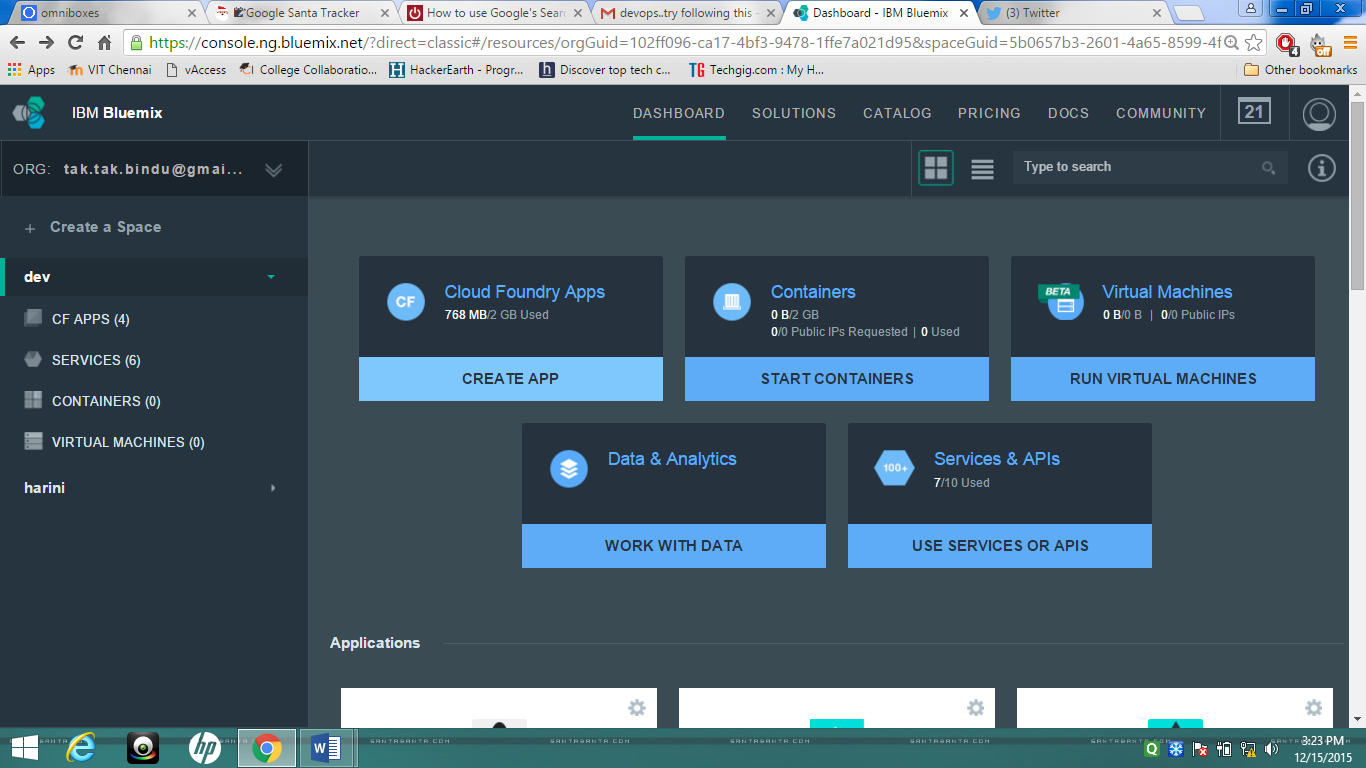
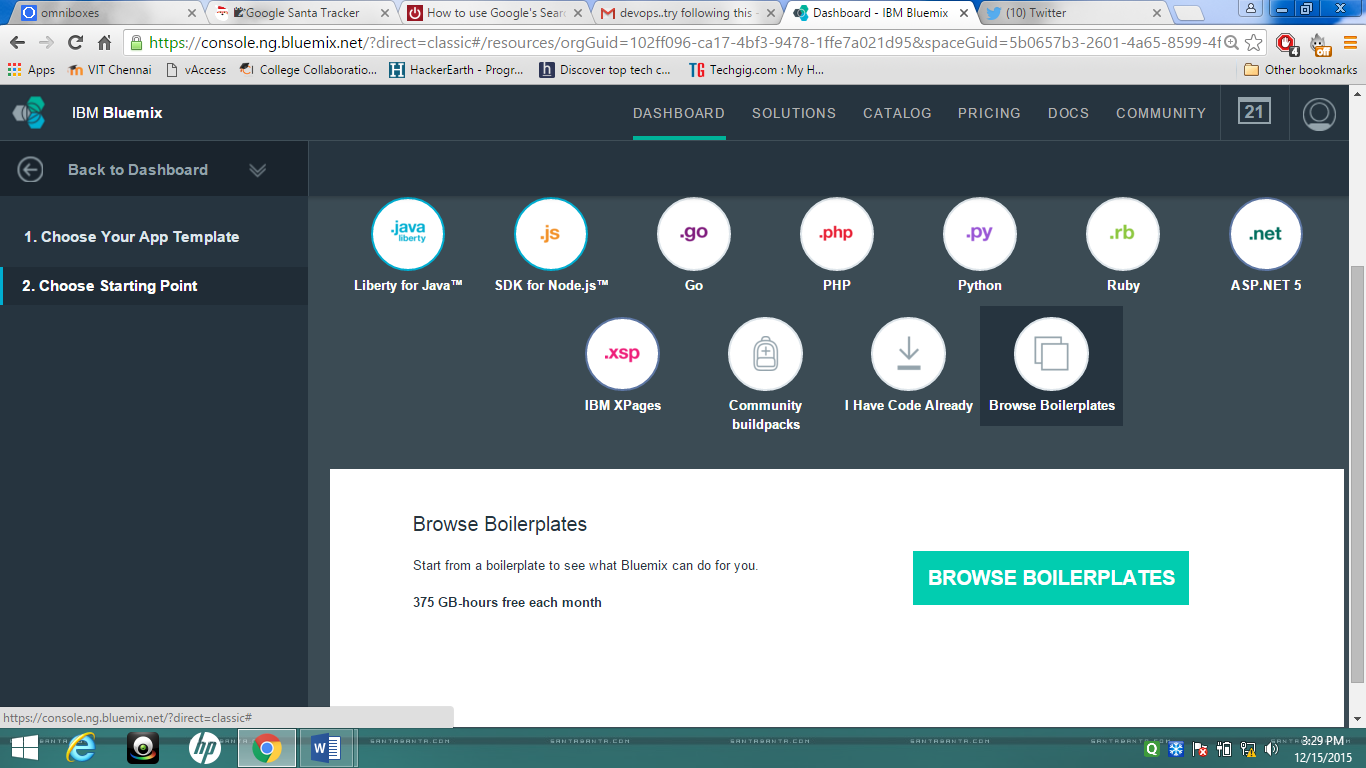
## Create Java Web Starter application on Bluemix

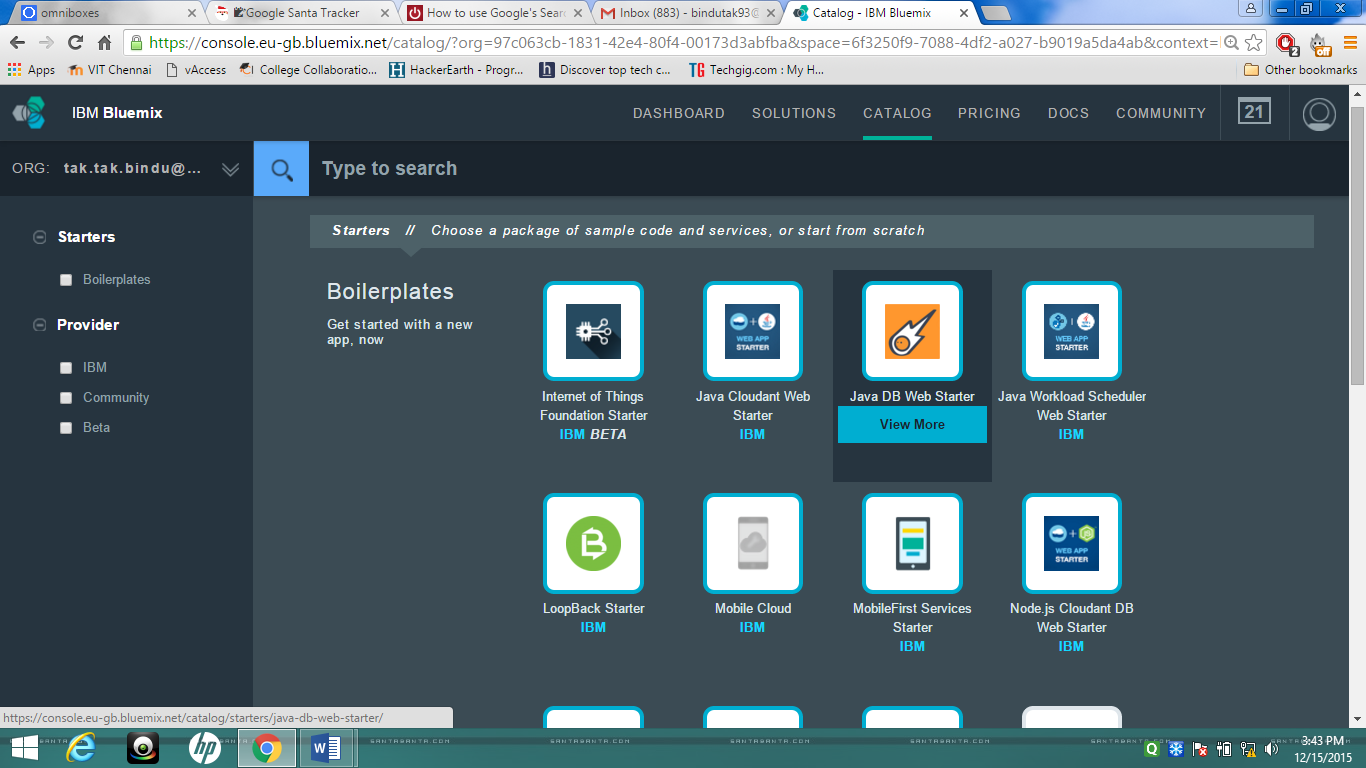
1. Login to Bluemix (<https://ace.ng.bluemix.net>) and navigate to your dashboard.
2. Click on the "CREATE AN APP" button



