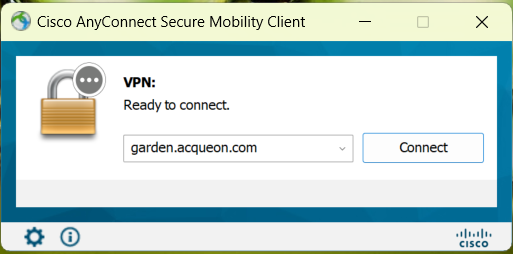
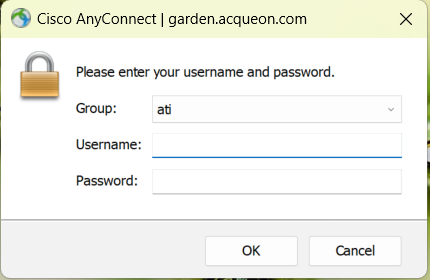
JENKINS   
  
Connecting to Jenkins via VPN and Accessing SVN

**Introduction**: This document outlines the steps to connect to Jenkins via VPN and access SVN repositories. By following these instructions, you will be able to navigate the Jenkins dashboard, access project folders, and utilize SVN repositories for your application builds. Here are some key points to remember when connecting to Jenkins with a VPN and SVN link:

* The VPN connection must be properly configured to allow access to the Jenkins server.
* Ensure that the SVN repository is accessible from the Jenkins server.

1. **Connect to the VPN.**  
     
   Open the Cisco AnyConnect Secure Mobility Client.
   * The Cisco AnyConnect Secure Mobility Client is a VPN client that allows you to connect to a remote network securely.  
     

Enter your username and password.

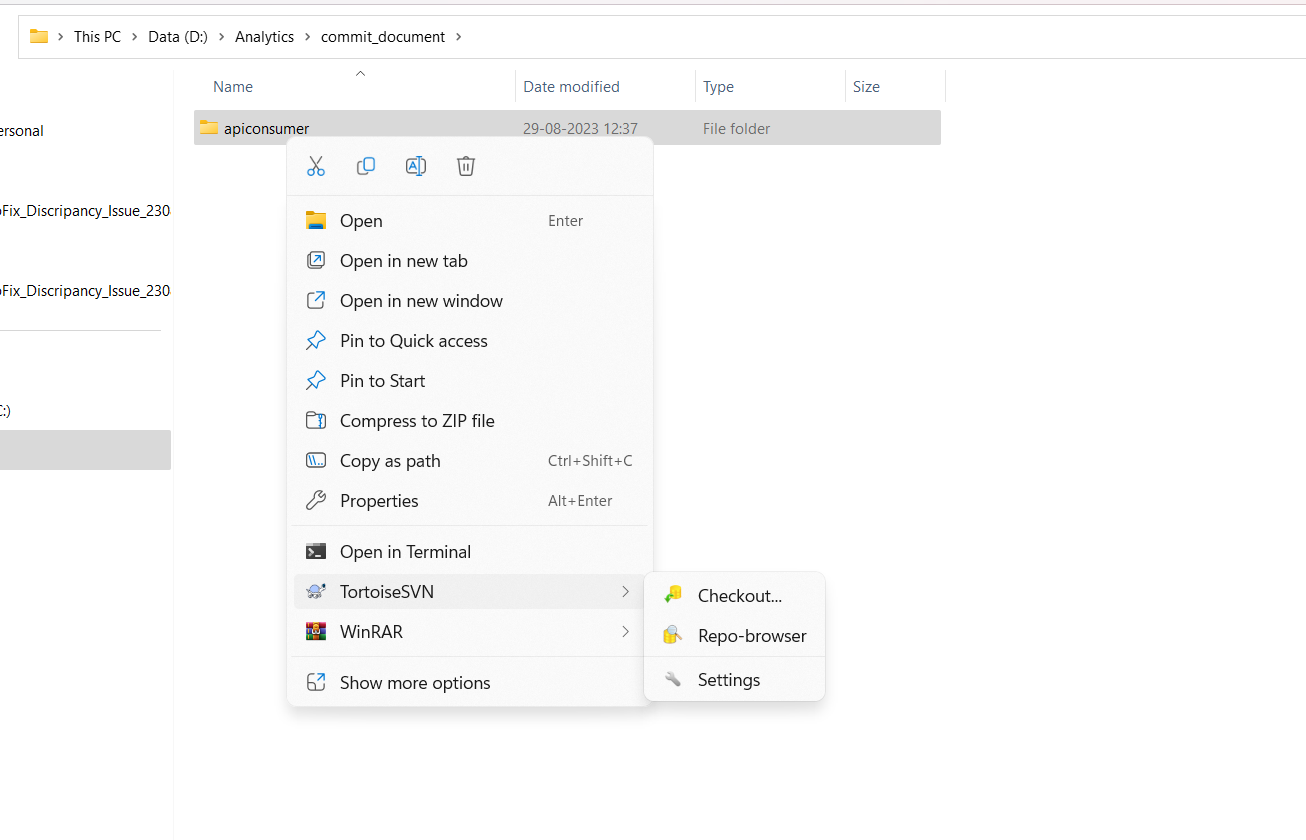
* + Your username and password are the credentials that you use to access the VPN network.  
      
