IBM Cloud Developer Certification Training

Understand capabilities of IBM Bluemix DevOps Services source code management for projects

Version: 2

Last modification date: 4 August 2015

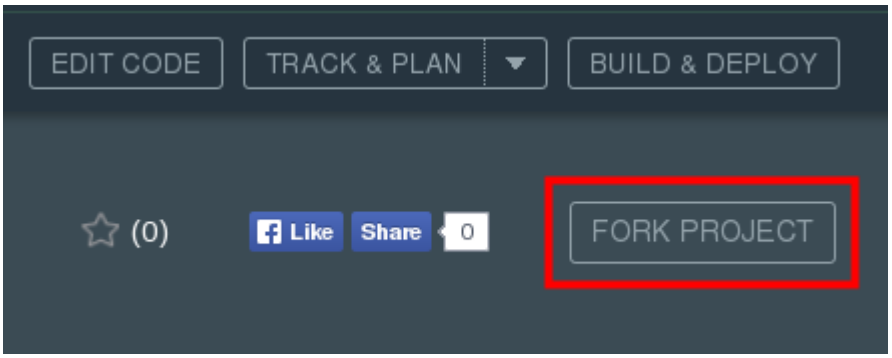
Owner: IBM Ecosystem Development

Exercise 5.4.0 - Lab Prerequisites

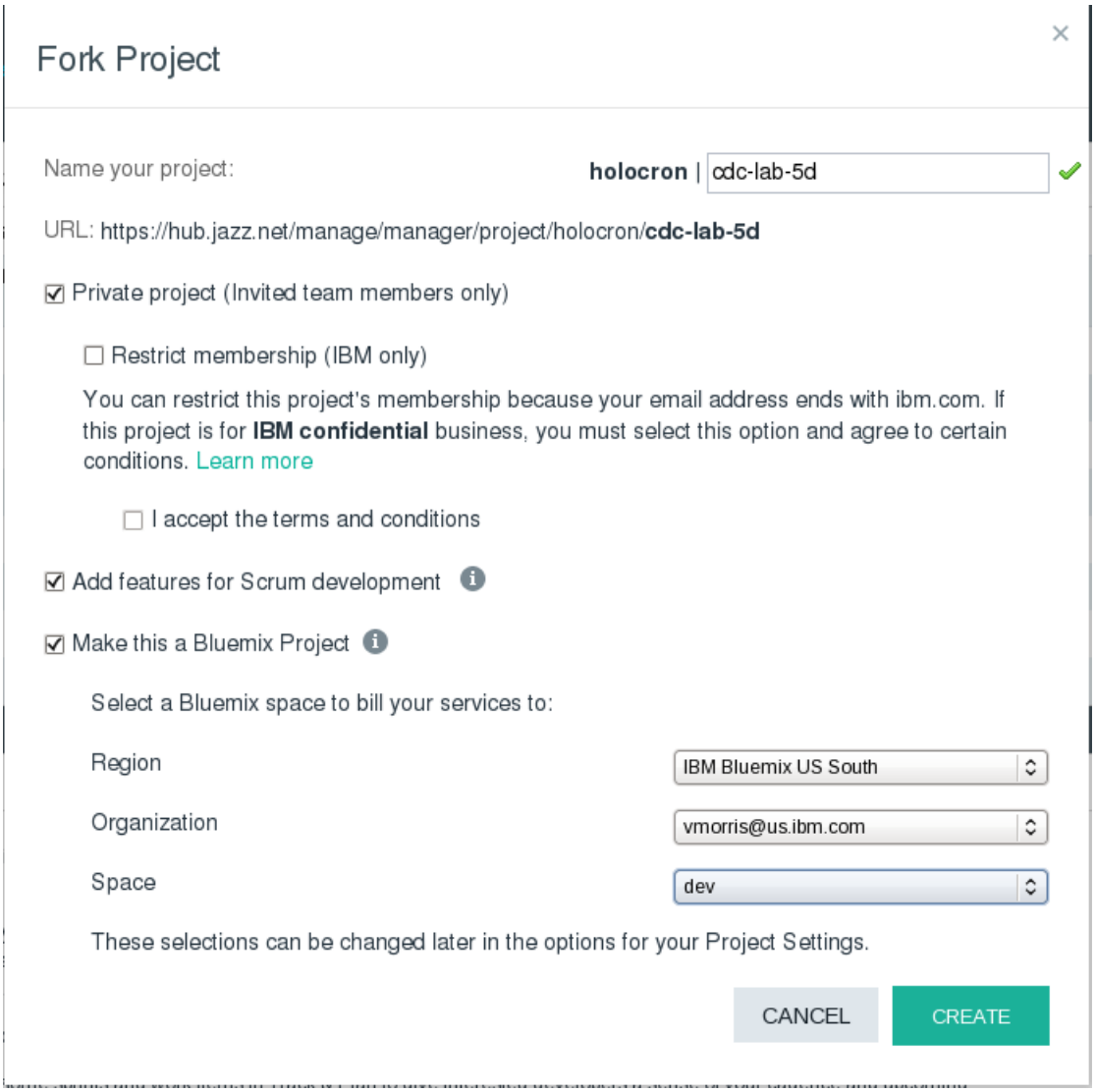
- IBM Bluemix account
  - Sign up for Bluemix: http://bluemix.net
- IBM DevOps account
  - Sign up for DevOps: http://hub.jazz.net
  - Use the same credentials as for Bluemix
- Supported web browsers:
  - Chrome, latest version for your OS
  - Firefox, latest version for your OS or at least ESR 31
  - Internet Explorer, versions 10 and 11
  - Safari, latest version for your OS

Exercise 5.4.1 - Getting Started

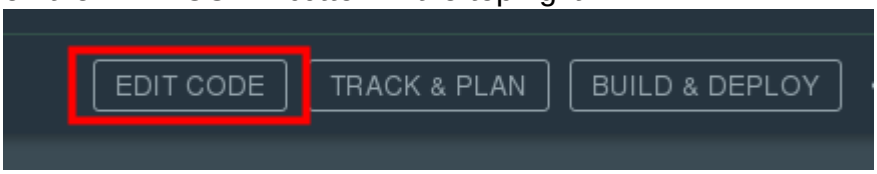
1. As in the 5.3 Lab, begin by logging on to IBM DevOps Services and fork the project located at https://hub.jazz.net/project/ecosysdevcnc/cdc-lab-5/overview into a new project.



