**Fast Start 2020 (v1.1)**

**Session:** NZS68T1185S - End to End Pipeline Part 3: Lab

**Presented by:** Nelson Lopez DevOps Acceleration Team/Raleigh (nelson.lopez1@ibm.com)

**Setup:**

Access to the latest zTrail image will be provided to the first 30 via slack(below) by Nelson 1 day before the lab. You are encouraged to test your connectivity before the lab. Any issues please slack me (see below).

**Slack Channel:** **#dat-lab**

**Slack Link:** <https://ibm-systems-z.slack.com/archives/CU3RA9C3Y>

**Lab Objective:**

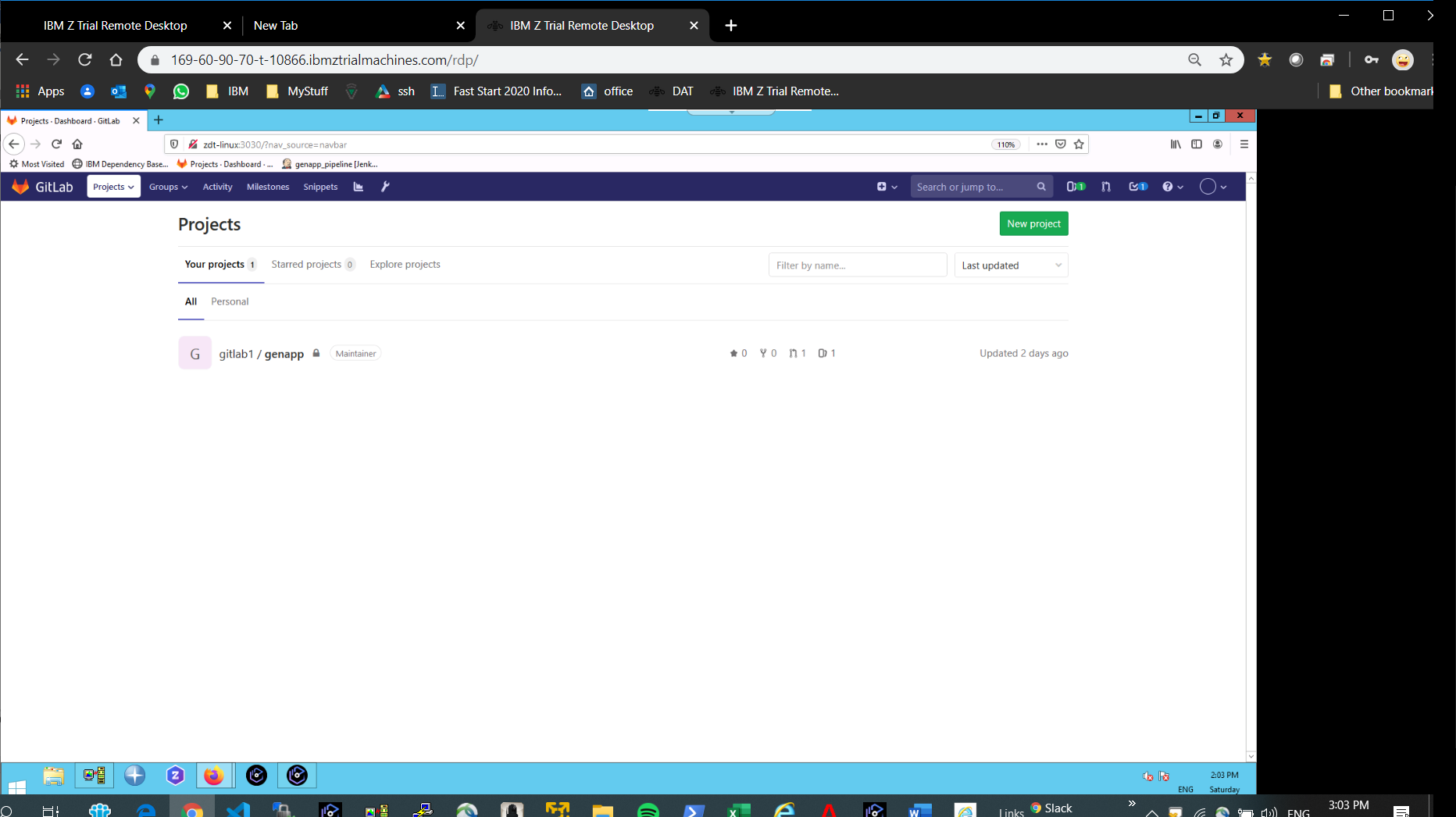
To migrate a sample mainframe application from MVS PDS’s to a GitLab repo. This is a hands-on lab that follows the migrate topic presented at “NZS68T0155SH End to End Pipeline Part 2 **Migration**, Packaging and Deployment Strategies, POC Cookbook”

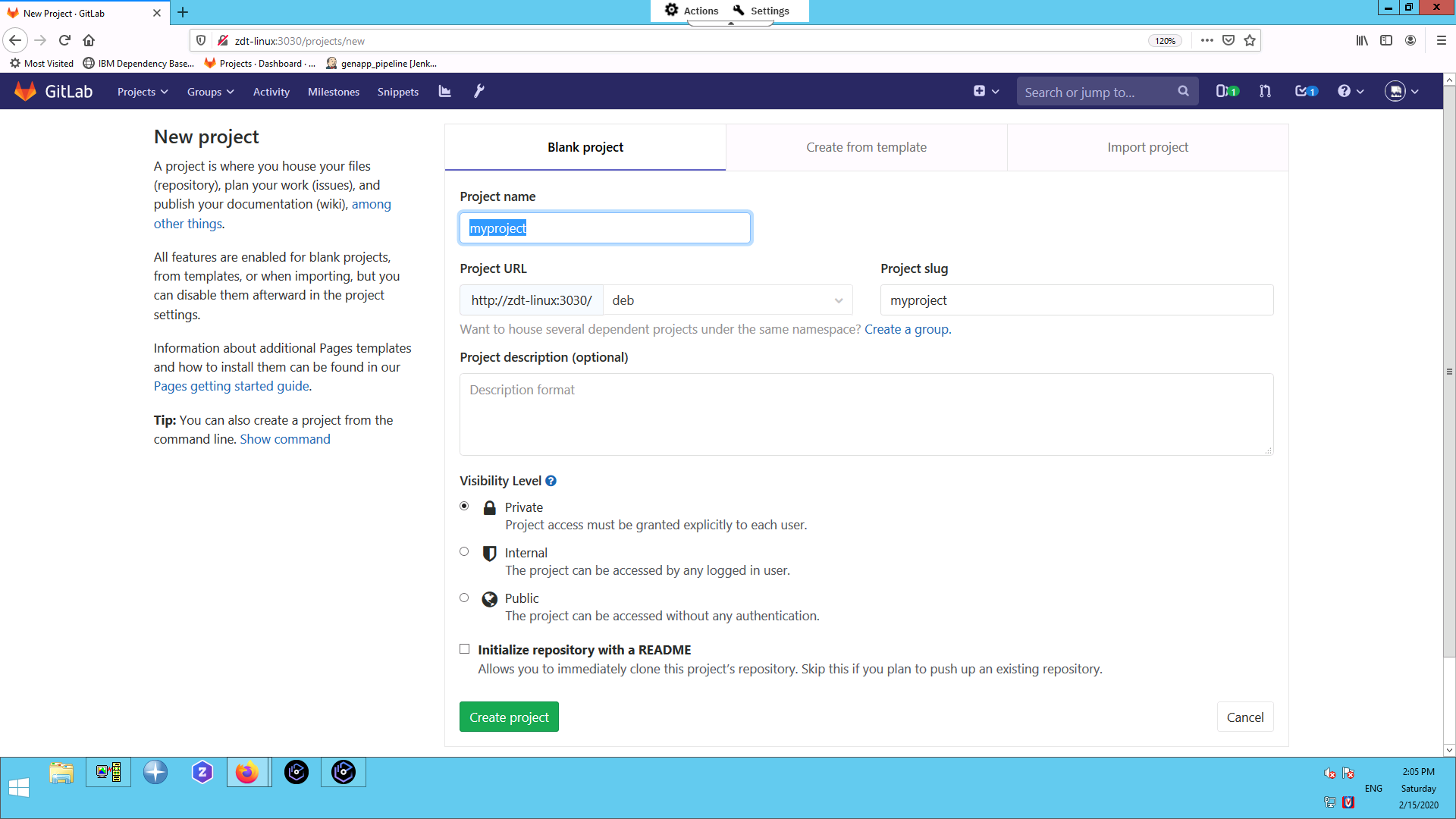
**Tools/skills/environment to be used:**

* Tools Used: IDz 14.2.1 , DBB 1.0.6 Migration tool , GitLab
* Skills leaned:
  + Running the DBB Migration tool on Unix Systems Services (USS)
  + Writing and executing shell scripts
  + Working with Git repo’s – create, clone, commit, add
  + USS command line processing
  + Basic understanding of USS file tagging
  + SSH terminal access
  + Cloning Repo’s in IDz’s Git perspective and viewing them in as z/OS projects.
* The environment will be a pre-define zTrail[[1]](#footnote-1) instance configured by Vijay Katoch, Suman Gopinath and David Bean.