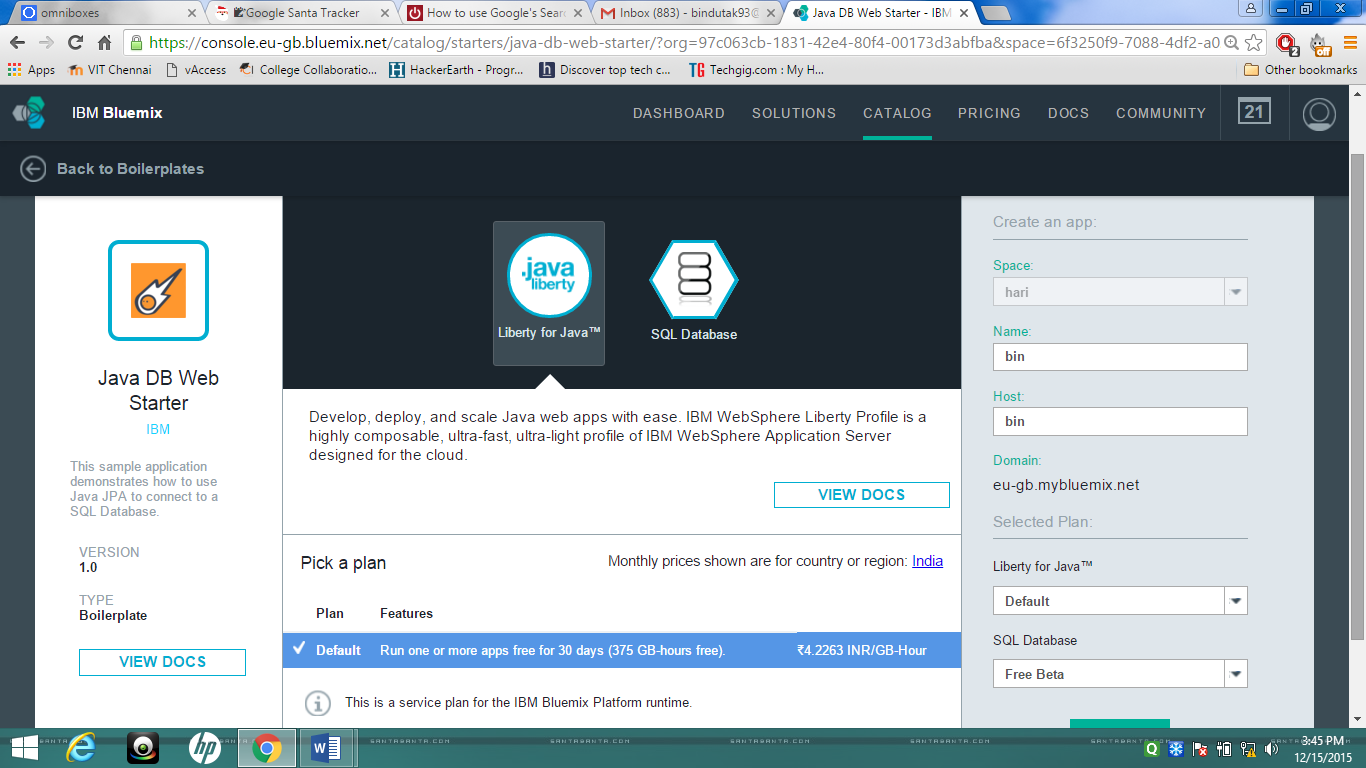
1. Select webApp.
2. Select Browse Boilerplates
3. Next window will appear. Click on Browse Boilerplates.



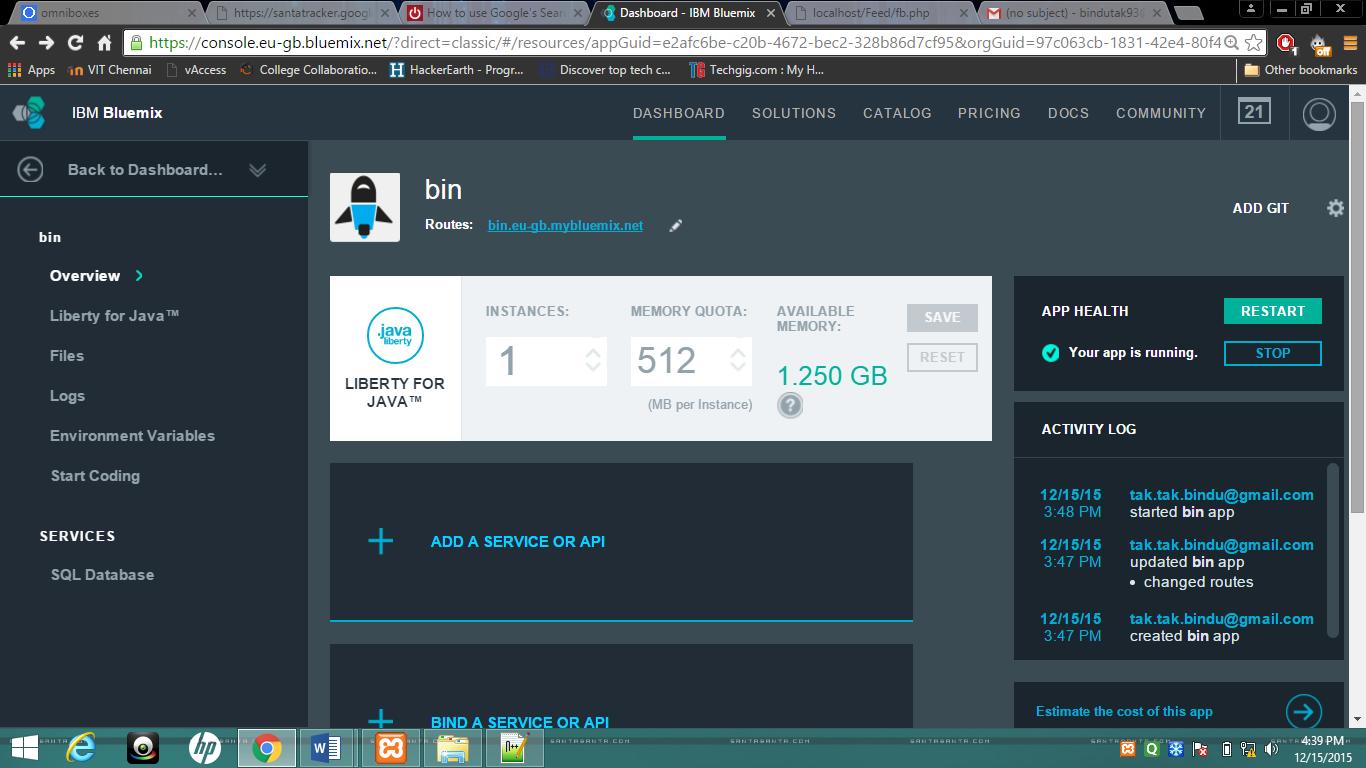
1. The Java Web Starter boiler plate uses the WebSphere Liberty run-time. The application created uses the Data Cache service



1. Choose an appropriate name and host for this application.



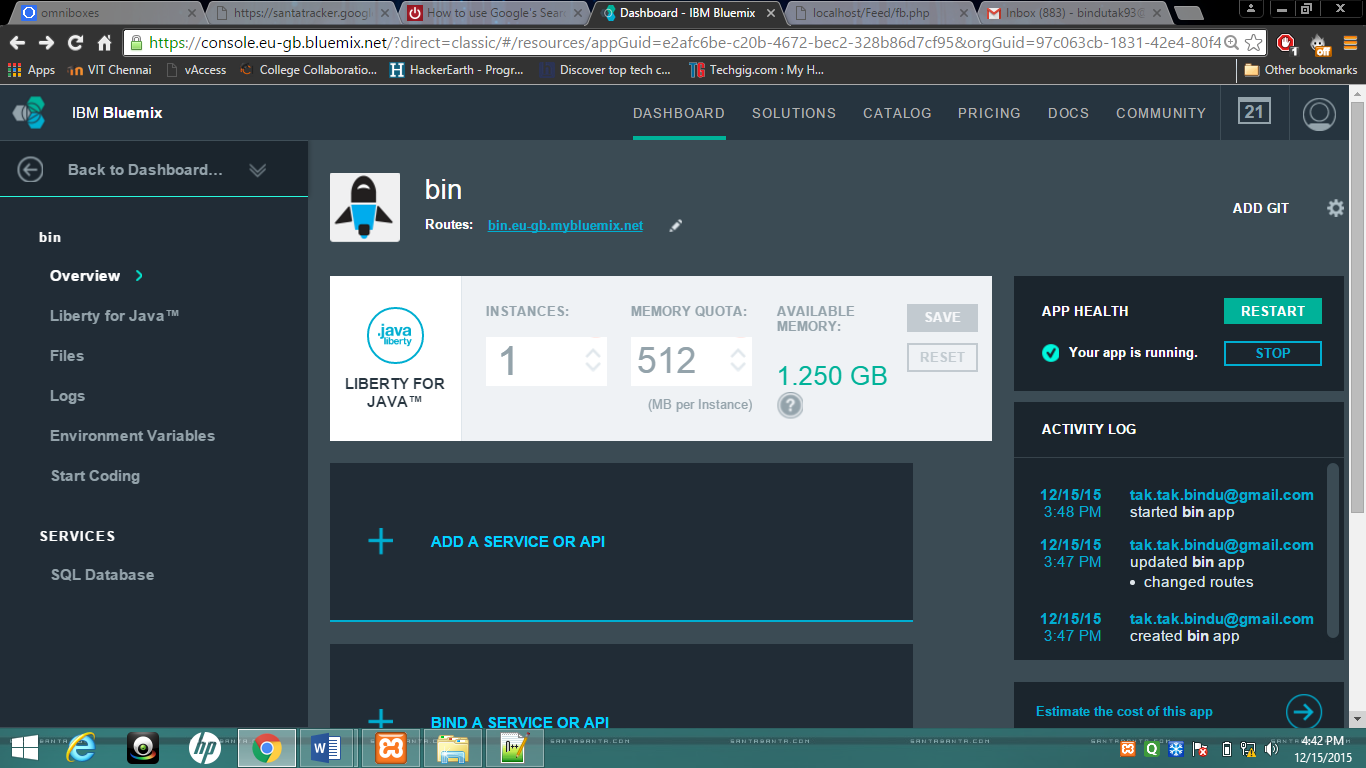
1. Click on the CREATE button to complete the application creation process.
2. Once the application has been created the application's overview page is displayed.



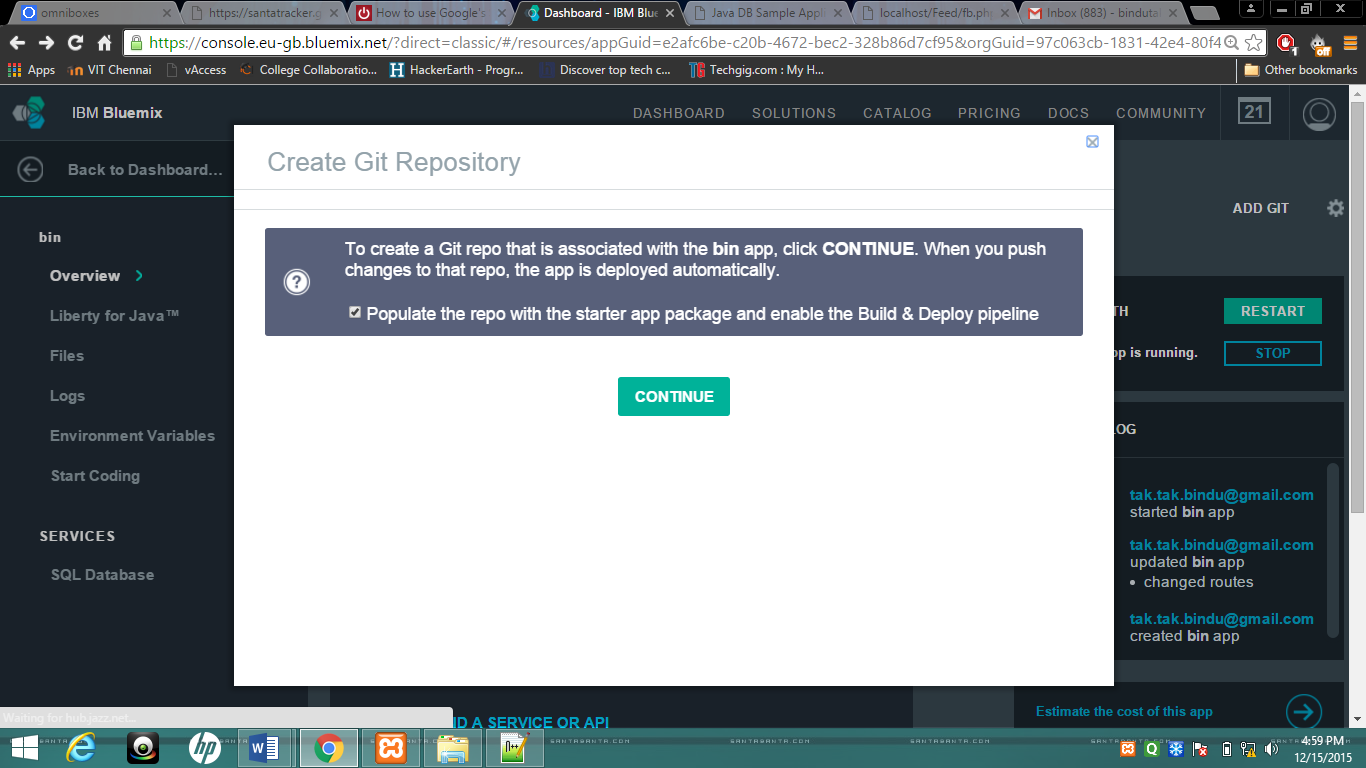
1. If you click on the “DASHBOARD”, the application can be seen on the dashboard. The application should be started.

