Location for activities:

Local system for access to:

http://jenkins.[redacted].local

http://octopus.[redacted].local

https://bitbucket.org/[redacted]/workspace/overview

Remote Desktop Protocol for access to: redgate.[redacted].local

*You must use the redgate.[redacted].local RDP machine to perform the following functions.*

RedGate Operations:

Start by RDP to redgate.[redacted].local

Open Visual Studio on the RedGate machine

Select Clone Repository, enter your repo location (for this example we will use the actual location for [redacted]DB)

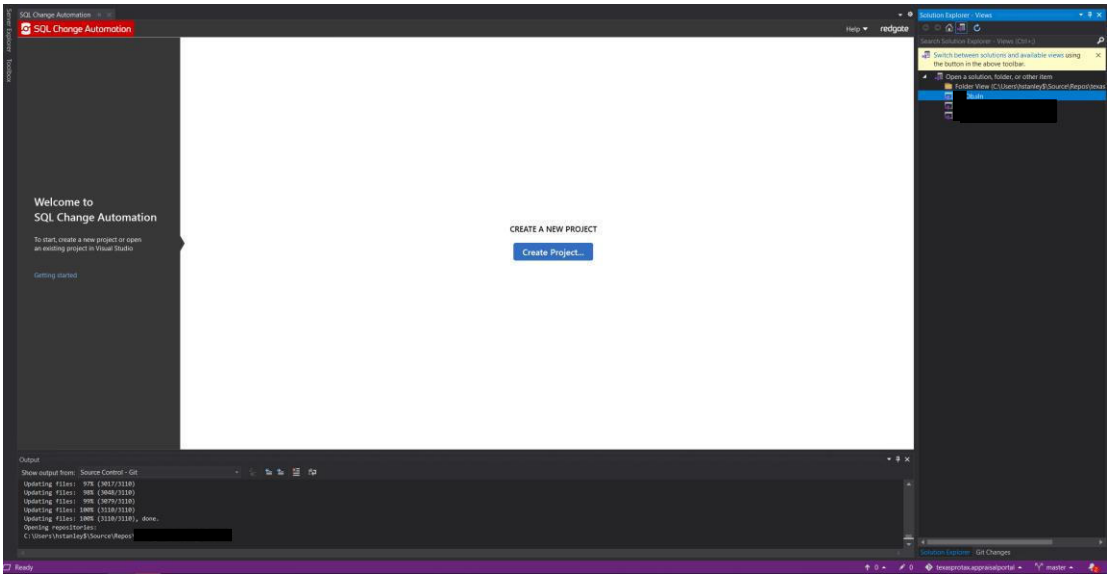
https://[redacted]@bitbucket.org/[redacted]/[redacted].[redacted].git

Click the Clone button.

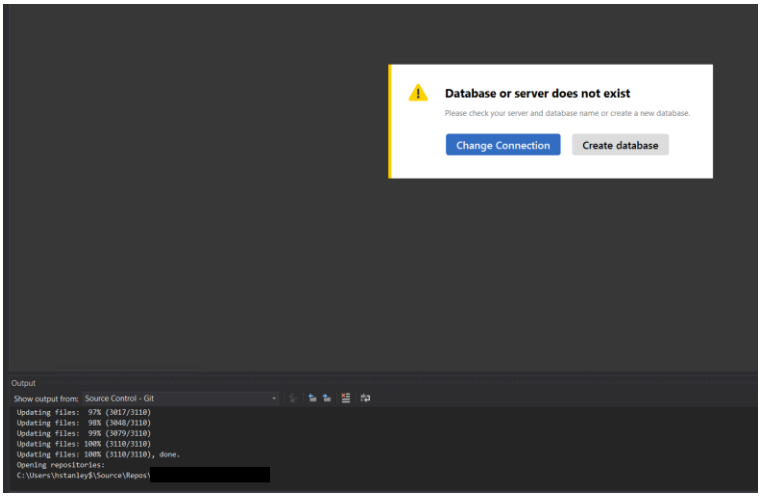
Depending on how your system is setup and which plugin you use for Bitbucket, click the sign in with web browser.

You should now be presented with this screen:

Select the [redacted]Db.sln to open the project.



You may be presented with the following message:




Click Change Connection,

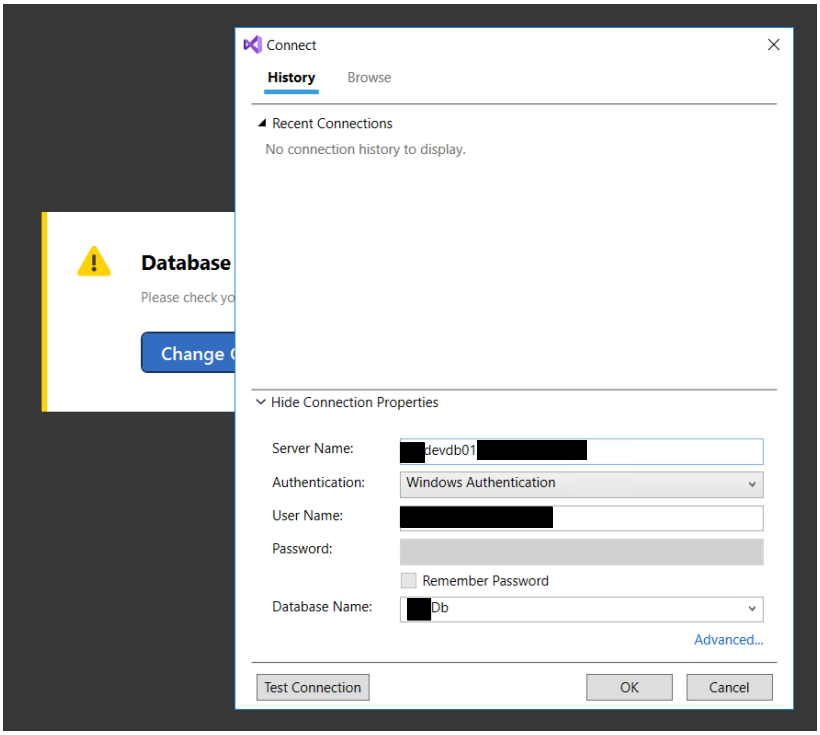
Server Name should be: [redacted]devdb01.[redacted].local

