Implementation -Sample war file to WAS using IBM UDeploy

Table of Contents

[1. Before you begin 3](#_Toc455062063)

[2. Agent 4](#_Toc455062064)

[3. Resource Creation 4](#_Toc455062065)

[A. Create Resource 4](#_Toc455062066)

[B. Map Agent to Resource 6](#_Toc455062067)

[C. Configure WebSphere cell 8](#_Toc455062068)

[4. Template Generation 11](#_Toc455062069)

[A. Create component for the template 11](#_Toc455062070)

[B. Map component 15](#_Toc455062071)

[5. Application creation (For Template) 17](#_Toc455062072)

[A. Create application 17](#_Toc455062073)

[B. Create process 17](#_Toc455062074)

[C. Environment Creation 19](#_Toc455062075)

[D. Map resources 20](#_Toc455062076)

[6. Deployment 22](#_Toc455062077)

[A. Create component 22](#_Toc455062078)

[B. Process creation on component level 23](#_Toc455062079)

[C. Map component 27](#_Toc455062080)

[D. Application Creation 28](#_Toc455062081)

[1) Create Application 28](#_Toc455062082)

[2) Map Component 29](#_Toc455062083)

[3) Create environment 29](#_Toc455062084)

[4) Add resources 31](#_Toc455062085)

[5) Create process (Application level) 33](#_Toc455062086)

# Before you begin

**The following setup and preparation is required.**

* Select or set up a WebSphere to use. The configuration template you create will use its configuration.
* Install an Urban Code Deploy agent on the appropriate WebSphere host. Note that you need the agent name during the procedures.

**Set up plug-ins in UrbanCode Deploy**

* Install the Middleware Configuration for WebSphere plug-in.
  + In UrbanCode Deploy, go to Settings > Automation Plugins.
  + Click Load Plugin.
  + In the dialog, click Choose File, then choose the MCWASPlugin-<version>.zip
  + Click Submit.
* Install the Application Deployment for WebSphere plug-in.
  + In UrbanCode Deploy, go to Settings > Automation Plugins.
  + Click Load Plugin.
  + In the dialog, click Choose File, then choose the ApplicationDeploymentForWebSphere-<version>.zip file.
  + Click Submit.
* Import the component template provided for Middleware Configuration for WebSphere. Do the following in UrbanCode Deploy. The <plugin\_home> directory is the top directory where you unzipped the downloaded UrbanCode Deploy plug-ins.
  + Click the Components tab.
  + Click the Templates tab.
  + Click Import Templates.
  + Select <plugin\_home>/imports/componenttemplates/Middleware Configuration for WebSphere.json. This template becomes the default template for the remaining process. You may use another template if you choose. It must already exist in UrbanCode Deploy.

# Agent

Configure the agent which is present on WebSphere server.

The agent should know where the WebSphere server application is installed on the machine so need to provide some configuration in that agent.

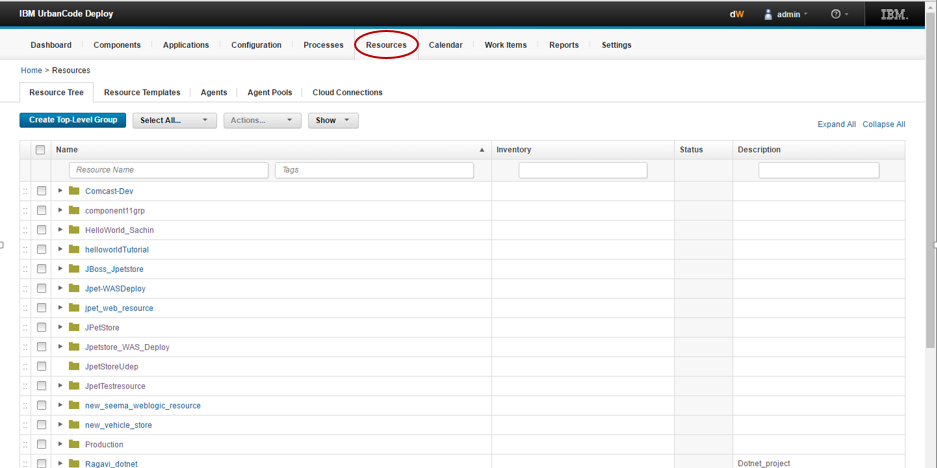
Select the agent which is installed on server machine 🡪 Go to configuration 🡪 agent property 🡪 add property