**Note:** In the application view on the DASHBOARD, Click on the application to return to the Overview page

1. The overview page includes a link to the Routes for the application. If you click on the Routes a browser page will open and display your application in it current state.



1. Click on ADD GIT



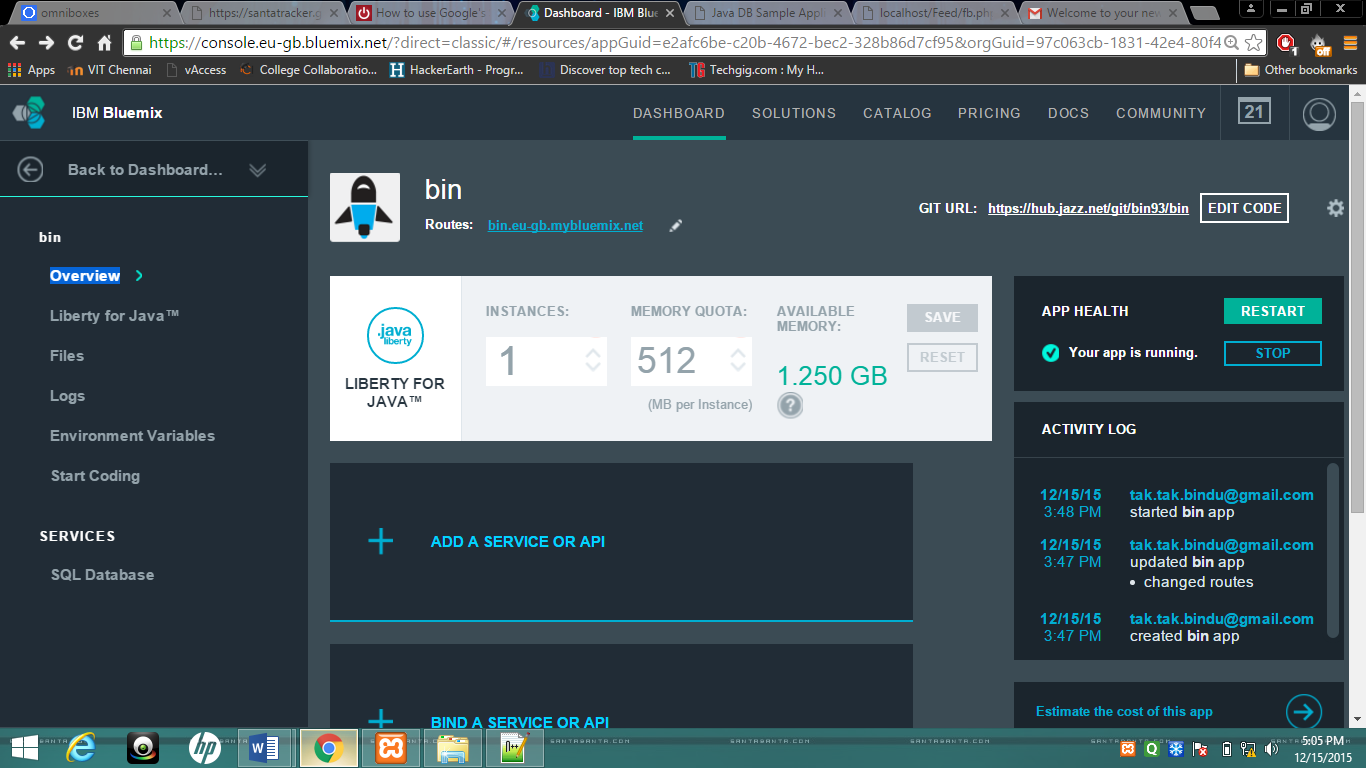
1. Click on the CONTINUE button to create a Git repository associated with your application. Ensure that you check the “Populate the repository with the starter application package and deploy it to the application.” checkbox.
2. The Git repository is successfully created and ready to use. Click Close to proceed further



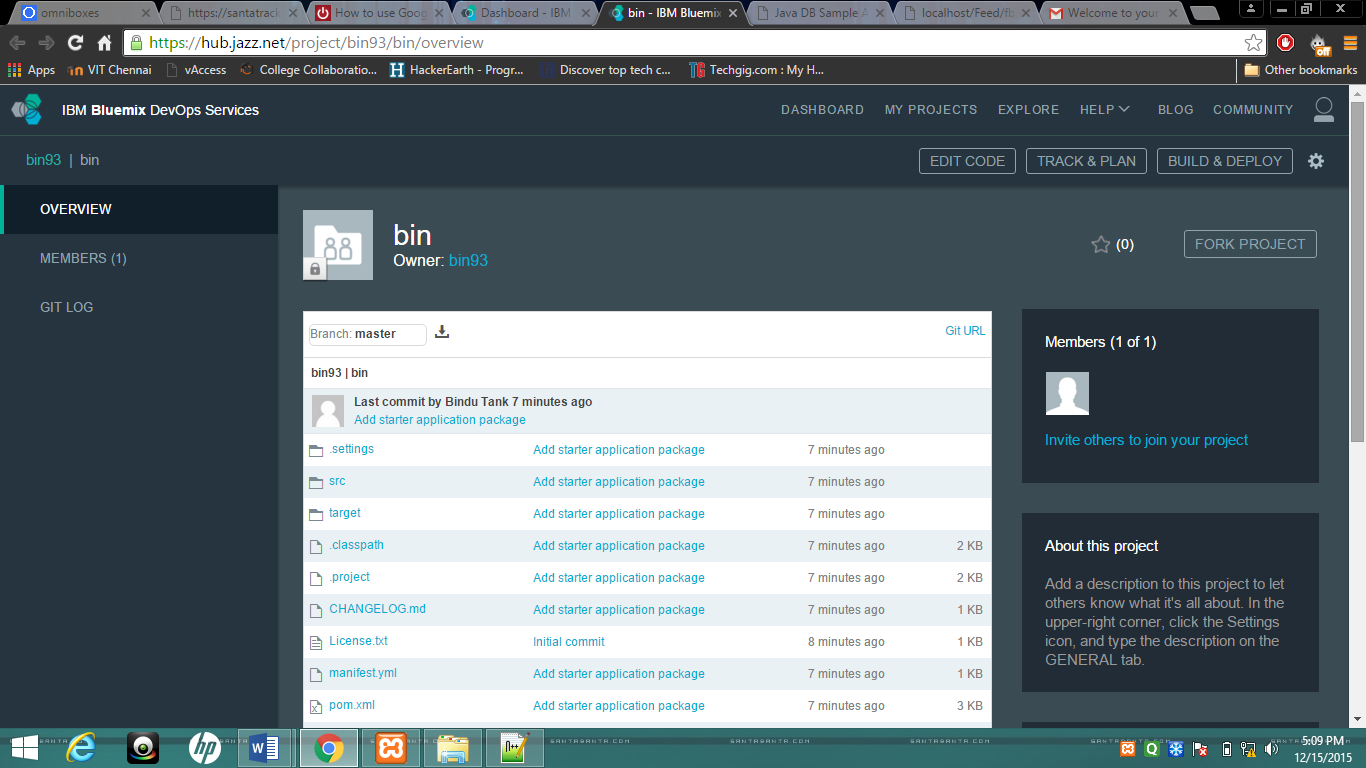
1. You will receive an email from the IBM DevOps Services email similar to the following detailing your project specifics and providing some useful links with tutorials on how to use your new Git repository.

|  |  |  |
| --- | --- | --- |
| |  |  | | --- | --- | |  | IBM DevOps Services  Powered by JazzHub | |
| Congratulations! Your new project is here:  Project URL: [https://hub.jazz.net/project/sdaya/LabExercis4-sdaya](http://sendgrid.softlayer.com/wf/click?upn=awxxXRTrFB8QupIumPNGrA0XxYJxKT6xPT0oLi22-2BvnEU2nIqZaNVcYLguRWka5XSA76utP41Kn-2B7BgttuCxPQ-3D-3D_CAkCtUVhogUP8-2B4DkpDTEsG9enp6DMCY0RM2gQKbfR-2BdB93G9wuUJoYmdNanN9v04ilktDl6EnEOzqCwhW9uTvNjznRet4-2F5BWPl7yqaDbQIskc9oTVBNUfy5k57sDzzsnJOXdDlEm5QH4G8fnjVxpAZYVYyVueWaIwZQW17R5Lpk1I1-2F67XPNh4RBakQNiUTKdHgAnJXjdOiKP-2BlksZgQ-3D-3D)  Git Repository URL: [https://hub.jazz.net/git/sdaya/LabExercis4-sdaya](http://sendgrid.softlayer.com/wf/click?upn=awxxXRTrFB8QupIumPNGrPJYJN-2FH7fR2KuS-2BcJE5Zg6LbUzy0Qe7K9om25-2FAg-2Bej0dF0Gz4vg7VVwYpFrgNeVQ-3D-3D_CAkCtUVhogUP8-2B4DkpDTEsG9enp6DMCY0RM2gQKbfR-2BdB93G9wuUJoYmdNanN9v0s7Qd2OzcLy-2BFVVgQHfiSnUqakXMKe6iDE9N5qAsguOJJwVuUOH4CJl4D-2FwUvdG9j9su9r9k7zIkCa2X6pBcpm8BNheEtjPgPuYtZFseOEipKjJeR-2BSQl8SM7FsKZDu2gm2h1U-2FFwW-2BXOTXypeSZFng-3D-3D)  Do you intend to code with Rational Team Concert client for Eclipse? If so, connect to your IBM DevOps Services project and clone your Git repository:  1. In your Rational Team Concert client for Eclipse, click **File** > **Accept Team Invitation**  2. Paste the text below into the Accept Invitation text field:  teamRepository=https://hub.jazz.net/ccm02 userId=sdaya userName=Shahir Daya projectAreaName=sdaya | LabExercis4-sdaya  3. Clone your Git repository using [EGit](http://sendgrid.softlayer.com/wf/click?upn=y3bCP1BIom5MJ-2BJU6M5lX2XgmbJJh3gKHEk4Ipv6qL-2F9IAqpdTsXFr9mDAaQYYwE_CAkCtUVhogUP8-2B4DkpDTEsG9enp6DMCY0RM2gQKbfR-2BdB93G9wuUJoYmdNanN9v03YP0IMcrXntSrJ8xABhHsMDQni-2FiraC8PMLJQUXDZQ-2FG6m9IZ46xO2Iyc8LQoFO1rXEmGI-2FGx-2Flk2GuIOkj2L2RVtauV-2BMZO-2F24ZyAaVUs-2FtdUJkvvjXtxCUglmHl4eJVJUjN50t60AW8a-2B7Natelg-3D-3D):  Open the "Git Repository Exploring" perspective. Click the icon "Clone a Git Repository and add the clone to this view". Complete the Clone Git Repository wizard using the information below.  URI=https://hub.jazz.net/git/sdaya/LabExercis4-sdaya User=sdaya Password=<*your password*>  Do you intend to code outside the IBM DevOps Services Web IDE with [git](http://sendgrid.softlayer.com/wf/click?upn=9V2tgSzR00a3EACzSlews2gms6xzjh4w4hDPfufqzgBshMsGD9bb4NyhIhe5IKpF_CAkCtUVhogUP8-2B4DkpDTEsG9enp6DMCY0RM2gQKbfR-2BdB93G9wuUJoYmdNanN9v08dnz6vQzzsD0cXGrMoYhN1JMjC2VeL3DgIpXch2RdlIMoEXrfrhocPNoNJsuZAVglCTjTvoEA5hP62SQQ63gpMrtsqCRone-2FDJrihoRyYVJVULh-2FIsp45m4FVO-2Fozfe80M0rKeUzzHugapM44JXkOw-3D-3D) or your favorite Git client?  1. To clone the Git repository using git: Run the command "git clone https://hub.jazz.net/git/sdaya/LabExercis4-sdaya". **Note:** The IBM DevOps Services Git repository is only available via the HTTPS protocol.  For more help choosing and getting started with your preferred client, see [Setting up Eclipse, Git, and RTC Desktop Clients to access IBM DevOps Services](http://sendgrid.softlayer.com/wf/click?upn=awxxXRTrFB8QupIumPNGrCYXCii8211MgnSv8r3vlvTC4zKEAW5oBwR-2FNe7H-2BL24_CAkCtUVhogUP8-2B4DkpDTEsG9enp6DMCY0RM2gQKbfR-2BdB93G9wuUJoYmdNanN9v0aTMxWrSAJ1hkGck7Bv2yze-2Fso9knKOyAfbpr-2F7NrP-2Fx4f9L5imzkDh-2B7heXx2SRCy9EQdtfjgICe4iDXkhHDRTUNEmndpRGhcfsawxuFnhgCNnra1QjD6B6S7gGRl-2BL6dJywGxllHwU-2FPCvQjFqidw-3D-3D).  Have questions? Check out [IBM DevOps Services help](http://sendgrid.softlayer.com/wf/click?upn=awxxXRTrFB8QupIumPNGrMXXl7NCzTesCu-2Bjjpj-2B9Ro-3D_CAkCtUVhogUP8-2B4DkpDTEsG9enp6DMCY0RM2gQKbfR-2BdB93G9wuUJoYmdNanN9v04Asu8S0xkOw2XutBs7dUcN-2BCApBpPsvjD4UY1sz0KABlo4-2FGEi-2BLu2pqx5-2BgHz-2FAHGwTwHwlyphz8gz5MaMbmpedeneHOtCxh3fmifQP6IBAmtcUrFwKnNNqN6b1zK4TqnoLsr8JbAWIXtbg-2Bd8xvg-3D-3D). |

