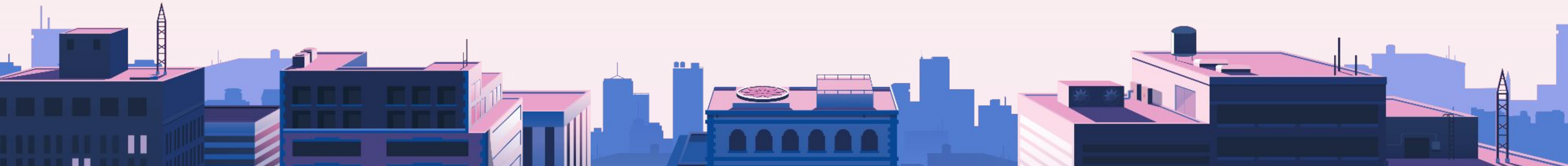
uat-org-test-and-deploy 6s

Pretty Badges for Status at a glance



Sample Repo

https://github.com/the1mattkaufman/sfdx-githubci



DevOps Dreamin'

Sponsored by Gearset

Thank you!