**DEPLOYMENT IN WEBLOGIC USING IBM UDEPLOY**

**Contents**

[**Steps to deploy simple java project in web logic using IBM UDeploy** 3](#_Toc454899623)

[**Step 1: Check-in the code in SVN repository** 3](#_Toc454899624)

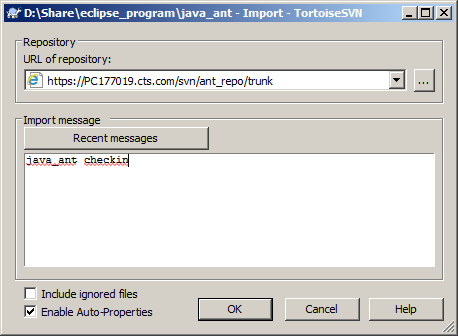
[**Step 2: Jenkins Configuration with Sonar and Ant** 5](#_Toc454899625)

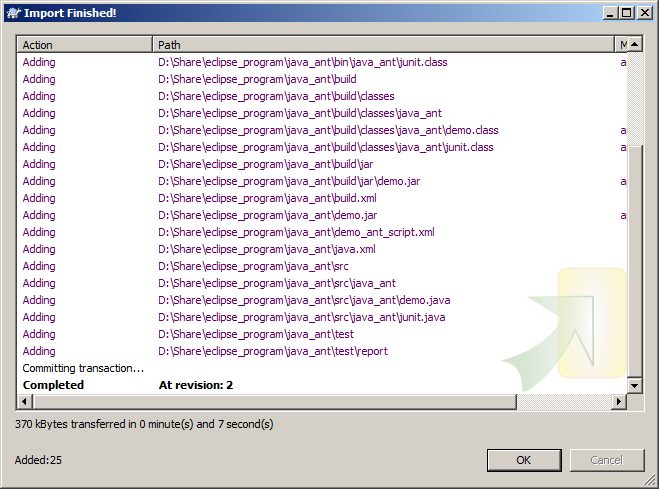
[**Step 3: IBM UDeploy Configurations** 10](#_Toc454899626)

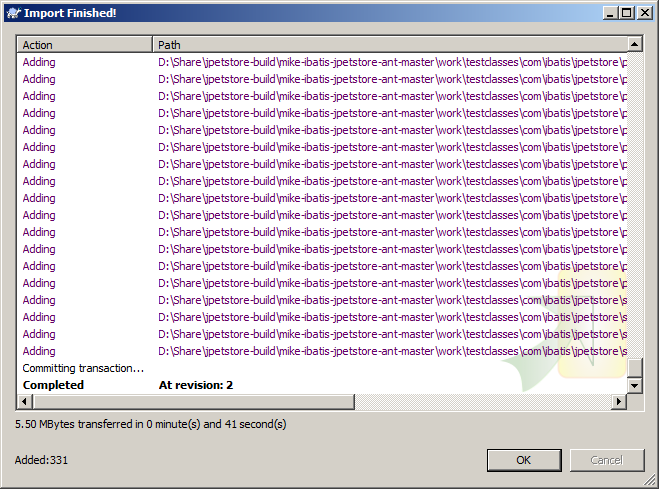
[**Deployment in web logic server.** 19](#_Toc454899627)

# **Steps to deploy simple java project in web logic using IBM UDeploy**

## **Step 1: Check-in the code in SVN repository**

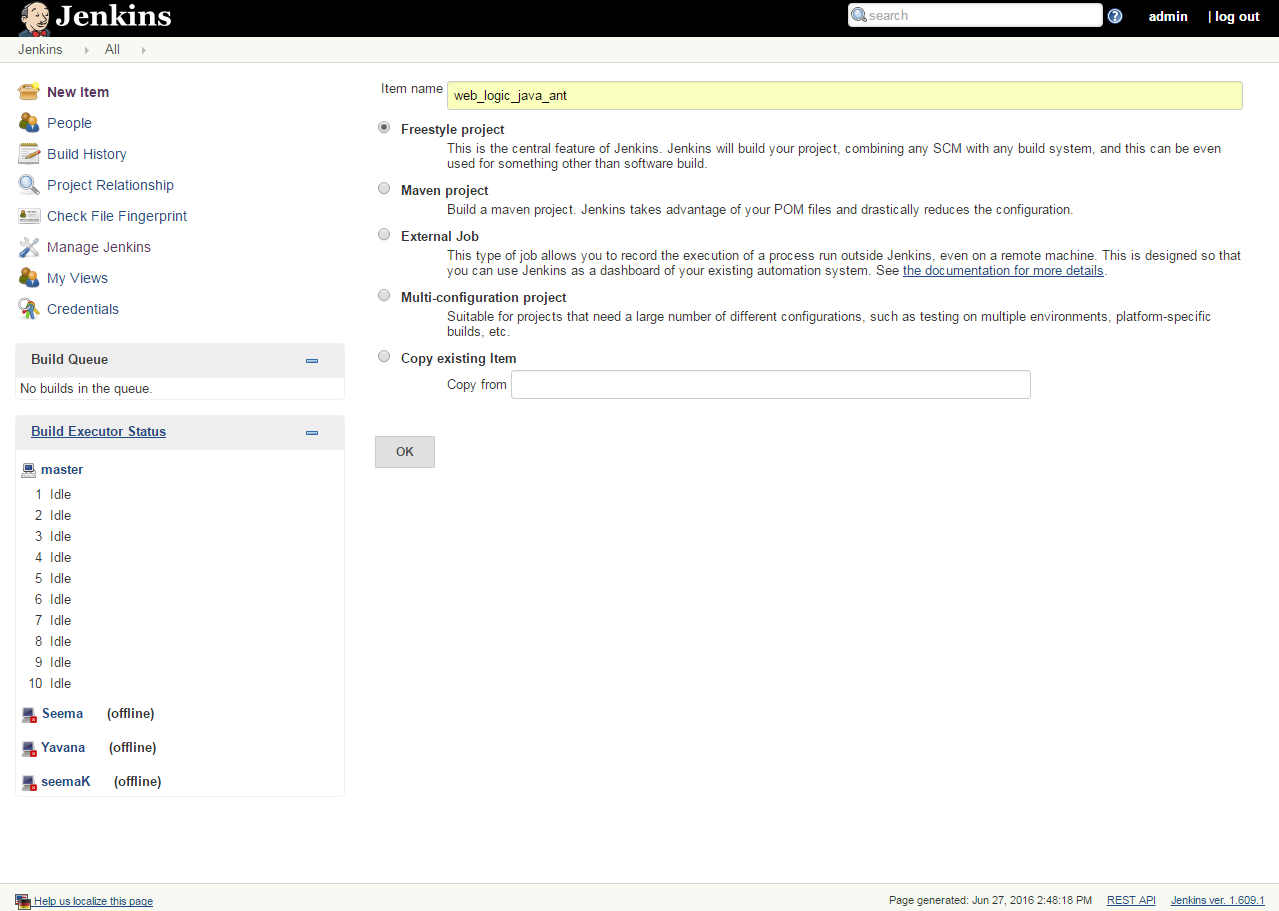






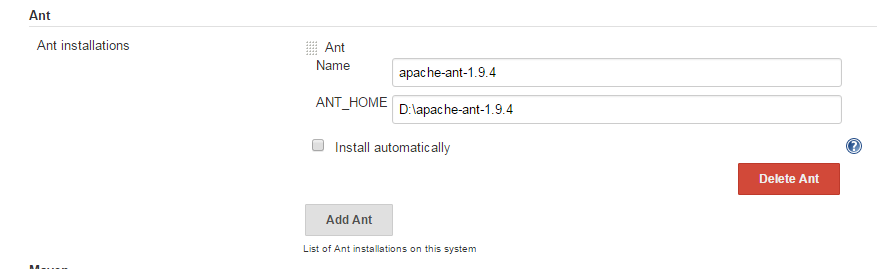
## **Step 2: Jenkins Configuration with Sonar and Ant**

