Continuous Delivery

Goals

This lab will guide you through building a BASIC continuous delivery pipeline using Jenkins, Artifactory, and Cloud Foundry.

Estimated time: 30 minutes

Exercises

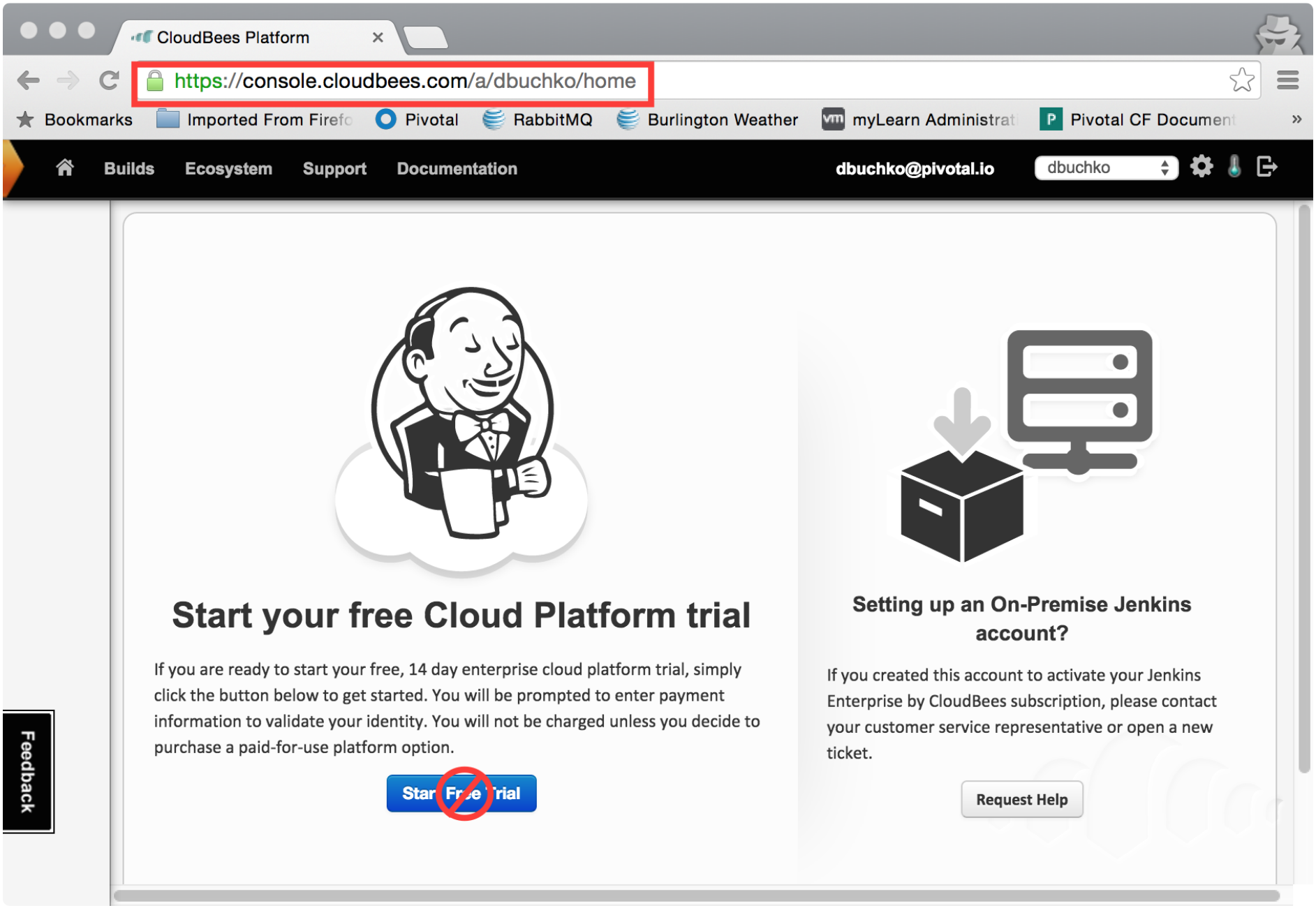
Fork an Existing Maven Project

For this lab, you can use any Github Maven project. If you don't have one, log into your Github account, navigate to the CloudFoundry-EnvironmentDemo (https://github.com/S2EDU/CloudFoundry-EnvironmentDemo) project, and fork it to your account.