Authentication should be: Windows Authentication

Username should be your username

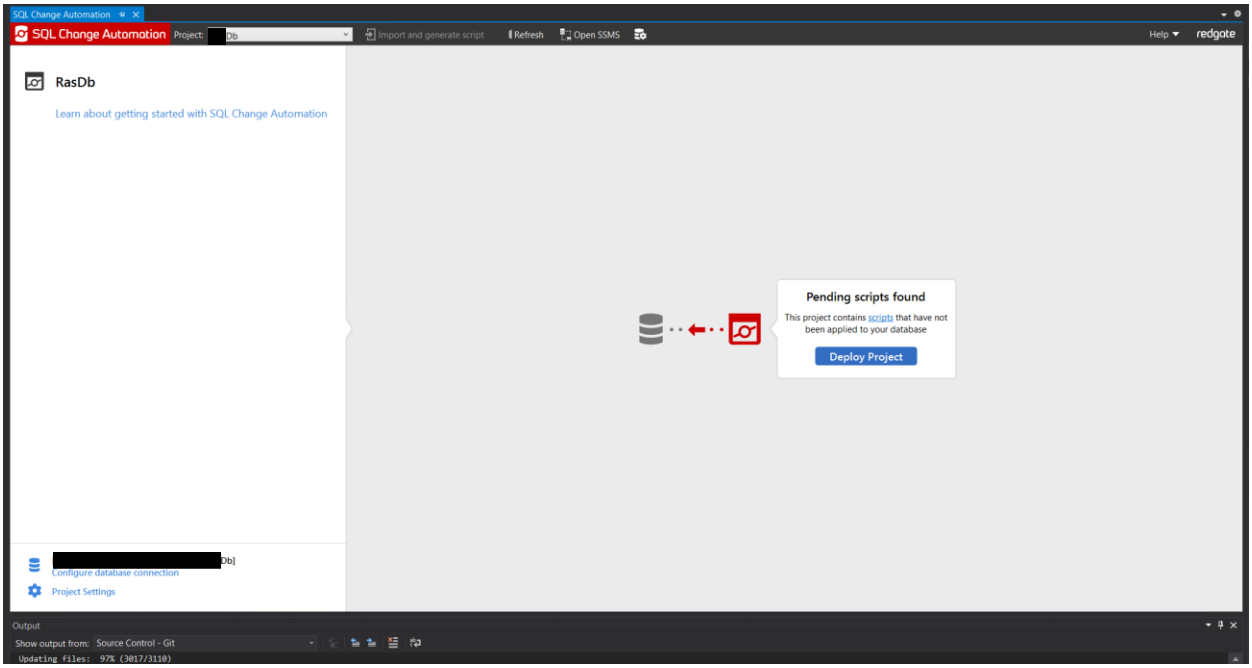
Database should be [redacted]Db

Click Test Connection to verify

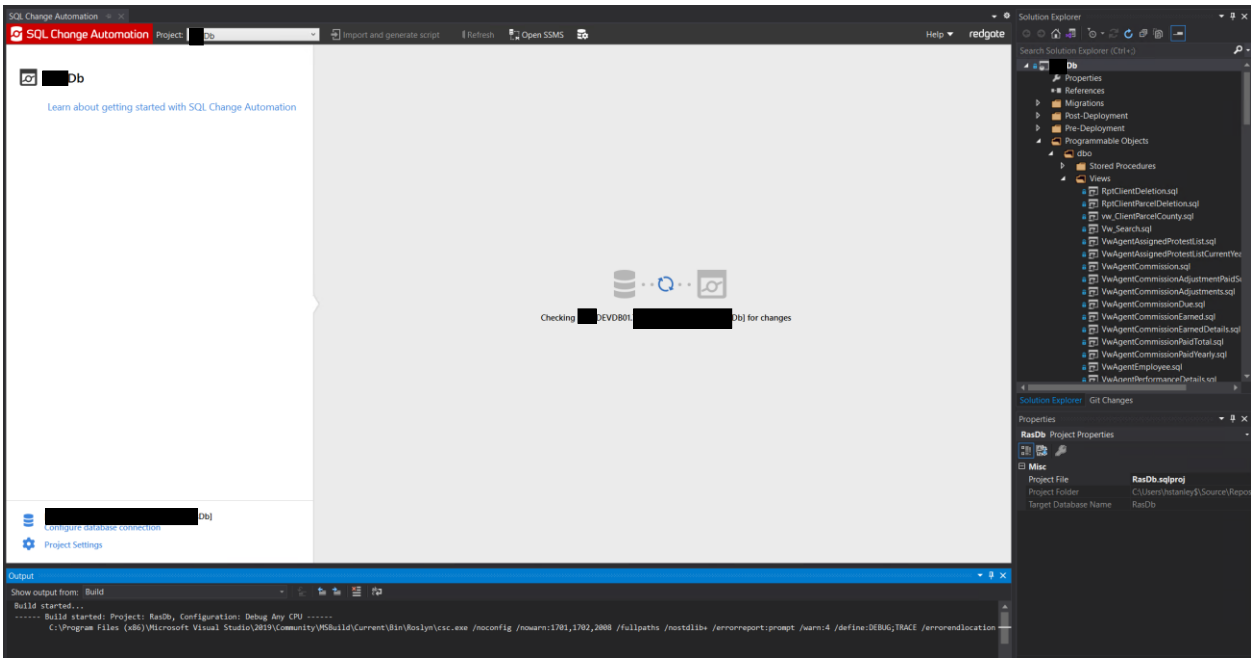



Click OK,

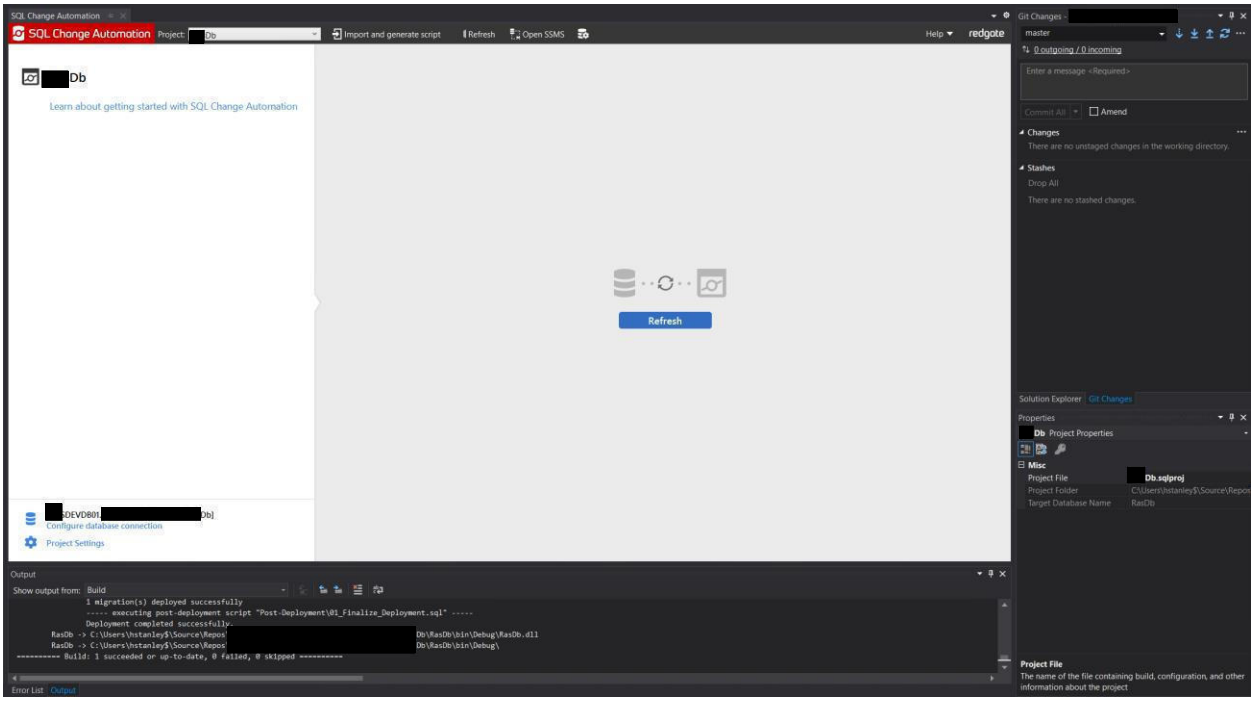
You may be presented with the following screen indicating that there are scripts that have not yet been run on the server but are in the project. They are located in the Error List towards the bottom of the screen:



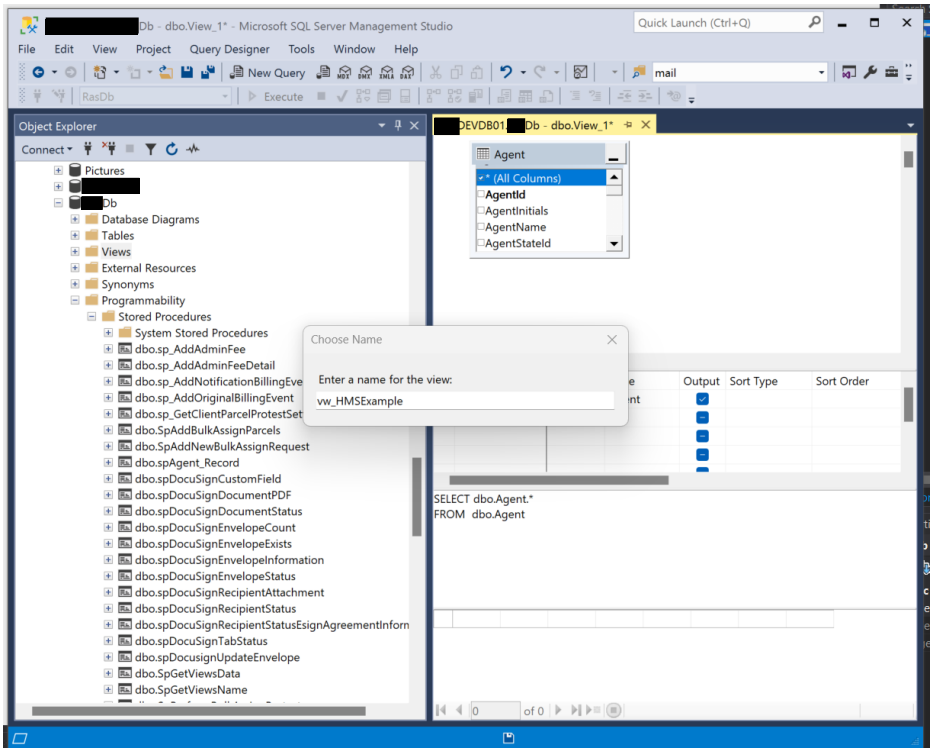

You'll need to resolve the errors by either removing the offending script or by Deploying the Project.



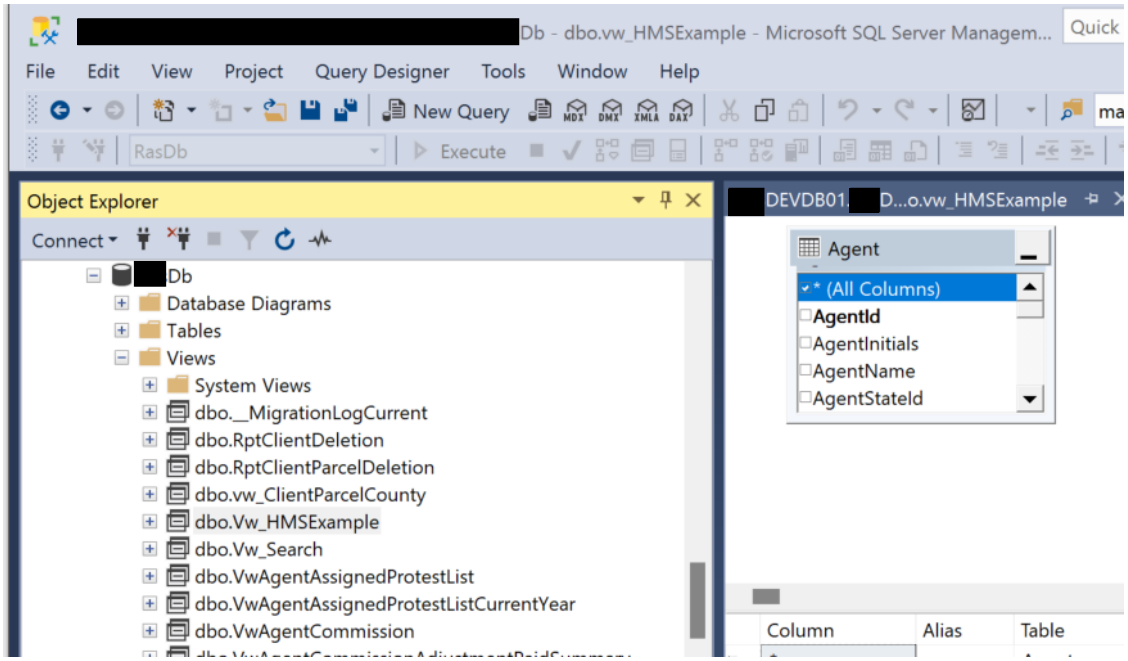
When it's done resolving, you should rest here:



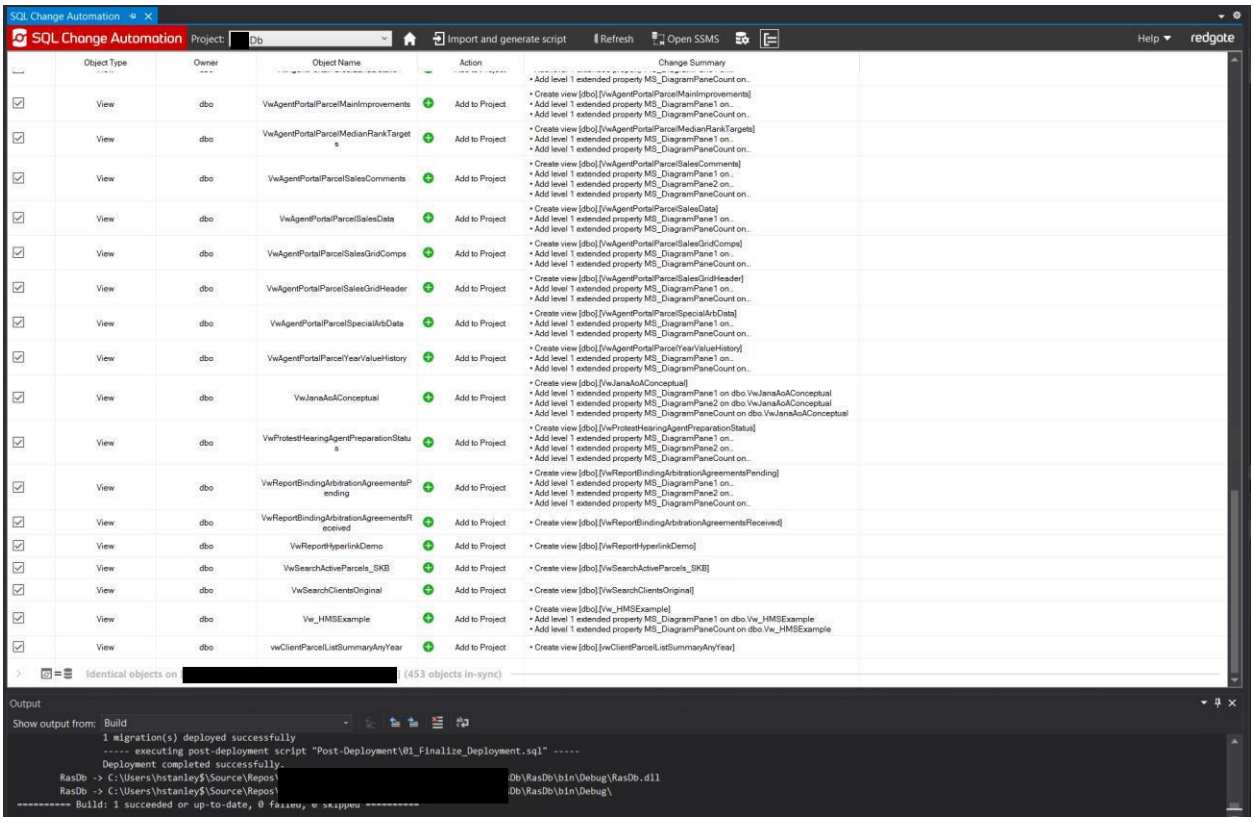
On your machine or on the RedGate Server, open SSMS (Sql Server Management) and create a test table on the ausdevdb01.[redacted].local database server. Stay within the database you are working with in visual studio. This example is in [redacted]DB and I've made a view called Vw_HMSExample – use Pascal case (not like in the picture!).




You can see it in the list of Views now after a refresh of the Views folder. (Notice that I fixed the error in type case)

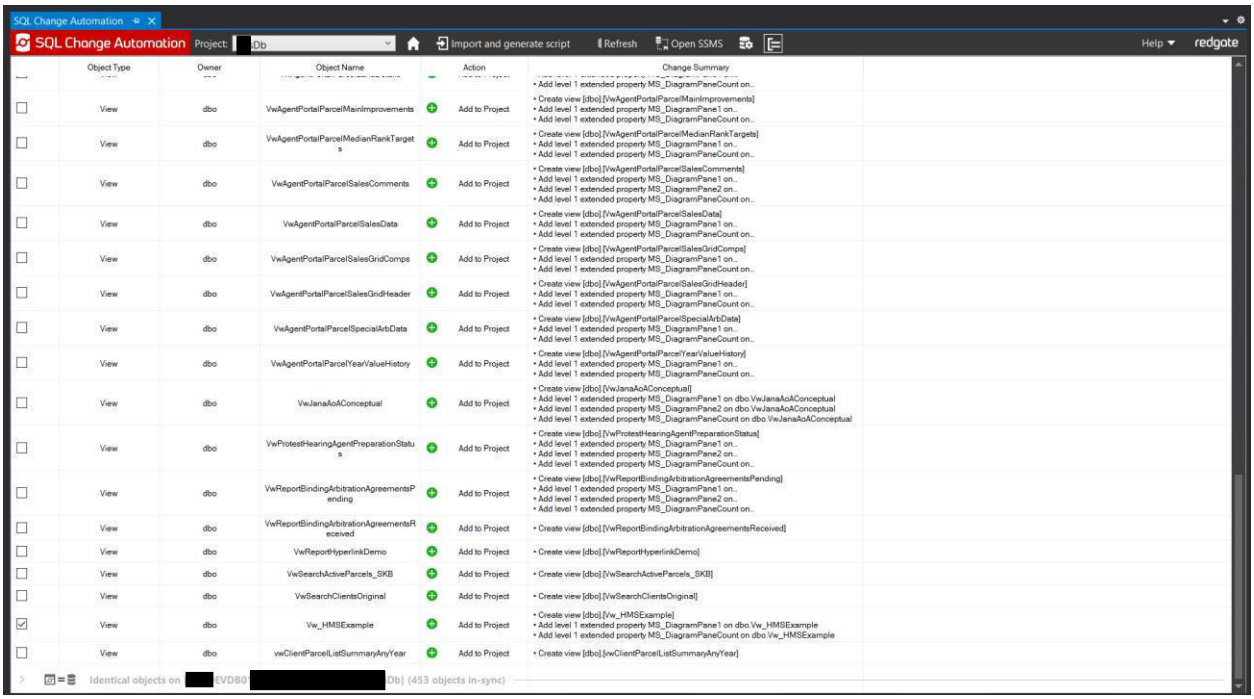


Go back to the RedGate server and click that Refresh Button displayed earlier. This will grab any changes between the code from Bitbucket and the [redacted]Db on ausdevdb01.[redacted].local (also known as the SourceDb and the DatabaseServer):

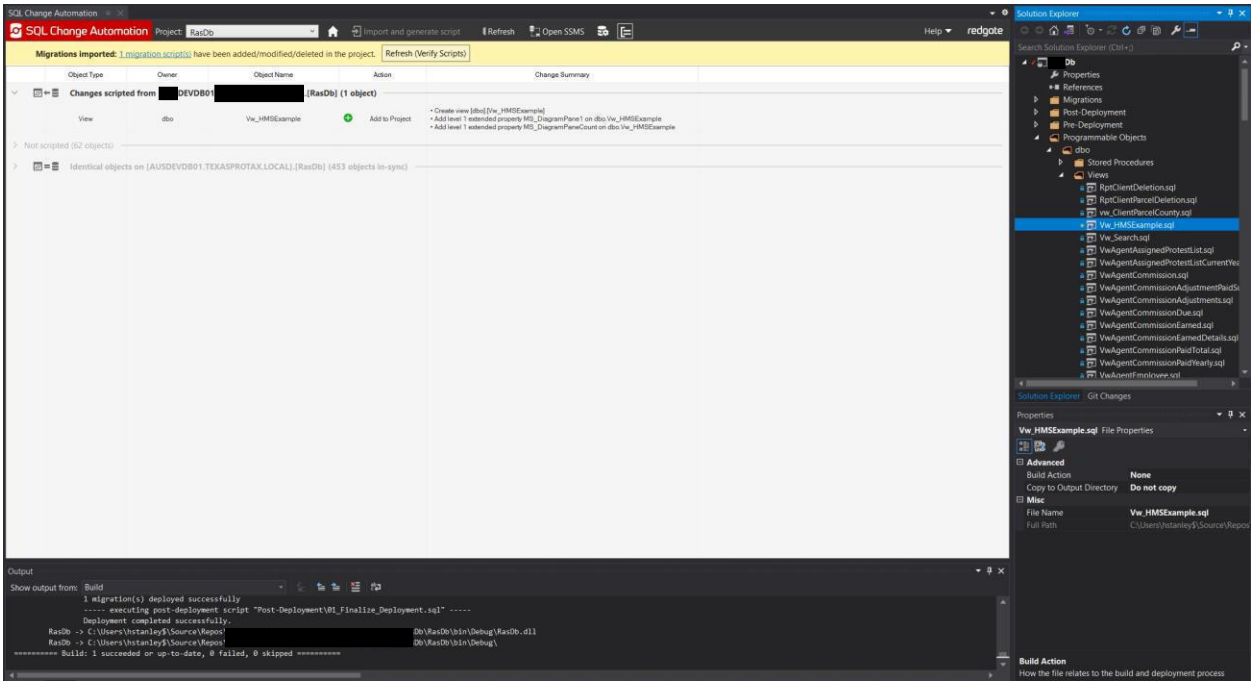


Select the item(s) you want deployed out of the Development Database (SourceDB) by selecting the checkbox (You can select them all by clicking on the group header then selecting a checkbox to any object. Deselect all items by clicking on a whitespace next to an object)