    

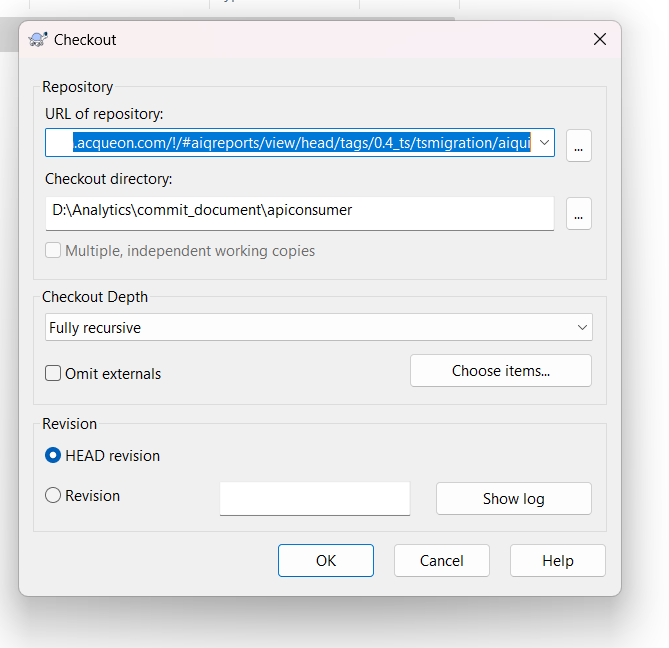
Click the OK button.

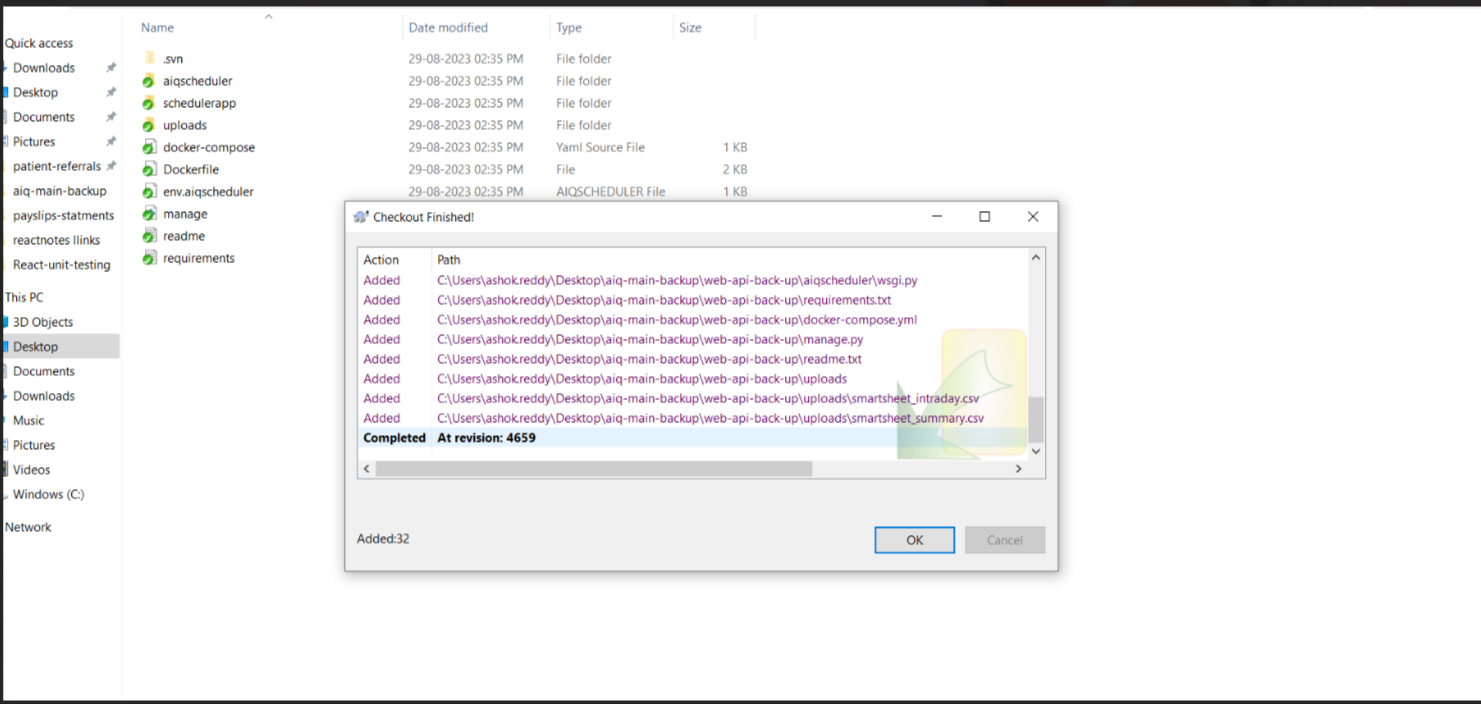
* + This will initiate the connection to the VPN network.

