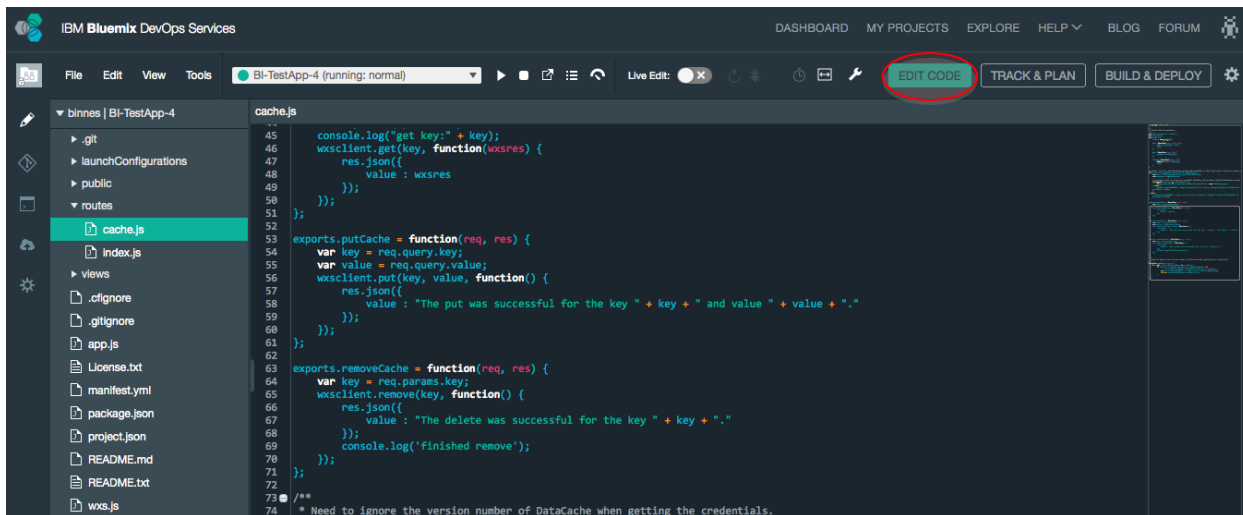# Getting Started with IBM Bluemix

Hands-On Workshop

Exercise 4b: Working in DevOps Services
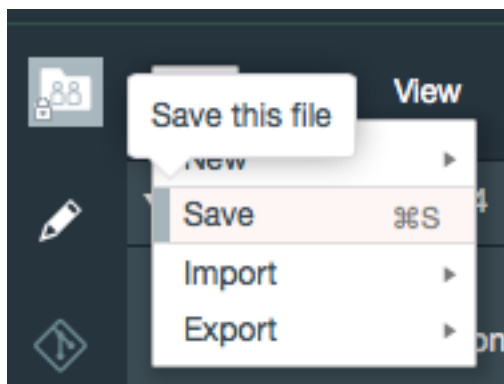
Working in DevOps services

1. In DevOps services, switch to the **EDIT CODE** section so that you can edit the code.

2. Open the `routes/cache.js` file and scroll to line 58 to find the two strings that are displayed on a successful put ("The put was successful …") or delete ("The delete was successful …") and change the strings to new messages:
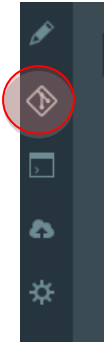
```
53   exports.putCache = function(req, res) {
54       var key = req.query.key;
55       var value = req.query.value;
56       wxsclient.put(key, value, function() {
57           res.json({
58               value : "The put worked for the key " + key + " and value " + value + "."
59           });
60       });
61   };
62
63   exports.removeCache = function(req, res) {
64       var key = req.params.key;
65       wxsclient.remove(key, function() {
66           res.json({
67               value : "The delete worked for the key " + key + "."
68           });
```