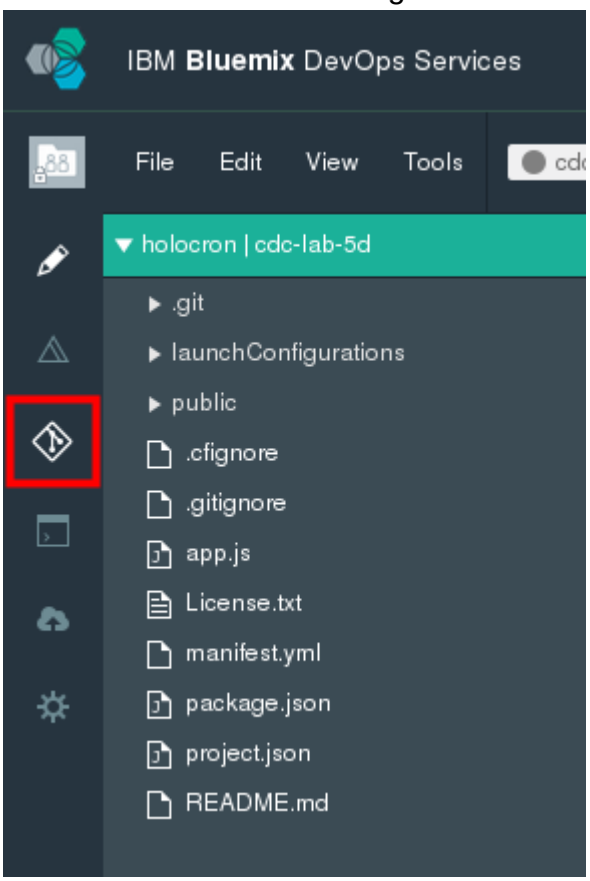
2. Give the new project a unique name, check all the boxes, and choose an appropriate Bluemix runtime configuration. Click on CREATE.



3. Once you see the message about successfully creating your project, click on the EDIT CODE button in the top right.

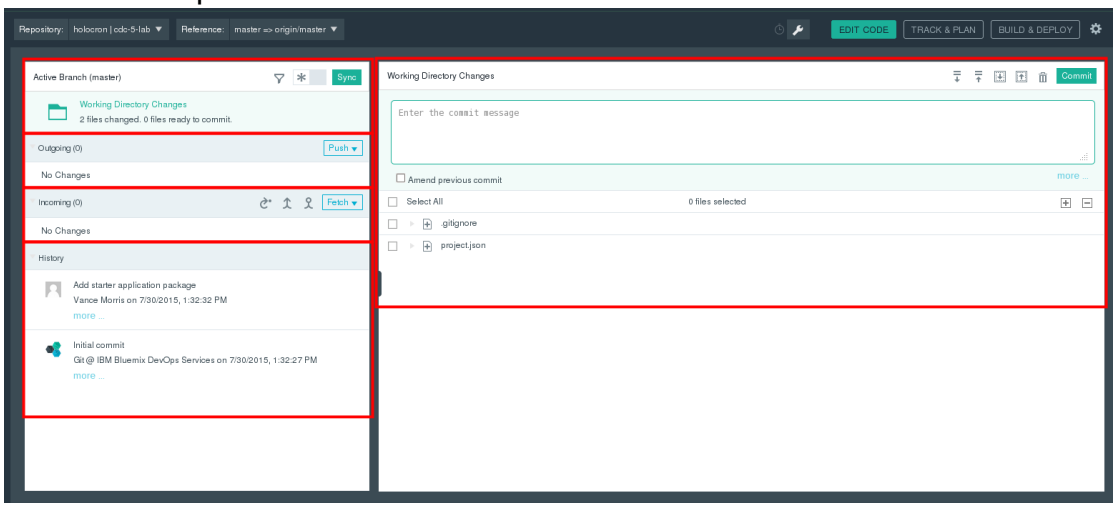


4. Wait a moment for the workspace to setup. Once the project is ready, click on the Git icon located along the left vertical column.