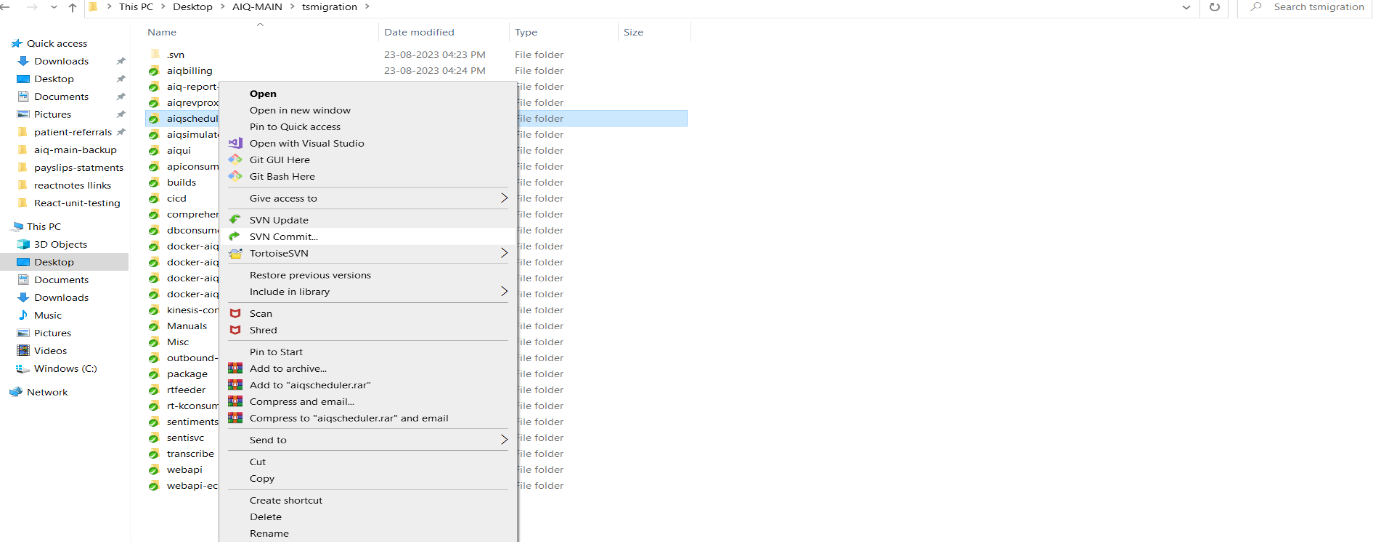
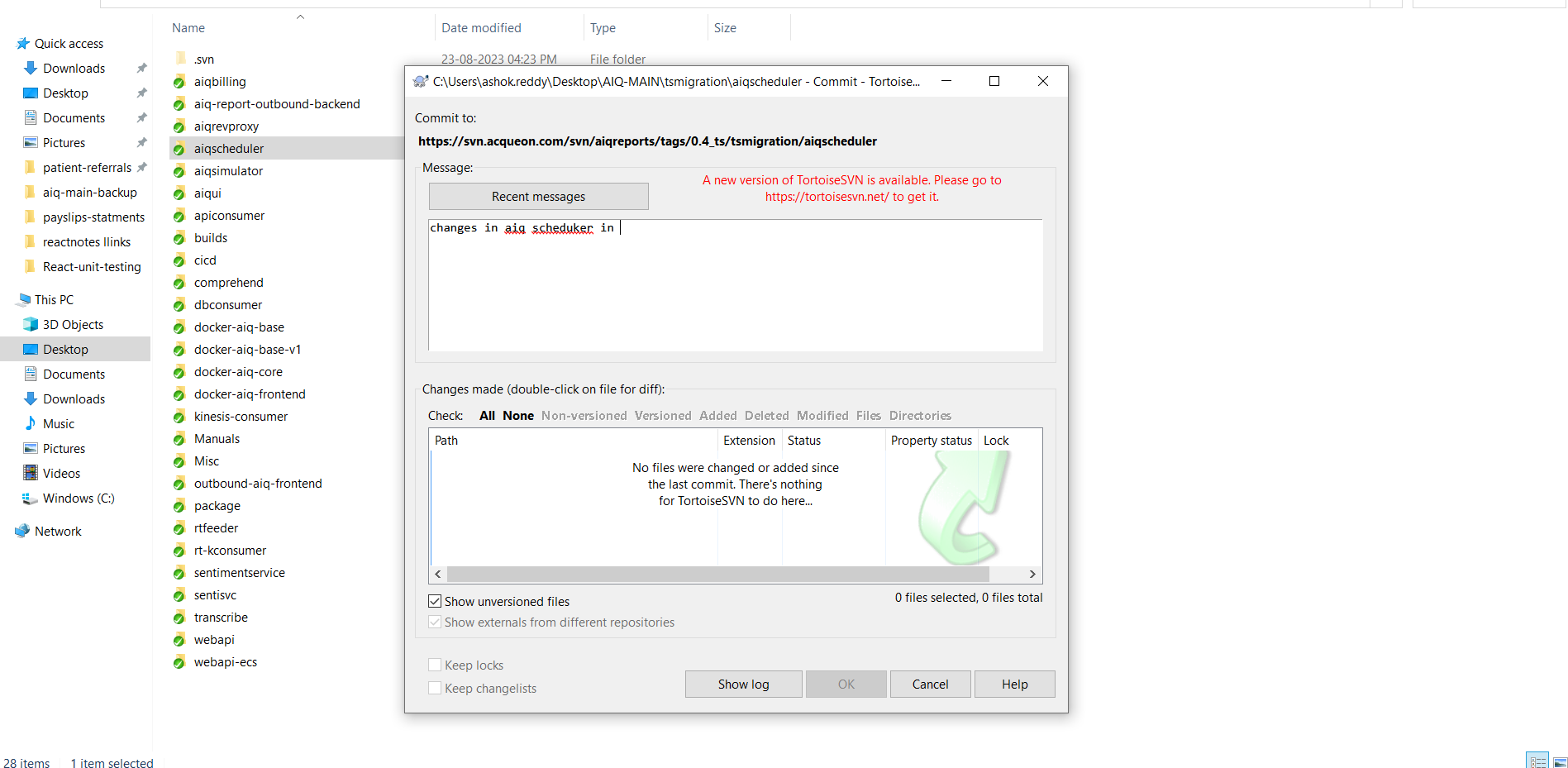
The VPN will connect and you will be able to access the resources on the VPN network.