Should add here the new property “wasadmin.path” and in value pass the path of directory where the WebSphere is installed. (Up to wasadmin.sh file)

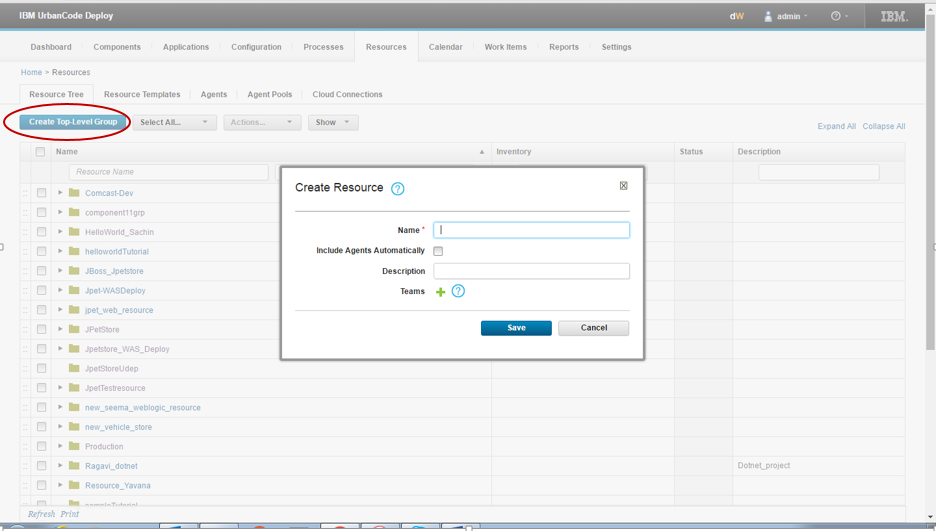
# Resource Creation

## Create Resource

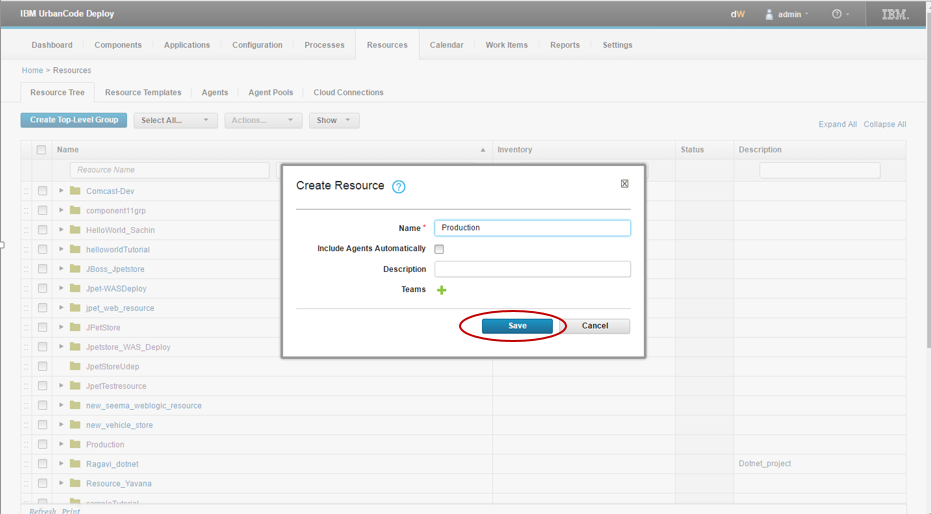
Click on resource tab



Click on create top level group

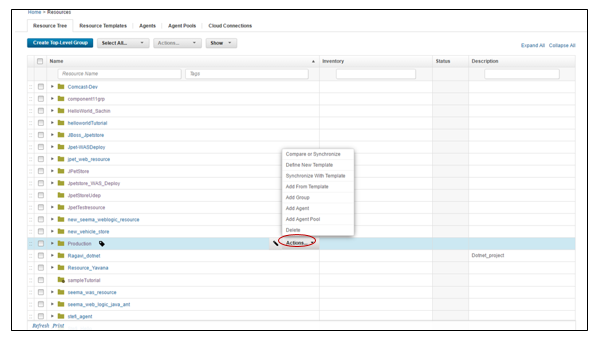


Click on create resource.