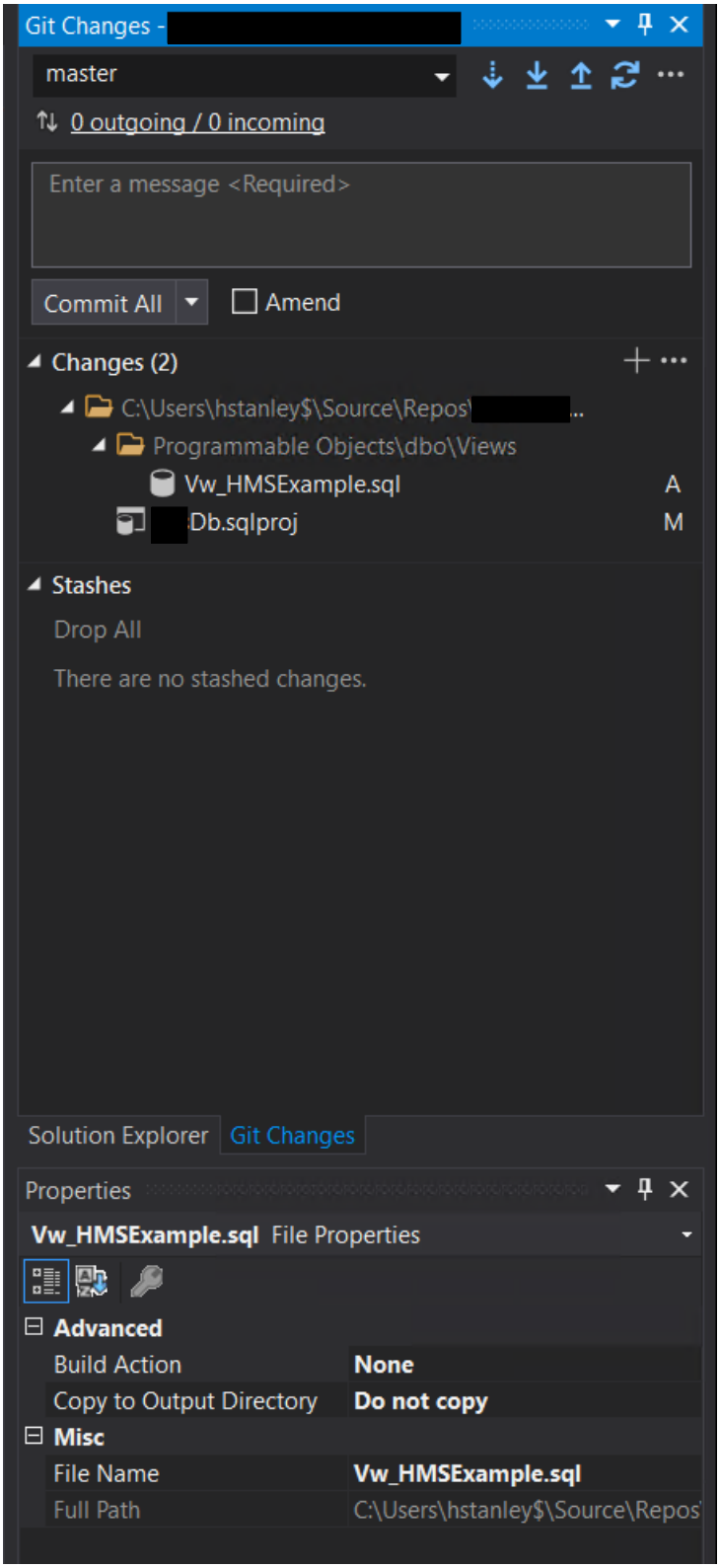
Here I've selected only the change created in this demonstration, then click the Import and generate script at the top of this screen.:



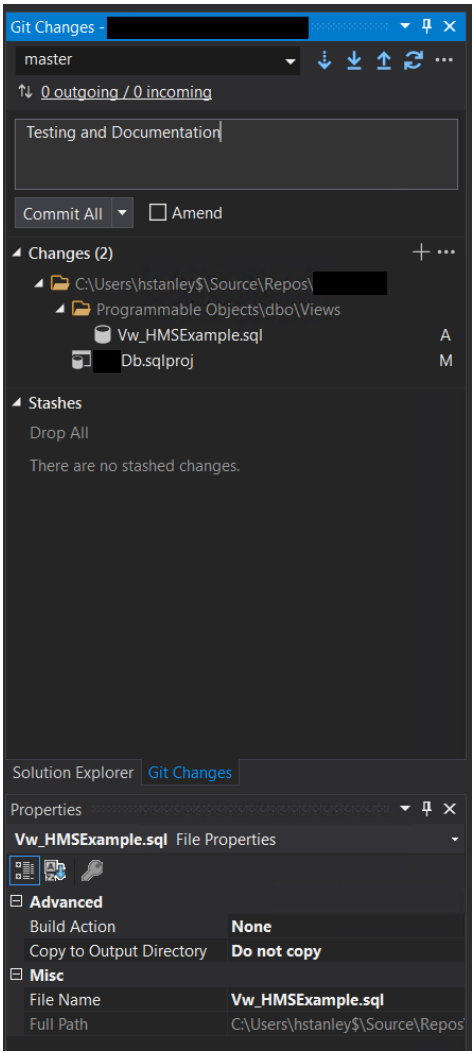

Now you can see, as indicated by the green plus symbol, that we have a new SQL Script Vw_HMSExample.sql under the Views Folder.



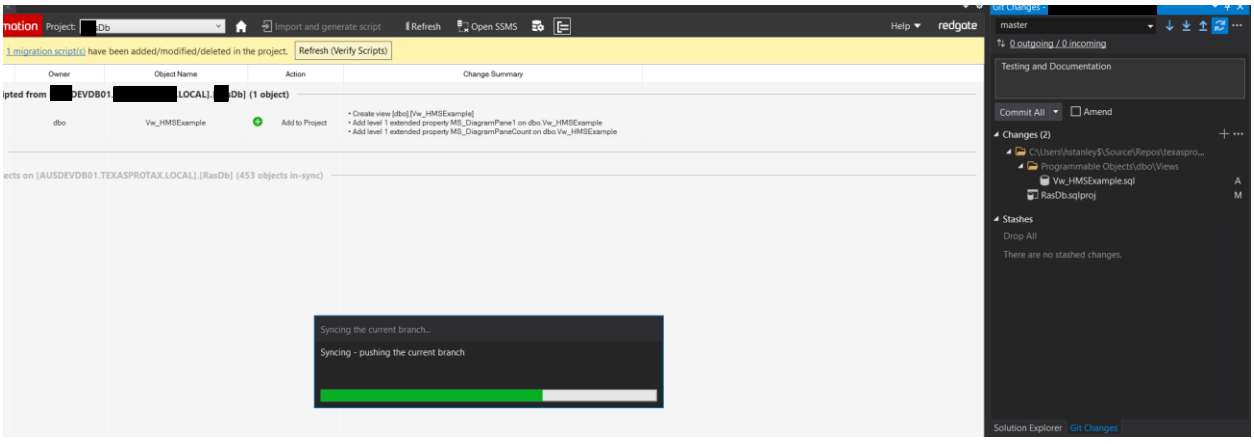
Let's click Git Changes,



Enter your Commit message, click Commit All, then select the Sync button (Pull, then Push)

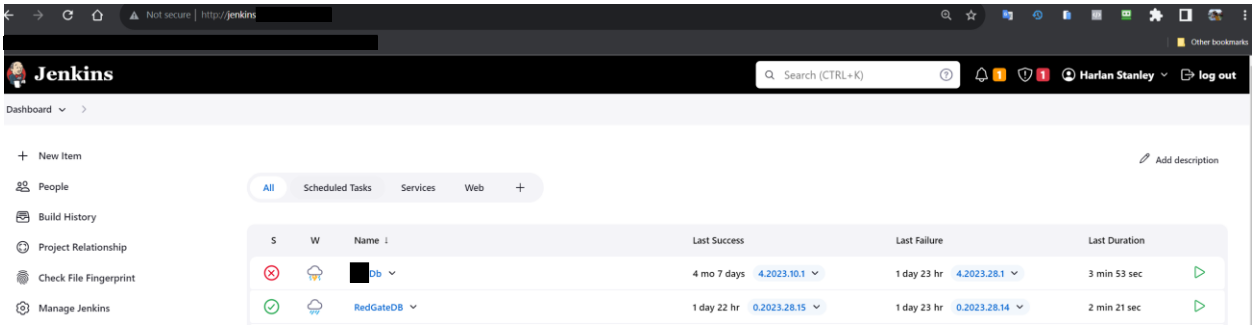



This will send the new script up to Bitbucket and we're done with the RedGate server, be sure to logout.



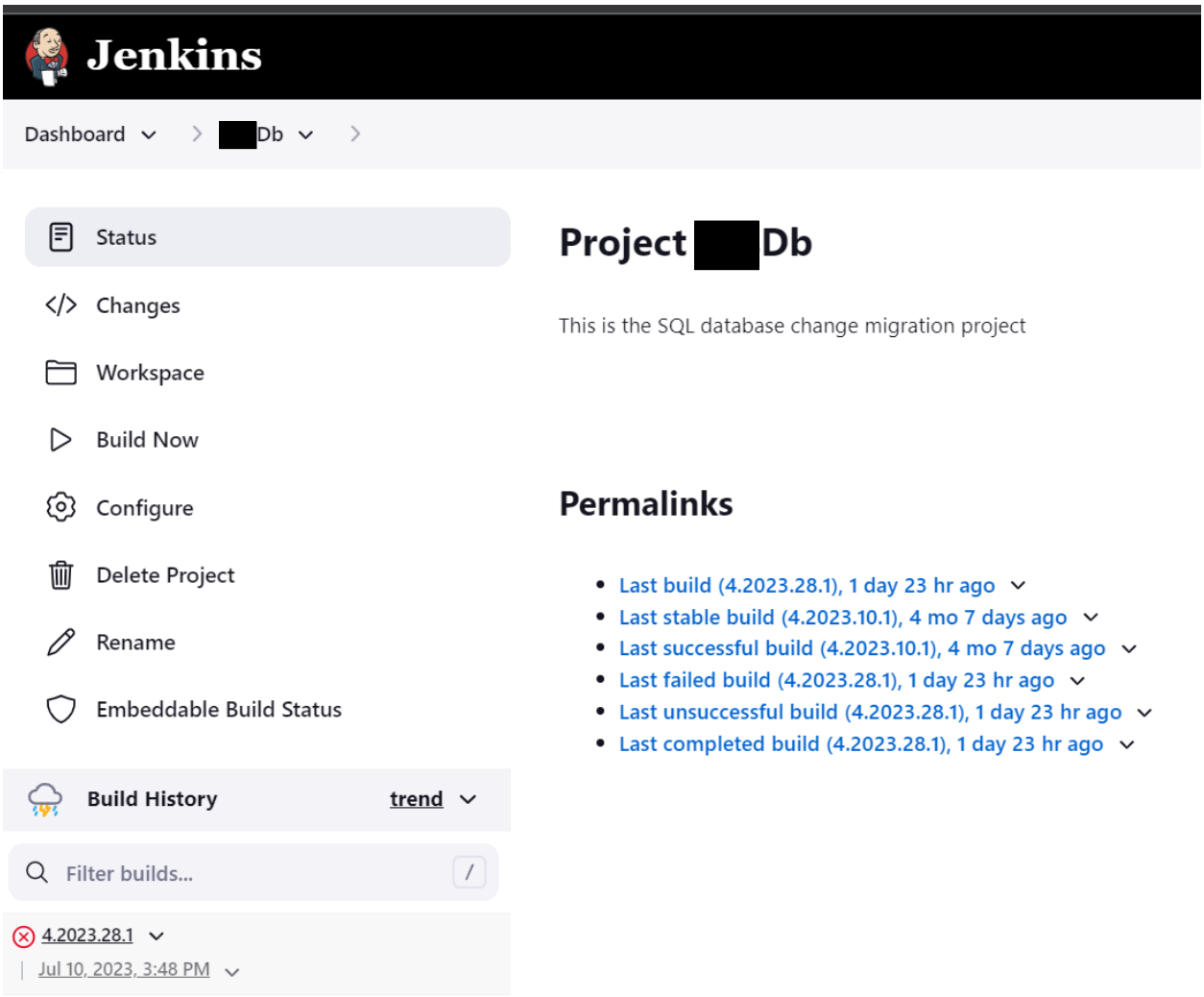

On your local machine let's switch over to Jenkins at [http://jenkins.[redacted].local](http://jenkins.[redacted].local)