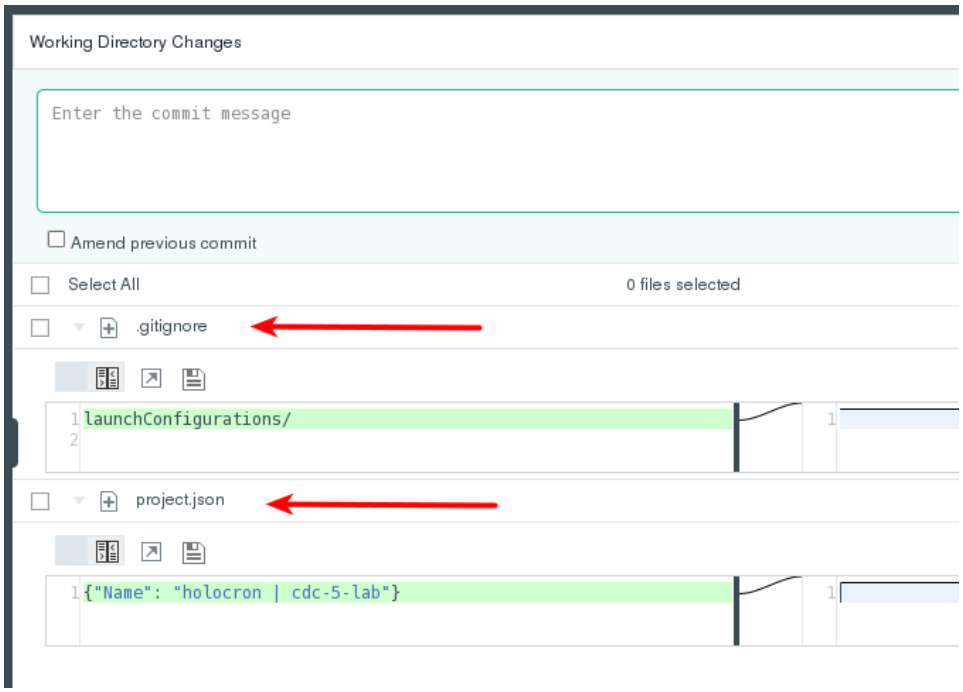


Exercise 5.4.2 - Adding Local Changes to the Git Repository

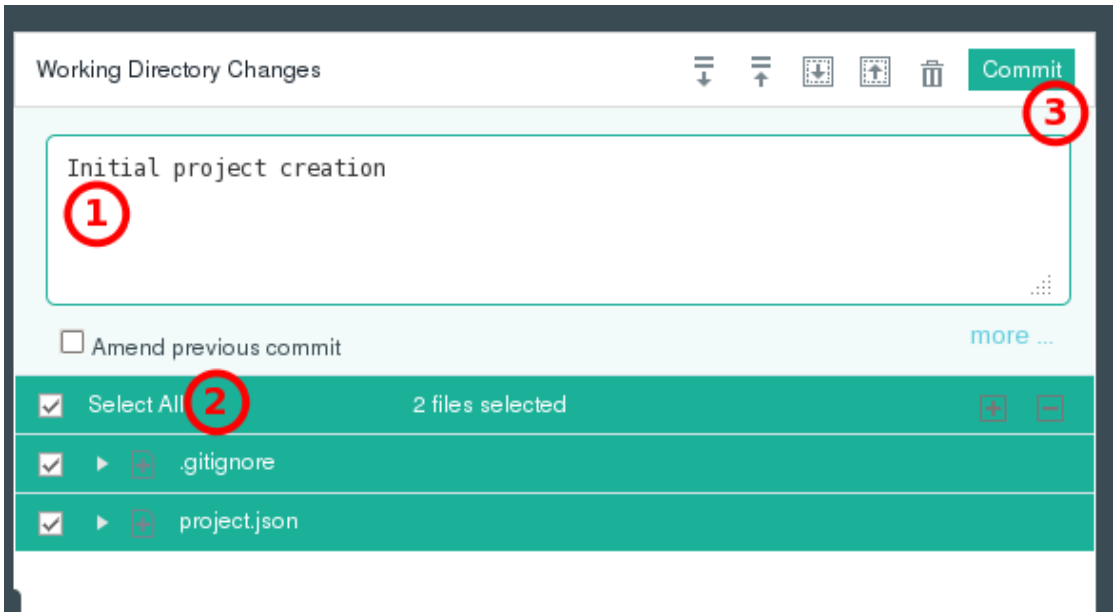
1. The Git Repository view, or "git view" is organized into two panes. The left pane shows any staged outgoing changes, any incoming changes, and the active branch's history. The right pane, taking up the majority of the screen, shows any working directory changes and provides the functionality to select changed files, enter a commit message, and stage the files for a push.



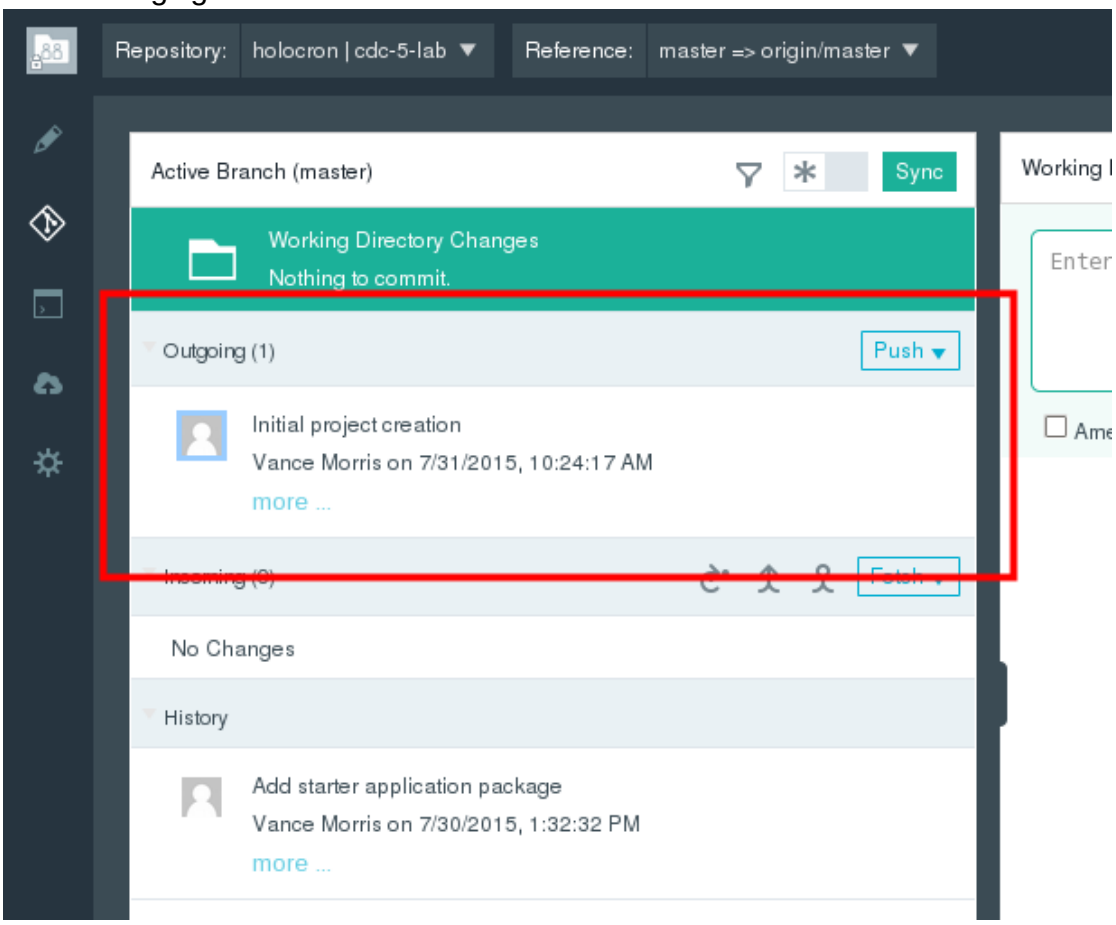
2. The process of forking a project causes changes to occur in the files. Notice there are two files listed in the right pane. Click on each one to get a view of what has changed.



