

Packaging Wizard

Setup Guide

Prerequisites

Before installing the package:

- **Enable My Domain:** Please follow [Salesforce documentation](#) to enable My Domain in the org where the Packaging Wizard will be installed.

Each of the following, while not required to install the Packaging Wizard, is necessary to set up and run packaging pipelines.

- **Enable DevHub:** The package must be run in a DevHub-enabled org. For details on enabling DevHub see the official documentation [here](#).
- **Link a Namespace:** If you haven't already done so, you will need to link a Namespace to your DevHub org. Documentation for linking and creating Namespaces can be found [here](#).

Heroku and GitHub are the tools used in the Packaging Wizard to provision and run CI/CD pipelines. In the future, the app will support more vendors.

- **Heroku Account:** [Sign up](#) for an account if you don't already have one.
- **GitHub Account:** You can sign up [here](#)

Quick Start

Configure Main Settings

Main settings

1

DevHub
Current organization's credentials which will be used as a DevHub

Connect

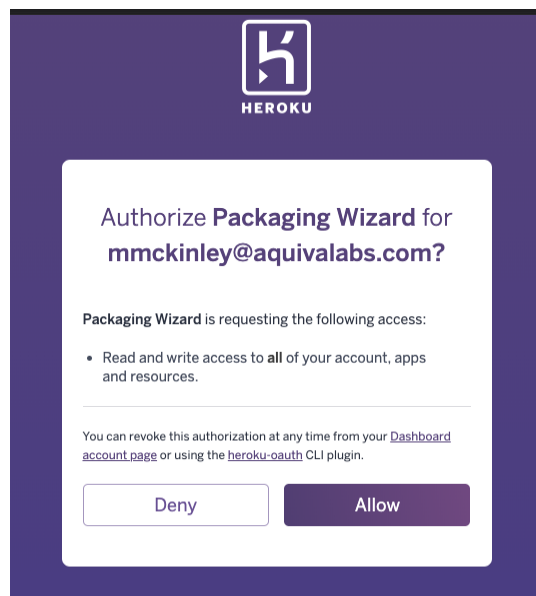
2

Connect Heroku
Heroku account which will host CI pipeline

Connect

1. **Connect the current org:** This requires the username, password, and security token of a user in the current org. **Make sure DevHub is enabled for this org.** This org will be used to spin up scratch orgs to run CI in.

2. **Connect Heroku:** If you have not yet logged into Heroku, you will be prompted to login. Next, you will be asked to Allow the Packaging Wizard read/write access. Click **Allow**.