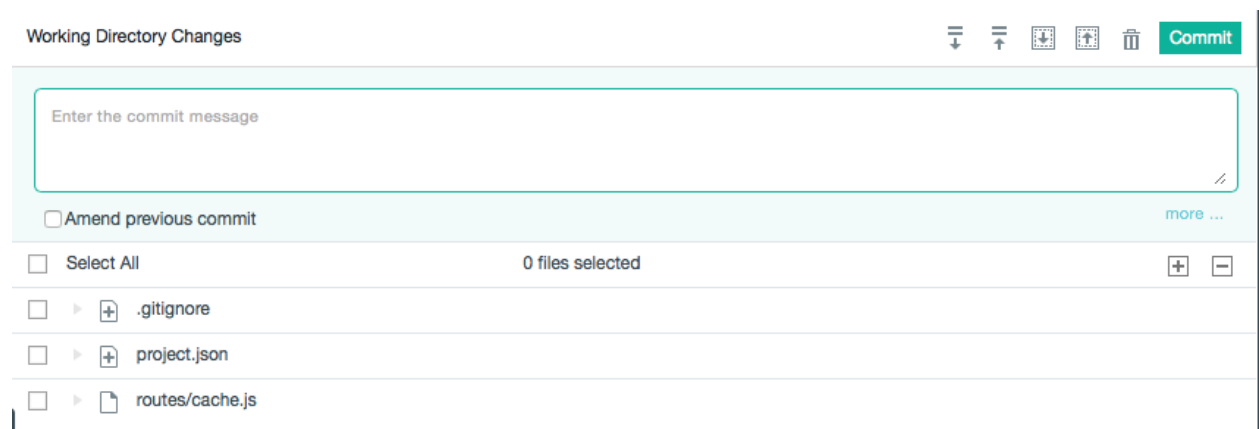3. Save the changes by clicking **File > Save**.

To get the changes pushed to Bluemix, commit the changes to the Git repository.
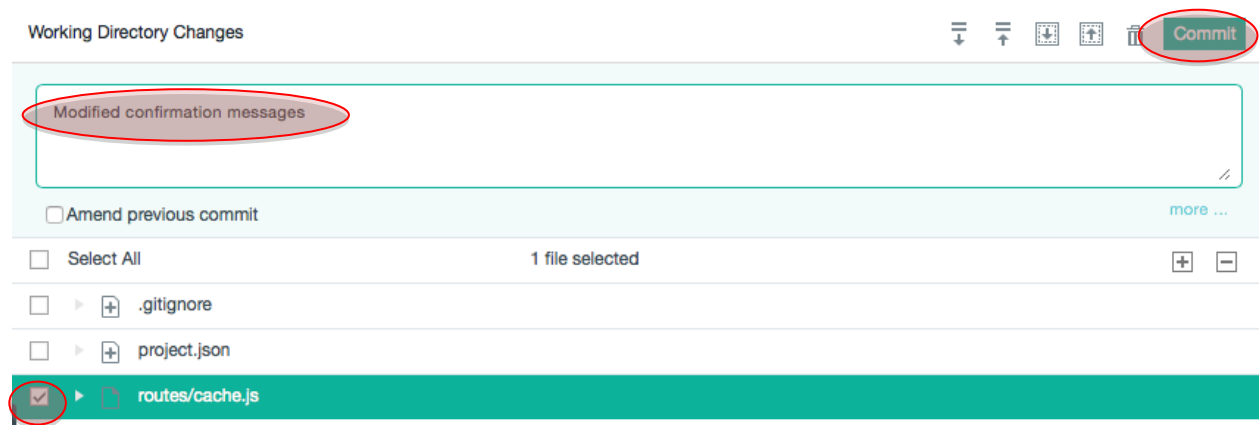
4. To work with the Git repository, switch into the Git section.