**Lab Work:**

**STEP 1 – Create an empty repo**

* Logon to your zTrial account and open the browser
* Got to the Gitlab site **zdt-linux:3030**
* Create a new project named “**myproject”**
* Minimize the browser for later



**Step 2 – Migrate a sample mainframe application to USS**

* Open IDz and select “Golden zCloud” remote system (right side)
* Under “My Home” , create a new file called “mig.sh” and cut/paste these lines:

**rm -rf myproject**

**git clone gitlab@10.1.1.1:deb/myproject.git**

**$DBB\_HOME/migration/bin/migrate.sh -r $HOME/myproject/myapp \**

**-m MappingRule[hlq:GENAPP,toLower:true,pdsMapping:false,targetDir:cobol,extension:CBL] "COBOL(LGIC\*)"**

**cd myproject/myapp**

**echo "\*.cbl zos-working-tree-encoding=ibm-1047 working-tree-encoding=ibm-1047 git-encoding=utf8" > .gitattributes**

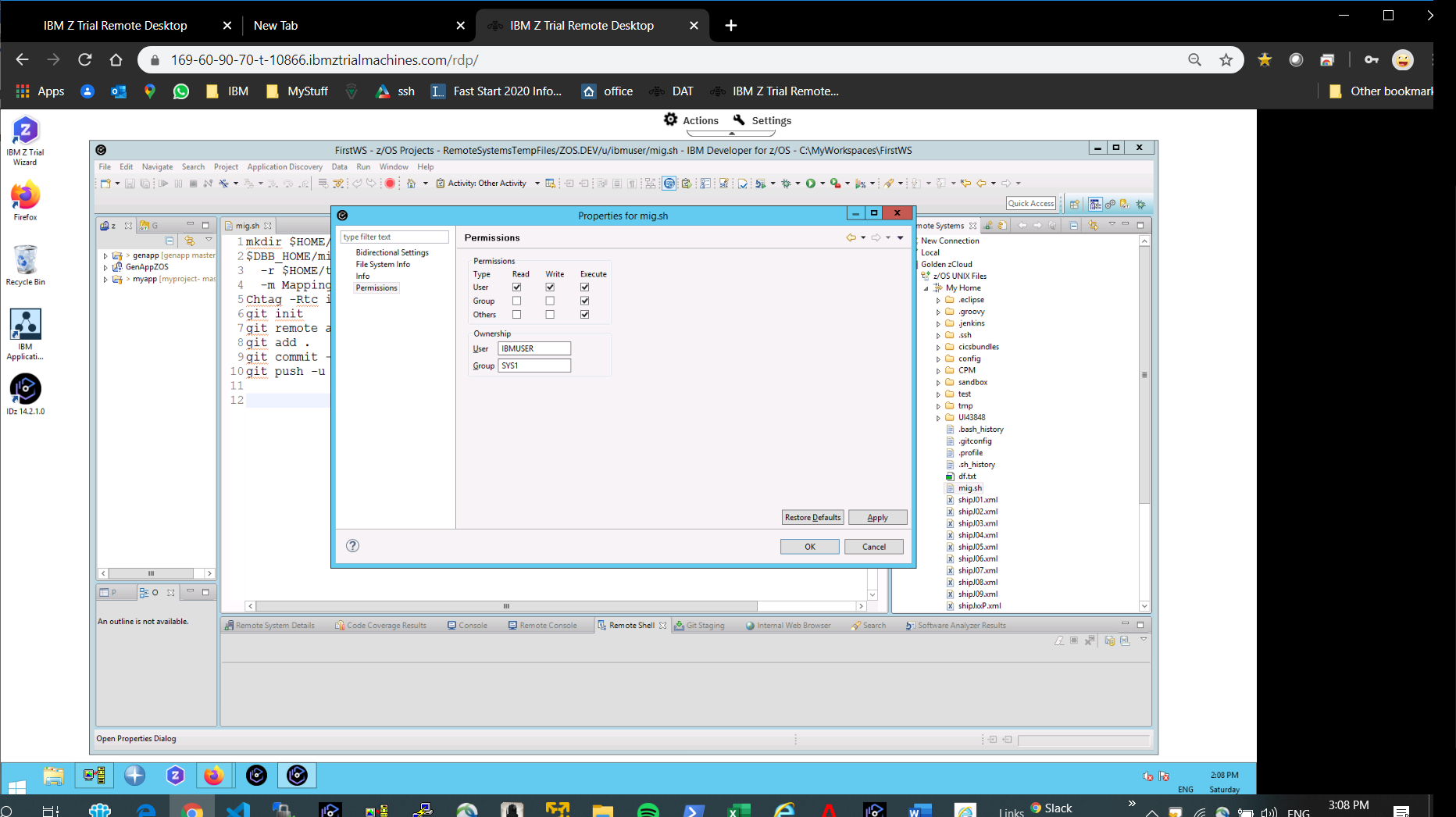
**echo ".gitattributes zos-working-tree-encoding=iso8859-1 working-tree-encoding=iso8859-1 git-encoding=utf-8" >> .gitattributes**

**git add .**

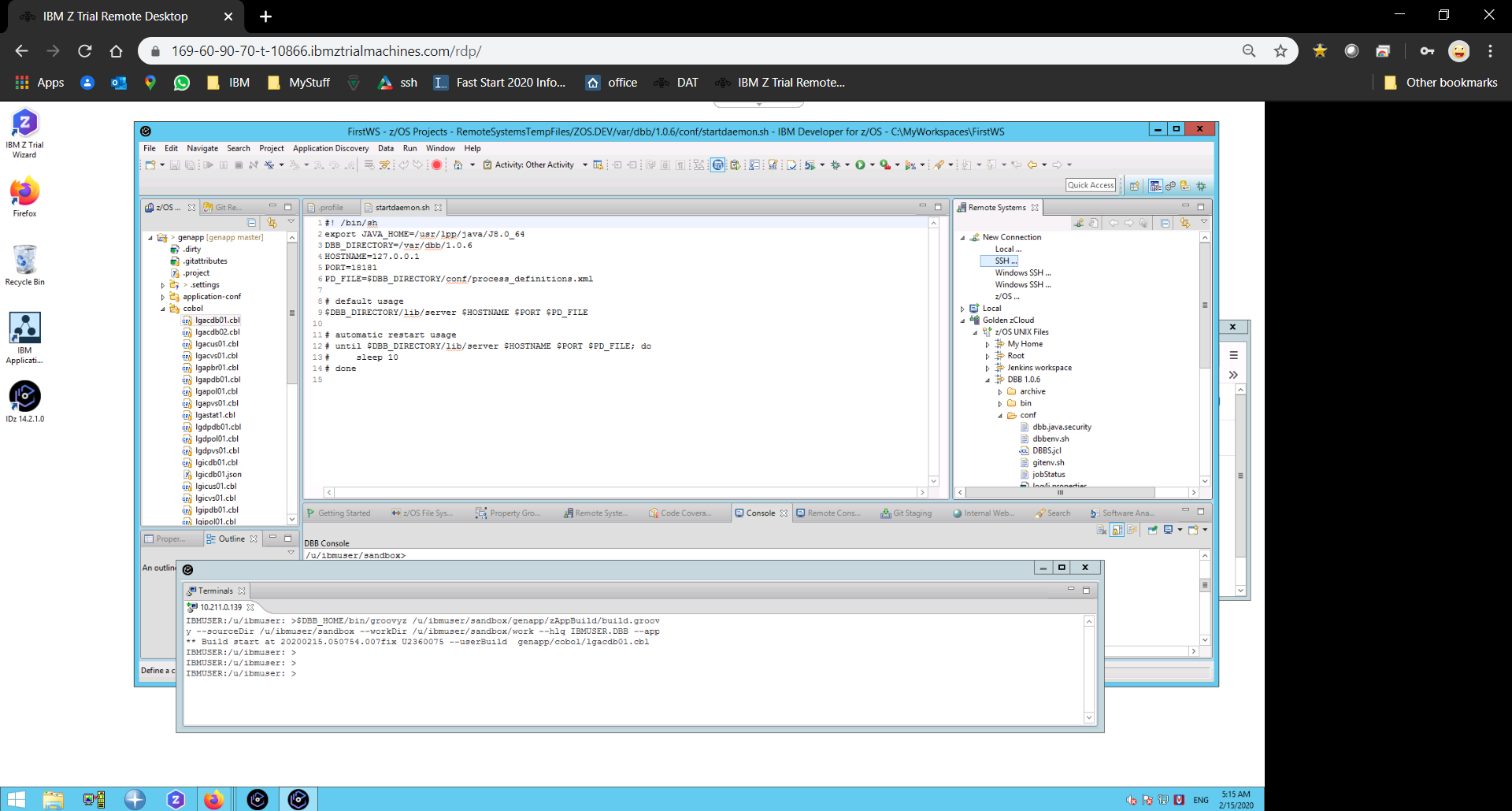
**git commit -m "Go team DAT!"**

**git push -u origin master**

* Save the file and change its permission to execute



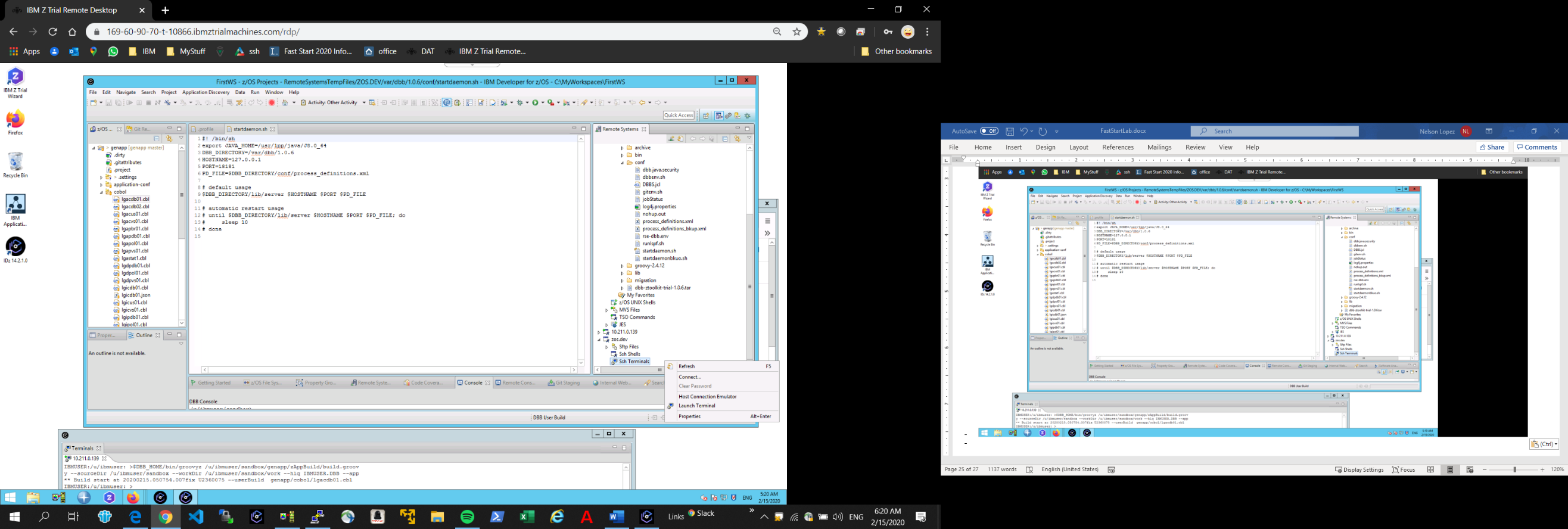
**Step 3 – Create an SSH terminal connection in IDz**

Create a new SSH connection from IDz’s “Remote Systems” pane.   