1. Create a freestyle Jenkins job.

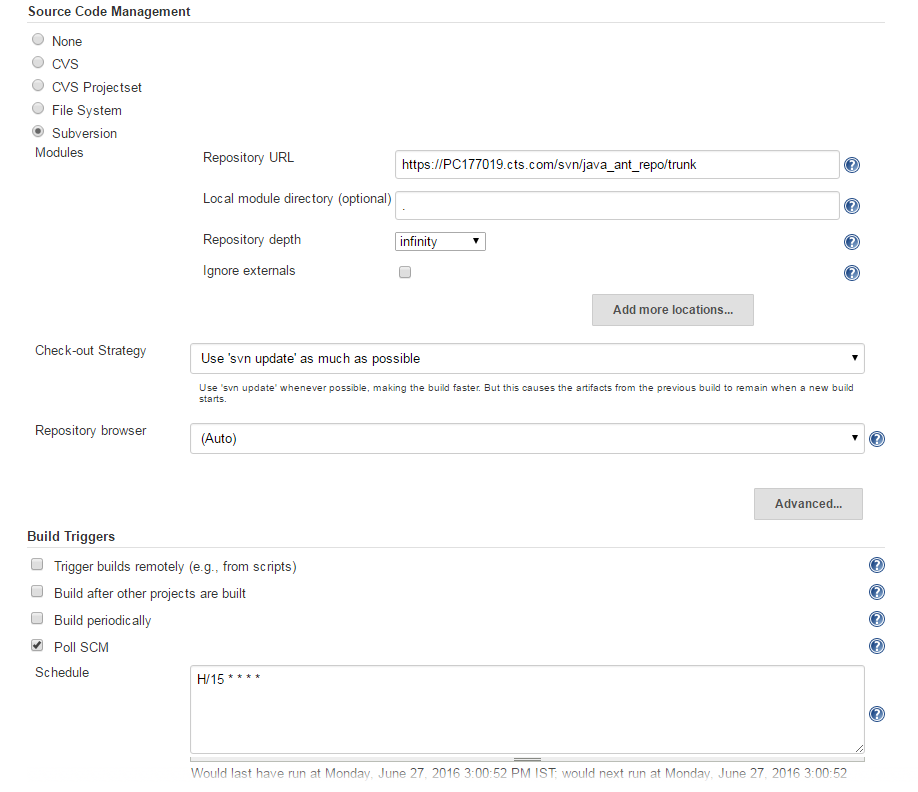


1. For invoking ANT script you need to have **Ant Plugin** installed in the Jenkins so first install it (if it is already installed then just skip this step).

Now configure ant with your Jenkins, (Set the Name and ANT\_HOME).

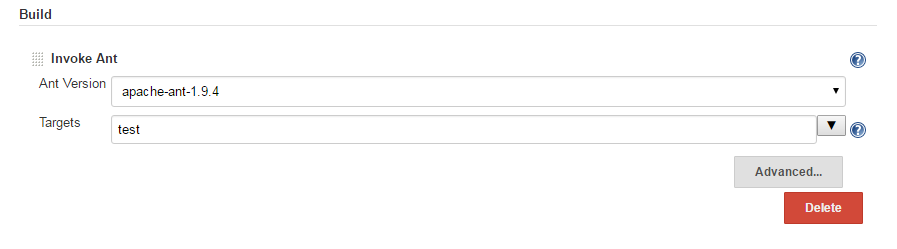


1. Configure SVN in the Jenkins job

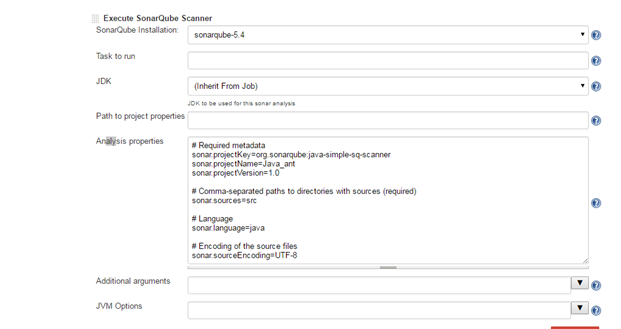


Take the code from the SVN repository by providing the SVN repository URL and poll the SVN every 15 minutes this will check for any code change in SVN every 15 minutes.

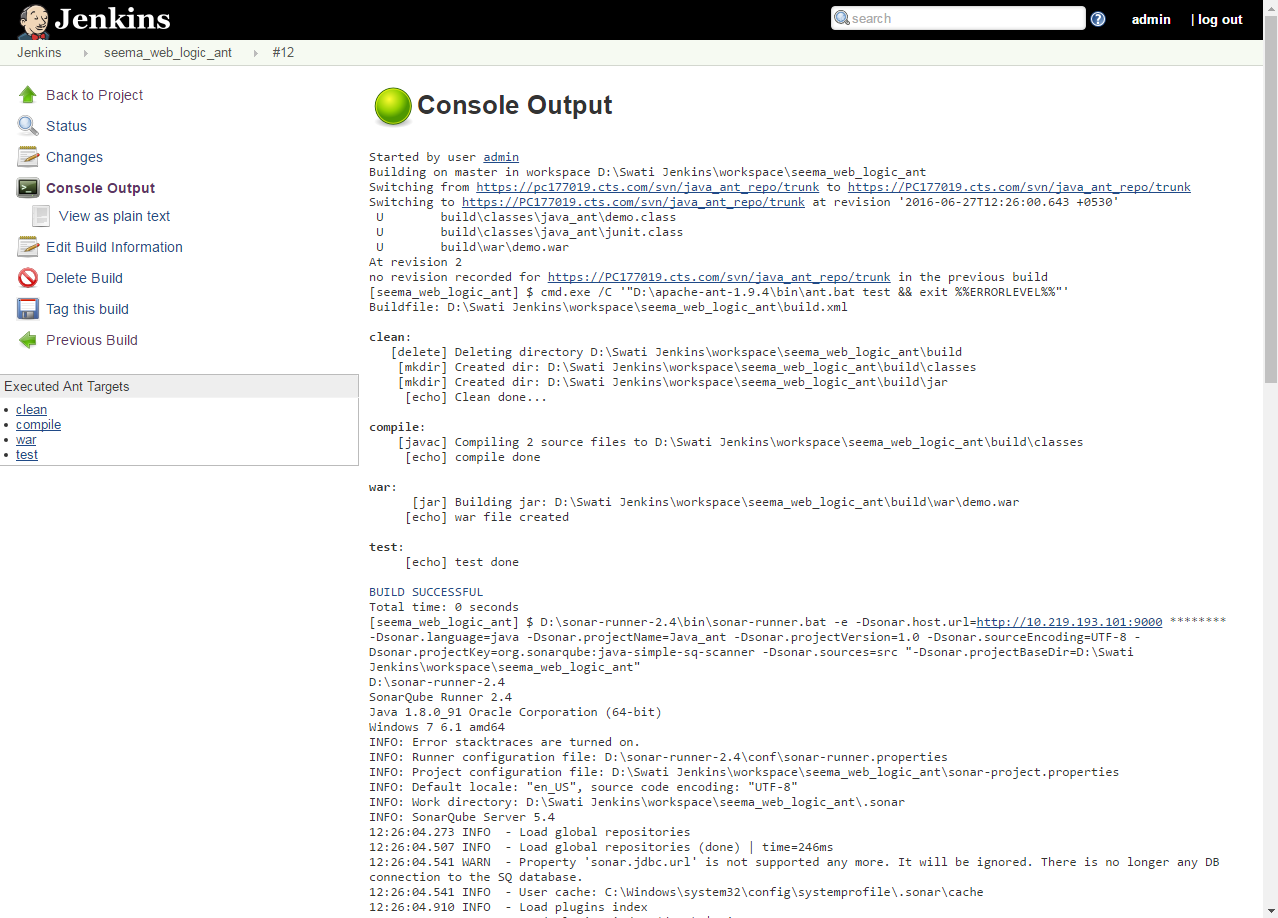
1. Add Build-step Invoke Ant 🡺 give the appropriate apache ant version and set the target as the last target given in your build.xml (ant script).