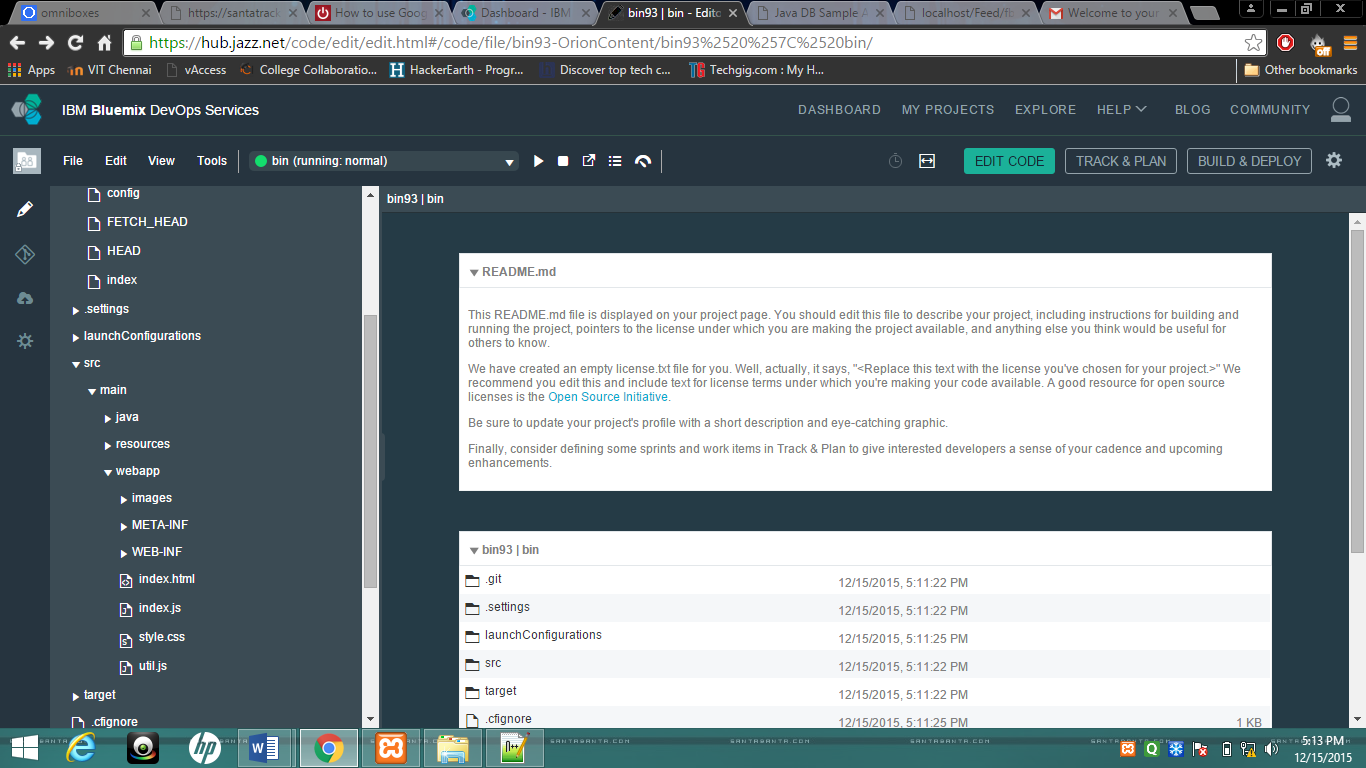
1. The overview page now shows the GIT URL and a link to access the project artefacts (using the Code link)

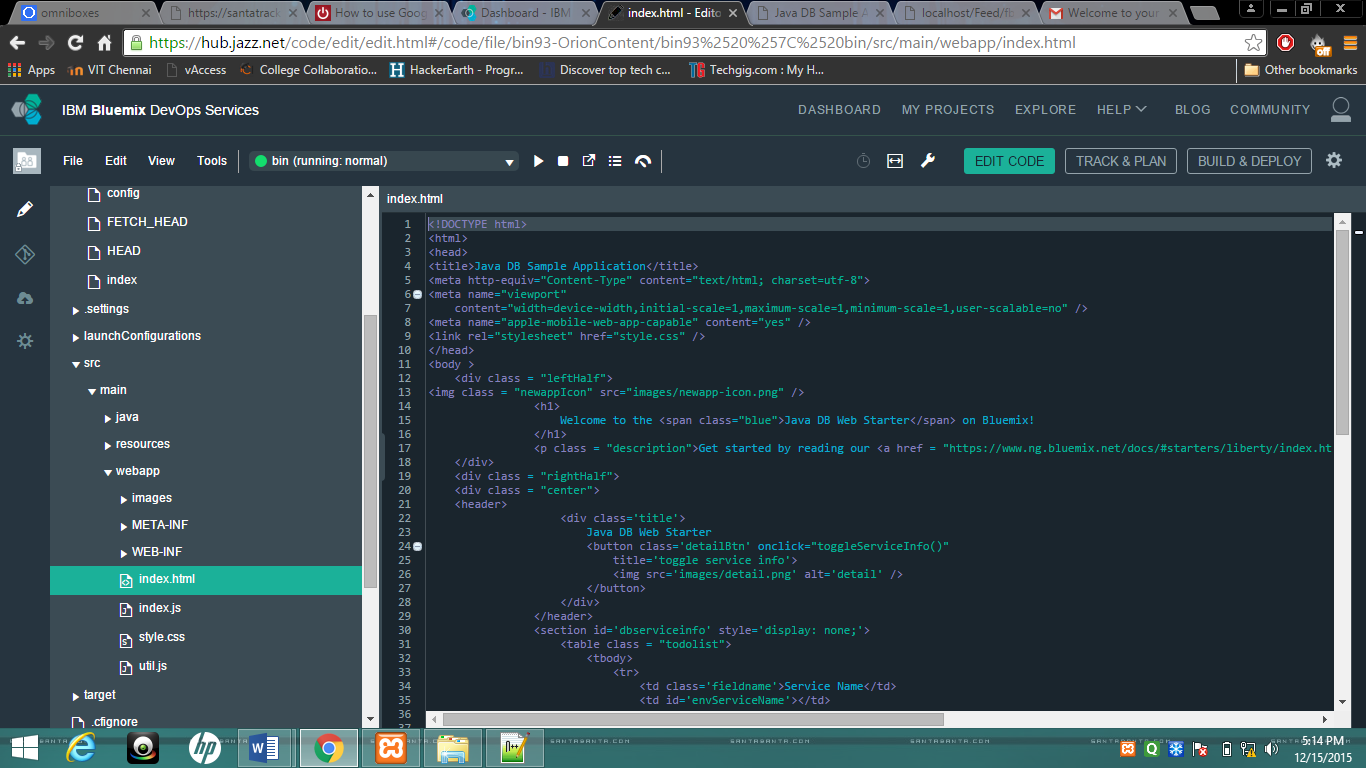


1. Click on the EDIT CODE link. A new browser window will open where you will see the project artefacts. This new browser window is the DevOps Services user interface that provides your Git Services.