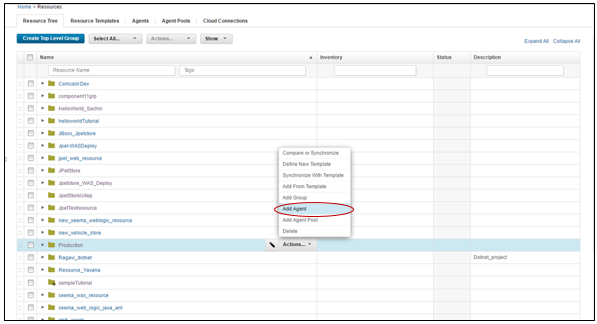


## Map Agent to Resource

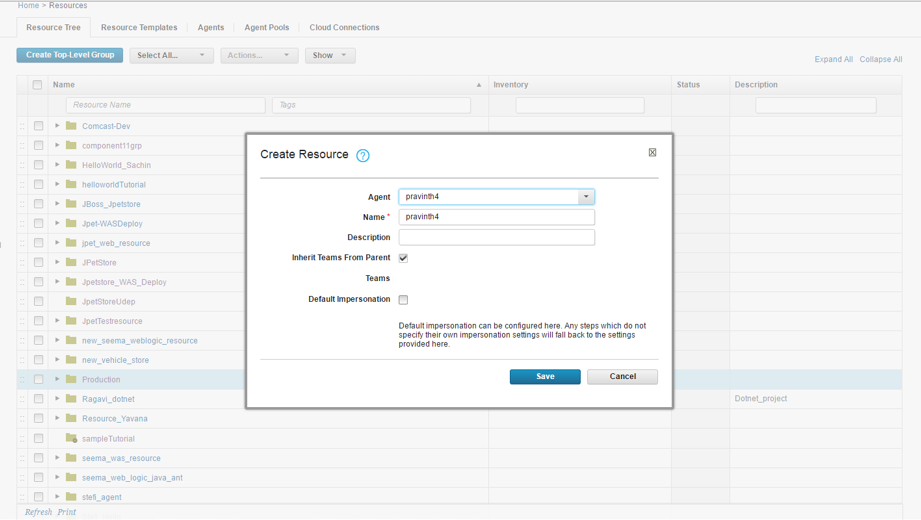
Go to the resource tab select your resource and click on action



Click on add agent and map agent with resource



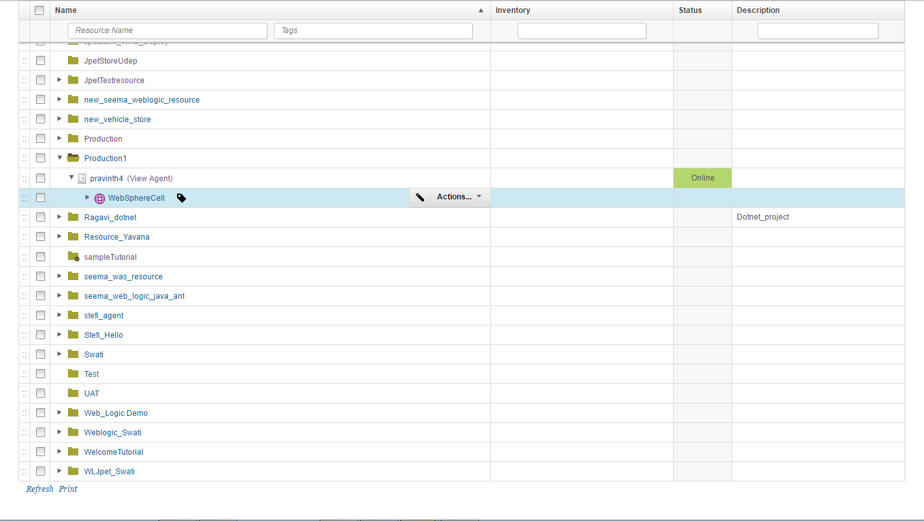
While mapping that agent with resource, should choose the agent which is present in the machine where the WebSphere server is installed.