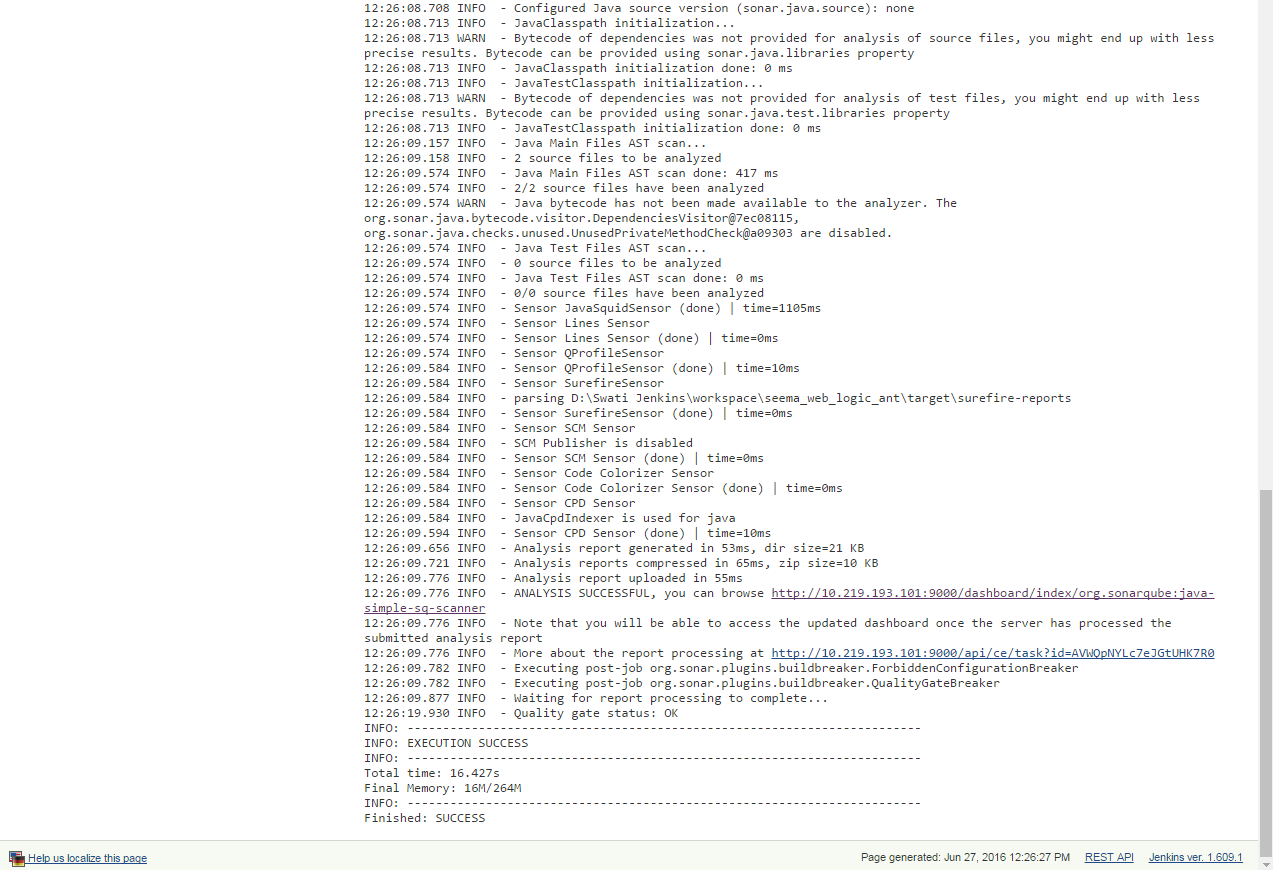


1. Add the execute Sonar as different build step and give the name of the sonar and add the property file (make sure that you have added the sonar.project property file in the folder which hold the source code (src) and build.xml).