Jenkins and Octopus Operation:

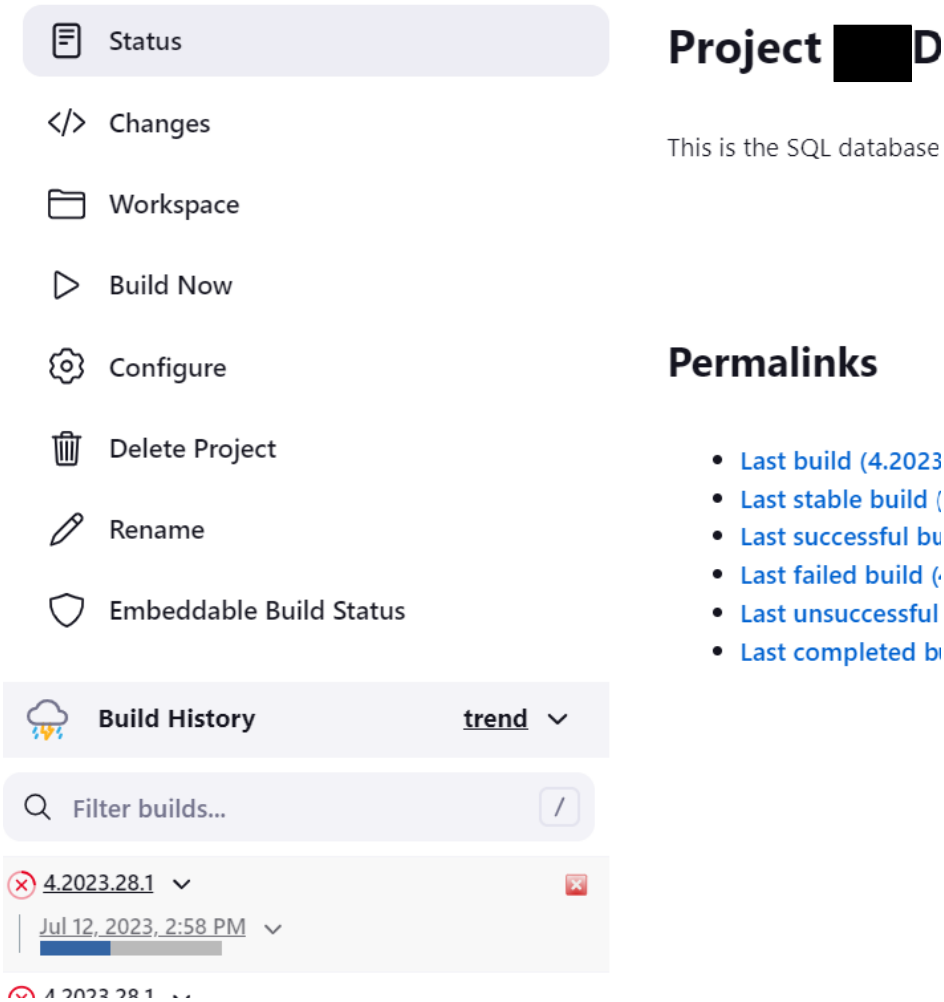


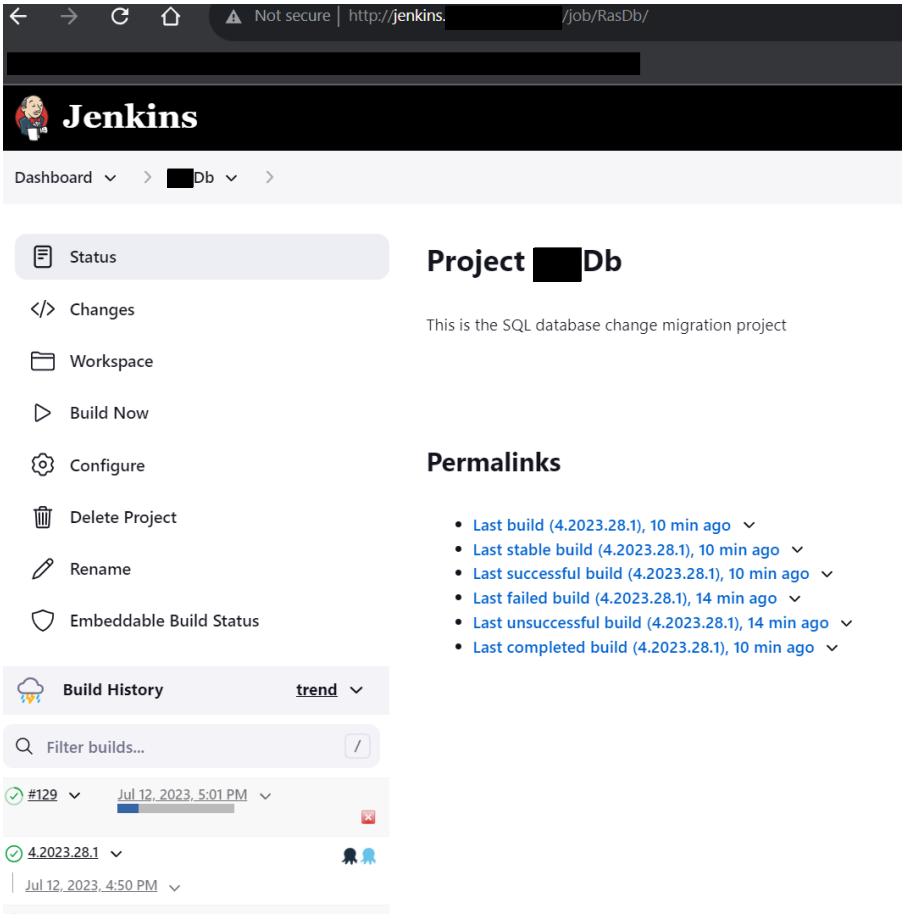

Select [redacted]Db from the menu

Click “Build Now”



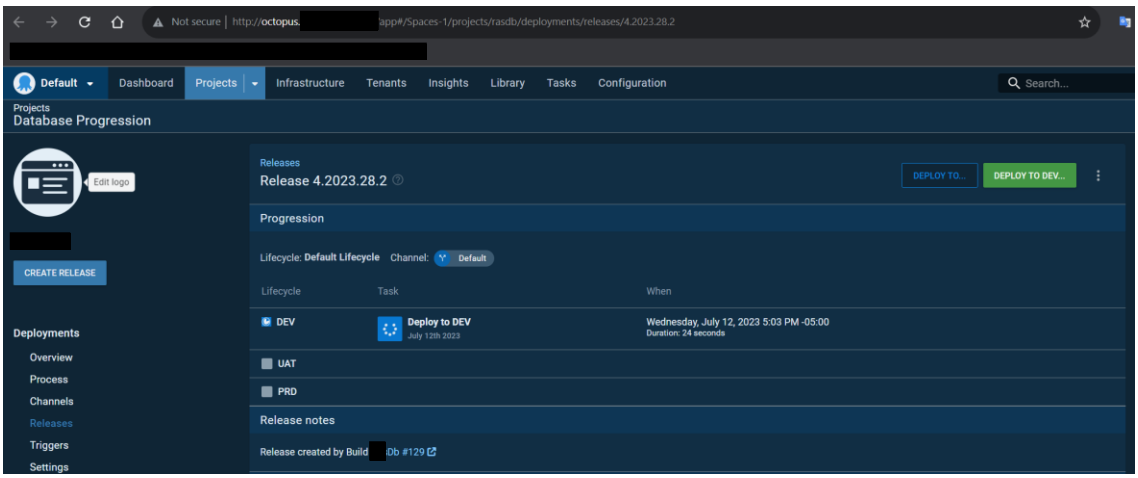
It takes a moment, but you should see your build start, and once the version name updates, If you click on the date link, you will be able to open the console stream where you can see the status of everything happening.






A green Check mark indicates that the build was successful. You can click on the Octopus icon, navy takes you to the Release screen and the light blue drops you in to the actual deployment page. We are done with Jenkins, let's move on to Octopus.