Enter the host name “zos.dev” and press “Finish”.

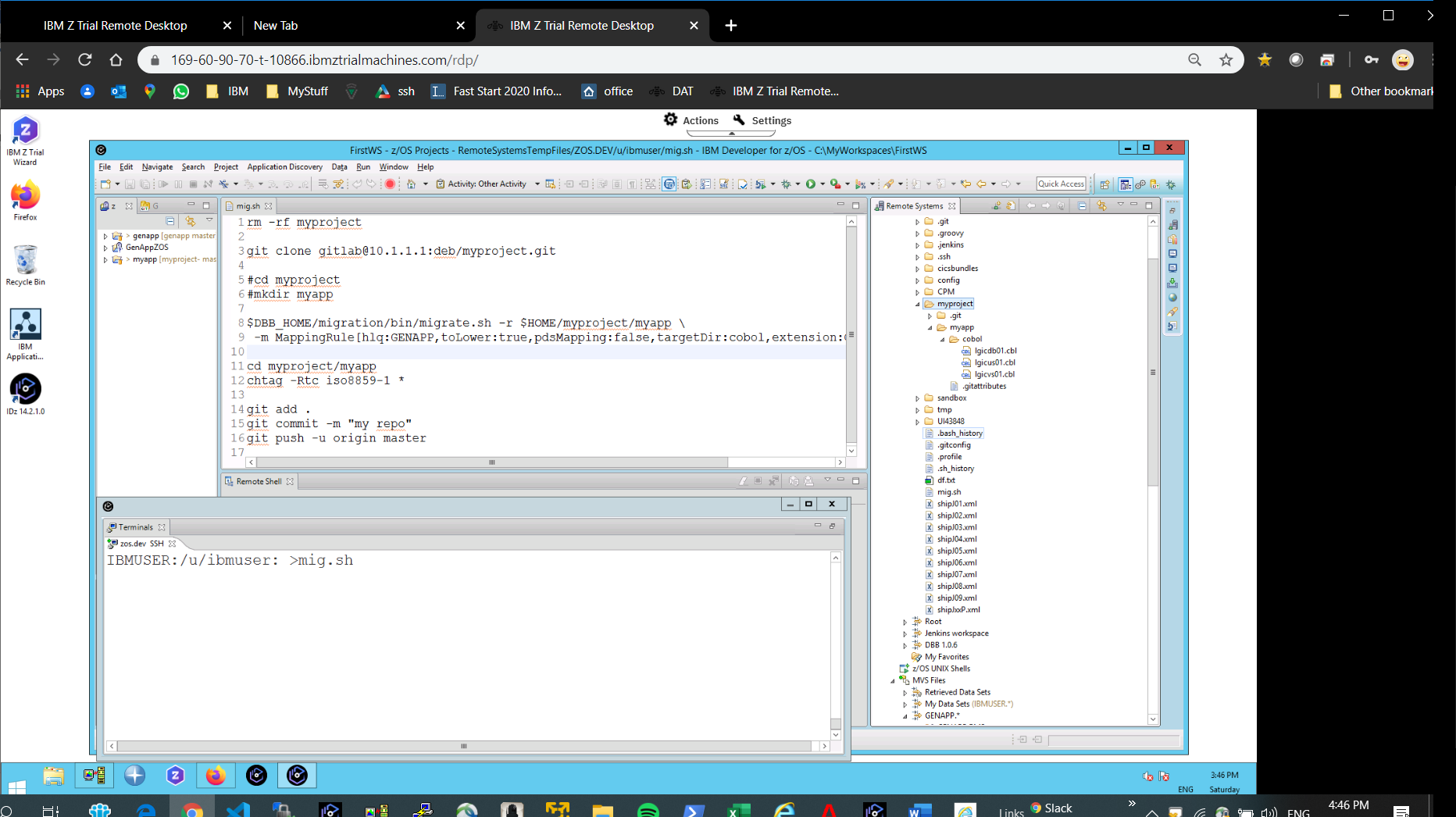
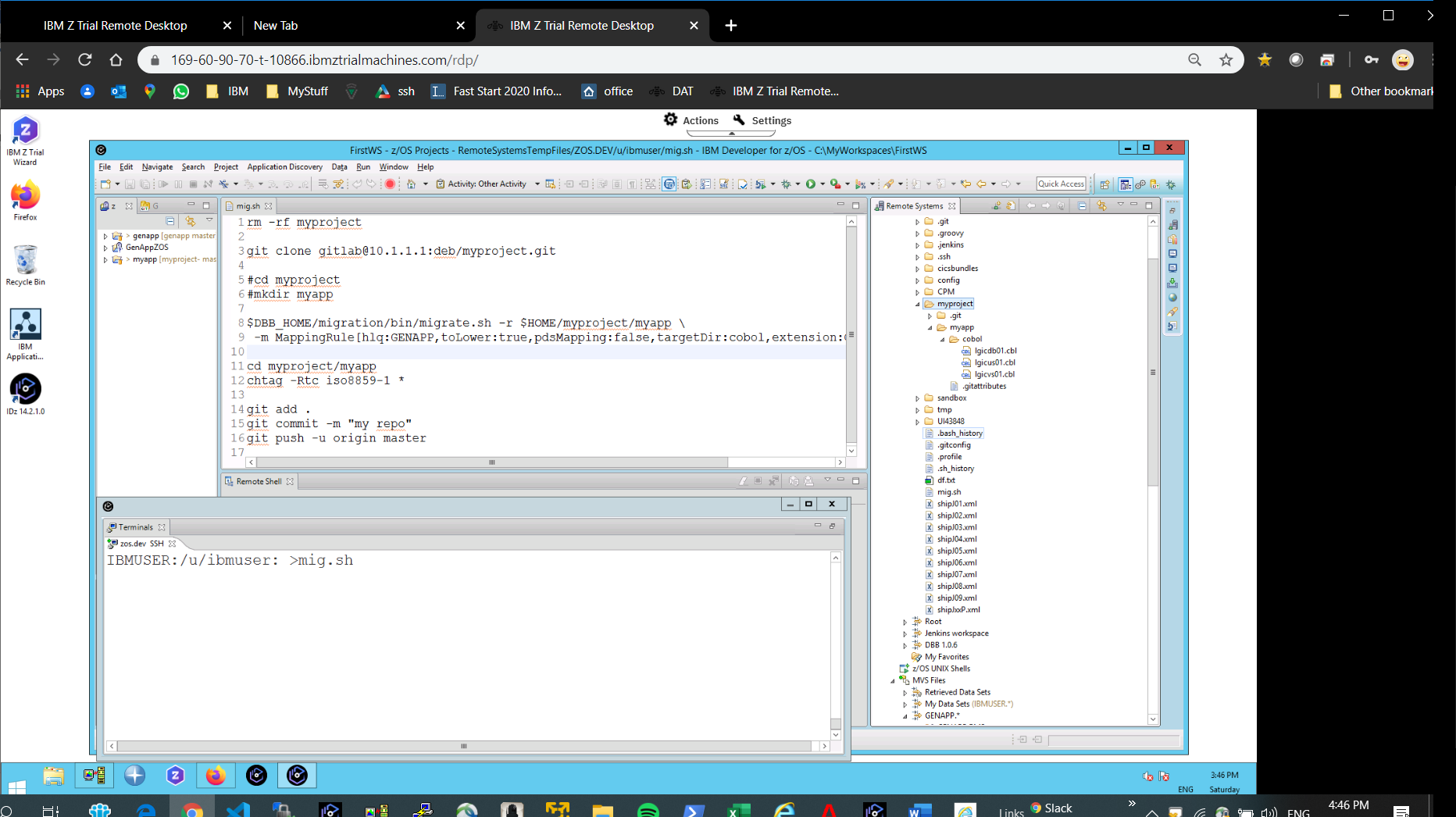
Under the new “zos.dev” connection, right click “Ssh Terminals” and in properties/subsystem enter 1022 for the port and “OK”

Right click on “zos.dev” again and “Launch Terminal” to start an SSH session in USS with the standard zTrial account (“ibmuser”password “sys1”) and follow the prompts.