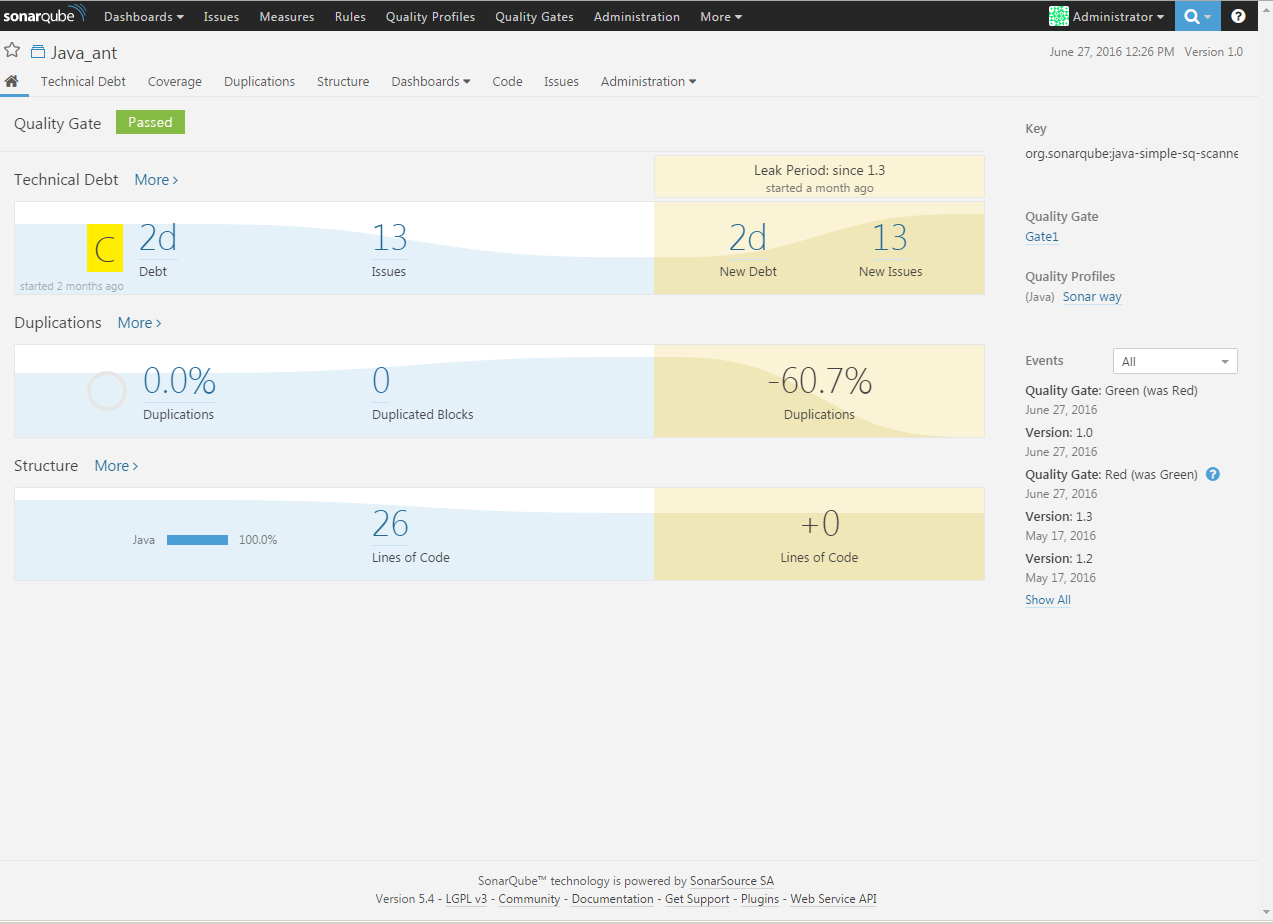


1. Build the Jenkins job.



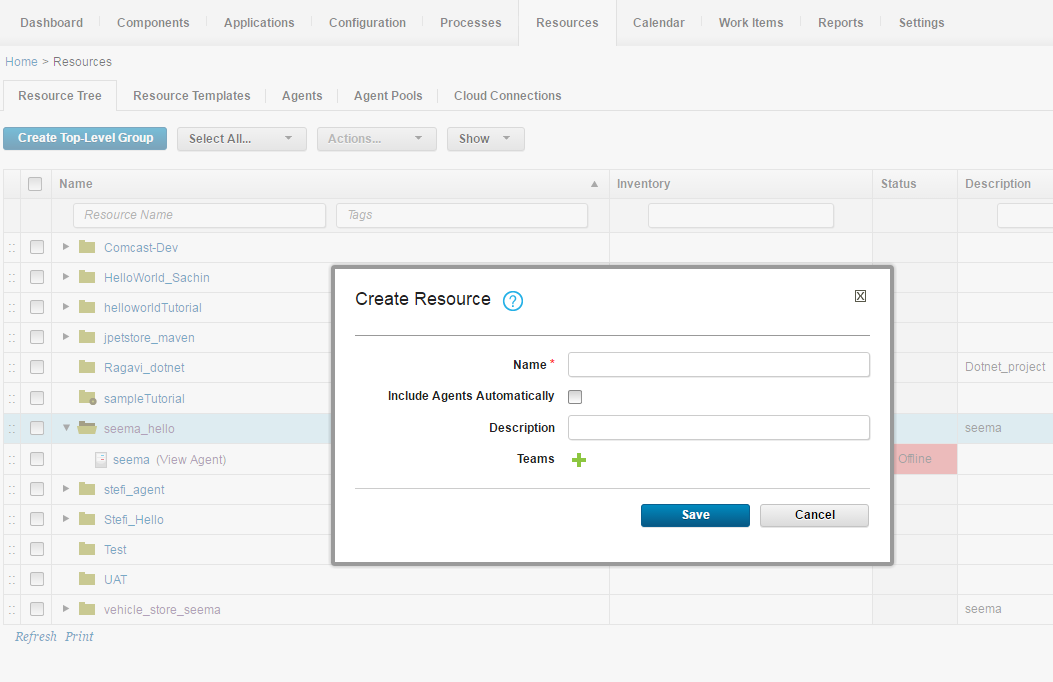


The link there in console output will redirect you directly to the sonar analysis page where you can see the quality gate passed.



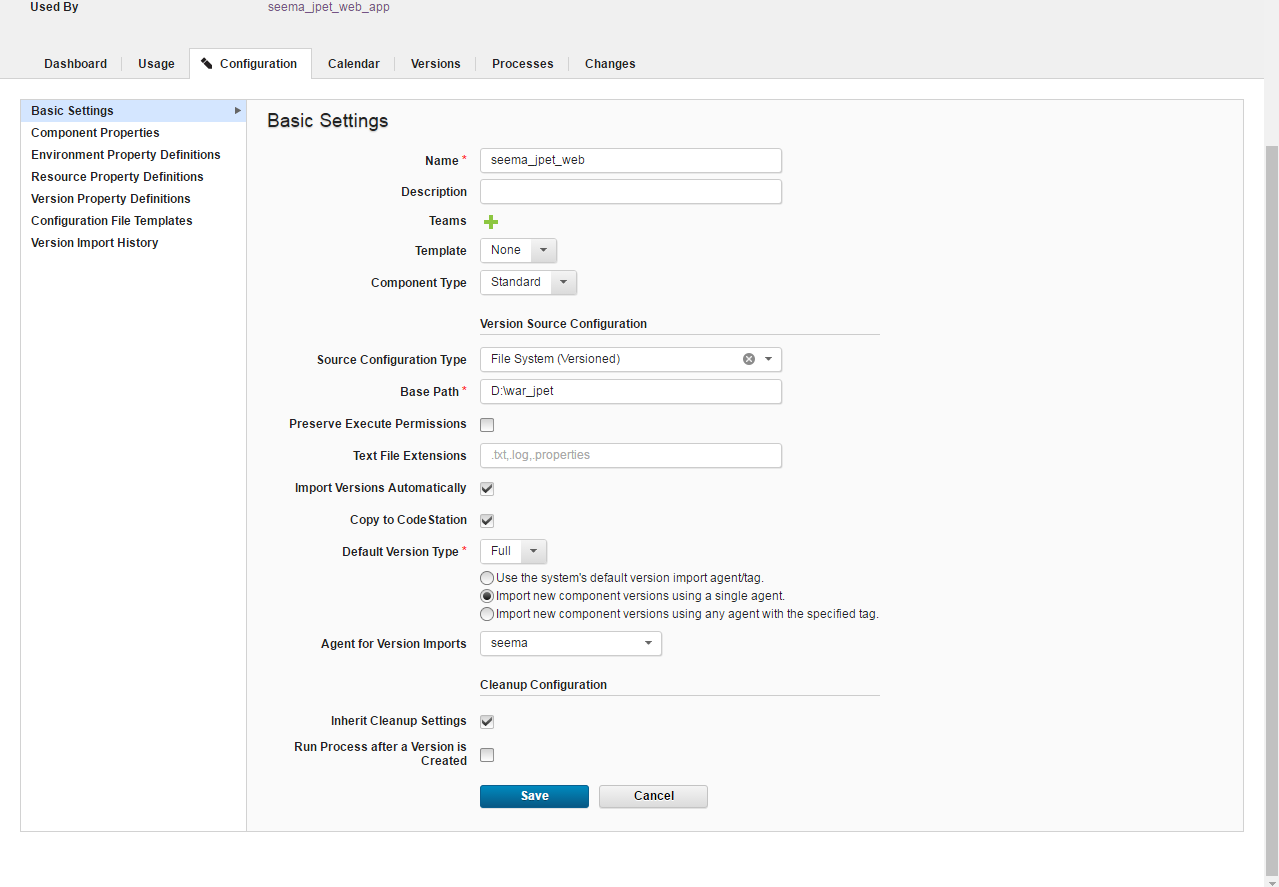
## **Step 3: IBM UDeploy Configurations**

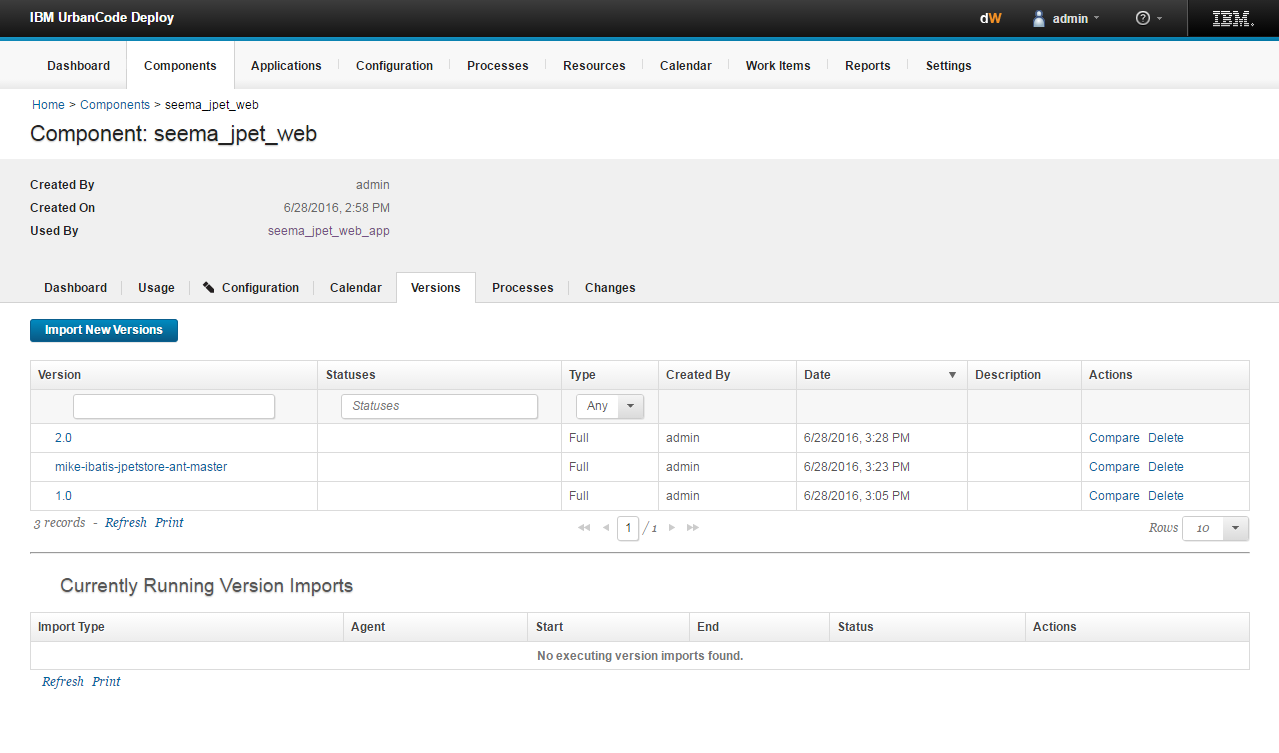
1. Create a resource group, and then add the IBM UrbanCode Deploy agent to the resource group, as shown in the following screen capture(click top level group)