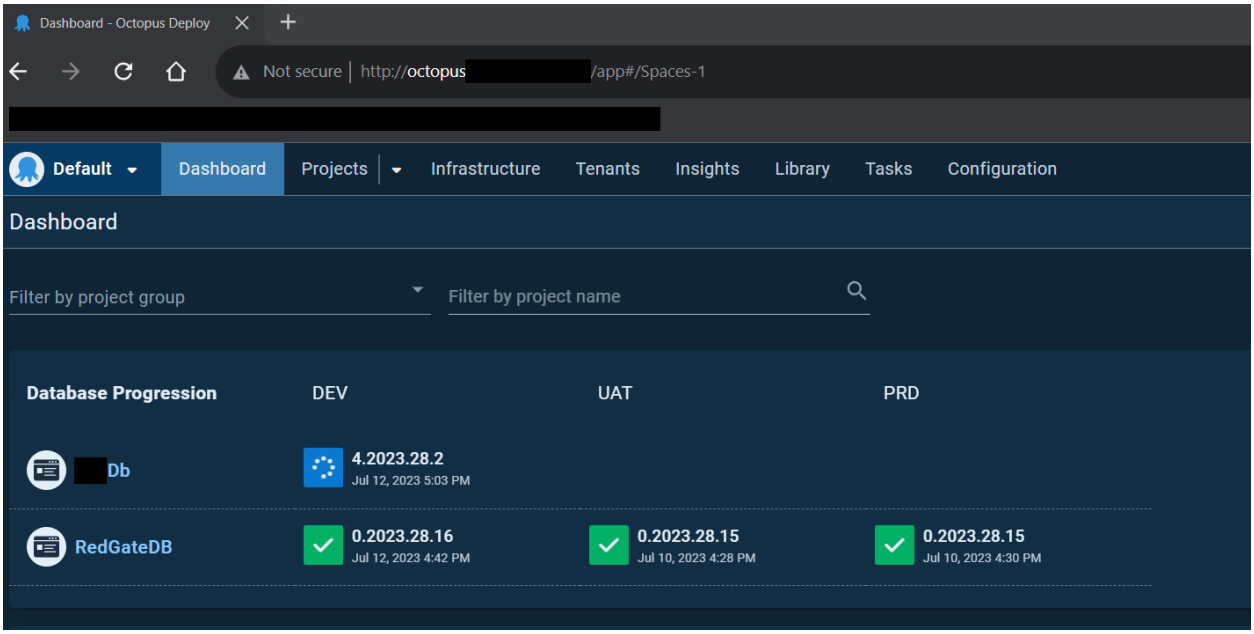
Alternatively, you can go to [http://octopus. .local](http://octopus. .local)



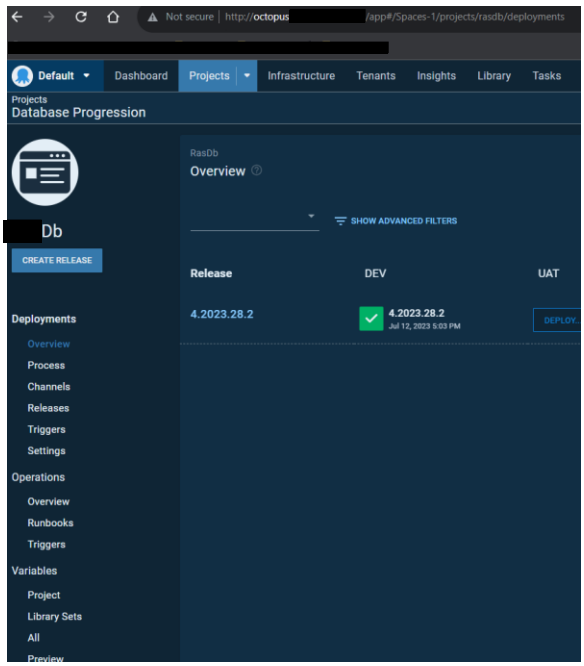

That blue box indicates that a deployment is happening. Green indicates that it was a success!

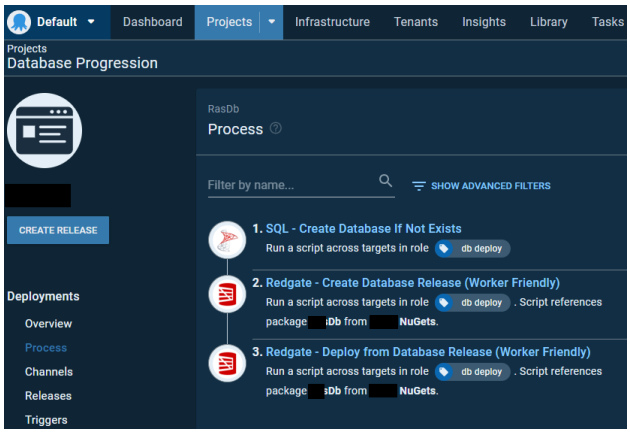
Octopus Configuration Overview:

Login to http://octopus.[redacted].local




Select the [redacted]Db Project, then on the next page, Click Processes.



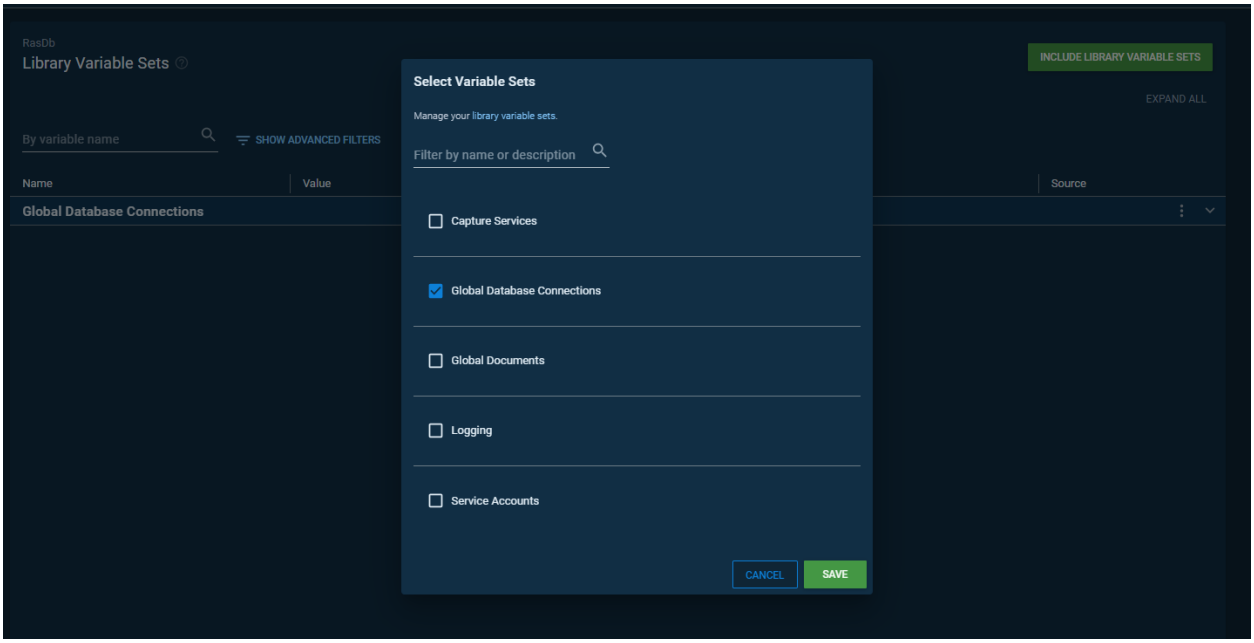




[redacted]Db Database Progression takes only three steps in Octopus. Remember, of course to select save after each operation. – Or cancel if you're just looking around.

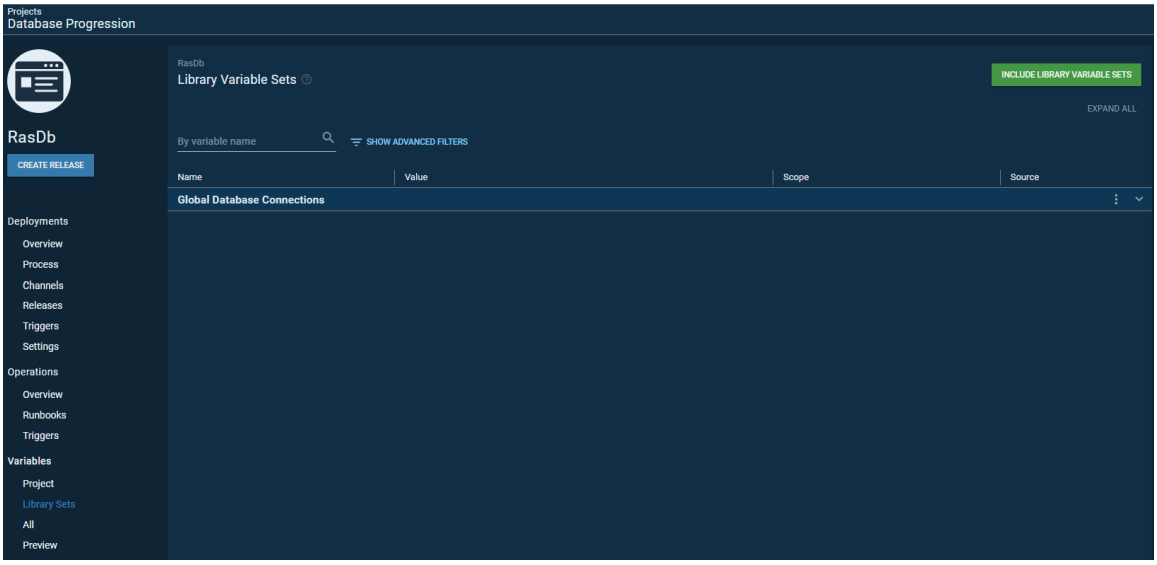
1. SQL – Create Database If Not Exists
2. Redgate – Create Database Release (Worker Friendly)
3. Redgate – Deploy from Database Release (Worker Friendly)

Octopus Variables must be set beforehand, #SourceDb, and #TargetDb are set in the Project Variables location.


#DatabaseServer is set in the Library Set "Global Database Connections" and must be subscribed to:




Click Save.



Each of these tasks have variable sets and configuration as described below.