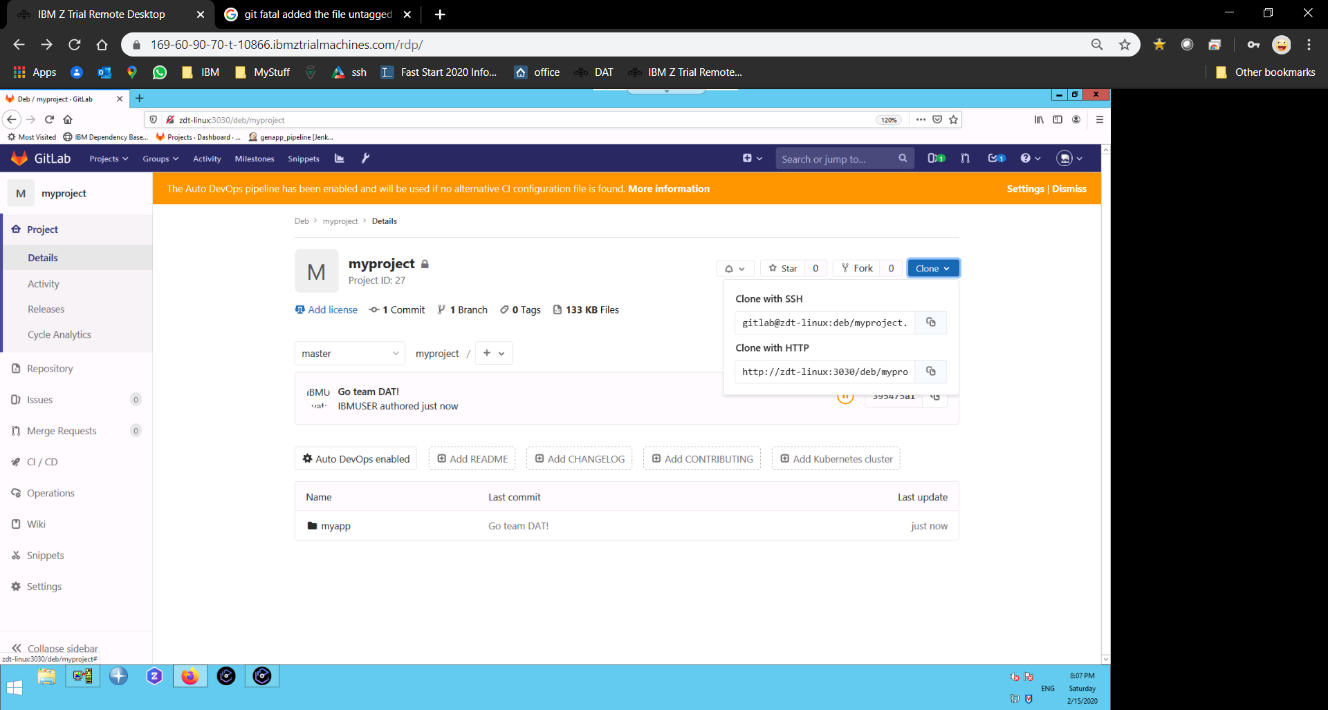
**Step 4 Run the DBB Migrate tool**

* From your SSH terminal enter these commands (the first line is a one-time cleanup on this zTrial image)
  + rm .ssh/known\_hosts
  + “mig.sh”
* The “mig.sh” script runs the DBB migrate tool. In this example, it copies members starting with “LGIC\*” from the MVS PDS “GENAPP.COBOL” into a new cobol folder under your USS home directory “myproject/myapp”. It also appends a “cbl” extension to each copied file.

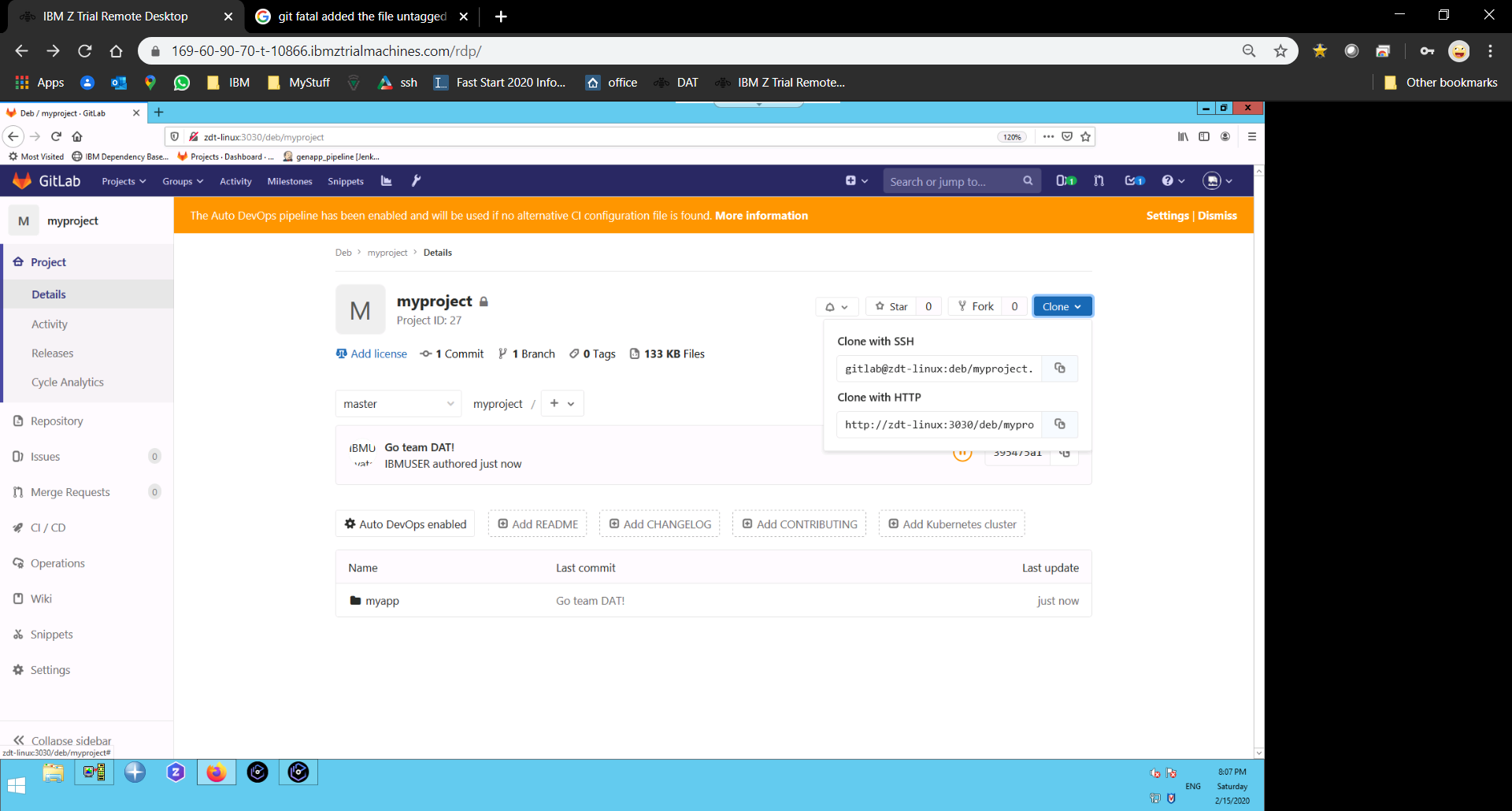


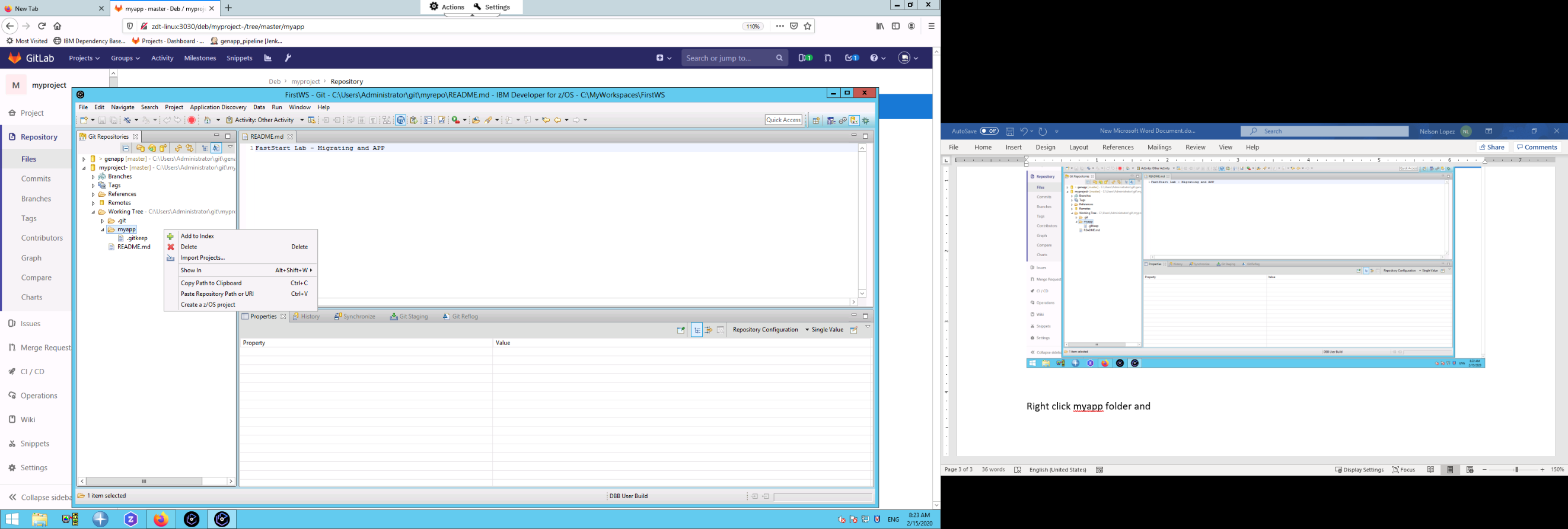
* Return to your browser and refresh your page to review and confirm that your migration was successful.