Establish a CloudBees Account

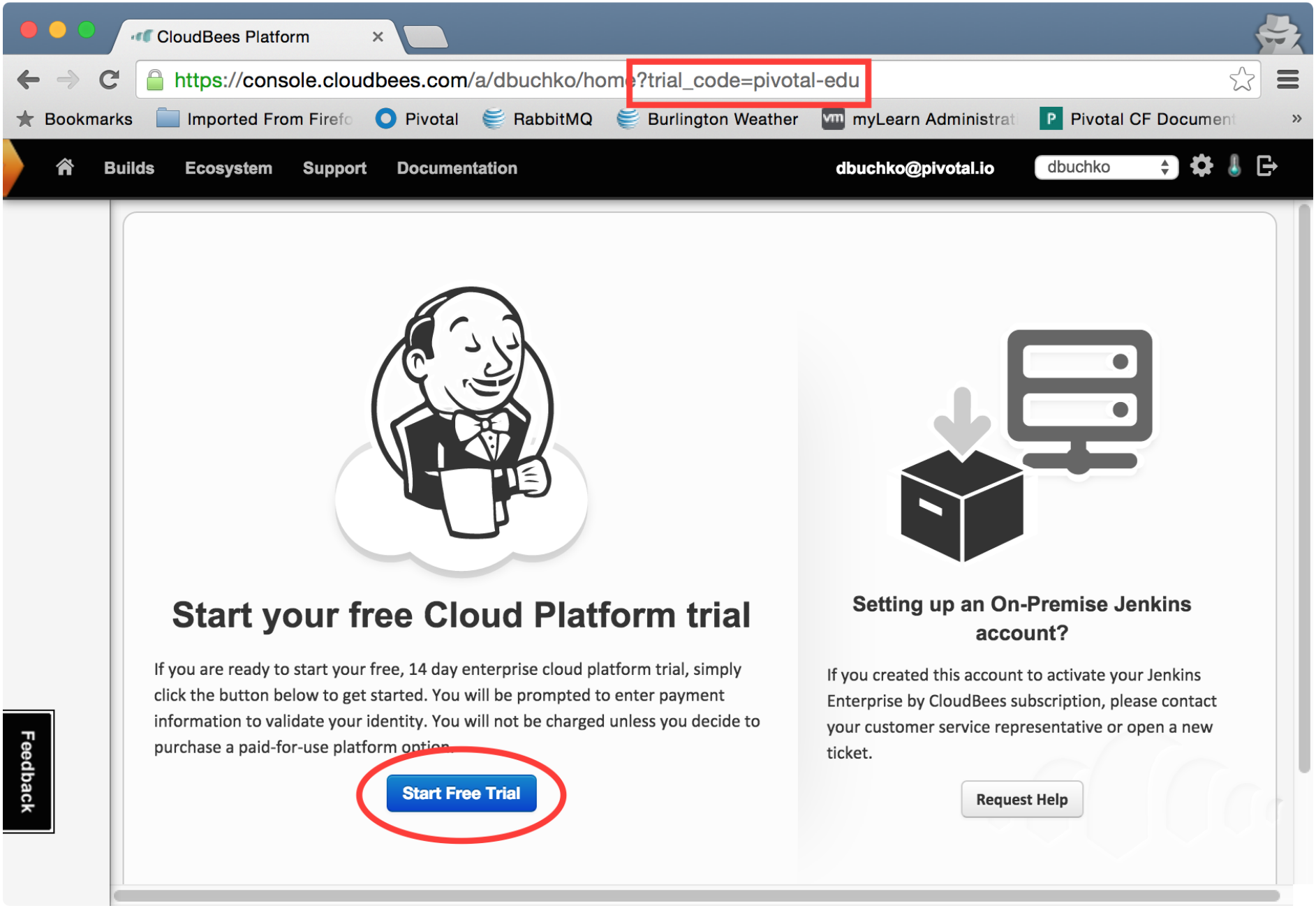
CloudBees is a SaaS provider for Jenkins, a tool for creating and managing automated builds.

1. Signup for access to CloudBees: https://grandcentral.cloudbees.com/login. Setup your account using one of the options available. After signing in, your should be taken to your account home page. Do **NOT** click "Start Free Trial" at this time.