Refresh it by clicking on refresh tag.

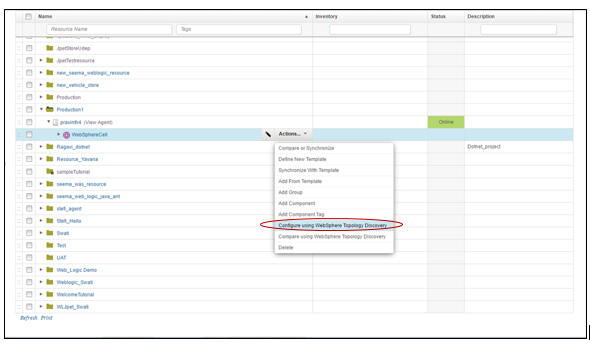


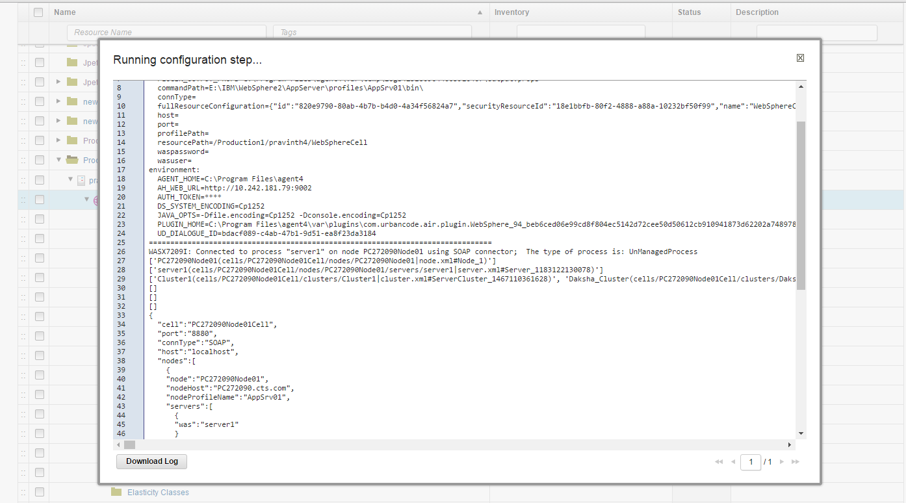
Now you will get web sphere cell.



## Configure WebSphere cell

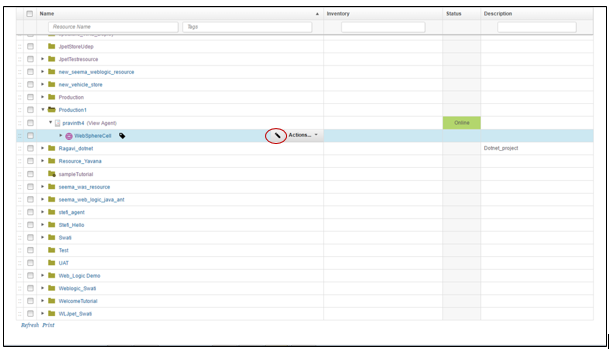
Now click on action tab 🡪 Select configure using WebSphere topology