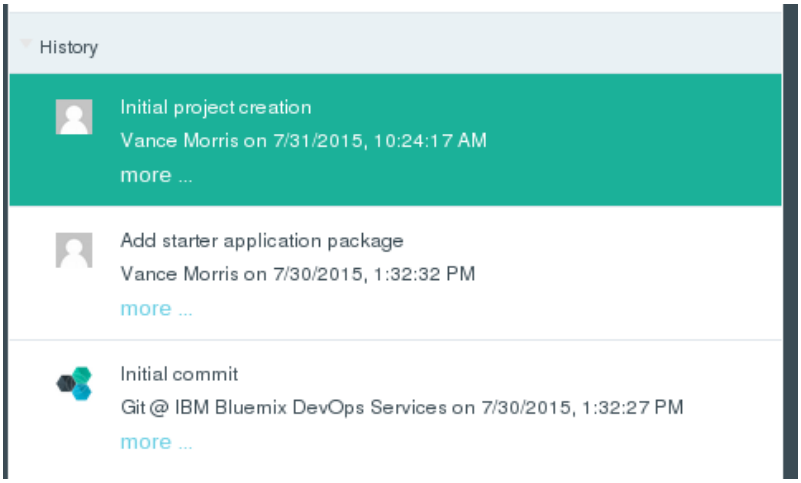
3. .gitignore is a file used by git to list all the files and directories in the project that should be ignored by git. Things like Bluemix launch configurations and build artifacts generally do not need to be tracked by git and should not be commited to the master branch.
project.json is a file used by DevOps Services to identify your project. It should remain unchanged and can be committed to the master branch.
Enter a commit message, something simple like "Initial project creation" is fine, check the box to Select All, then click the Commit button in the top right.



4. Notice that the working directory changes pane is now empty of any files, and a new entry is made in the Outgoing section of the left pane. This is called "staging" a commit.

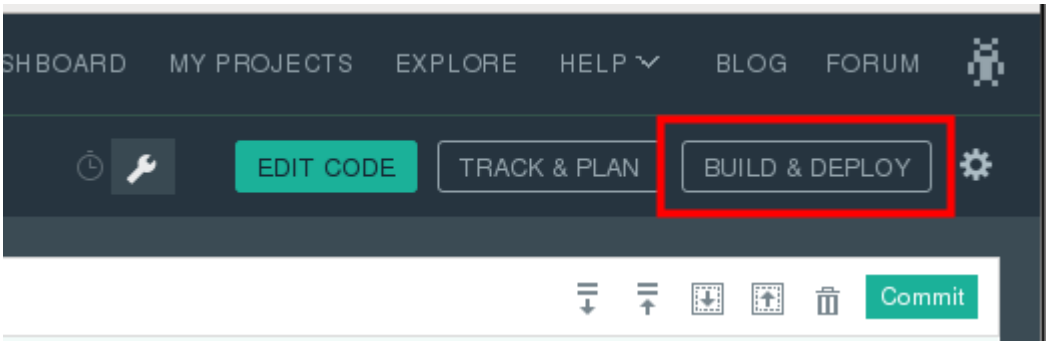


5. The master branch is still unchanged, and will remain so until you push the outgoing commit. If there were any incoming changes from other branches, you might need to merge them together with your outgoing changes, and you would have the opportunity to do so before the push completed. Click on the Push button now and note that the History is updated to include your commit. Any other branches would be able to pull this commit and it's changes into their local working copy.

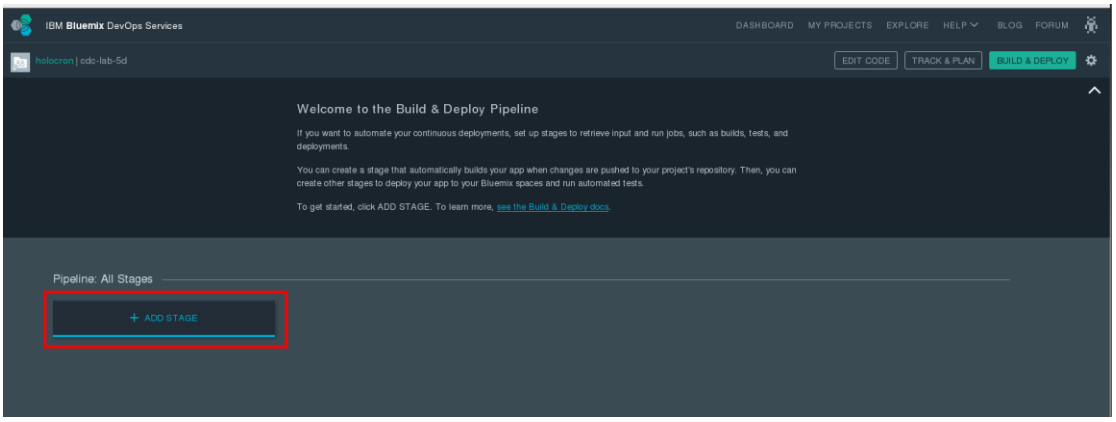


Exercise 5.4.3 - Verify the integrity of code delivered to the repository with a build

1. Next we will configure DevOps Services to perform a build of the code and have the build process trigger off a successful push to the master branch. The builder will watch the master branch for any new pushes, and when detected, it will automaticaly begin building the project into a deployable package.
   Click on the BUILD & DEPLOY button in the top right.