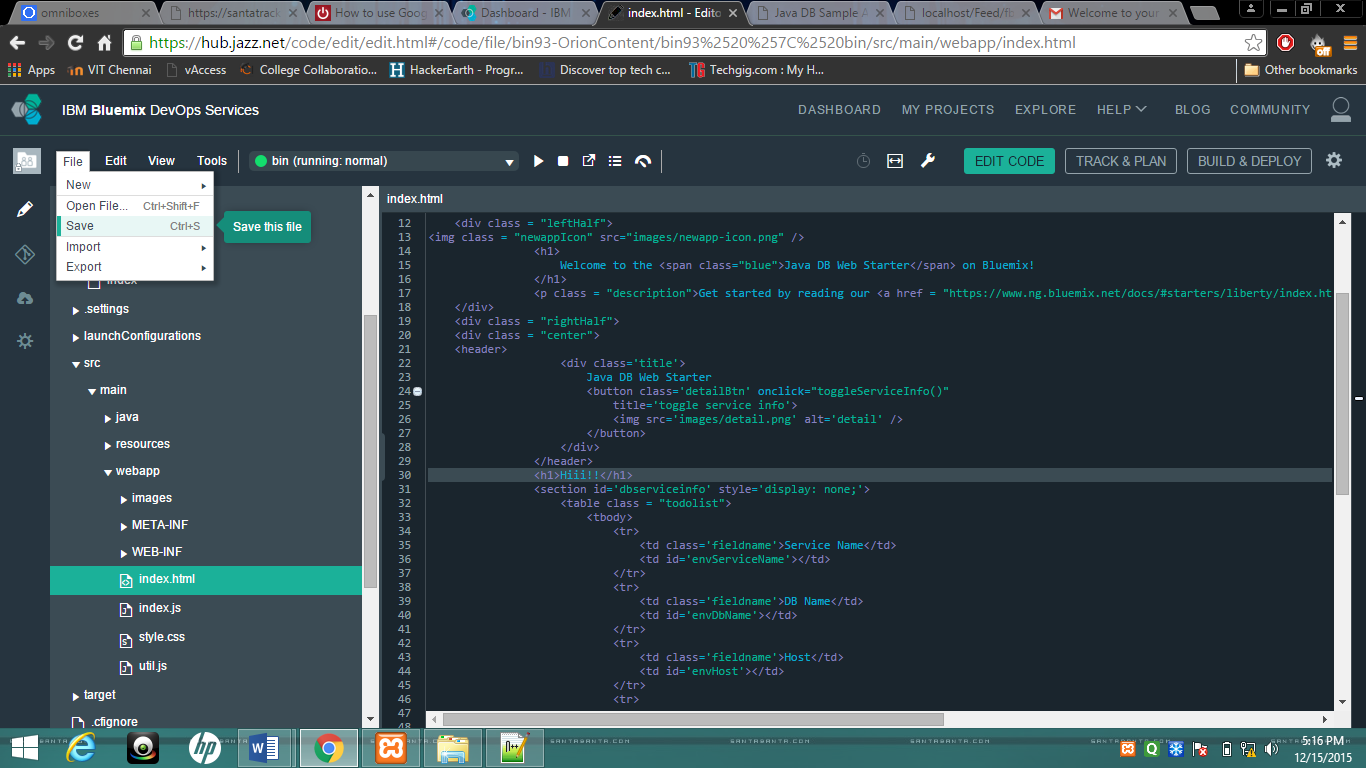


1. Expand WebContent in the navigation bar and locate index.html

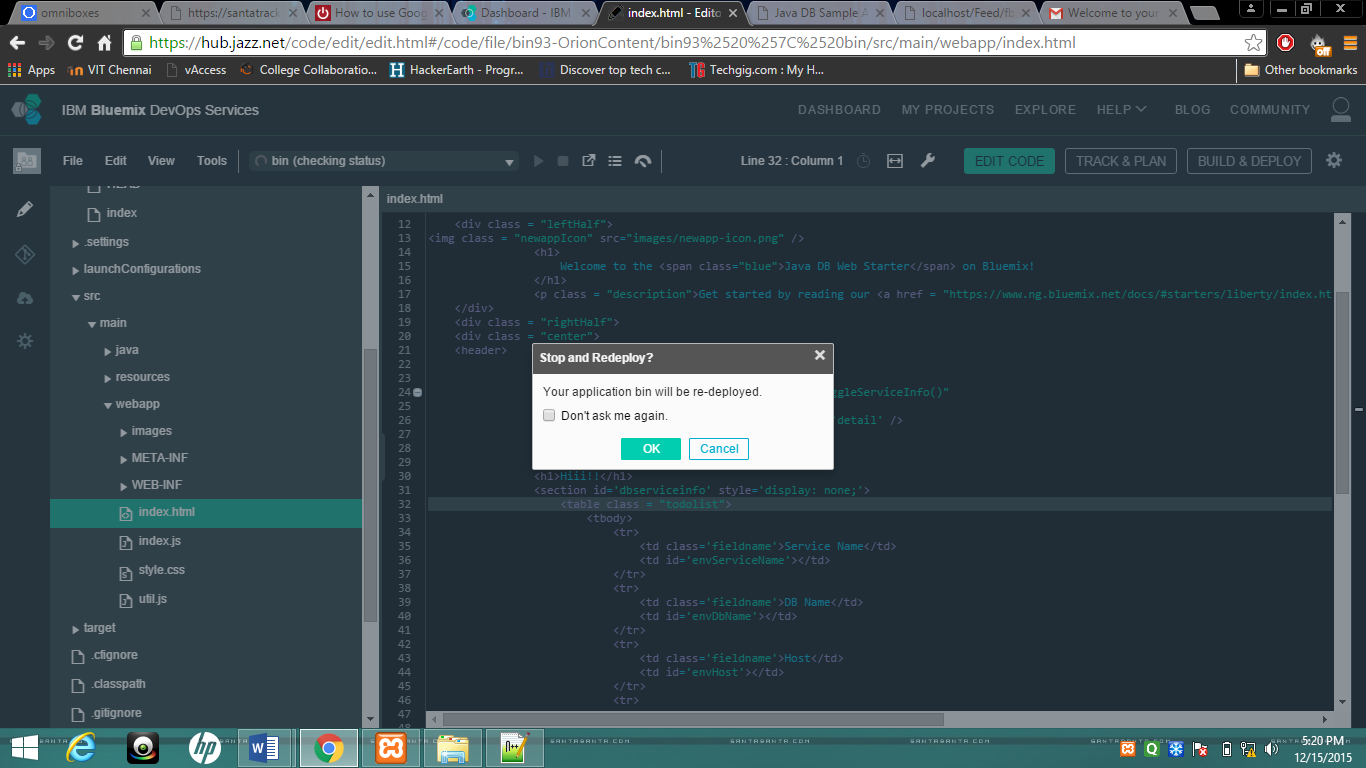


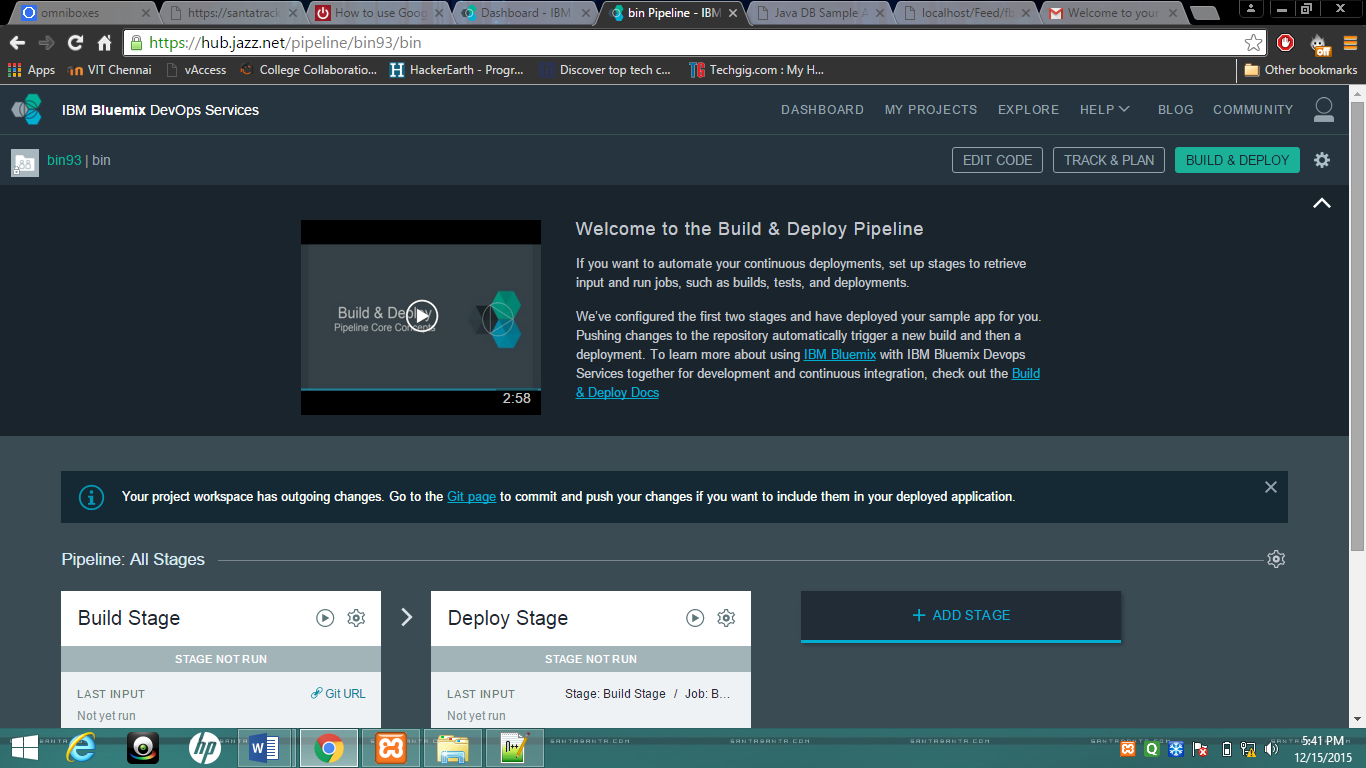


1. Make minor changes to the index.html file to explore the build and deploy functionality. Locate the body onload line and change the header to read differently. As an example add a "test" phrase as suffix. Also change the web forms label names: The changes are shown in bold text below.