The files that have local changes are shown in the **Changed Files** section. When a project is first added to DevOps services, some configuration files are updated automatically, in this case, the first 3 files in the list.
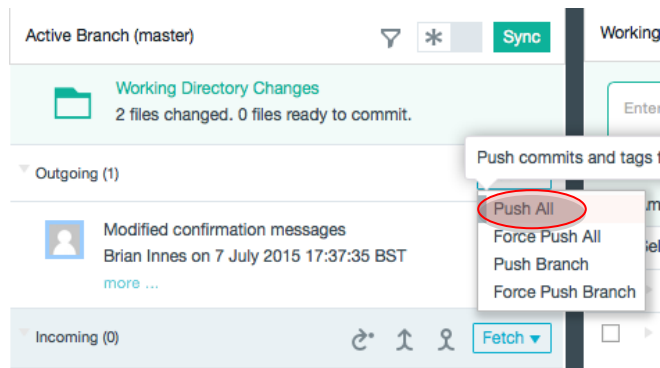
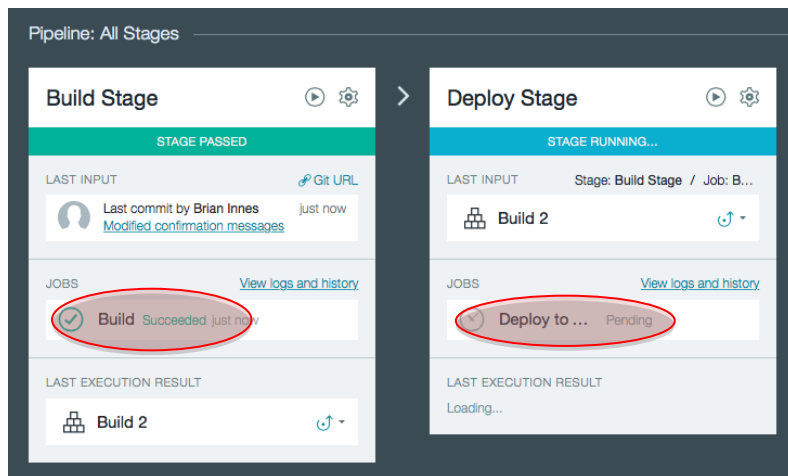5. Select the JavaScript file, add a commit message, and then click **COMMIT**.

The commit was made to the local branch. The builder works from the remote branch, so you must push the changes back to the remote branch.
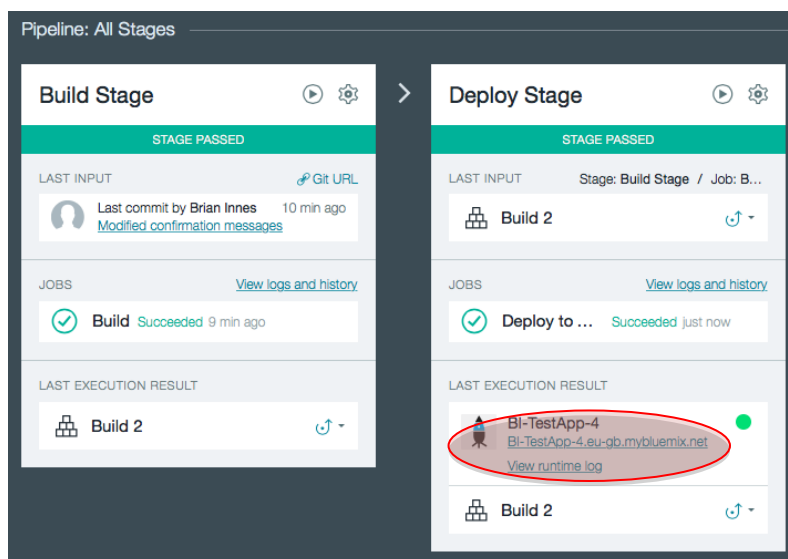
6. In the left column, expand the **PUSH** menu and click **Push All**.

If you quickly jump to the BUILD & DEPLOY window, you can see that a build has automatically started, which will then be automatically deployed to Bluemix if the build is successful.

7. After the deploy is completed, test the application to verify that the code changes are running. Launch the application by clicking the link in the deploy stage.

Tip: If a build fails, click the build in the Builder where you can see a detailed build history. You can also access the logs, files, and details of changes included in the builds. The build history and logs can help you determine why a build failed.