CloudBees Platform
https://console.cloudbees.com/a/dbuchko/home
Bookmarks
Imported From Firefo
Pivotal
RabbitMQ
Burlington Weather
myLearn Administrat
Pivotal CF Document
Builds
Ecosystem
Support
Documentation
dbuchko@pivotal.io
dbuchko
Start your free Cloud Platform trial
If you are ready to start your free, 14 day enterprise cloud platform trial, simply click the button below to get started. You will be prompted to enter payment information to validate your identity. You will not be charged unless you decide to purchase a paid-for-use platform option.
Setting up an On-Premise Jenkins account?
If you created this account to activate your Jenkins Enterprise by CloudBees subscription, please contact your customer service representative or open a new ticket.
Request Help
Feedback

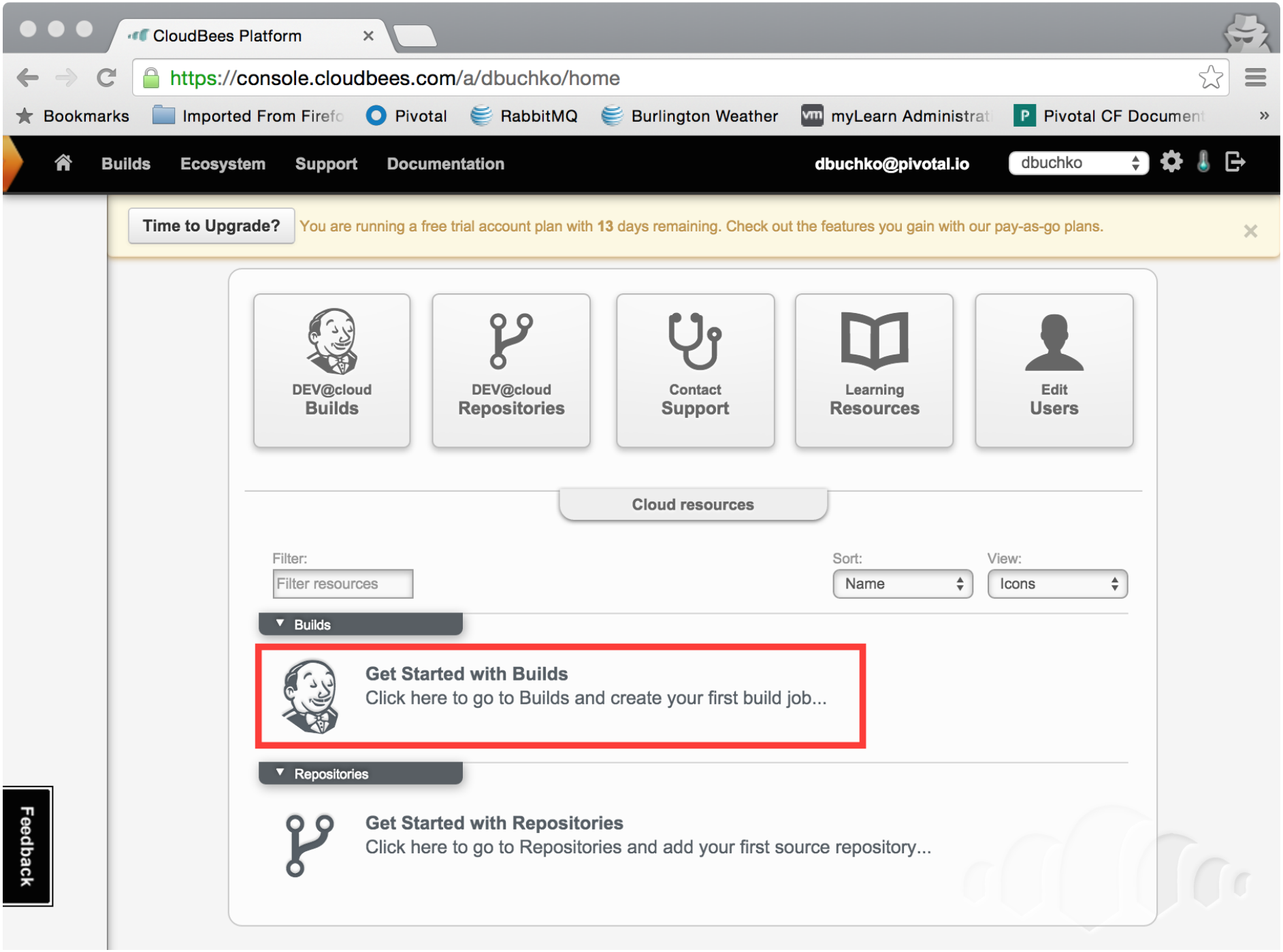
2. In your browser, append `?trial_code=pivotal-edu` to the end of the URL. So for example, your home page URL will be something like https://console.cloudbees.com/a/XXX/home (where XXX is your organization name). Your modified URL will be https://console.cloudbees.com/a/XXX/home?trial_code=pivotal-edu. Hit enter to submit the new

URL.