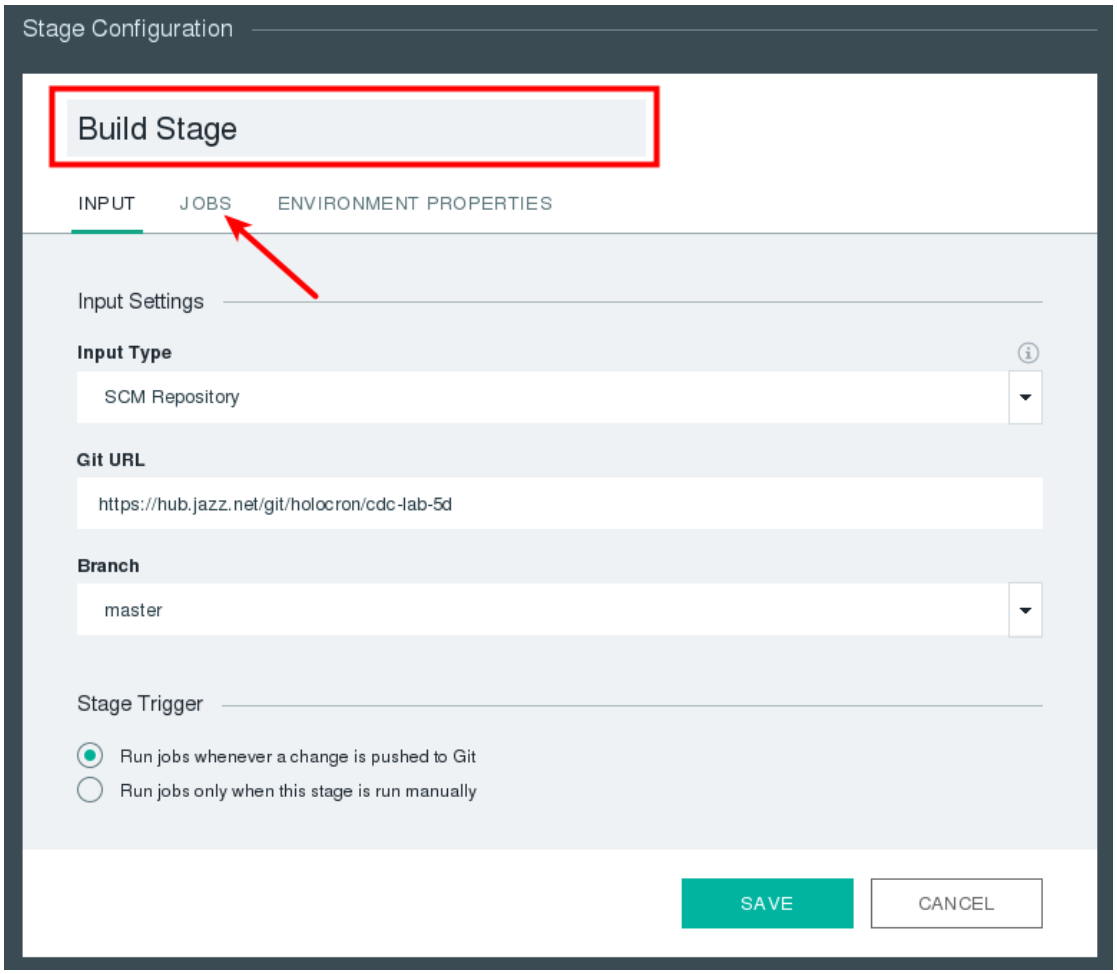


2. The buiild and deploy pipeline view gives a high level view of the various stages that are configured in the pipeline. Click on the ADD STAGE button now.