**2.Importing Source Code from SVN:** Before you begin, ensure that your project's source code is imported from the SVN repository. Follow these steps to import your source code:

1. Create a new file or directory in your local workspace.
2. Hover over the SVN option and click "Checkout."  
   
3. Provide the SVN repository path from where you want to pull the source code.



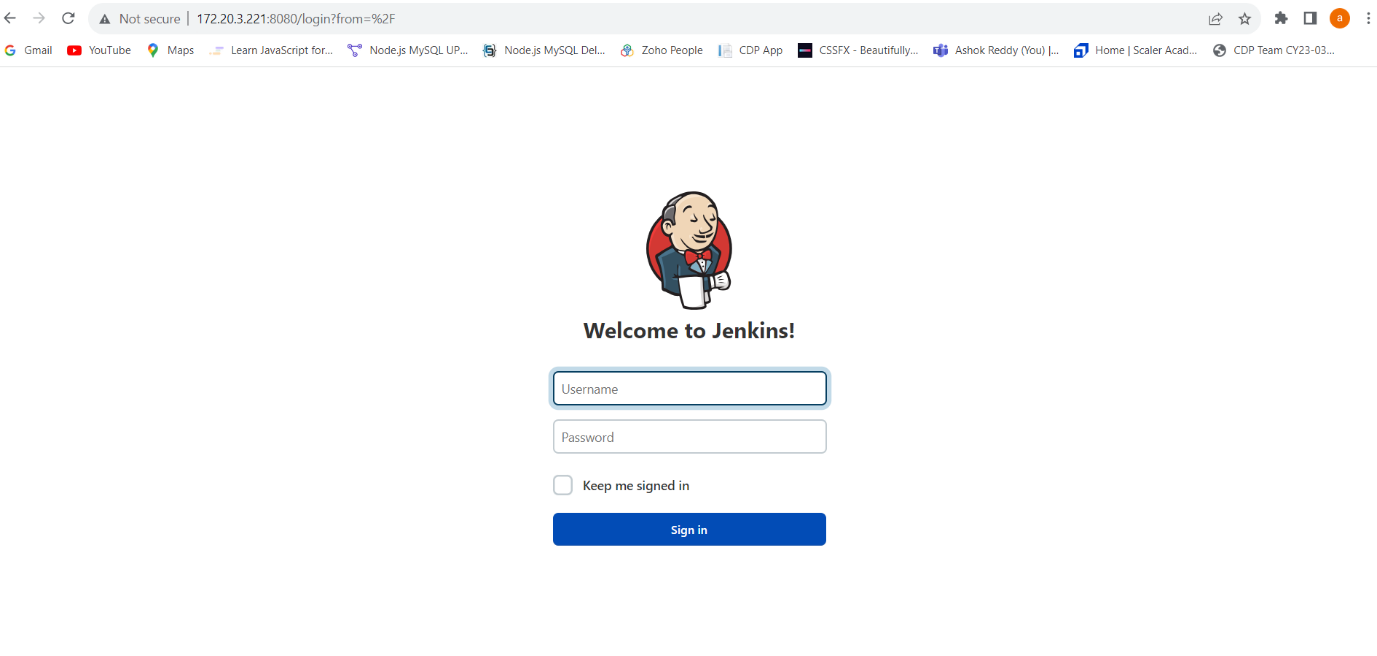
4. press ok and next ok.and you can see the source code in your local system  
  