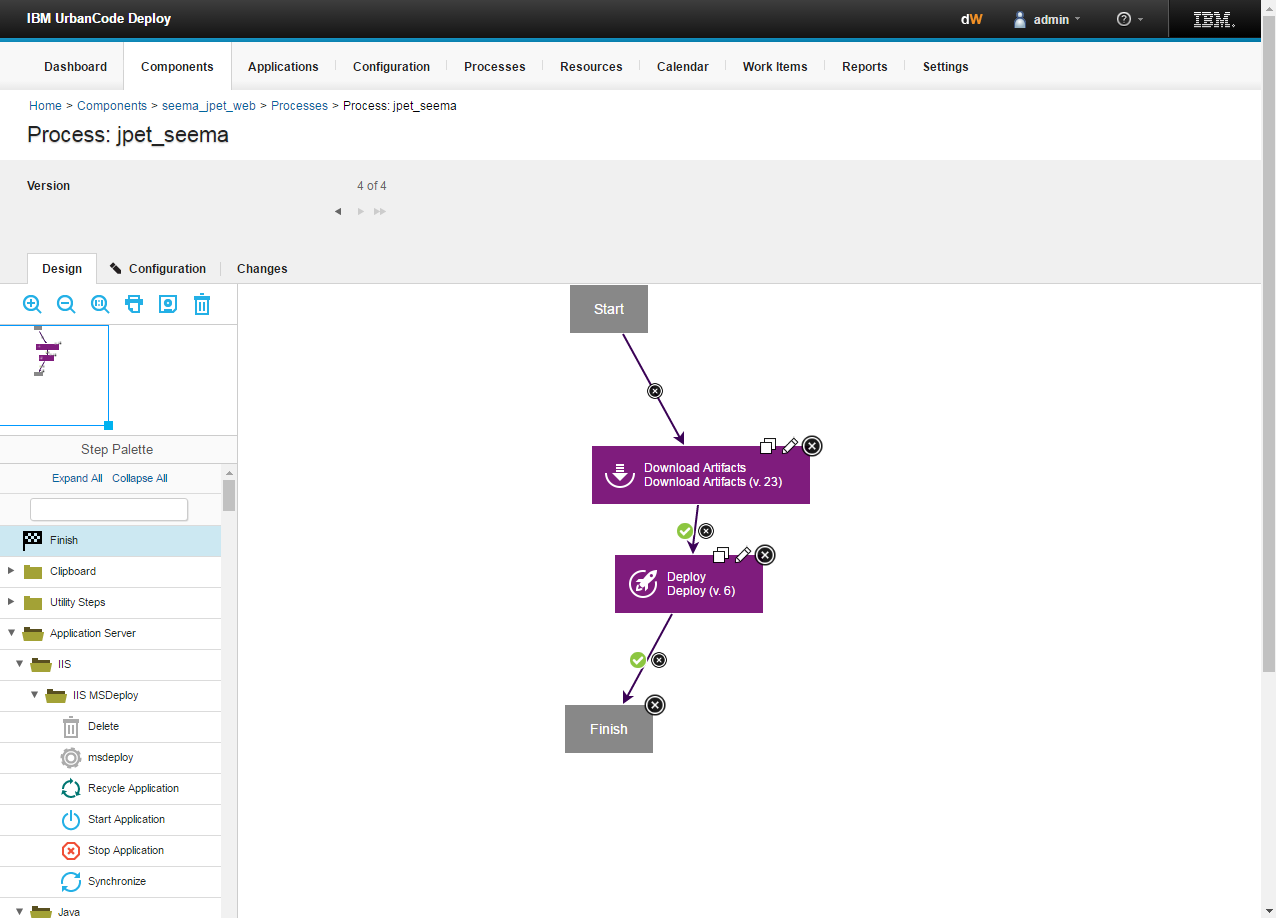


1. Create components in IBM UrbanCode Deploy

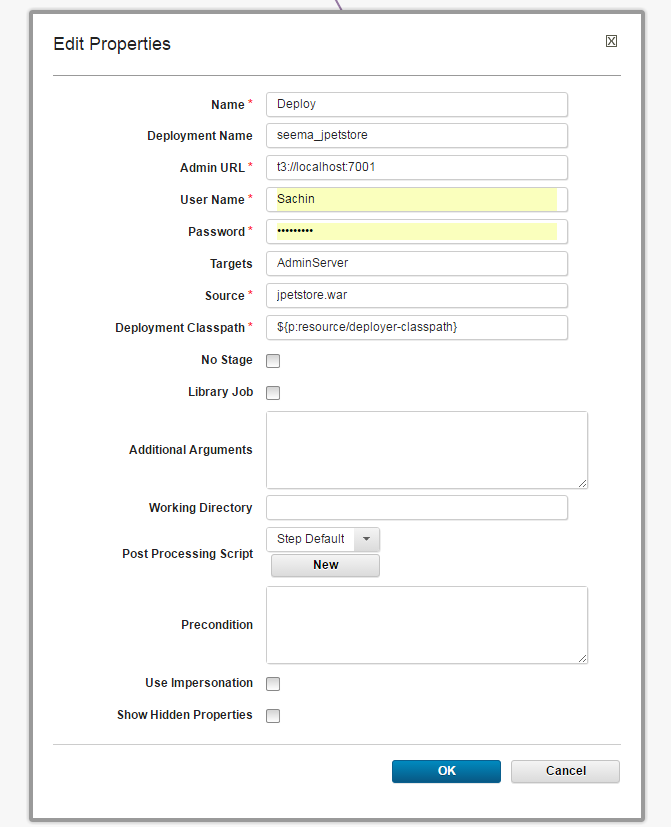
* Click the **Components** tab and then click **Create Component**. In this window, you specify where the artifacts for the component are.
* In the **Name** field, give your component name.
* In the **Source Configuration Type** list, select **File System** and pass the required repository URL.
* Click **Save**.