1. SQL – Create Database If Not Exists

Be sure to select the Target Role to “db deploy”SQL Server is set to the variable #{DataBaseServer} and Database to create is set to #{TargetDb}

2. Redgate – Create Database Release (Worker Friendly)

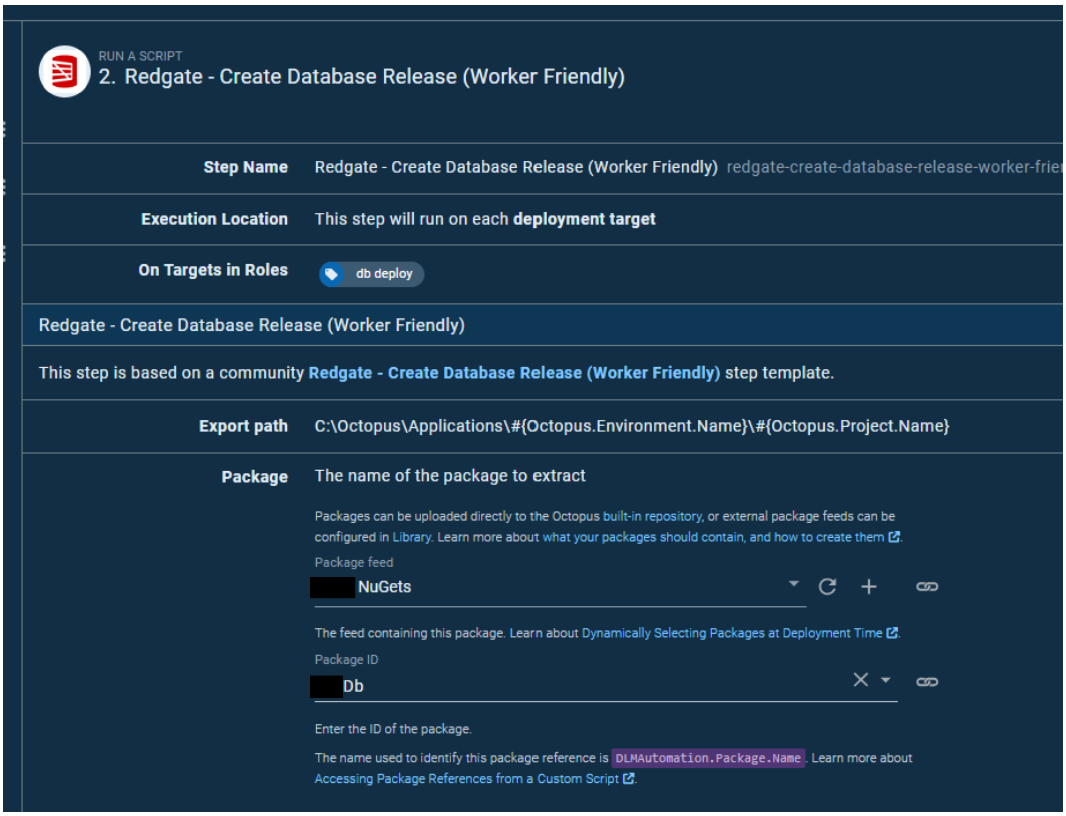
Again, set Target Role to “db deploy”

Export path is set to C:\Octopus\Applications\#{Octopus.Environment.Name}\#{Octopus.Project.Name}

(This path is located on the Redgate Server, redgate.[redacted].local)

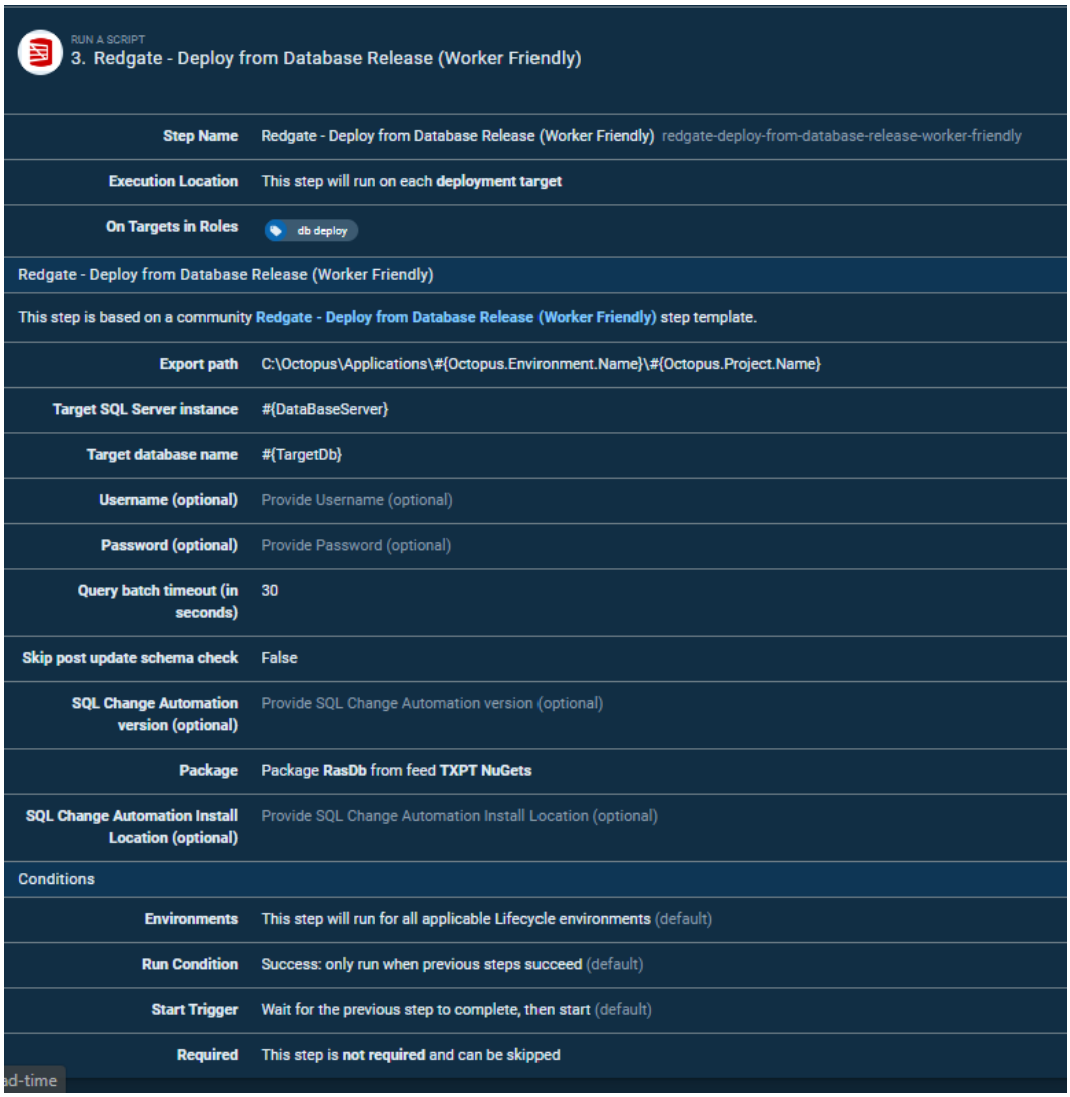
Target SQL Server is set to the variable #{DataBaseServer} and Database to create is set to #{TargetDb}

Pay attention to the Package selection you will need to select the dropdown to get [redacted] NuGets and then pick the [redacted]Db from the Package ID dropdown:



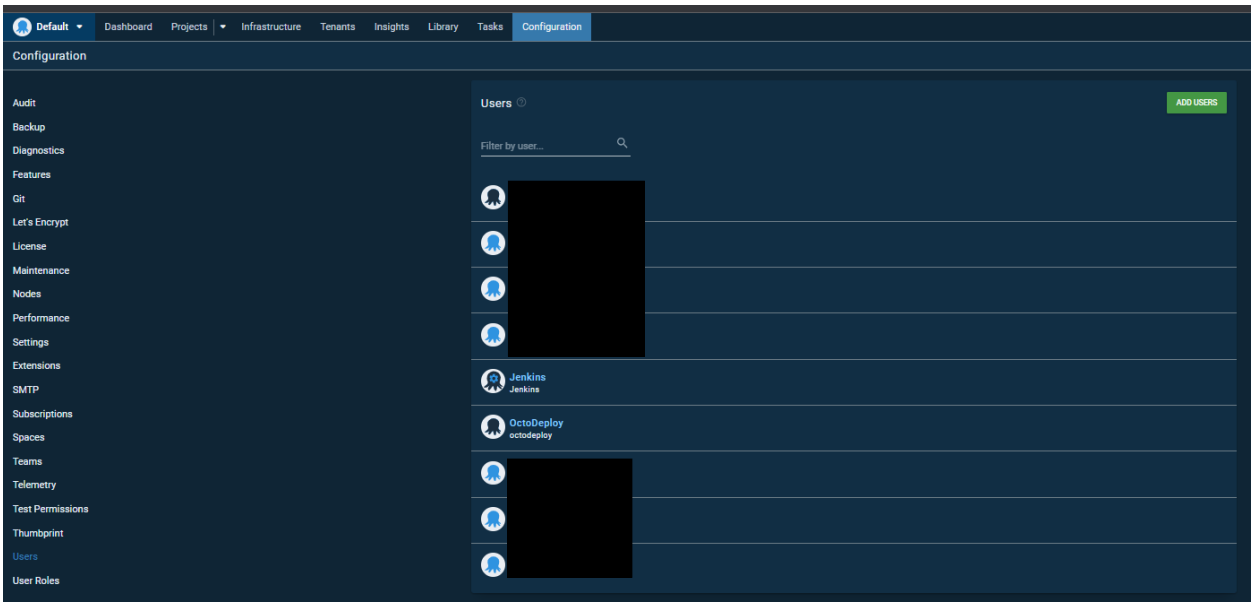
3. Redgate – Deploy from Database Release (Worker Friendly)

This is setup much the same way as Step 2, set the Roles, Export Path, Target SQL and Target Database, and finally the Package selection.

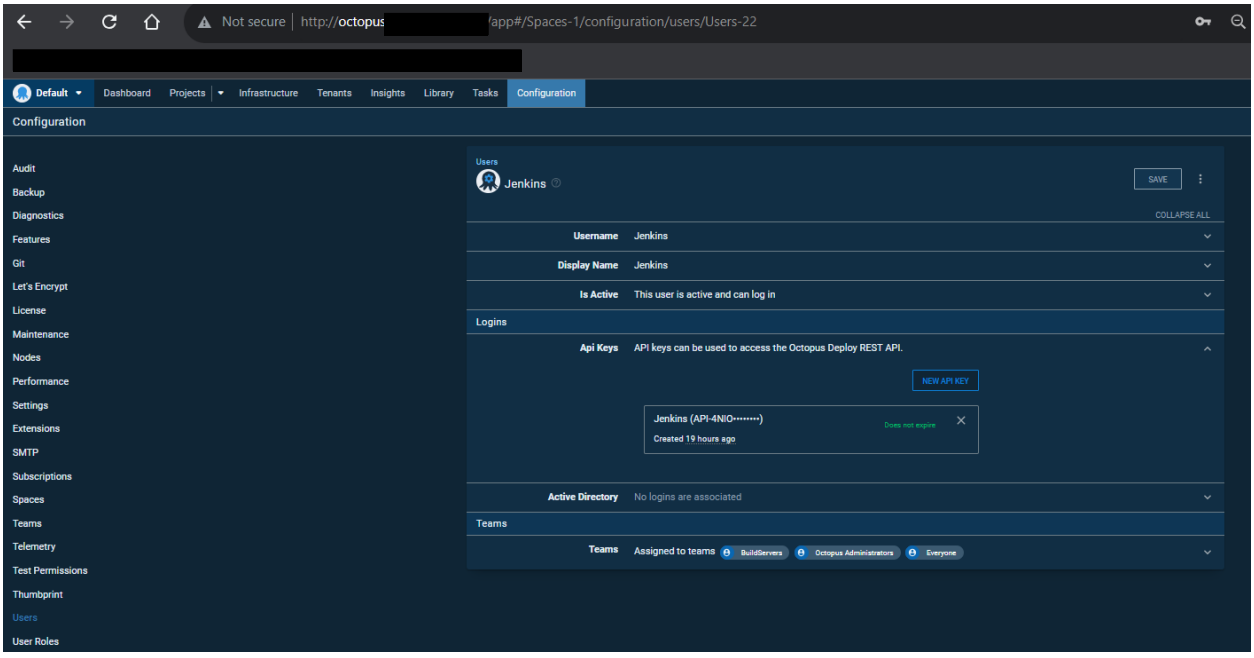


Octopus API User Configuration Overview:

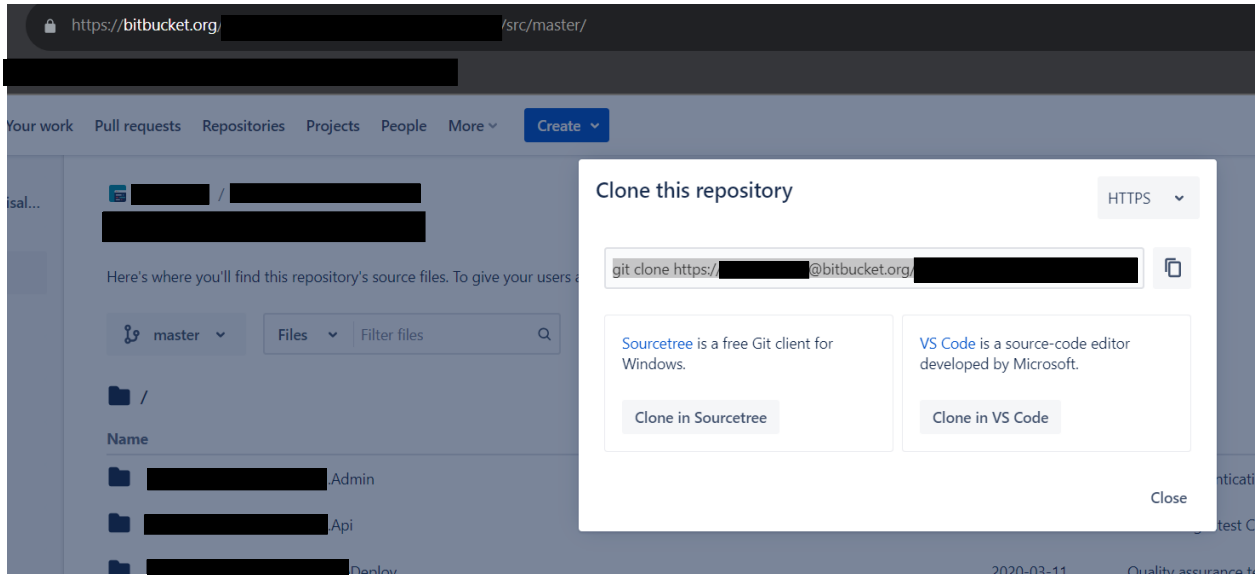
Creating an API Key for Jenkins to login to Octopus is done through the Octopus Configuration Window:




Once the Key is created, it is no longer viewable, you can create multiple keys for a user, the key for Jenkins is recorded at the end of this document. Do not change or delete the key for the Jenkins user!



BitBucket Git Clone Configuration:



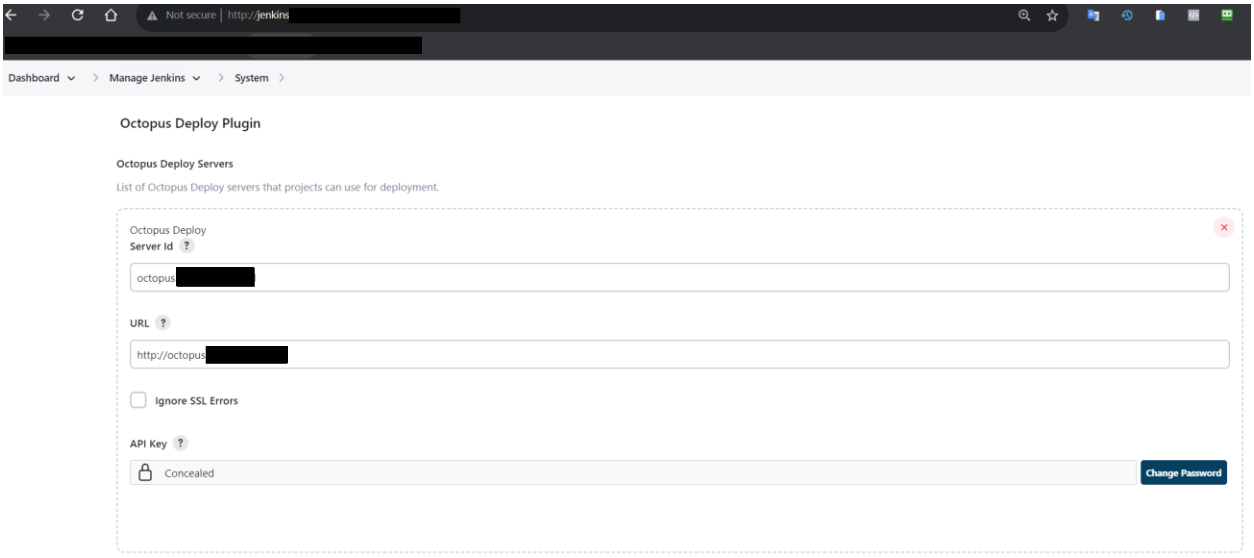
Jenkins Configuration Overview:

[redacted]Db Config [Jenkins].pdf

Please reference the above attached [redacted]Db Config [Jenkins}.pdf it has the configuration pages printed out from Jenkins.

Octopus API Key for Jenkins Authentication to Octopus NugetServer:

The following screenshot is for the Octopus Deploy Plugin on Jenkins. It requires an API key related to the Jenkins account within the Octopus application. The Key is below the screenshot.

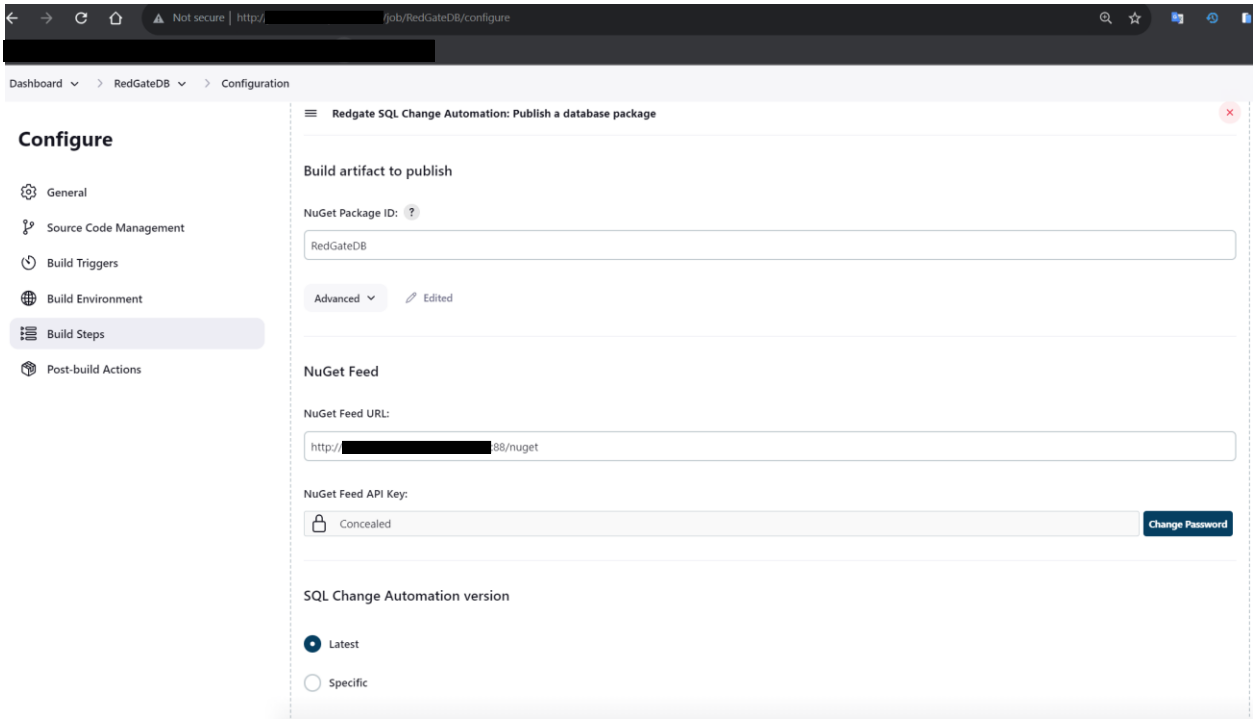



**Octopus Jenkins Key: API-[redacted]**

Another thing to keep in mind is that this API key is located in D:\Build_Tools\Scripts\OctoBuild.ps1 on [redacted]devbox01.[redacted].local and is used for all the other various applications within the CI/CD system.

Jenkins builds will not work if this key is changed on the Octopus server.

There is another API key which is specific for the Nuget package manager loaded on [redacted]devbox01, it uses the API key below the screenshot. This key is generated by the Octopus Nuget Package Manager and is located within the configuration file for the IIS web application. "[redacted]devbox01 - D:\nuget_site\web.config (yeah… I know)




**[redacted] NuGet Server API: [redacted]**

Also keep in mind is that this API key is also located in `D:\Build_Tools\Scripts\DotNetBuildWeb.ps1` on [redacted]devbox01.[redacted].local and is used for all the other various applications that use the [redacted] Nuget Package Manager within the CI/CD system.

Jenkins builds will not work if this key is changed on the octopus server.

Octopus JSON Files for [redacted]DB:

[redacted]db-process.json [redacted]db-variables.json