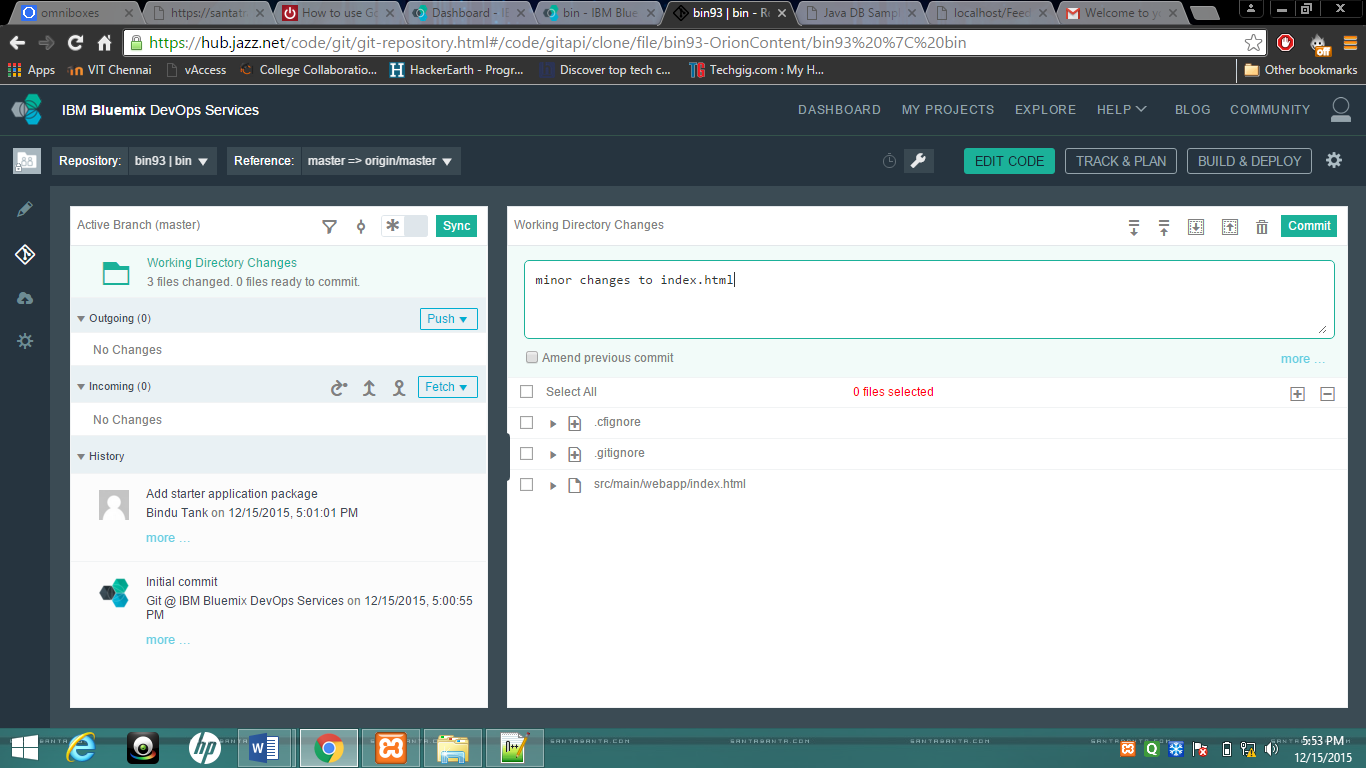


1. Click on BUILD & DEPLOY. You will see a screen similar to the following.

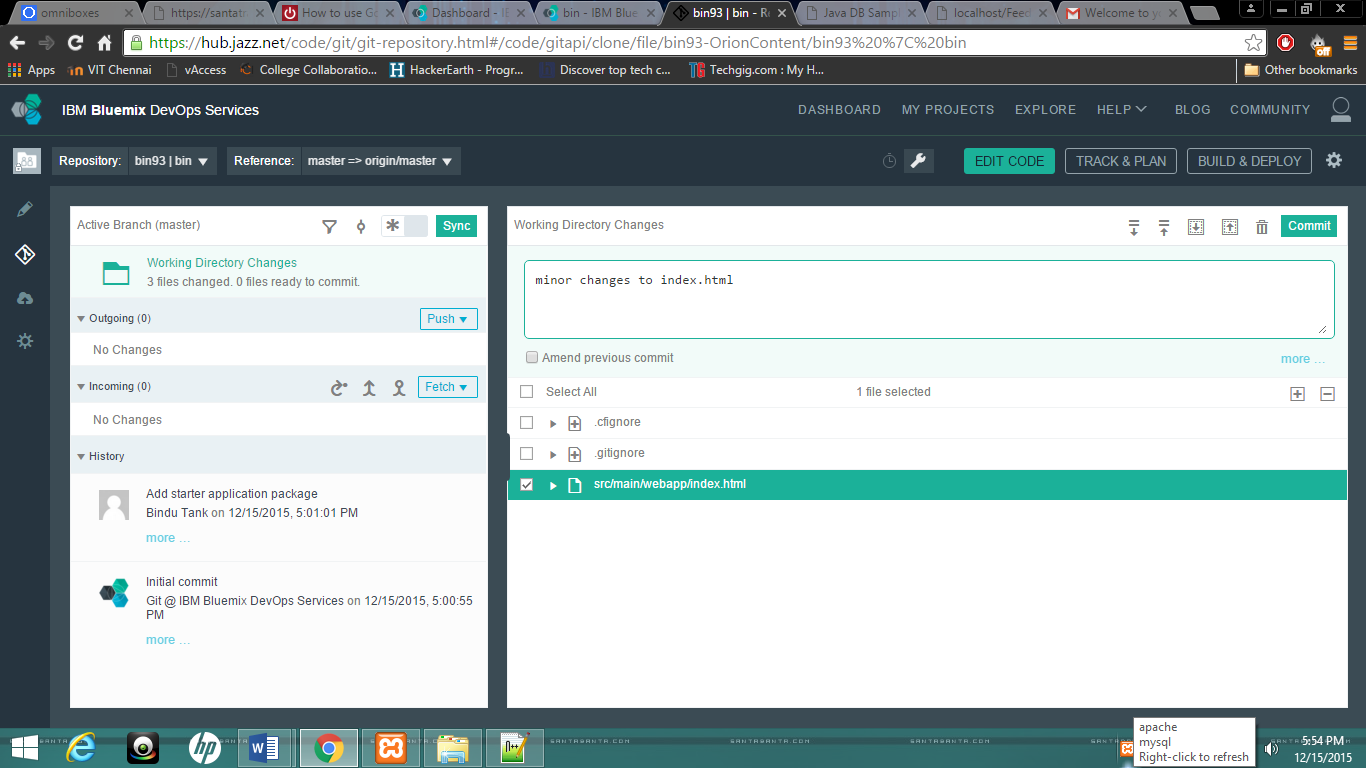




1. IBM DevOps Services has identified that there is an outgoing change.
2. Click on the “*Git Repository View”* link to identify the outgoing changes and commit them. The build process would use the checked in artefacts.
3. Check the WebContent/index.html. Specify a commit message and click on COMMIT. The changes are committed.

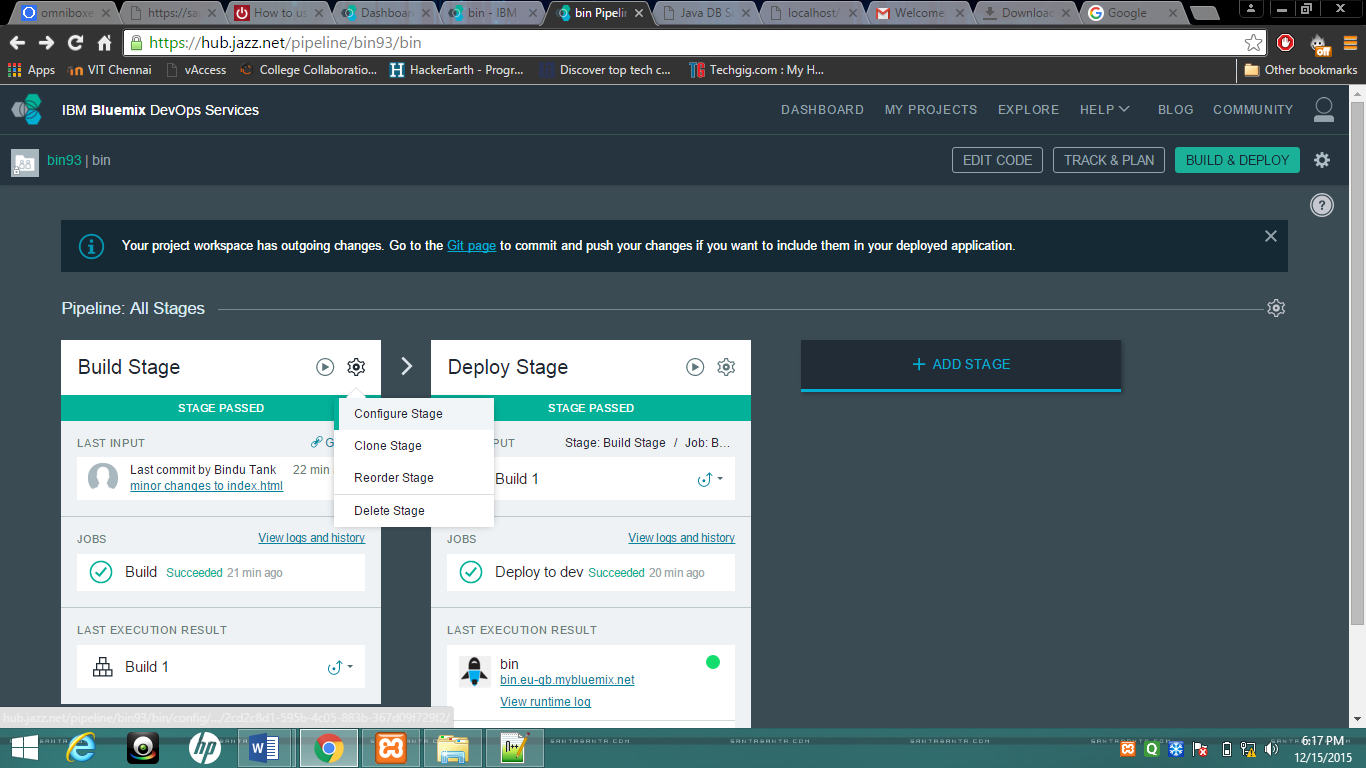


1. Push the changes to the remote origin master by clicking on the PUSH button.
2. Now the local and remote Git Repositories are in Synch as seen by No Incoming and Outgoing commits

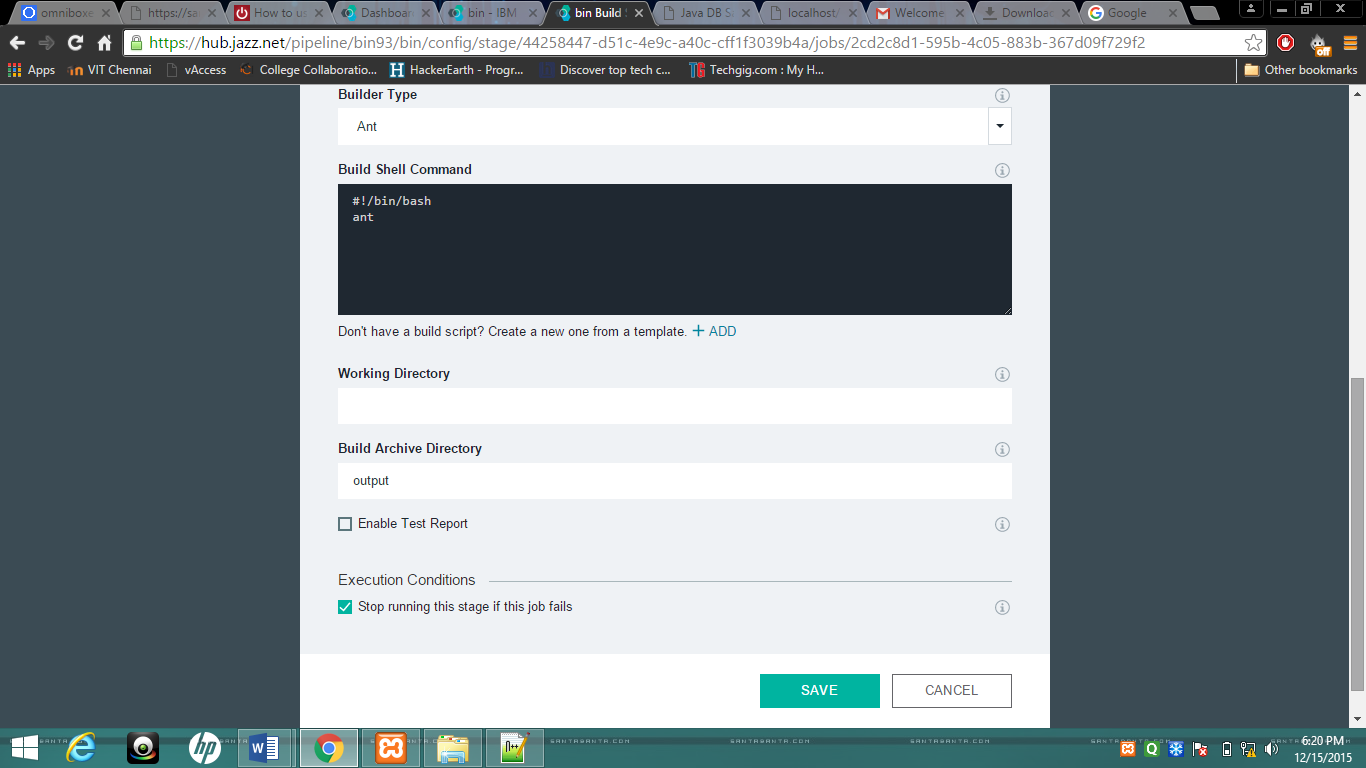


1. Click on BUILD & DEPLOY
2. Click on the gear button at the top . You will see the following screen showing a pipeline with a Build stage and a Deploy stage. The *Builder* and *Deployer* will need to be configured.