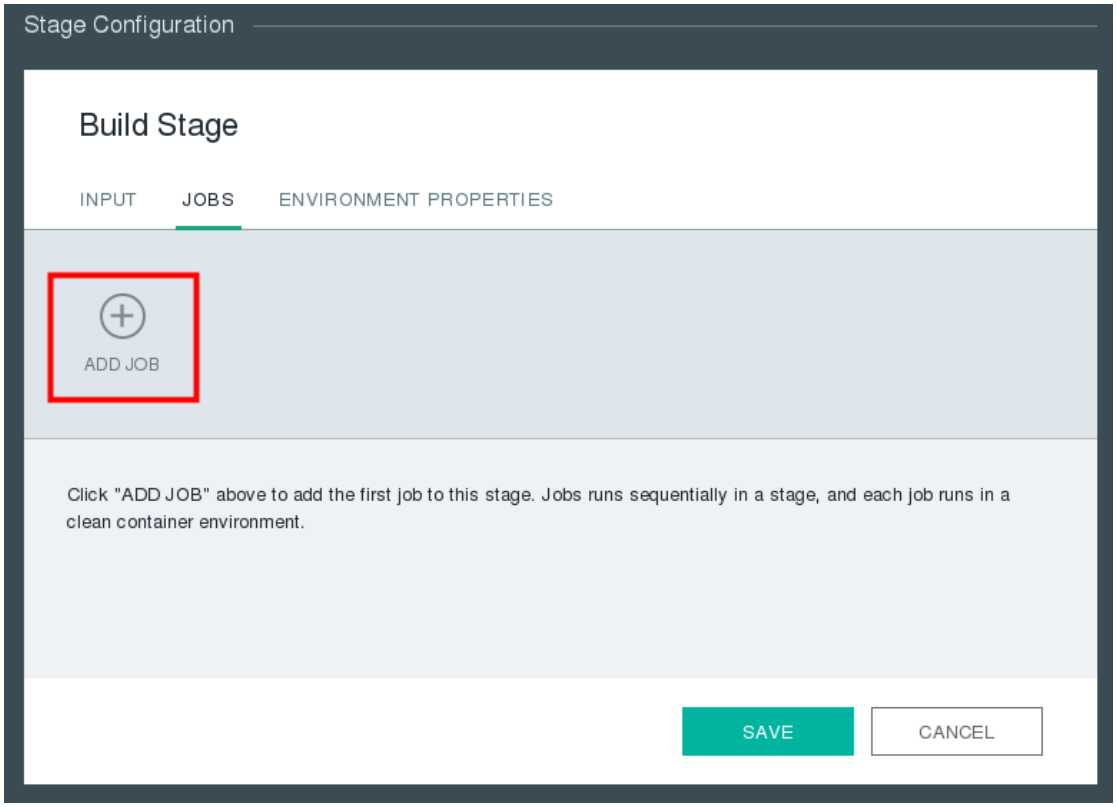


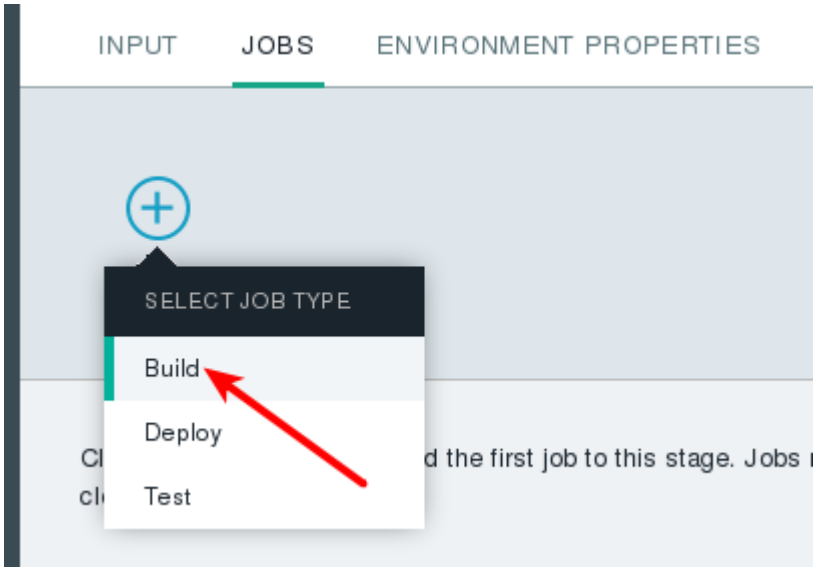
3. A stage consists of a series of jobs that run in sequence and the input for those jobs. Stages can be configured to run automatically based on triggers.
   The input can be either the project SCM repository or build artifacts from a preceding stage. Input applies to all jobs in the stage. When a stage is run, the input is fetched. The files are placed in the working directory before each job starts.
   Jobs perform the work for a stage, such as building, deploying, and testing. The jobs in a stage run sequentially, and each job runs in a clean container environment. Files do not persist across job executions, so any dependencies required by the jobs must be provided in the stage input or installed as part of the job.

4. Give the stage a good name to identify it, then click on the JOBS tab.
   Note that the input for the stage is the SCM repository that contains the master Git branch of your project. Also note that the default has the stage run whenever any change is pushed to Git.