  
  
**3.SVN Commit dialog box**: This dialog box is used to commit changes to files that are under SVN version control. To commit a file to SVN, you would do the following:

1. Open the folder of your project.
2. Go to the code file that you need to commit.
3. Right-click on the file in which you made changes and select "SVN Commit".  
     
   
4. In the SVN Commit dialog box, enter a commit message.  
     
   
5. Click the "OK" button.

This will commit the changes to the file to the SVN repository.  
Once the changes have been committed to the SVN repository, the results should be built in Jenkins.

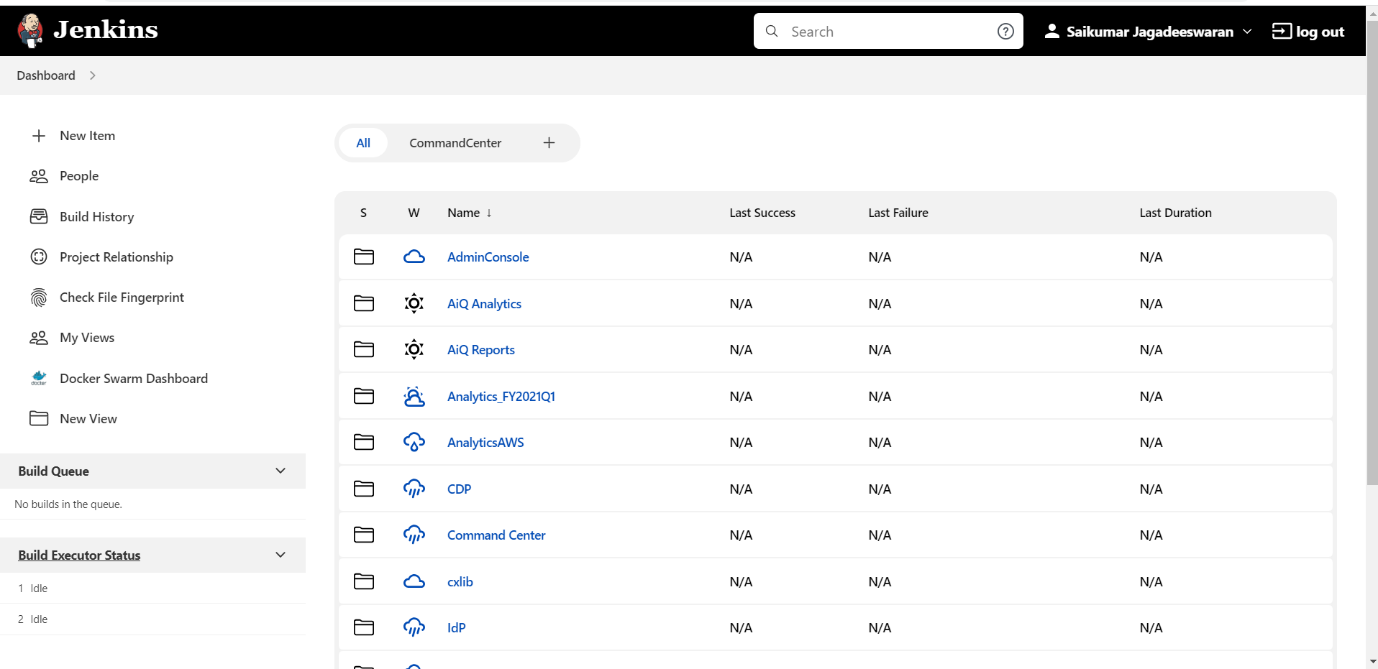
**3.Access the Jenkins Dashboard**:   
Open a web browser and go to the Jenkins URL-

* In the browser's address bar, replace "localhost" with the IP address of the Jenkins server:  
  <http://172.20.3.221:8080/>
* Press "Enter" to navigate to the modified URL.
* Provide your Jenkins instance username and password.
* Click the "Login" button to access the Jenkins dashboard.