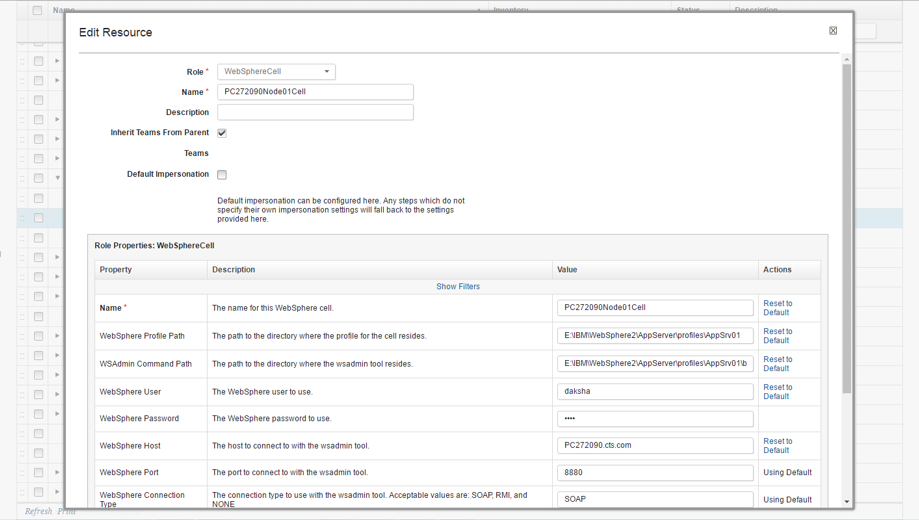




It will configure and show as connected to server.

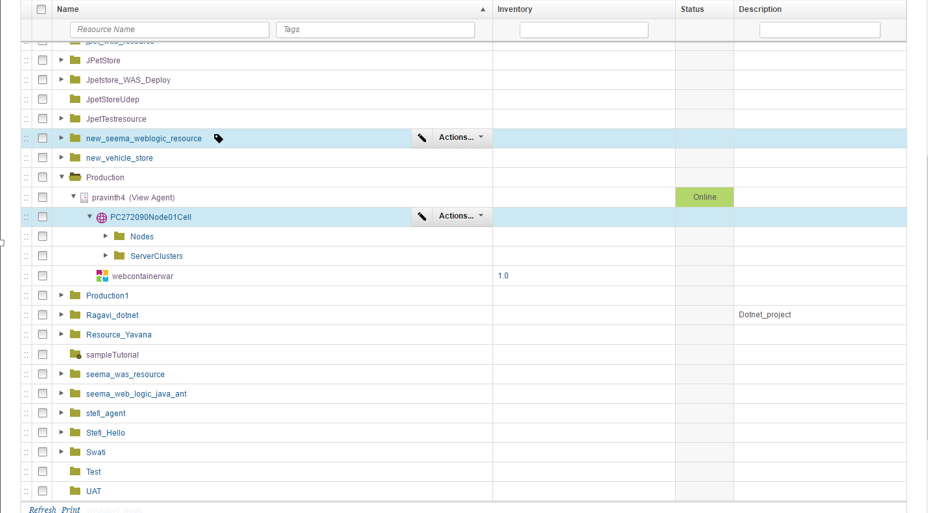
Click on edit to configure the properties of WebSphere cell.





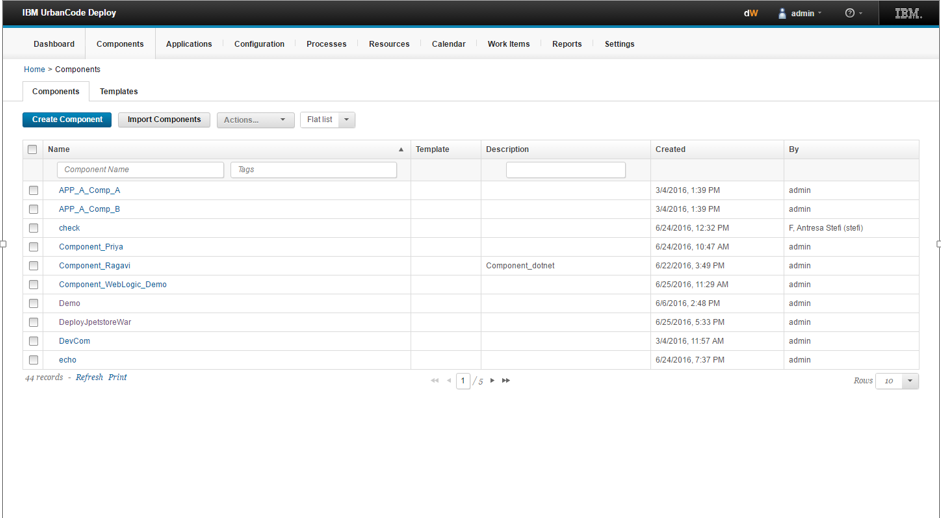
* + 1. The Name of WebSphere cell will appear automatically when you configure it with WebSphere topology.
    2. WebSphere profile path – give the path of root location of WebSphere server.
    3. WSAdmin command path - give the path of root location of WebSphere server. (up to bin folder)
    4. User – provide the username.
    5. Password – provide the password.
    6. WebSphere host – Provide the hostname.
    7. Port – provide the port to connect with WSAdmin tool. (by default it is 8880)
    8. Then click on save to save this configuration.