Click on configure Stage

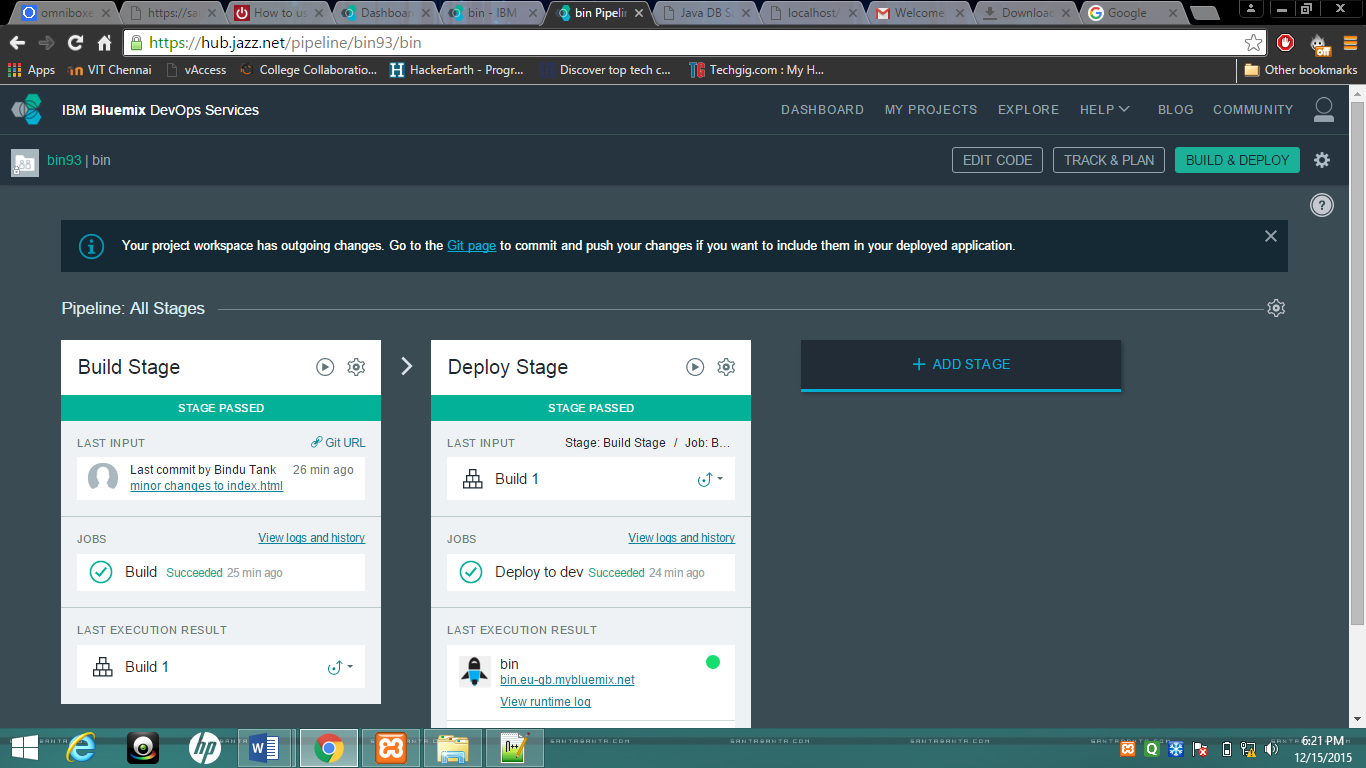


1. Click on the gear icon in the upper right hand corner of the Builder. Examine the builders that are available.

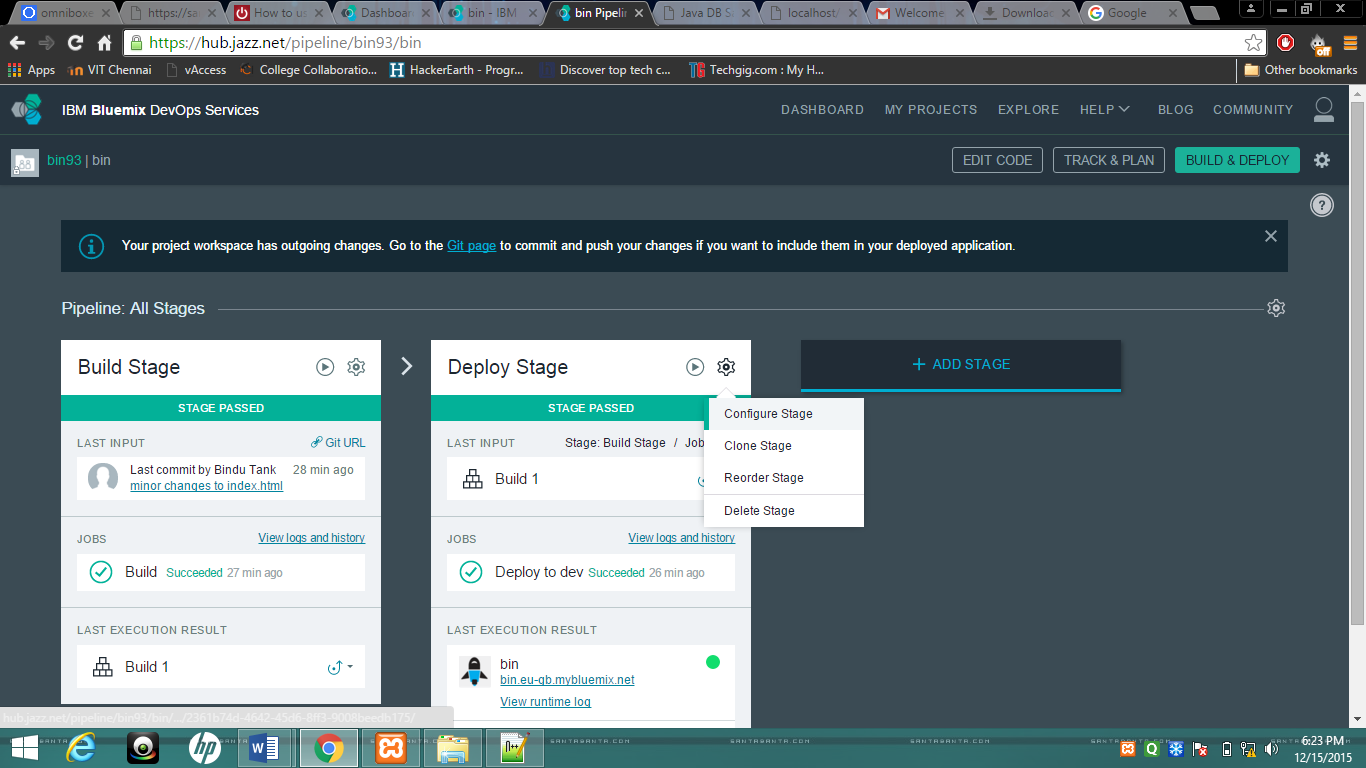


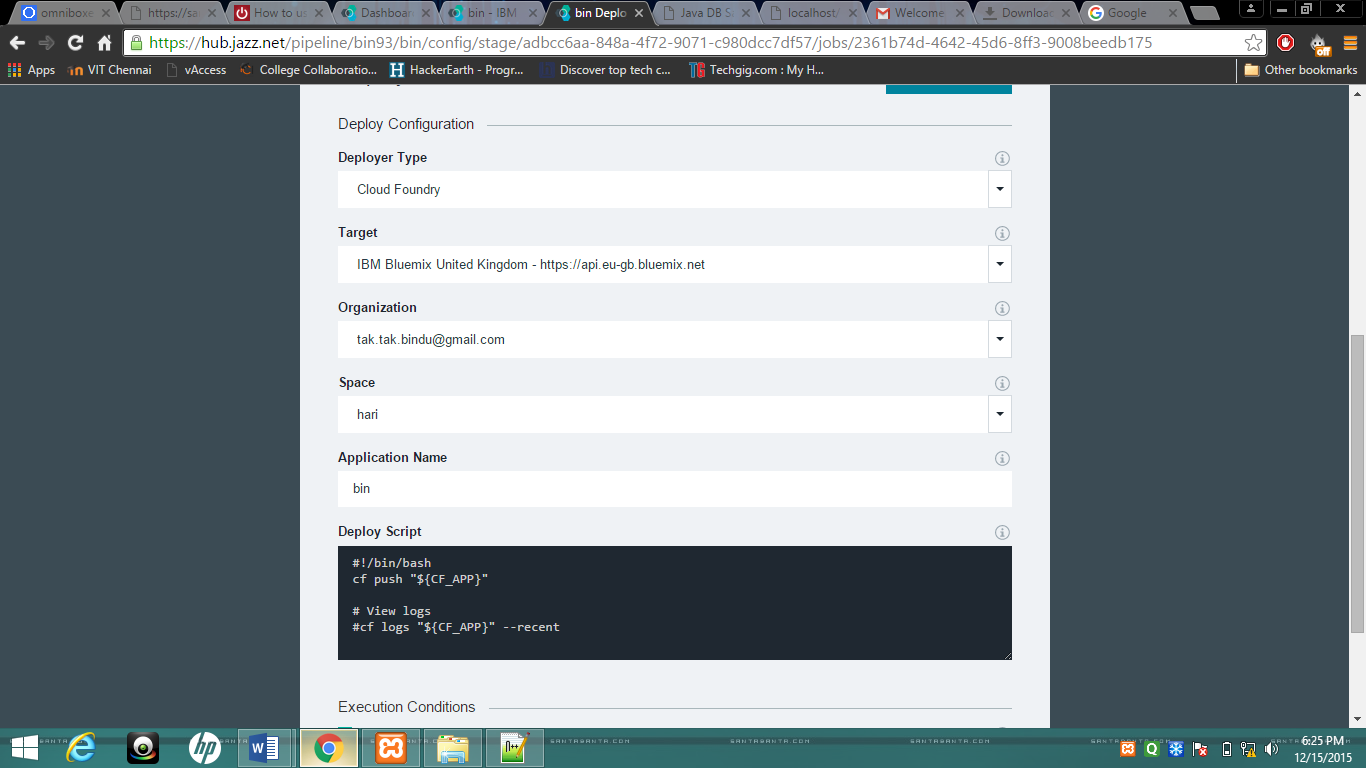
**Note:**  Maven has been recently introduced.

1. Choose *Ant* as the Builder. Accept the defaults for the others in the Builder configuration.
2. Click on SAVE to complete the Builder Configuration. If you get an error message regarding the “Build archive directory”, remove the value “output” from the field and try again.
3. Click on the gear icon in the upper right hand corner of the *Deployer* to configure the Deployer.