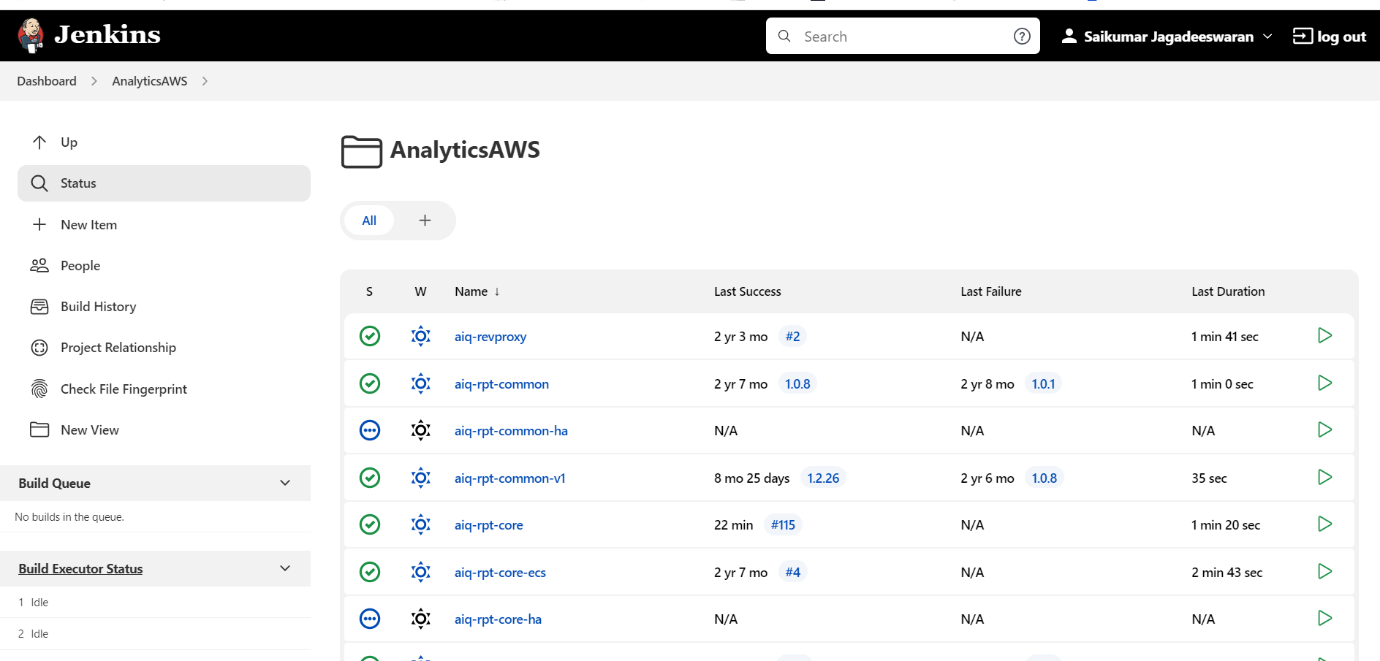
3. **Navigate to the Project Folder**

1. When you login to Jenkins, you will see a list of all the projects that are configured on the Jenkins server.