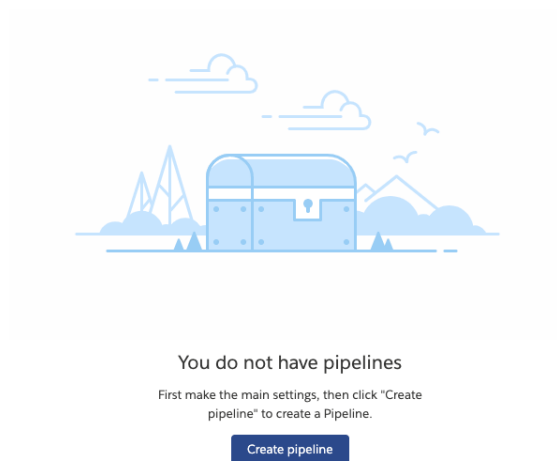


Connect External Orgs

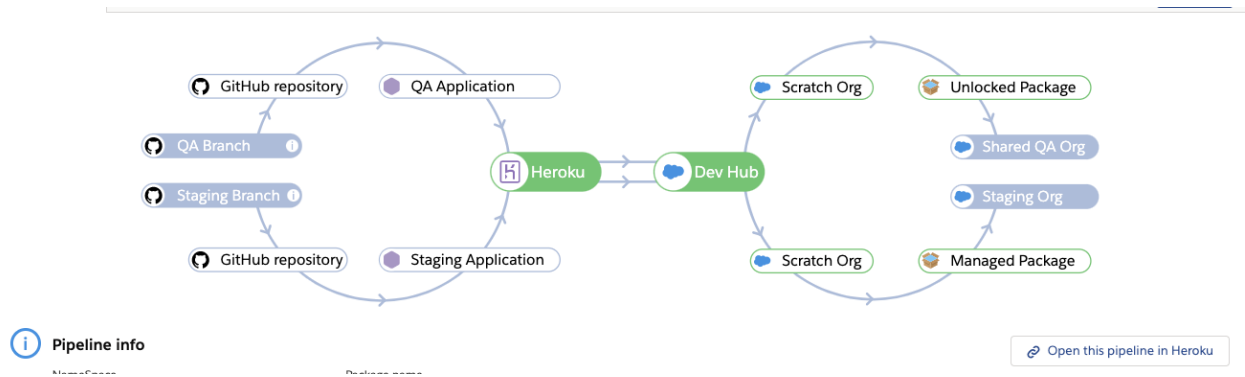
Connected orgs	New
Staging	
QA	

3. Connect SFDC Orgs: You can create as many connections as you like by clicking New. The best practice is to connect two orgs – one for the QA/develop branch and one for the Staging/release branch of your package. In a future step, you will select orgs from this list as target orgs for package installations.

Create and Configure the Heroku Pipeline



3. Create the Heroku Pipeline: Give your pipeline a name and associate a namespace with your package. This Namespace will be used as the “namespace” attribute in the sfdx-project.json file when building your package. Once complete, a Heroku Pipeline will be created in your Heroku instance. After creation you should see the following:



4. Connect GitHub to your Heroku Account: Your Heroku account may already have an associated GitHub account in which case you can skip this step. Otherwise, you will see a prompt with a link to your Heroku Connection settings. Click the provided link and choose the Github account containing the repository with the source code for your package.

⚠ Your Heroku account does not connected to GitHub. Please connect. [Open Heroku connections settings](#)

Pipeline info

Namespace
packwiz

Package name
myDreamPackage

[Open this pipeline in Heroku](#)

5. Select a Repository and Choose Branches: After choosing your repo you will select the following two branches:

- i. **QA Branch:** This is typically the *develop* branch – the branch representing the stable versions of source in the development lifecycle. Any commit to this branch will be incorporated into an a version of an Unlocked Package and installed into the QA Org.
- ii. **Staging Branch:** This is typically the *master* or *main* branch – the branch representing the release lifecycle. Any commit to this branch will be incorporated into a managed package version and installed into the Staging Org.

1 Repository

Select repository

mattaGIT/dreamhouse-sfdx ▼

QA Branch

develop ▼

Staging Branch

master ▼

6. Choose a Buildpack: To start, it is recommended to choose the “Default” buildpack. You can learn more about buildpacks [here](#). In our case, a buildback is a set of scripts responsible for packaging your source into a deployable package.

2 Buildpack

Buildpacks are responsible for transforming deployed code into a slug, which can then be executed on a dyno.

Choose buildpack

☐ Default

☐ Custom

7. Map an Org to the QA Branch: This can be any org you connected previously in step 2. However, it should be an org where you intend to test the **unlocked** version of your package.

3 QA

Organization which will be used for manual testing activities during development

Connected org

Select connection ▼

8. Map an Org to the Staging Branch: This can be any org you connected previously in step 2. However, it should be an org that you intend to test the **managed** version of your package in.

4 Staging

Organization which will be used for the acceptance testing before distribution

Connected org

Select connection ▼

9. Save your configuration. Click the blue save button in each setup step to save your work (steps 5-8).

Note: you can update these sections in the future. All changes are automatically reflected in the Heroku pipeline.

The screenshot shows the Heroku Pipeline configuration interface. At the top, a green notification banner reads "Save successfully!". The "Pipeline" section shows "NameSpace" as "packwiz" and "Package name" as "myDreamPackage". The "Repository" section shows "Select repository" as "mattaGIT/dreamhouse-sfdx", "QA Branch" as "develop", and "Staging Branch" as "master". The "Buildpack" section is partially visible, showing "Buildpacks are responsible for transforming deployed code into a slug, which can then be executed on a dyno." and "Choose buildpack". A large, semi-transparent watermark "Setup completed!" is overlaid on the right side of the form.

10. Enjoy!