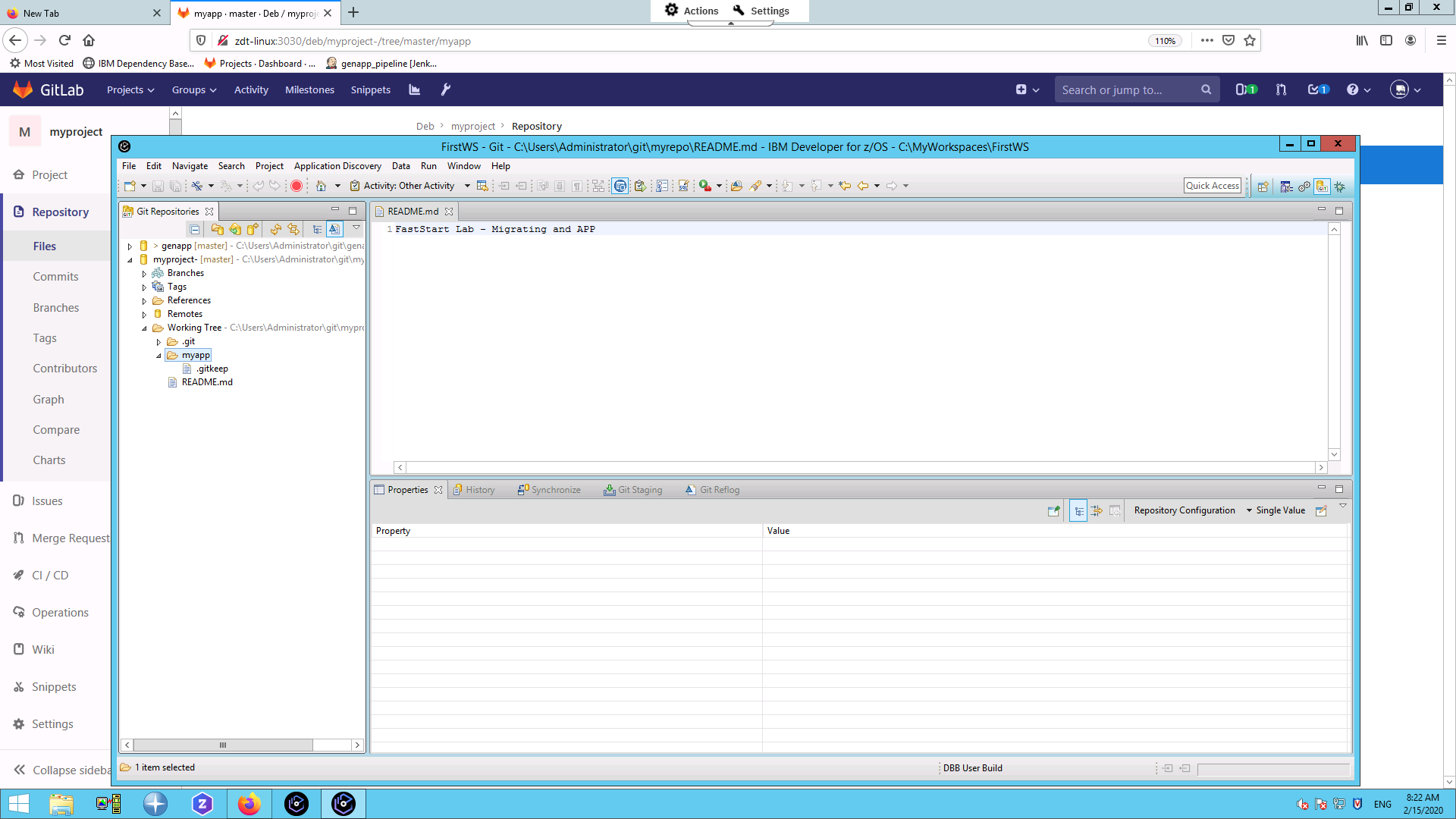
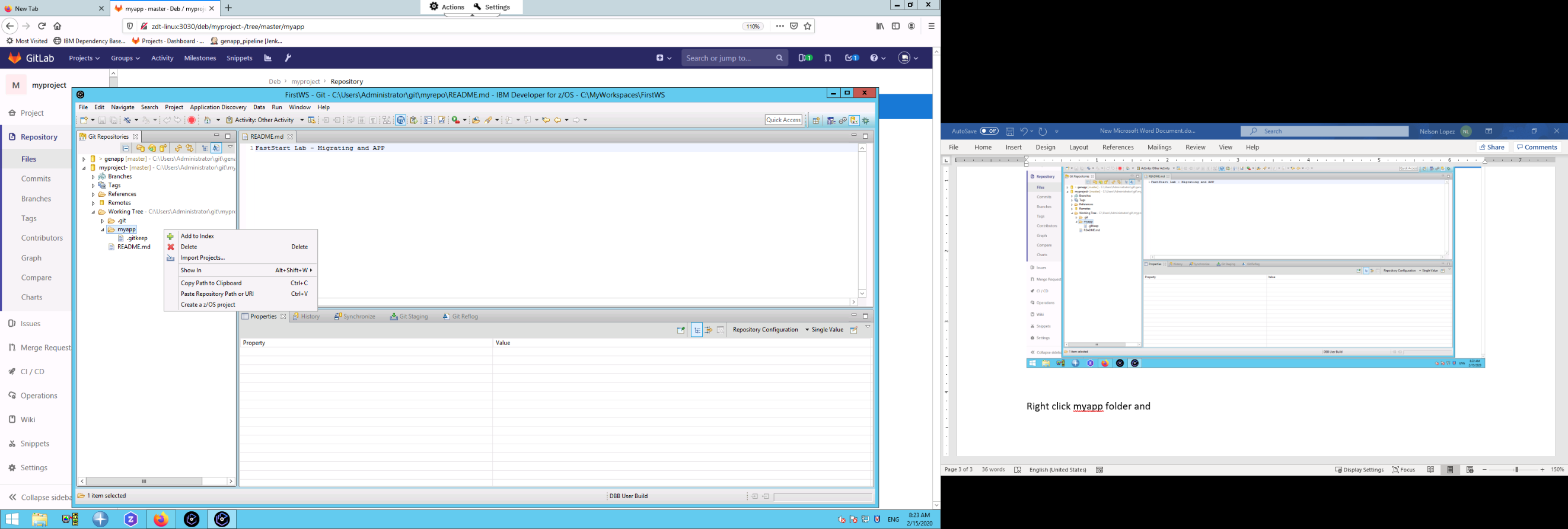
**Step 5 – Clone your new repo in IDz**

From your Repo’s GitLab project page, click on clone and cut/paste the “clone with SSH” url text