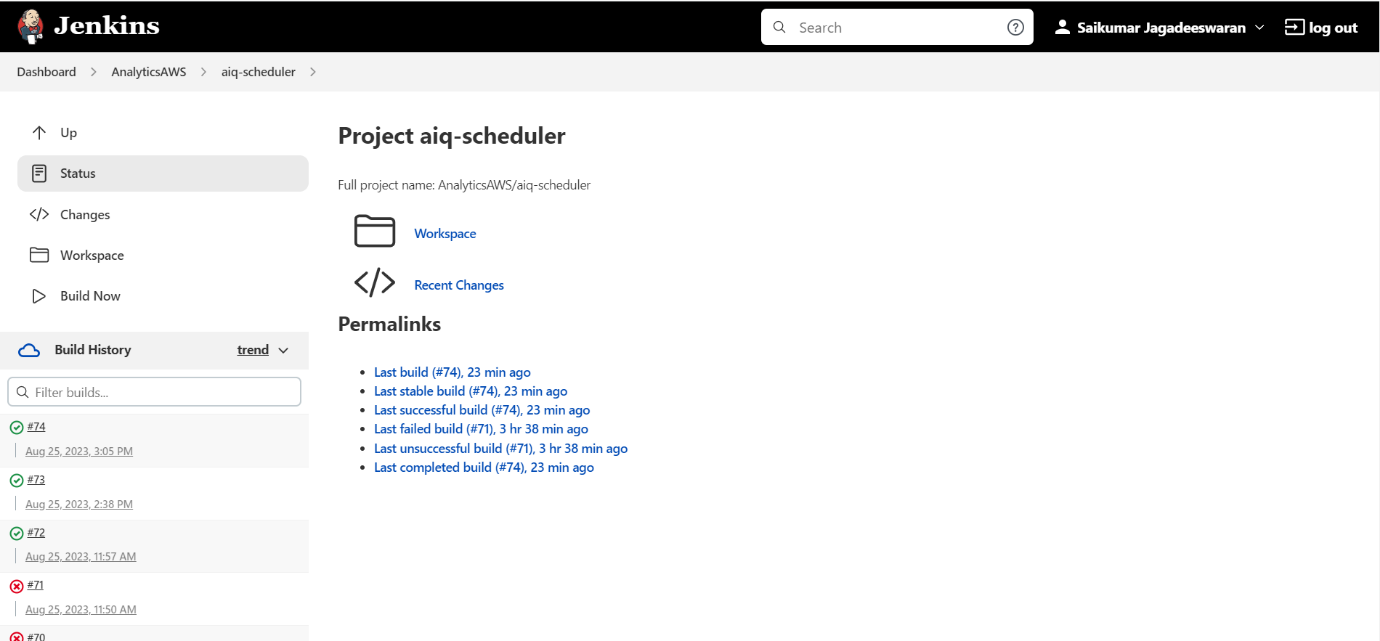
For example: On the Jenkins dashboard, locate the folder called "**AnalyticsAWS**".  
  




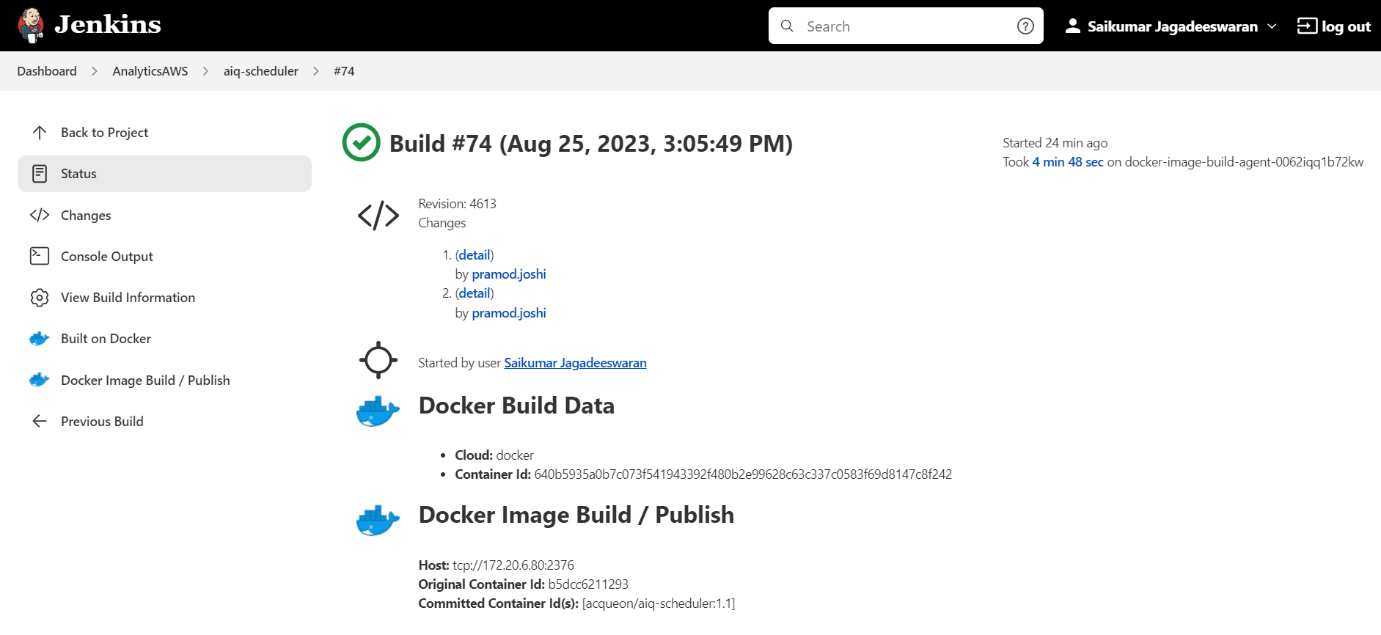
Click on the folder called "AnalyticsAWS".