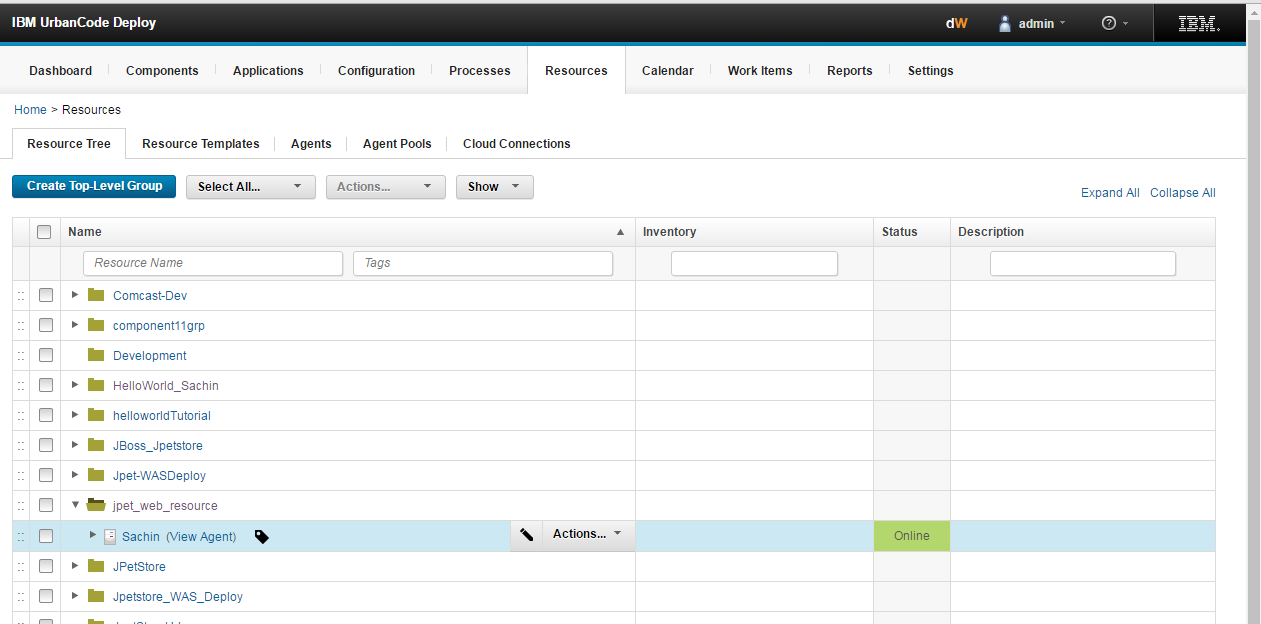
1. Go to versions and import the new version 
2. Go to process and create a new process with following configurations:



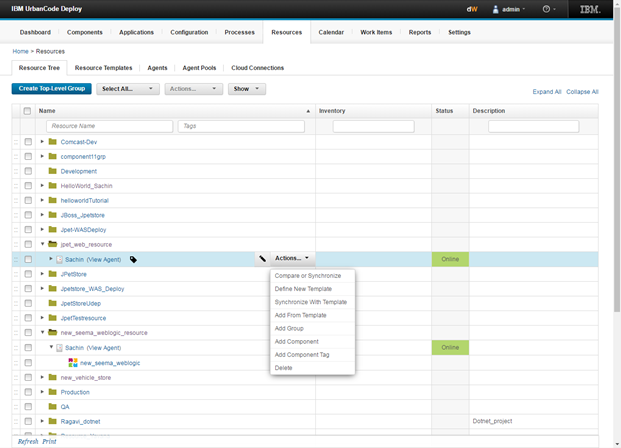
* Firstly, download the artifacts and apply the default properties.
* Then, add Deploy step with the below mentioned configurations:
* **Deployment Name:** the name of the project with which it will get deployed.
* **Admin URL:** Give the URL of the web logic you are using (like here, **t3://localhost:7001**).
* **User Name & Password:** Give the username and password of the web logic server.
* **Target & Source:** Pass the target as AdminServer and in source give the complete name of your war file.
* **Deployment Classpath:** Set it as default (as given).

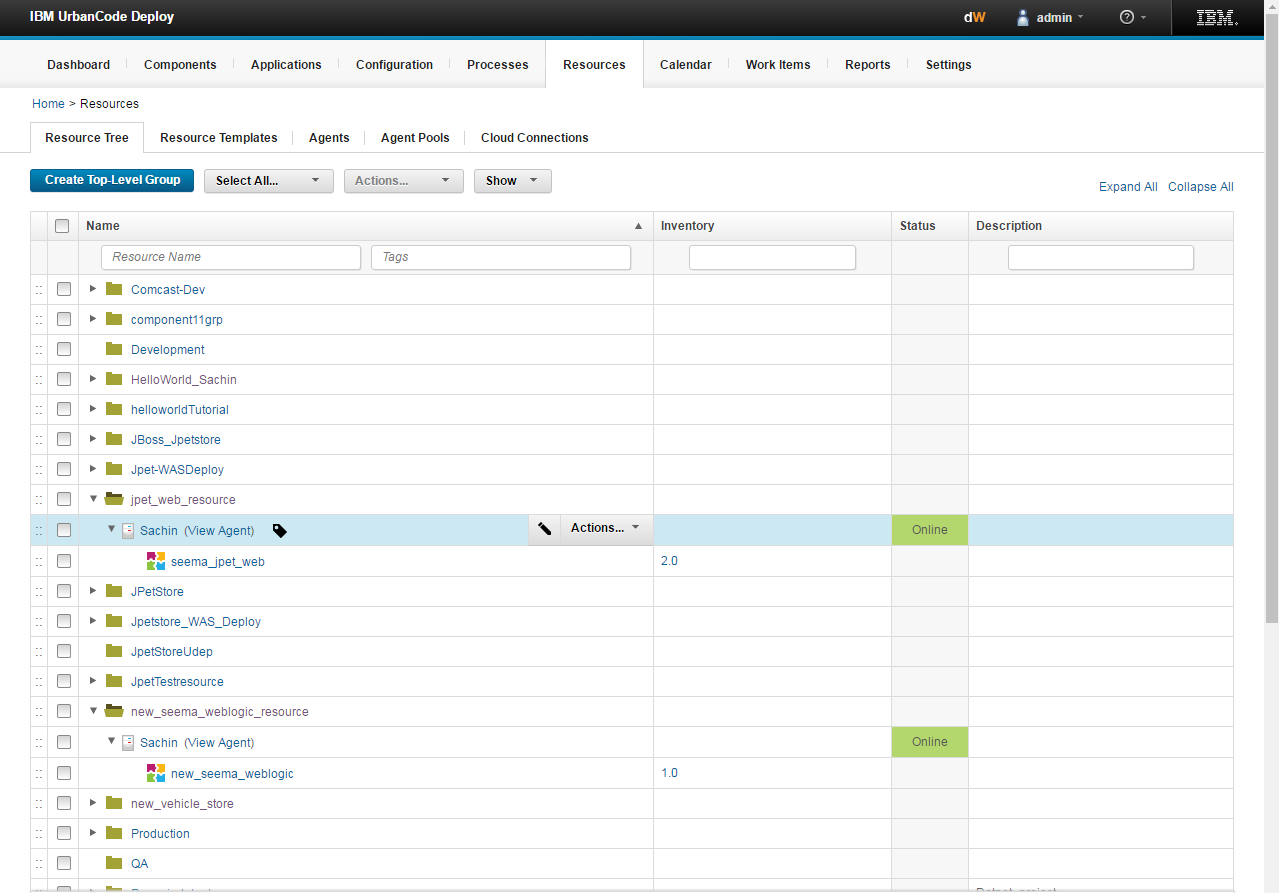


1. Go to Resources tab 🡺 Create **Top Level Resource 🡺** Map the agent on which you want to run your application.

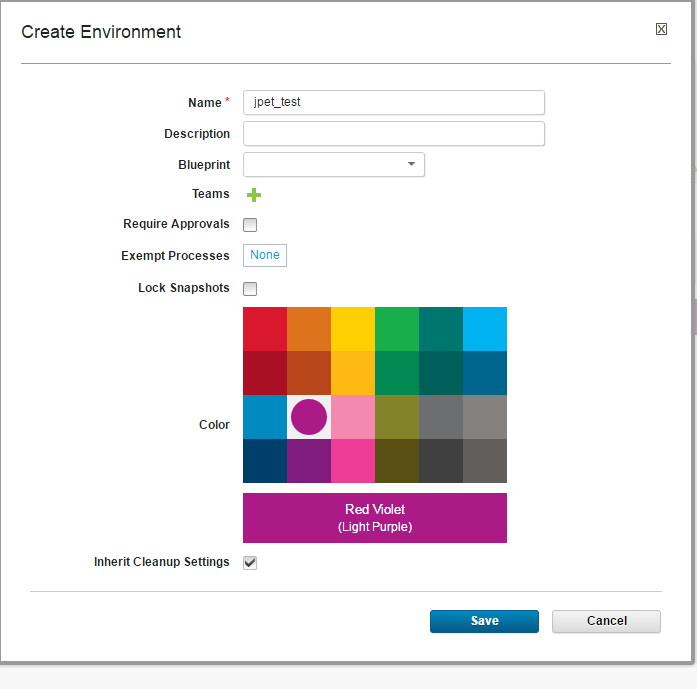


Then add the component to the agent.