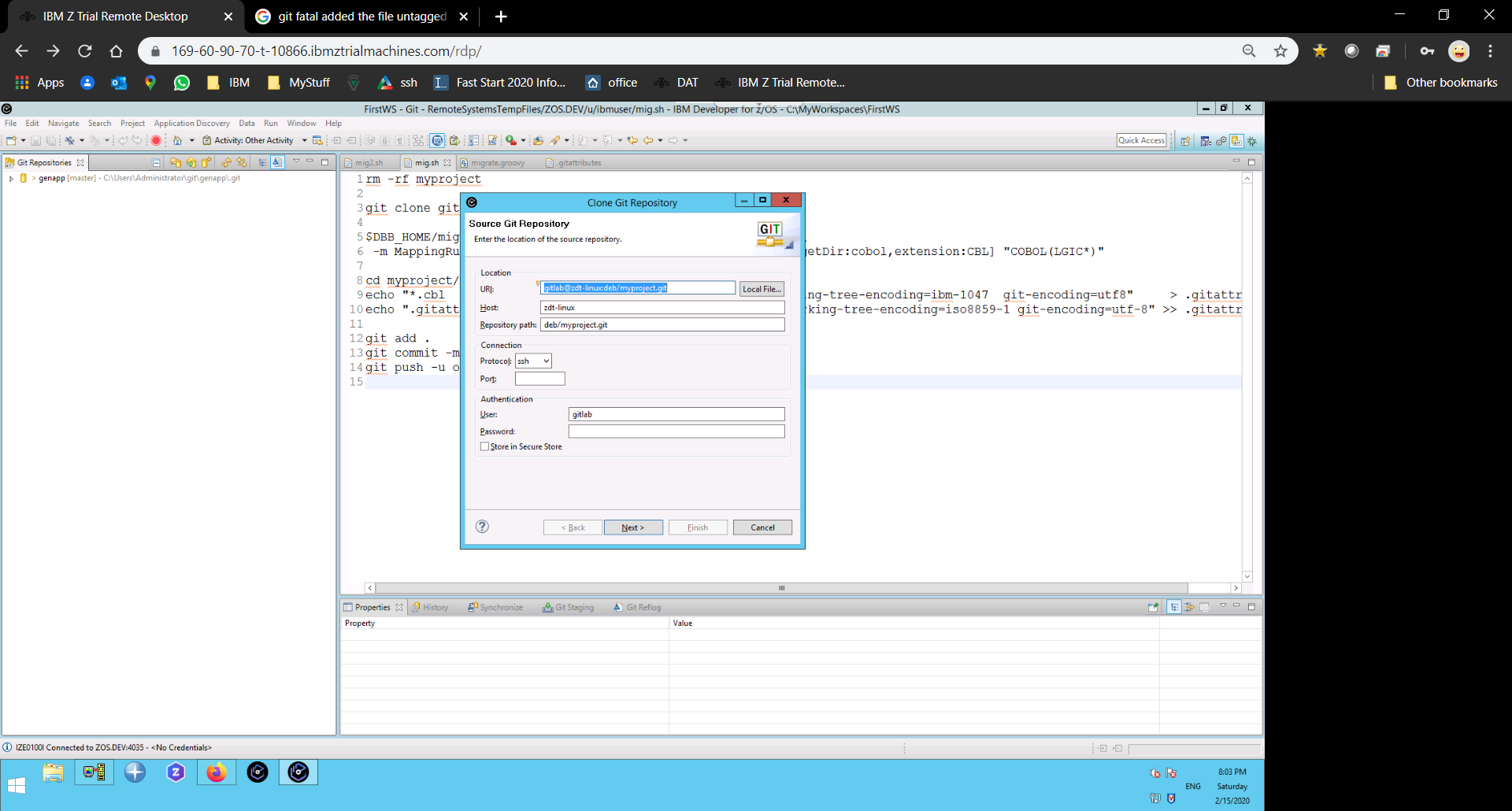
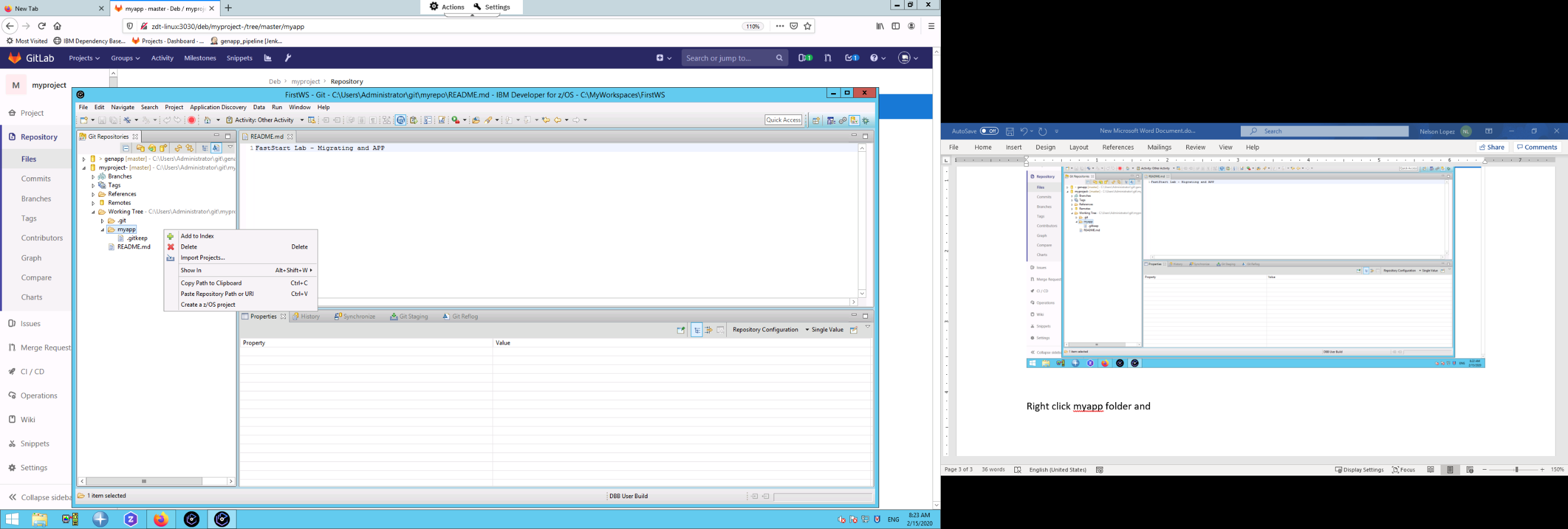
**gitlab@zdt-linux:deb/myproject.git**



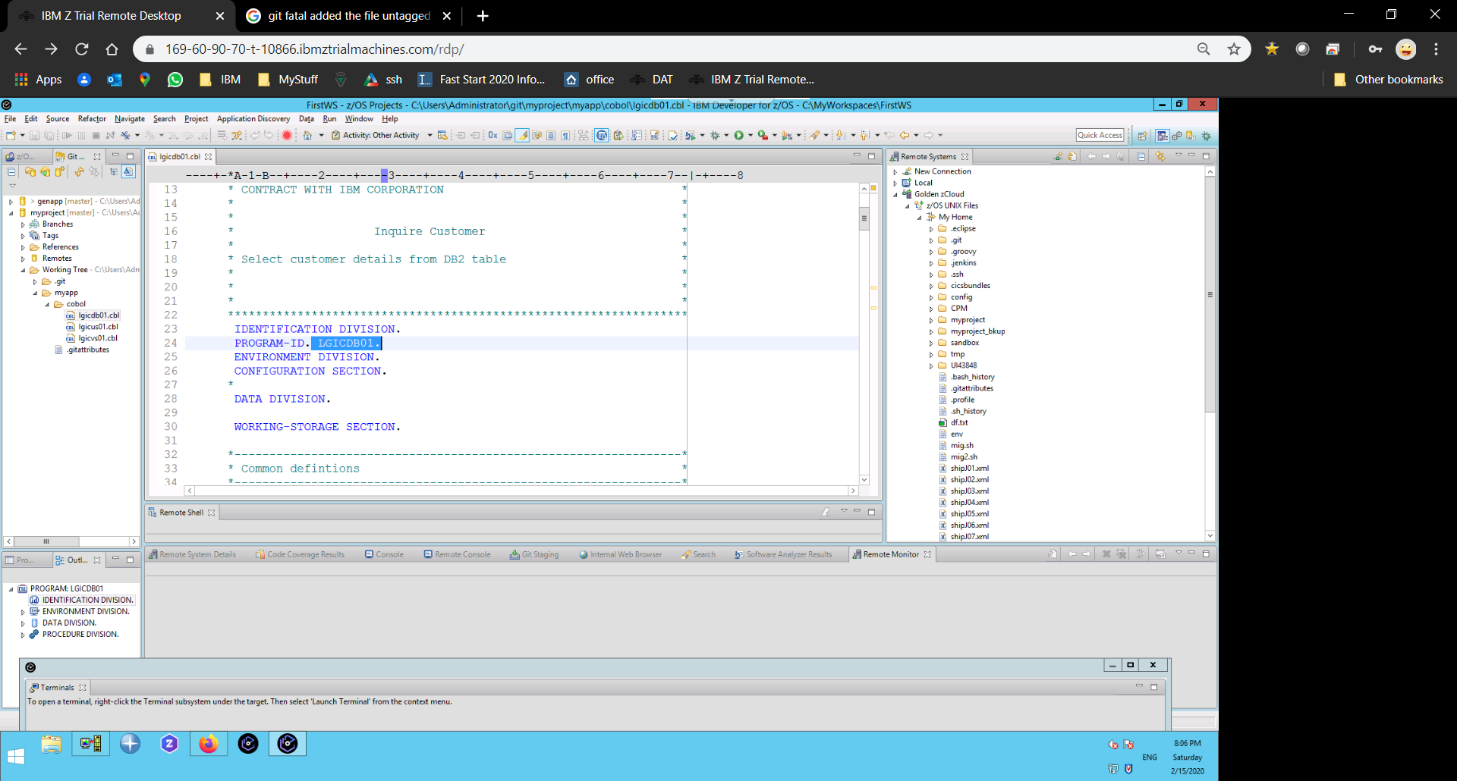
In IDz’s Git perspective , select the “clone icon”



On the next page, your repo’s URL should be pre-filled. If not, paste it and follow the prompts.

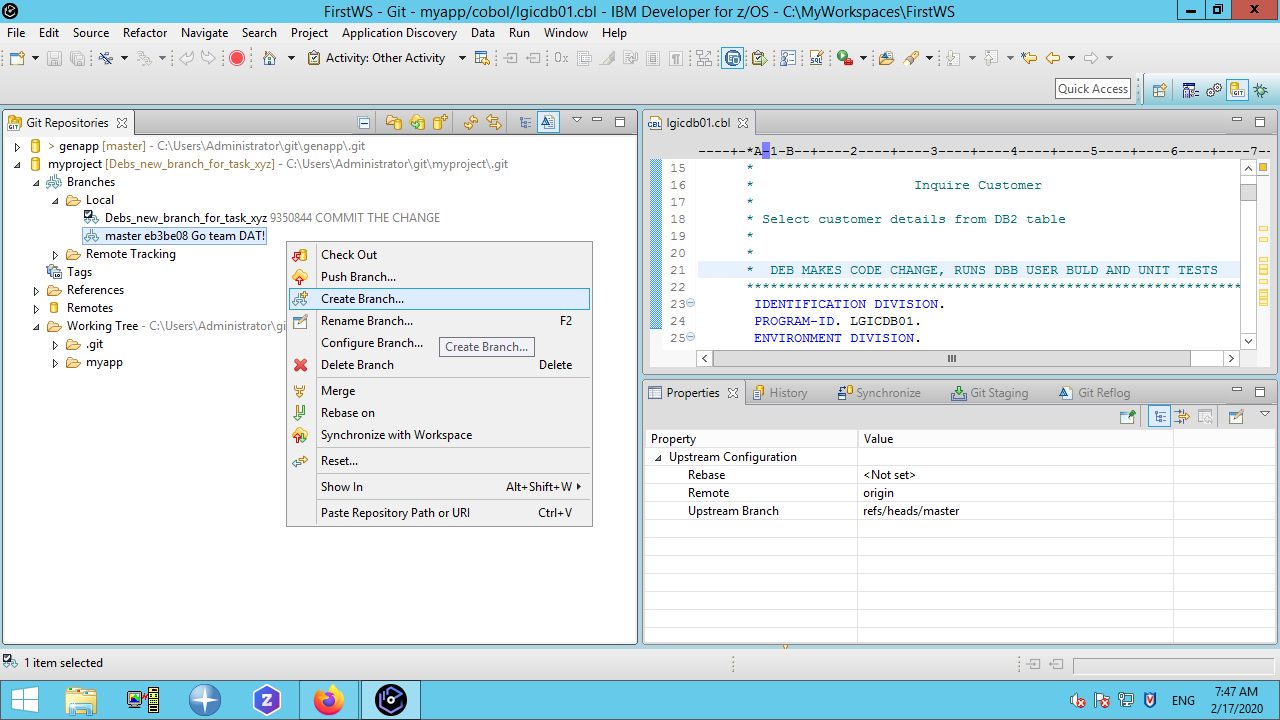


In the Git perspective, navigate to your cloned folder “myapp” and right click to “Create a z/OS project”