1. This will open the folder and you will be able to see all of the containers that are associated with the project:  
   
2. Now click on the name of the Container Project.  
    **To Build:** Navigate to the 'Build' option on the project page. Initiating the build process will incorporate the updated code from SVN into your project. You can monitor the progress and results of the build in the console output.

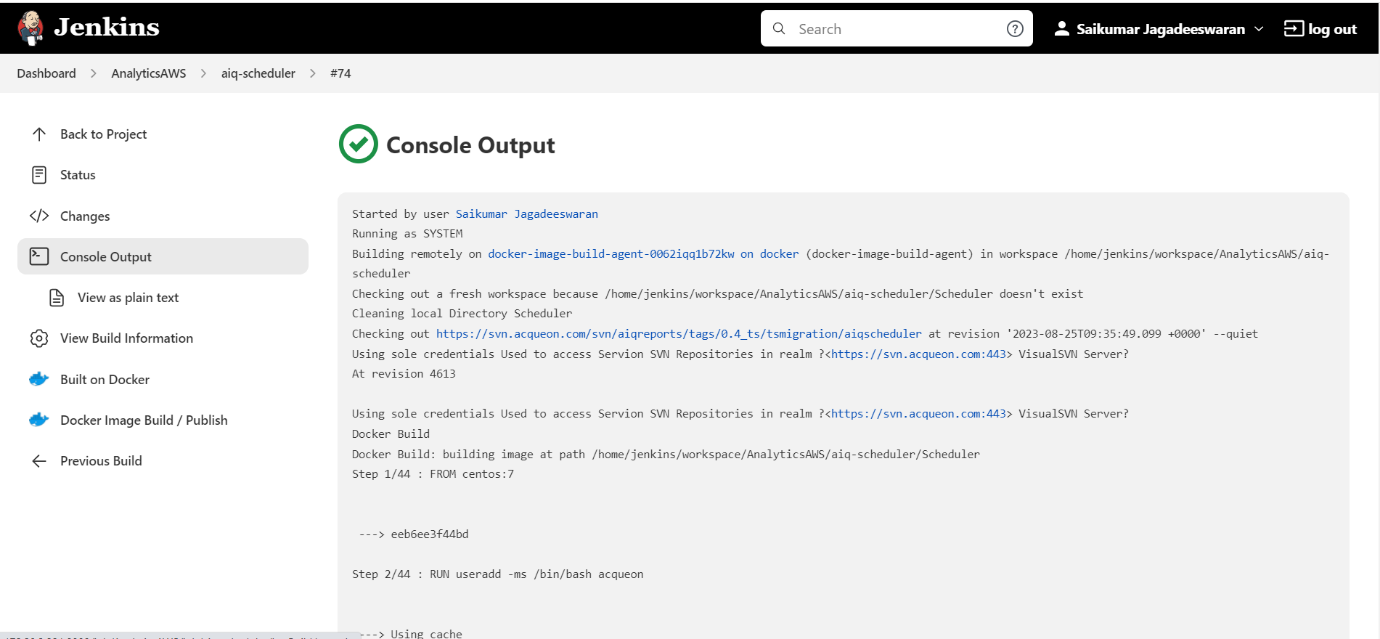
To find the last build of the aiq-scheduler container, go to the project page and click on the "Last Build" link.





1. When you click the "Last Build" link, you will be taken to the page for the last build of the container. On this page, you can find the console output for the build.  
     
   



e) In the "Console Output" tab, you will see the entire process of the build, including any errors or warnings that occurred.  
  
  
The console output can be quite large, so it may be helpful to scroll through it or search for specific keywords. If the console output is too large to fit on the screen, you can click on the "Expand" button to view it in a separate window. You can also download the console output as a file by clicking on the "Download" button.

The console output can be used to troubleshoot problems or understand how the build works. For example, if the build fails, you can check the console output to see the error message. You can also use the console output to see what steps were taken during the build.

In addition to the build process, the console output may also include information about pulling data from SVN, building containers, and pushing them to the Docker Hub. If you need to make changes to the project source code, you can copy the SVN link from the console output and paste it into an SVN repo-browser.

Once you have made the changes to the source code, you can rebuild the containers and push them to the Docker Hub. To pull the containers from the Docker Hub to your local machine, you can use the docker pull command. Be sure to stop the services before pulling the containers, so that the services are not interrupted. Once you have pulled the containers, you can start the services again.