# GitHub GHEC to GHEC EMU Migration

We plan to migrate all the old GitHub repos from **[source-org]** to a new **[Client]** enterprise org in GitHub. The new **[Client]** org will have Enterprise Managed Users. This means that all our users will be automatically created from our **[Client]** login. All repos will be migrated automatically.

## When

The migration will start on **[Start Migration Date]** and end on **[End Migration Date].**

## What's Changing

1. **GitHub URL**

**Old**: https://github.com/**[source-org]** https://github.com/**[target-org]**

**New**: https://github.com/**[source-org]**

1. **User Id**

**Old**: created by each developer when signing up for GitHub.

* e.g. **[Primary Contact GitHub Handle]**

**New**: automatically created by **[source-org]**. Format is firstname-lastname\_atclient.

* *e.g. sam\_peddamatham\_atclient*

1. **Repo name**

We need to follow standard naming convention in the new GitHub instance. Please see full details at Source Control Policy.

In short, we will prefix each repo with engineering- to the existing repo name.

* e.g. an old repo named HelloWorld in **[source-org]** will be named engineering-HelloWorld during migration.

All new repos created in the new GitHub must follow the naming convention outlined in Source Control Policy

1. **Team name**

GitHub team names will follow the same convention as repo name. All exiting teams will be migrated to the new GitHub with an engineering- prefix.

## Actions Needed Your Team

The following actions need to be performed by each team after the migration.

1. **Login to new GitHub**

GitHub allows login with multiple accounts at the same time. It might be less confusing if only one login is used at a time. Login with the new **[Client]** enterprise by entering the new user id.

Click on 'Sign in with your identity provider'. You shouldn't need to enter a password. After entering the user id, it will detect that this is setup for SSO.

A screenshot of a login form

Description automatically generated

**Team leads:** Please make sure each developer is able to login to the new GitHub before the migration. For any issues, please contact **[Primary Client Contact]**.

1. **Local repo folder**

All existing repo folders will have to be updated so that it points to the new GitHub account. This can be done in one of two ways.

Clone repo from new GitHub account

1. Delete or rename the old repo folder
2. Login to new GitHub
3. Search for the repo and copy the clone URL
4. Clone it
   * Using command-line:

git clone https://github.com/source-org/repo.git

* + Using UI tools e.g. Visual Studio, VS Code, Sourcetree etc.

Update the remote in the old repo

Given the old repo is https://github.com/source-org/hello.git and new repo is https://github.com/target-org/engineering-hello.git

* Command line

git remote set-url origin https://github.com/source-org/repo.git

* UI tools
  + This functionality can vary depending on the tool being used

1. **Update pipelines in Concourse, Jenkins**

This affects only the pipelines which pull repos from the old GitHub

* Update repo URL in the pipeline to the new GitHub URL
* Update any old GitHub auth token used by the pipeline

1. **Update pipelines in GitHub Action**

We will need to update the common variable and secrets names in the workflow for each repo that uses GitHub Action to build and deploy.

Details - TBD