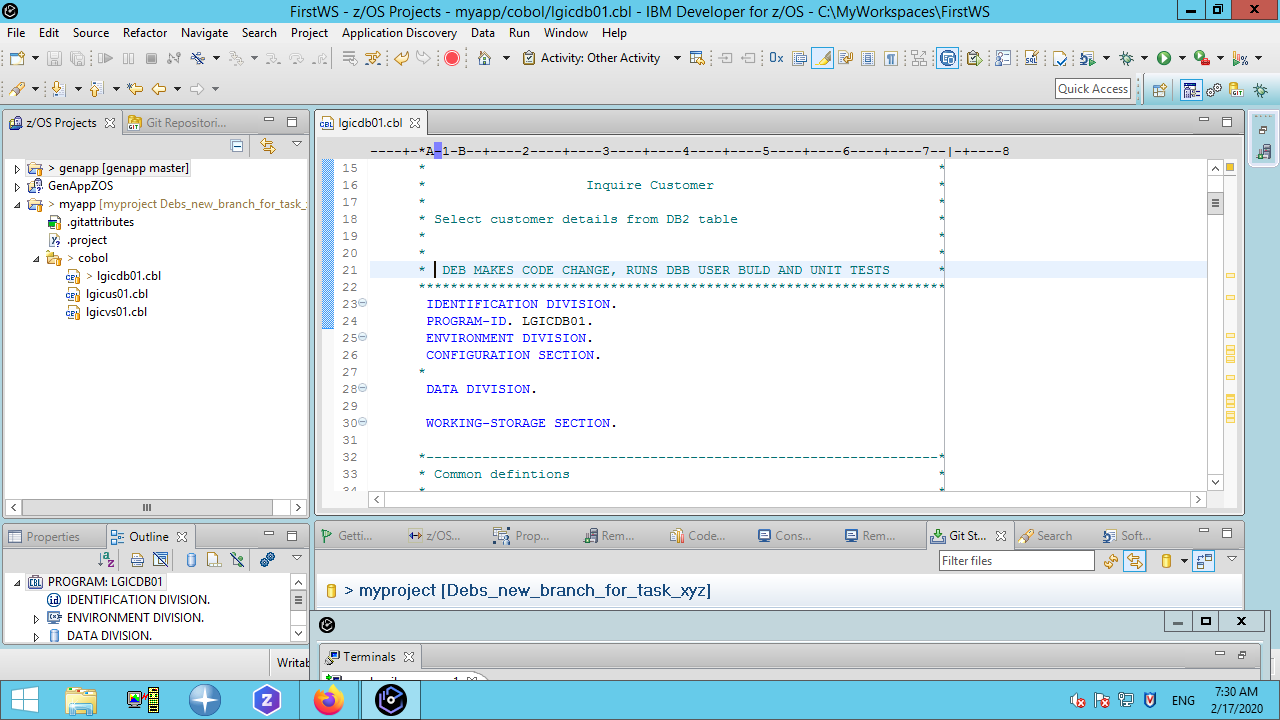
From the z/OS perspective, you can now view and work on your migrated mainframe application!

**Extra credit: Create a branch in IDz and push a change to GitLab**

From your Git prespective, right click “myproject/Branches/Local/master …” and select “Create Branch”



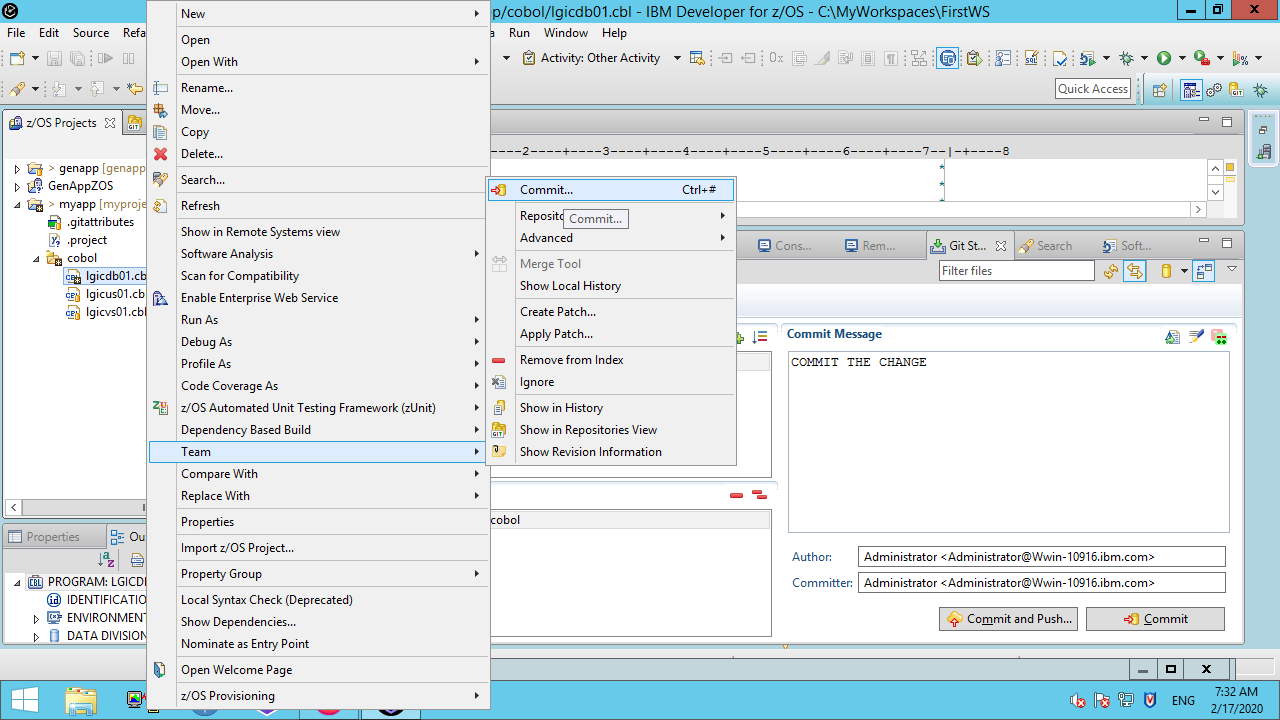
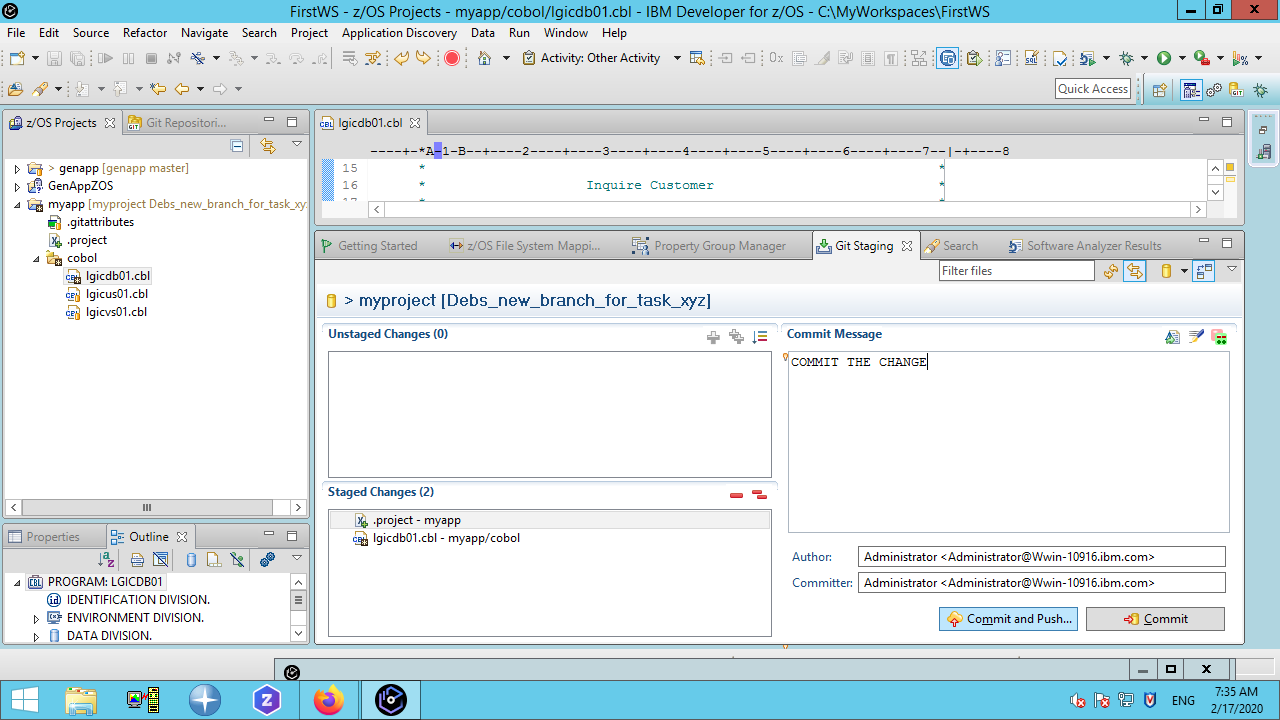
Under the z/OS perspective, make a change to a program and save it.