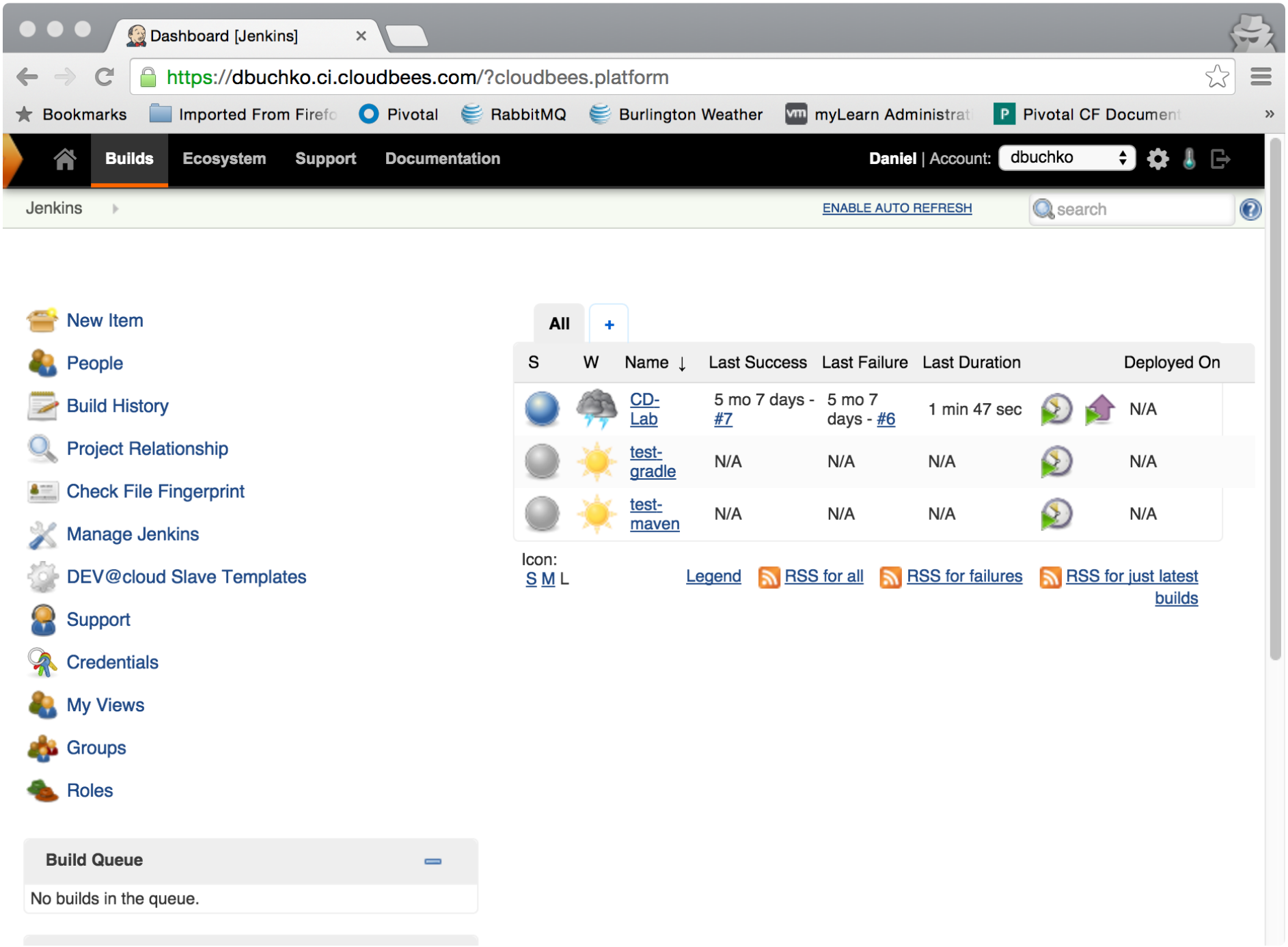
3. **Now** click on "Start Free Trial".

4. Click on the "Get Started with Builds" link.



5. At this point a blue progress bar appears while your Jenkins instance is provisioned.

   Warning: Registering for the trial often takes longer than the time to refresh the page. If the previous page appears, you may have to click the "Get Started with Builds" link again. Wait a short while and click the house icon at top left of the page. Eventually it should say your trial has been accepted and take you to the home page.