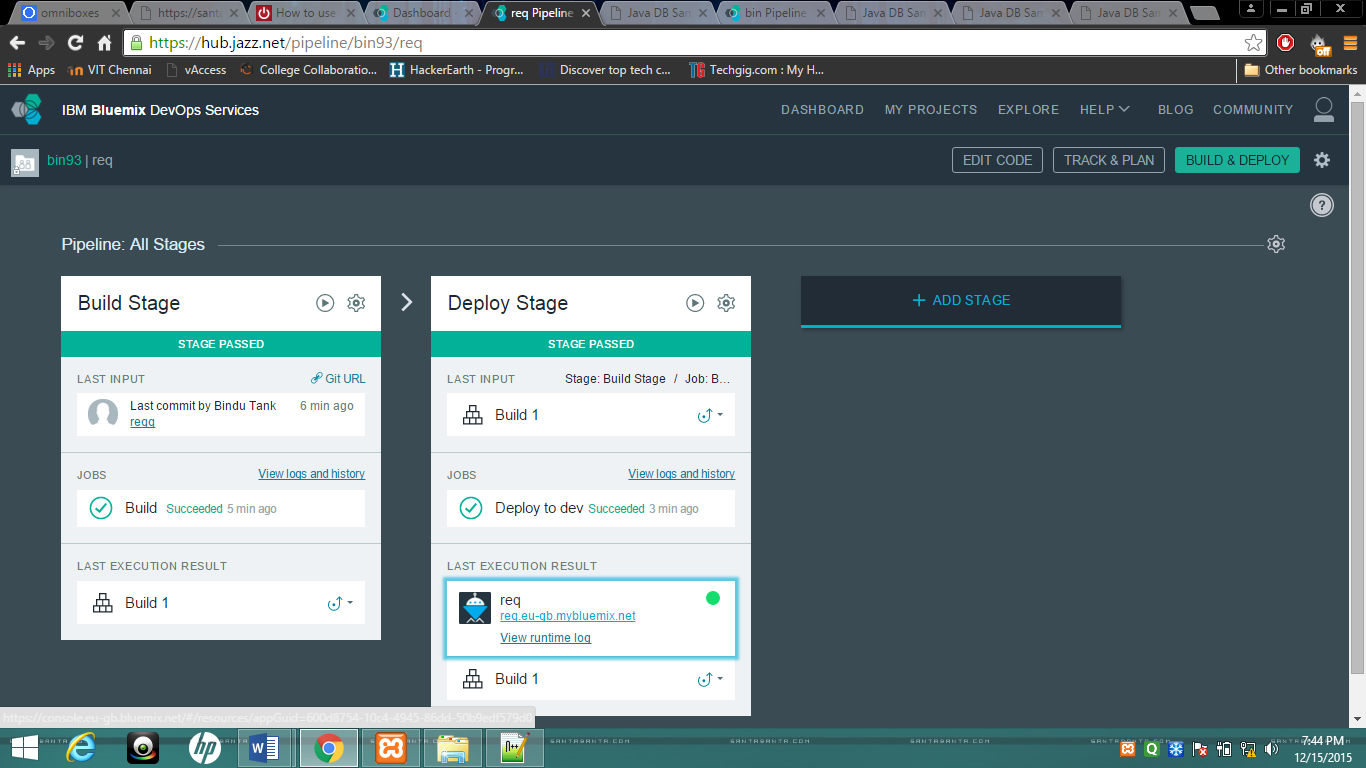


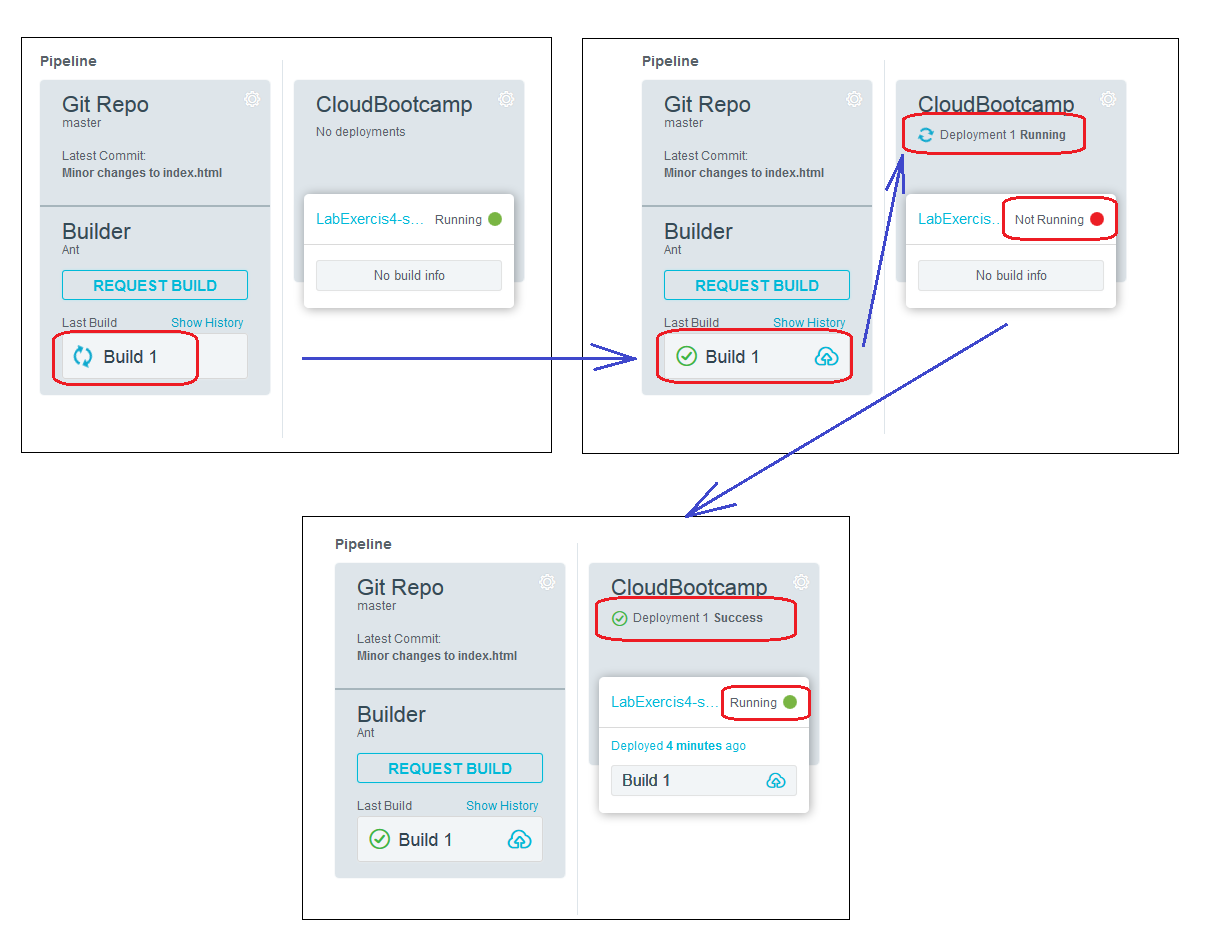
1. Specify the space that you work with. We worked with the "CloudBootcamp" space so specified this in the Deploy configuration. Accept the defaults for the rest of the Deployer configuration and click on Save. Now you have configured both the Builder and Deployer.





1. A Build is kick started and it is automatically followed by a Deploy. Build and Deploy are two discrete activities (stages in the pipeline) and each of them has their own status. Observe below that the build is successful and has been successfully deployed. The illustration below shows the progression from build to deploy.
2. Click on the URL to see the changes.



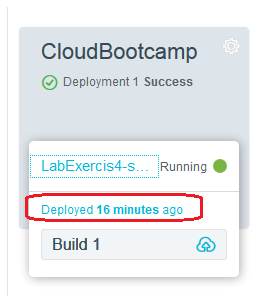


1. The success of Deploy is communicated with the message "*Deployedt*" while the keyword "*Success"* is used to state that the Build is successful.
2. Click on the application URL link to confirm that the changes you made to the index.html were successfully deployed.

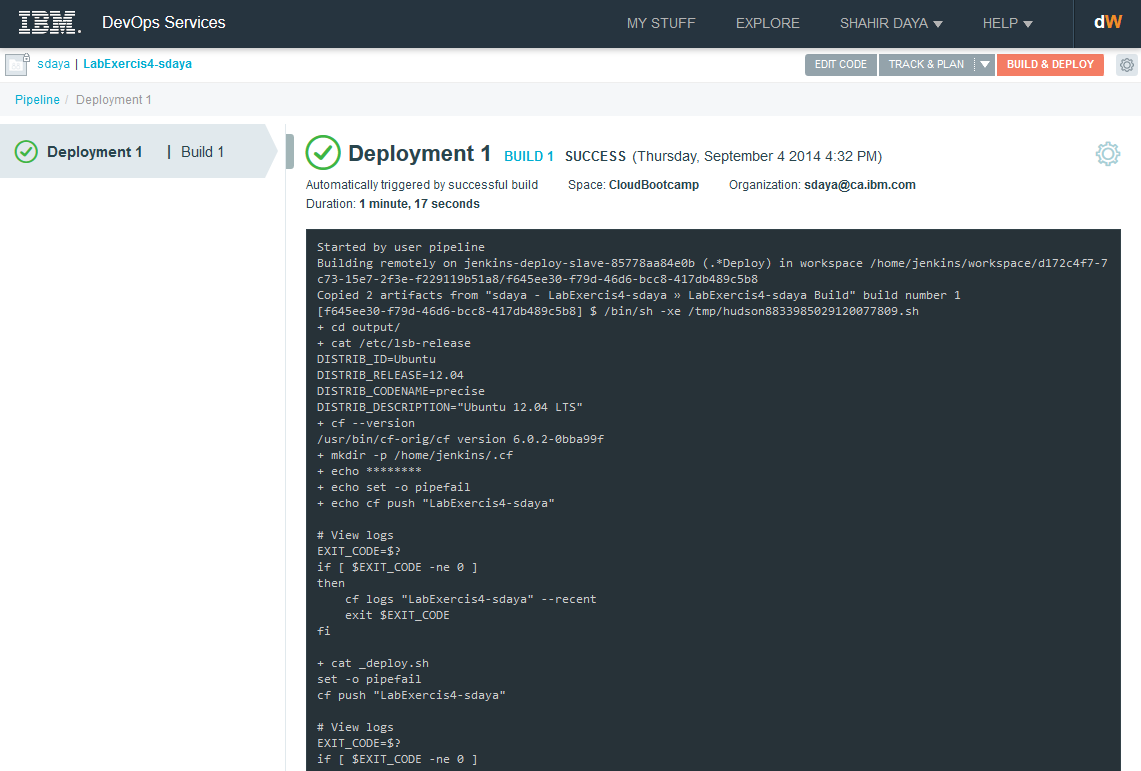
## Viewing the Build and Deploy logs

In this section, you will view the build and deploy logs.

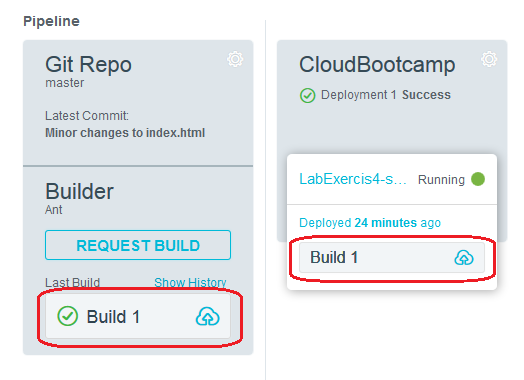
1. The Deployment logs can be seen by clicking the *Deployed* or *Failed* message (depending on the outcome of the Deploy step). In our case the Deploy was successful. Click on *Deployed*



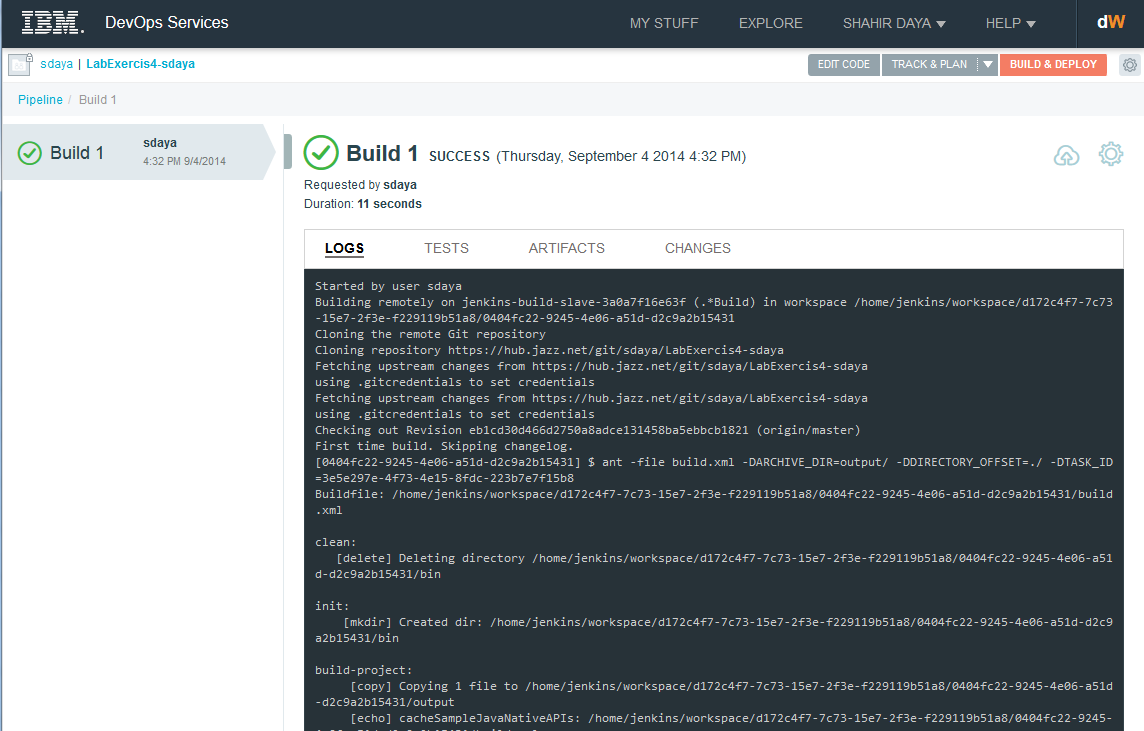
A screen similar to the following will be displayed.



1. Now let's view the history of our Build step. This can be viewed by clicking on Build 1 link. It is available in two places as marked in the figure below.



1. Examine our Build Logs



1. The builder uses "Jenkins" under the covers to perform the application build. Explore the other tabs; TESTS, ARTIFACTS and CHANGES.

In this exercise you created a java web starter app on Bluemix. You moved the artefacts to a Git repository provided by IBM DevOps Services. You made a change to a project artefact. The changes were built and deployed to Bluemix.

# References

1. Hub.jazz.net, (2014). *Learn - IBM DevOps Services*. [online] Available at: <https://hub.jazz.net/learn> [Accessed 4 Sep. 2014].