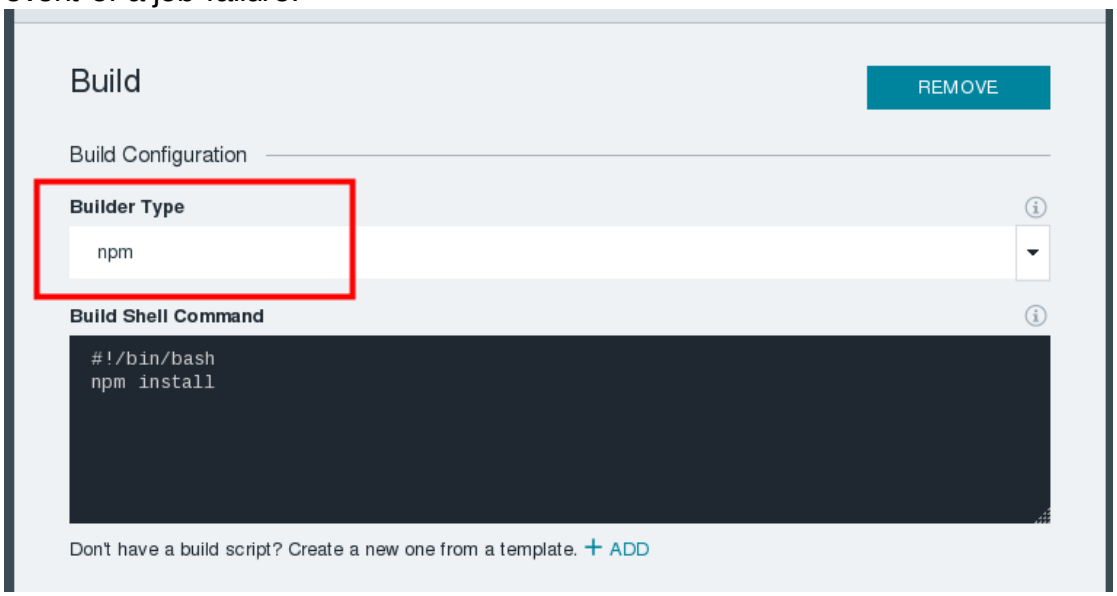


5. In the JOBS tab, click on the plus sign to add a job, then select Build.

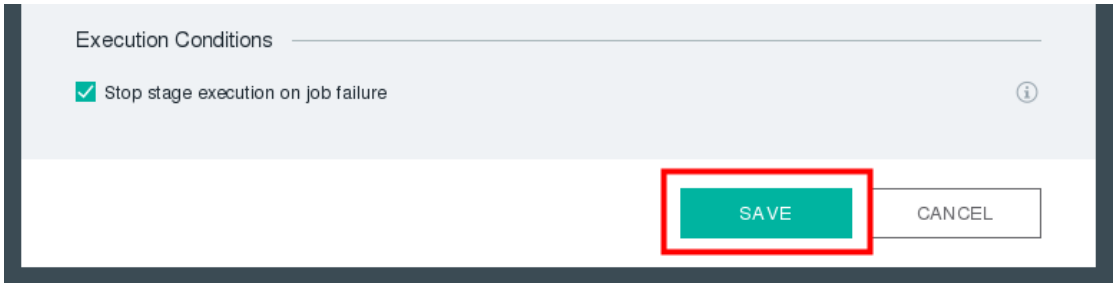




6. Under Builder Type, select "npm".
   Note that many builder types are available to facilitate your project's specific needs, and futher configuration options include working directory, build archive directory, and an option to stop the stage execution on the event of a job failure.

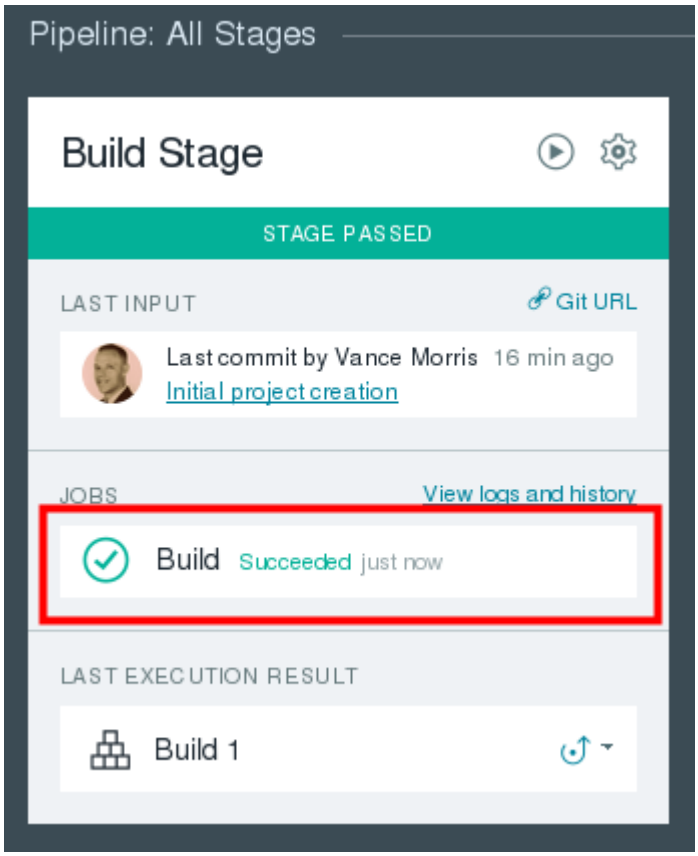


   Scroll to the bottom and click SAVE.

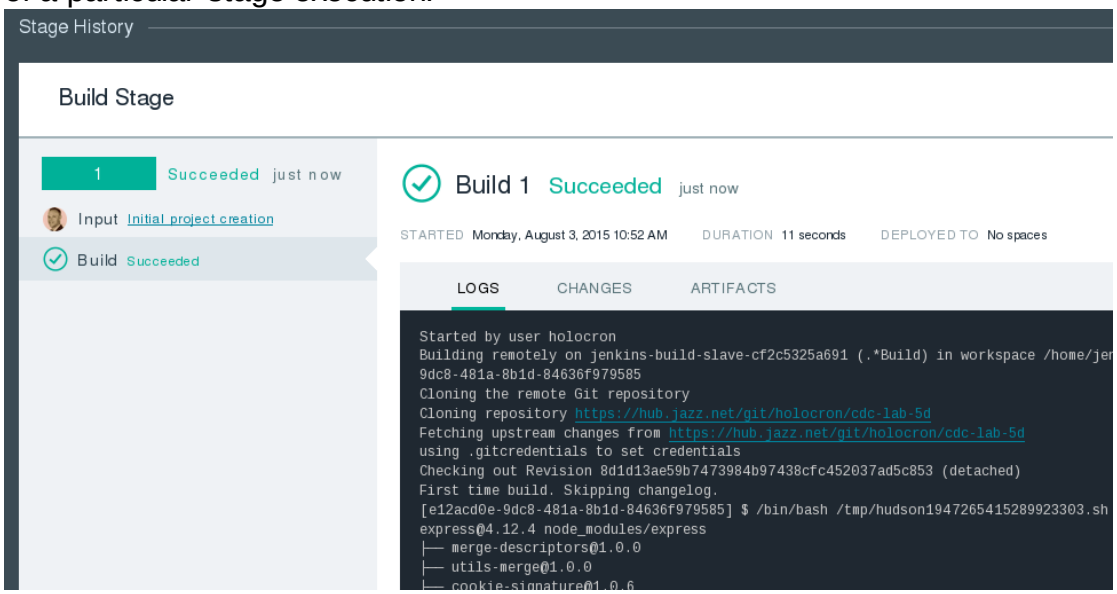


7. We are automatically returned to the build and deploy pipeline overview and the new stage is displayed. It is possible to start a build automatically by clicking the play button within the stage. Do this now.

8. The stage starts running, and the build job status changes from Pending, to Queued, to Running, to Succeeded. To view the logs from the build, click on the job status message.