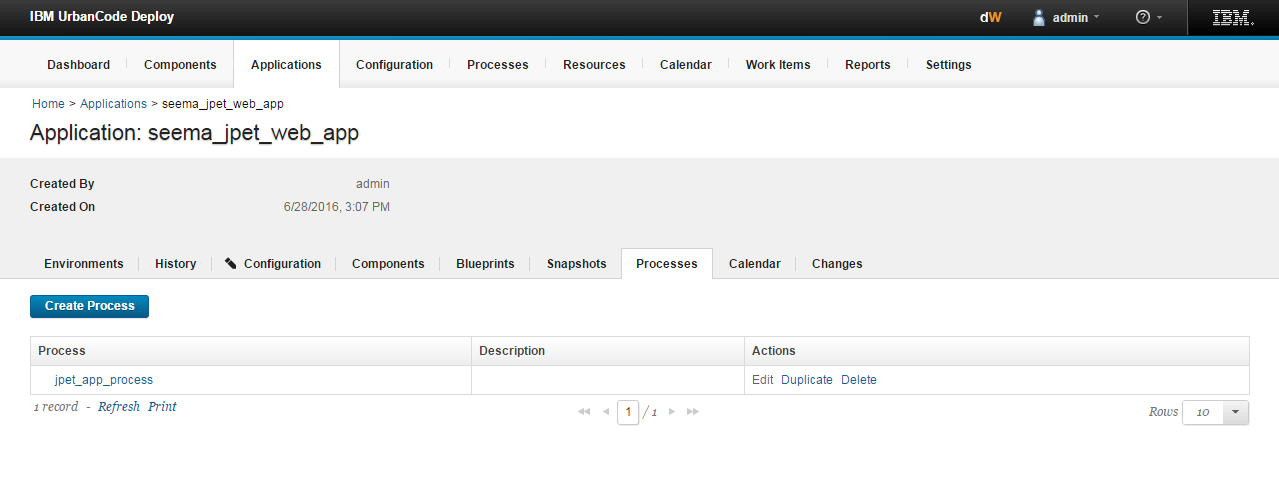




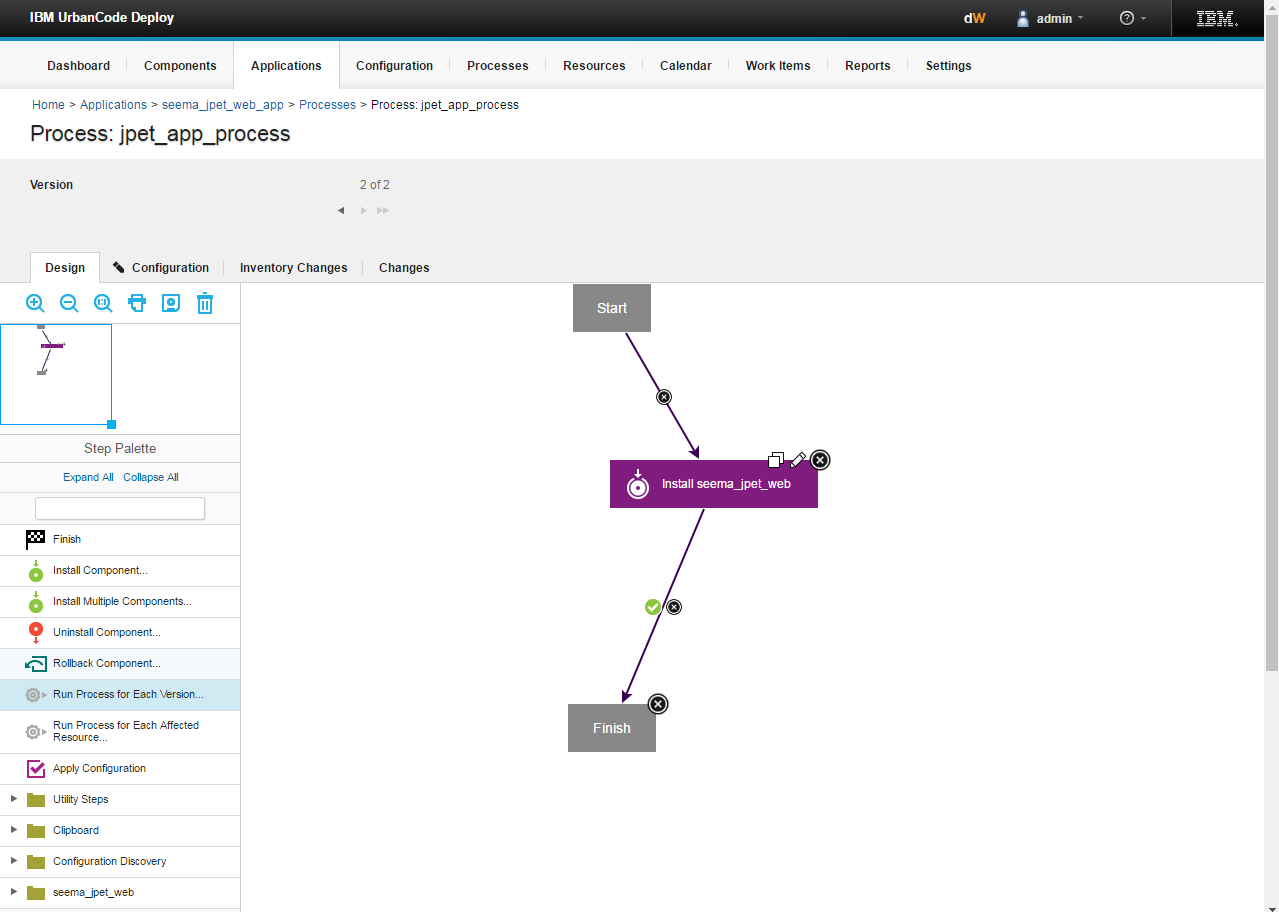
1. Go to applications and create an environment



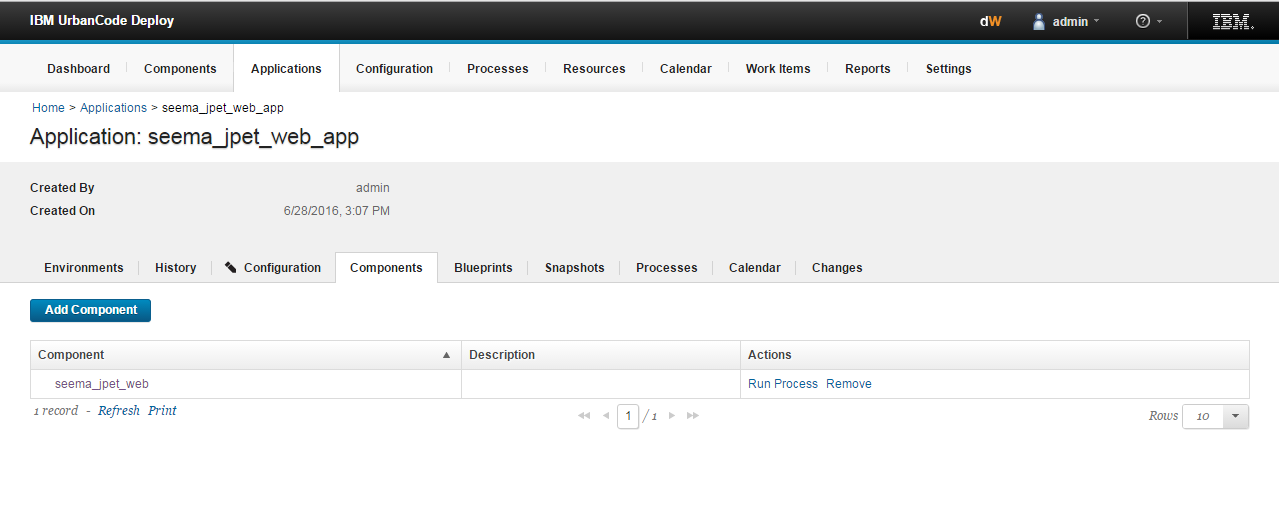
1. Go to Processes in applications and create a new process 🡺 click on edit and complete the flowchart.