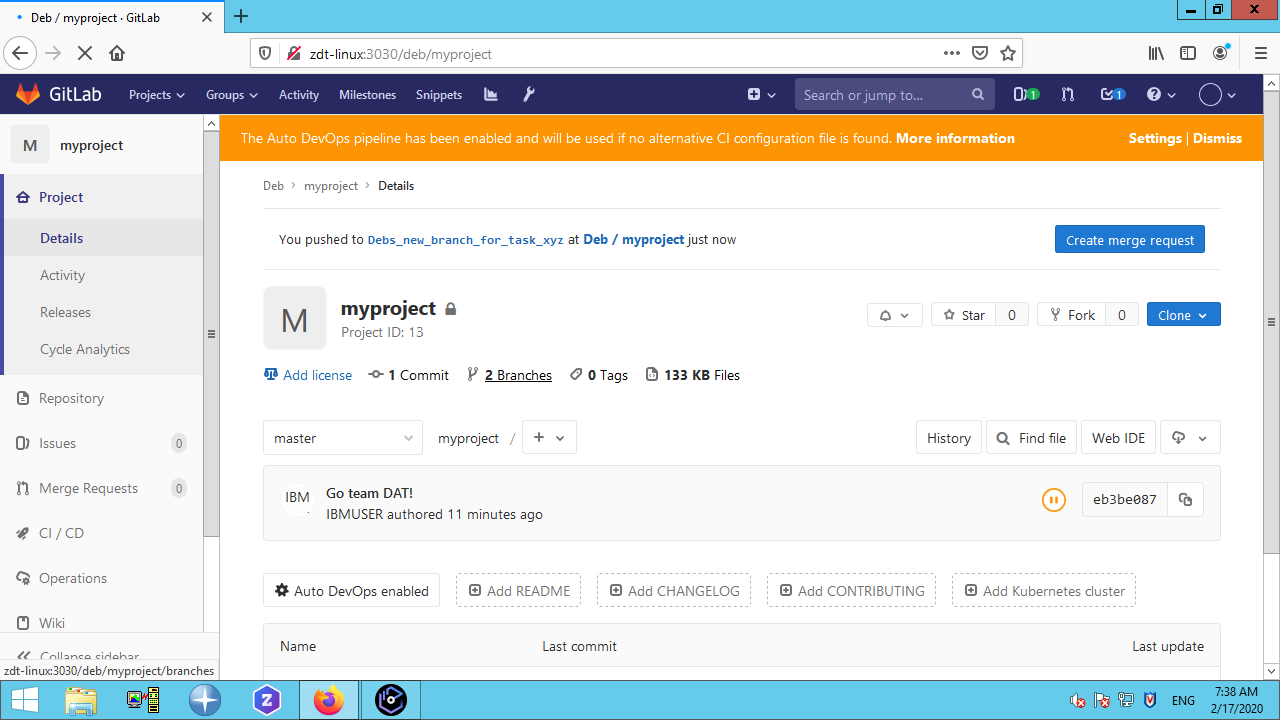
*Note: Before making a change, the developer would have performed an Impact Analysis to determine which program(s) to change. Also, after the change, the developer would run a DBB User Build and Unit test. In this lab, we’re skipping those steps and jumping to pushing the change to Gitlab.*

**Commit your change:**

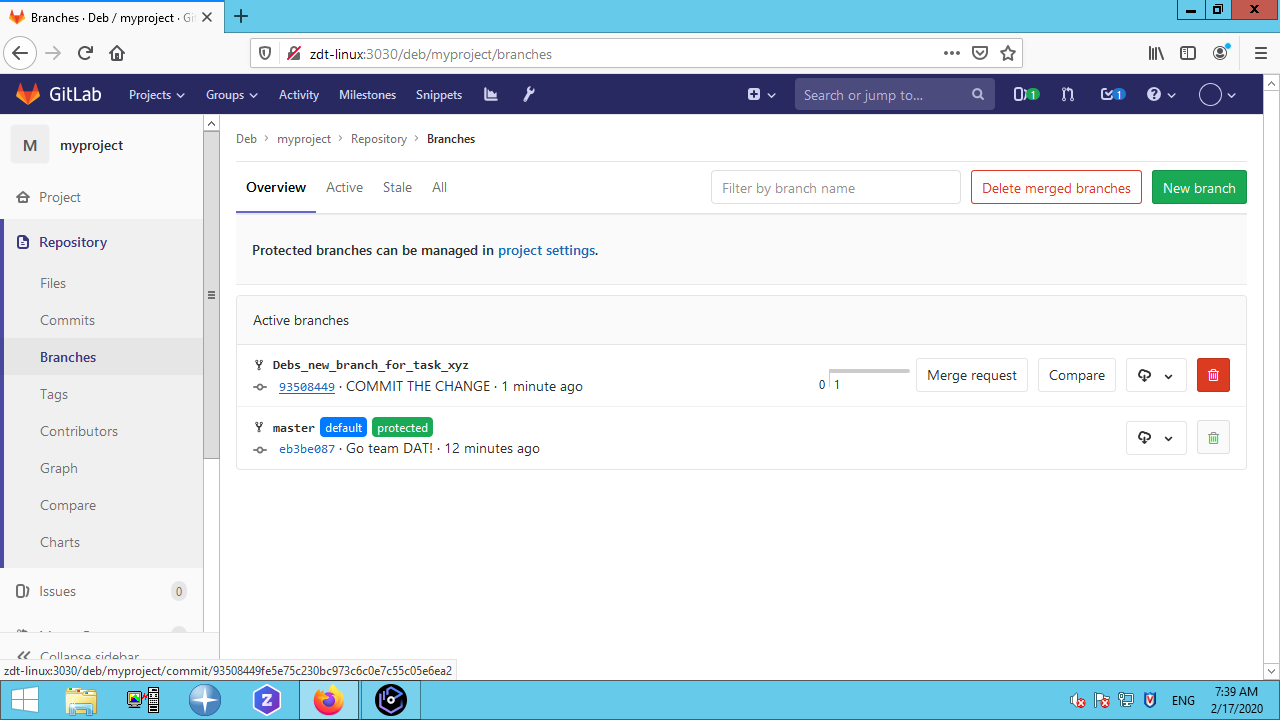
* From the context menu select “Team/Commit”.
* In the “Git Stage” window press the “green ++” icon to stage the change(s)
* Enter a “Commit Message”, press “Commit and Push” and follow the prompts.



From the GitLab “myproject” page, click on “Branches”



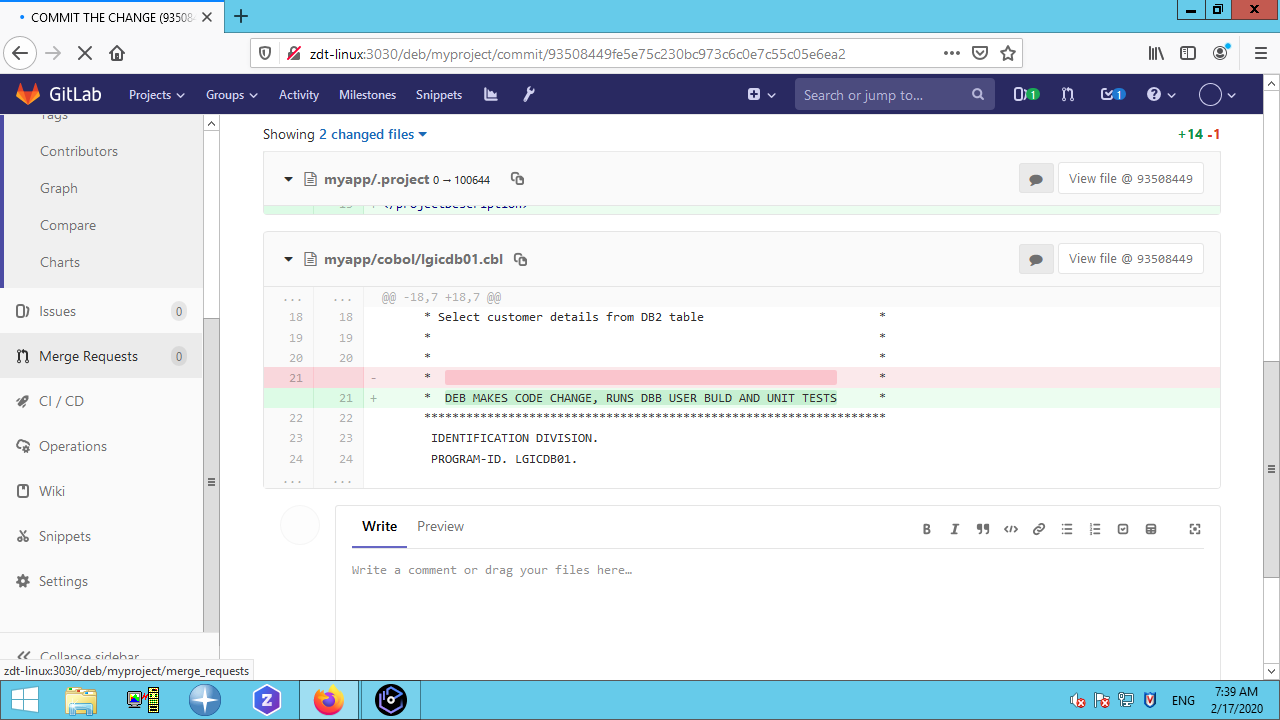
Select the new branch’s “Hash Key” link



Scroll to view the changes related to this branch

Red lines = original

Green = new



At this point a GitLab “Merge Request” can be submitted to trigger a pipeline and run a DBB build, automated unit tests, code coverage and peer review.

*NOTE: While this example is using GitLab, the same principles can be applied to GitHub or BitBucket.*

1. VPN is not required for zTrail access [↑](#footnote-ref-1)