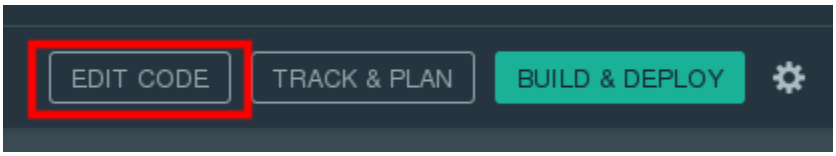


From the Stage History view, we can see the history of all previous executions of this stage, and download any artifacts generated as a result of a particular stage execution.

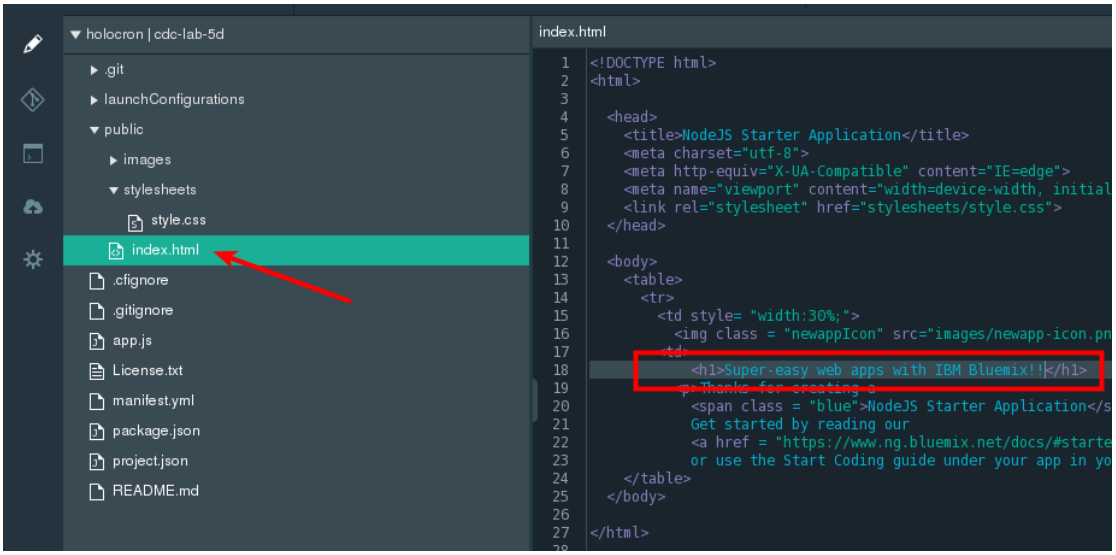


Excercise 5.4.4 - Trigger the Build Stage by Pushing to the Git Repository

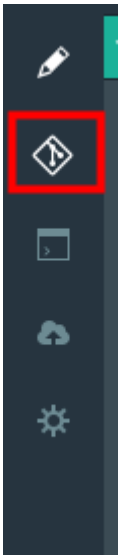
1. Return to the Web GUI editor by clicking on EDIT CODE in the top right.



2. Alter the project slightly by opening the public/index.html file and change the <h1> tag to something new.

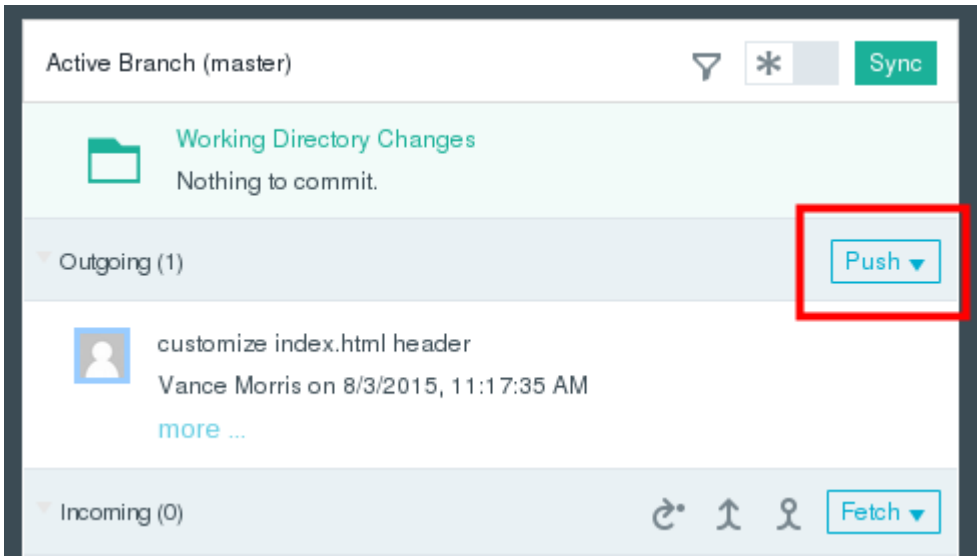


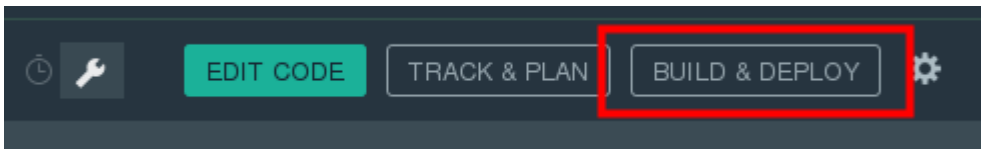
3. Switch to the Git repository view by clicking the Git icon in the left column.



4. Note that the SCM detected a change in the working directory. Enter an informative commit message, check the box next to index.html, then click the COMMIT button.

5. The outgoing commit is now staged, and ready to push. Immediately after pushing the commit, switch over to the BUILD AND DEPLOY view. You will need to be quick to see the builder stage automatically start, but if you miss it, you can always view the details of the build in the history view. Click on Push, then immediately click on BUILD AND DEPLOY.





Observe the automatic execution of the build stage.

This completes the IBM Cloud Developer Certification Section 5.4 Lab.