Dashboard [Jenkins]
https://dbuchko.ci.cloudbees.com/?cloudbees.platform
Bookmarks
Imported From Firefo
Pivotal
RabbitMQ
Burlington Weather
myLearn Administrat
Pivotal CF Document
Builds
Ecosystem
Support
Documentation
Daniel | Account: dbuchko
Jenkins
ENABLE AUTO REFRESH
search
New Item
People
Build History
Project Relationship
Check File Fingerprint
Manage Jenkins
DEV@cloud Slave Templates
Support
Credentials
My Views
Groups
Roles
All
+
S
W
Name ↓
Last Success
Last Failure
Last Duration
Deployed On
CD-Lab
5 mo 7 days - #7
5 mo 7 days - #6
1 min 47 sec
N/A
test-gradle
N/A
N/A
N/A
N/A
test-maven
N/A
N/A
N/A
N/A
Icon: S M L
Legend
RSS for all
RSS for failures
RSS for just latest builds
Build Queue
No builds in the queue.

6. Follow the directions to install the Cloud Foundry CLI into your instance, as described here (http://documentation.cloudbees.com/docs/cje-user-guide/cloudfoundry-cli.html) (including this section here (http://documentation.cloudbees.com/docs/cje-user-guide/cloudfoundry-sect-config.html)).
7. Navigate to `Manage Jenkins` > `Manage Plugins` > `Installed`, and verify the GitHub Plugin is installed. If not, click on the `Available` tab and check the box for the GitHub Plugin. Click on the `Download now and install after restart` button. Then manually restart Jenkins from the `Manage Jenkins` page.

Create the Initial Build Job

1. Navigate back to Jenkins Home.
2. Click `new item`, give it a name and select `Build a Maven project` Then click `OK`.
3. Under `Source Code Management`, select `Git`, and supply your forked repository URL and credentials
4. Under `Build Triggers`, select `Poll SCM`. In the `Schedule`, enter the CRON formatted string such as `H/5 * * * *`.
5. Under `Build Environment`, check `Set up Cloud Foundry CLI`.

   Cloud Foundry CLI version: Select the latest version in the dropdown. If there is no dropdown, then go back and do the steps here (http://documentation.cloudbees.com/docs/cje-user-guide/cloudfoundry-sect-config.html), paying particular attention to this:

   *Note: **if the dropdown list of available versions is not visible** and replaced by an input box, you have to trigger the reloading of the update center: Click on Manage Jenkins then Manage Plugins then go on the Advanced tab and click on* `Check Now`.

   If you had to do the above step, un-check the `Set up Cloud Foundry CLI` box, save your job, then re-check and save the job to make sure the version is picked up.

   Set the following parameters:

API EndPoint: `https://api.run.pivotal.io`

API Credentials: Add your PWS credentials

Organization: `<your organization>`

Space: `<your space>`

6. Add a post-build step "Execute Shell".

   Select `Run only if build succeeds`

   Command: Enter the `cf push` command (relative from the project root directory) to push the application to PWS.

   eg: `cf push dbuchko-env-demo -p target/cf-intro-environment-demo-0.0.1-SNAPSHOT.war`

7. Save the config and try running the build by clicking ``Build Now".

Github Set Up

1. Go to `https://github.com/<YOUR-ACCOUNT>/your-workspace` > Settings > Webhooks & Services > Services > Add a Service > Jenkins (Github plugin)
2. Jenkins hook url: `https://<YOUR_ACCOUNT>.ci.cloudbees.com/github-webhook/`
   - Make sure you include the trailing slash. The integration between github and jenkins will not work without it.
3. Commit a change to git and watch the magic

Beyond the Class

The CD exercise above is very simplistic and should be expanded for real projects.

Artifact Repository

Ideally, you want to build your artifacts (jars/wars) and publish them to a repository like Artifactory.

- Artifacts should be versioned to match the app deployments on PCF.
- All pushes to PCF should be using the same artifacts. Artifacts should be built once and used throughout the lifecycle.

Code promotion

- Jobs should be established in Jenkins to deploy/promote code to different phases like dev to test to prod.
- Jobs should use the same artifact published to Artifactory.
- Jobs can be triggered automatically or manually but should be fully automated. There should be no manual steps beyond clicking "build now".

Notification

Jenkins supports many notification plugins. It is important for code owners to be aware of build status.

Last updated 2015-11-30 10:39:49 PST