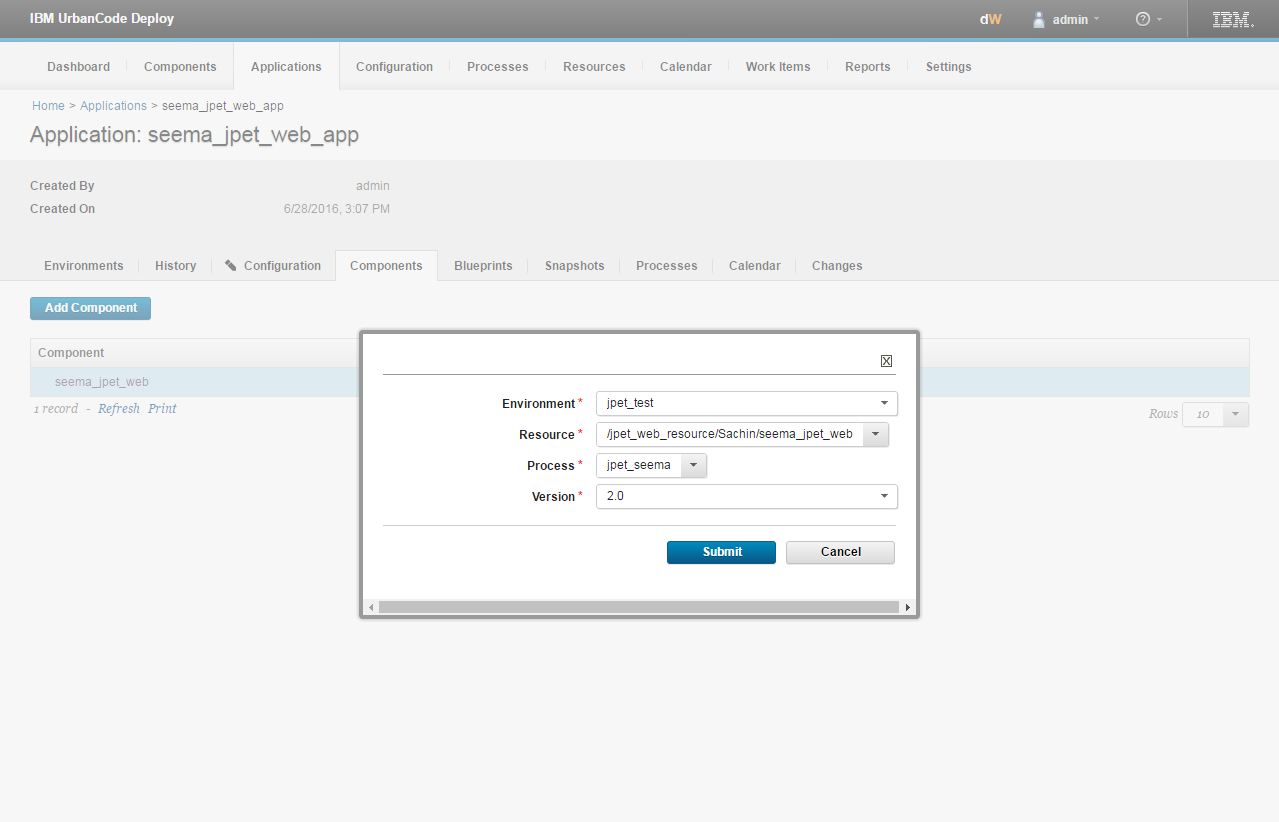
Install the component over here (just drag the install component to this).



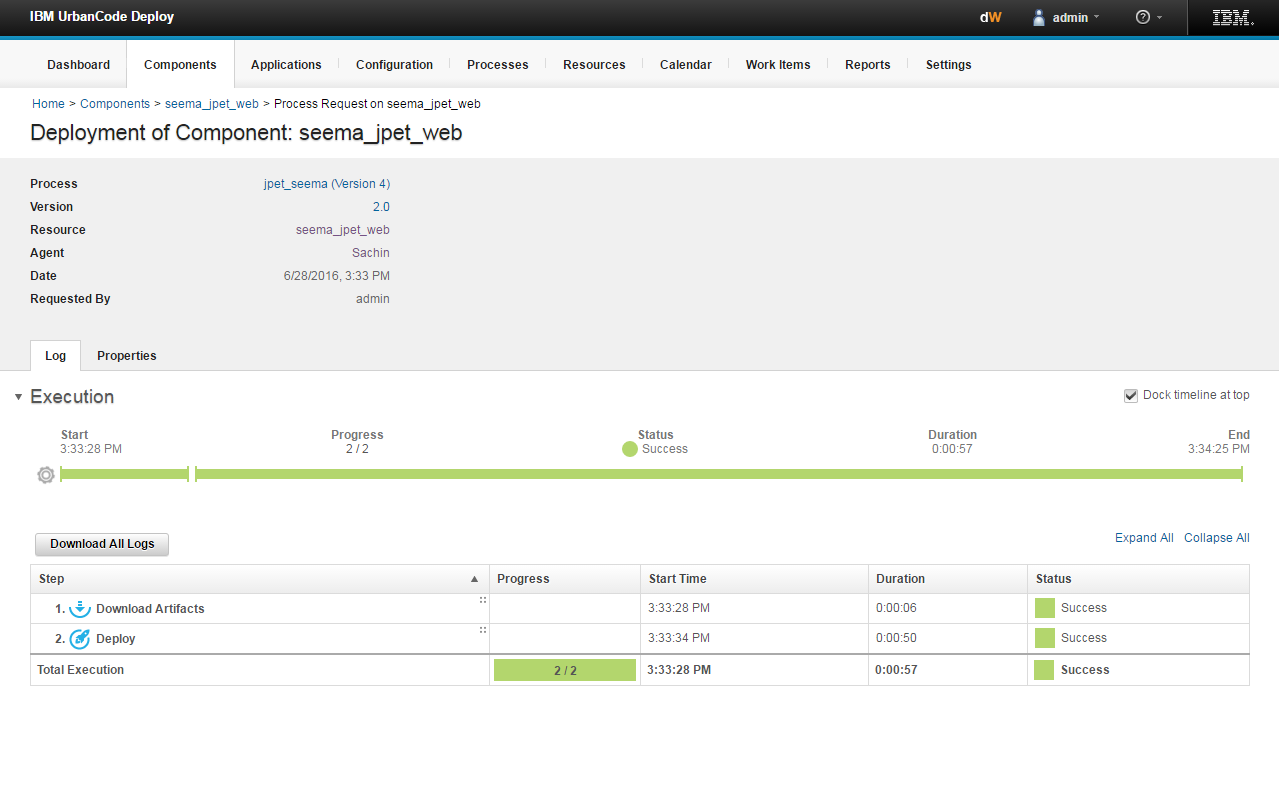
1. Go to the components in the applications and add your component.



Click on **Run Process (in versions select the appropriate version and click submit).**



**RUN PROCESS for deployment success.**



# **Deployment in web logic server.**