You will get that node and server cluster.



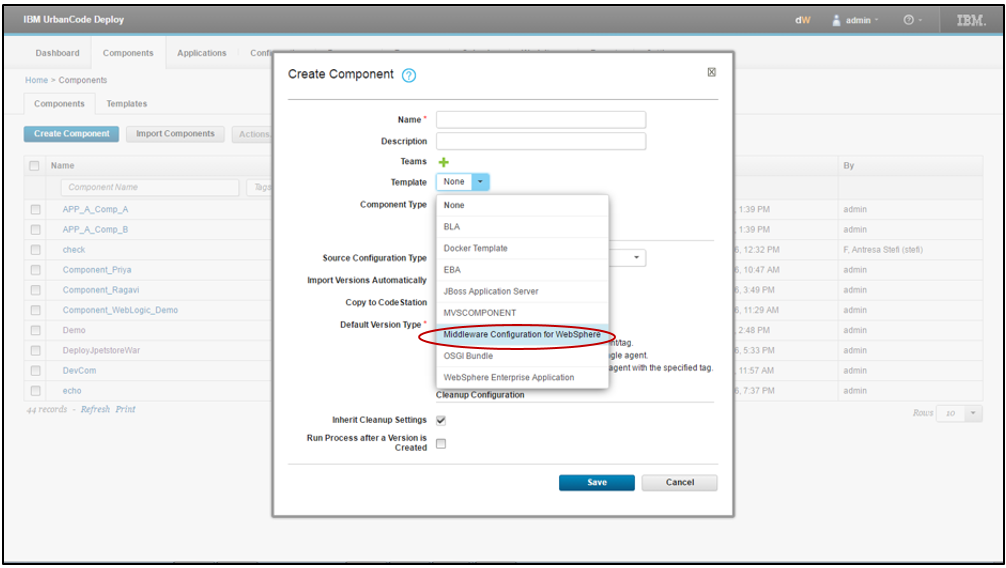
# Template Generation

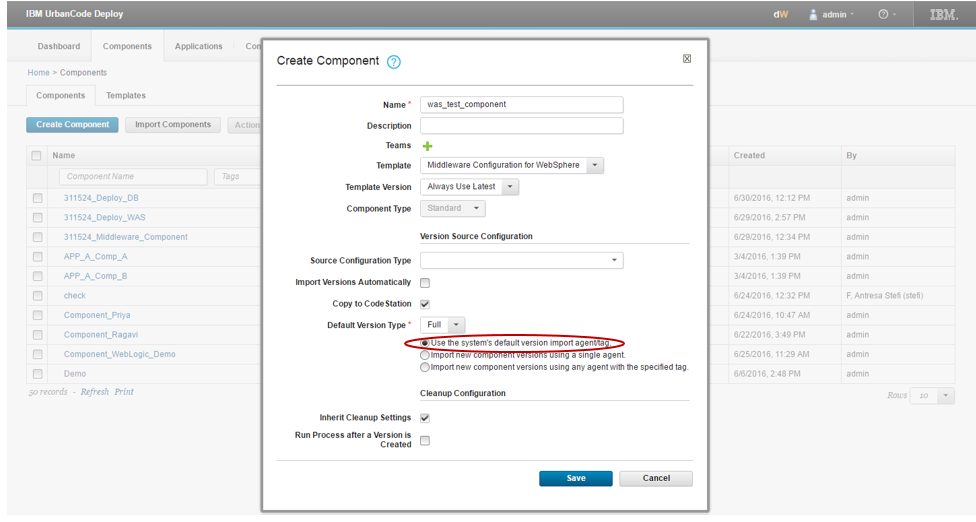
## Create component for the template



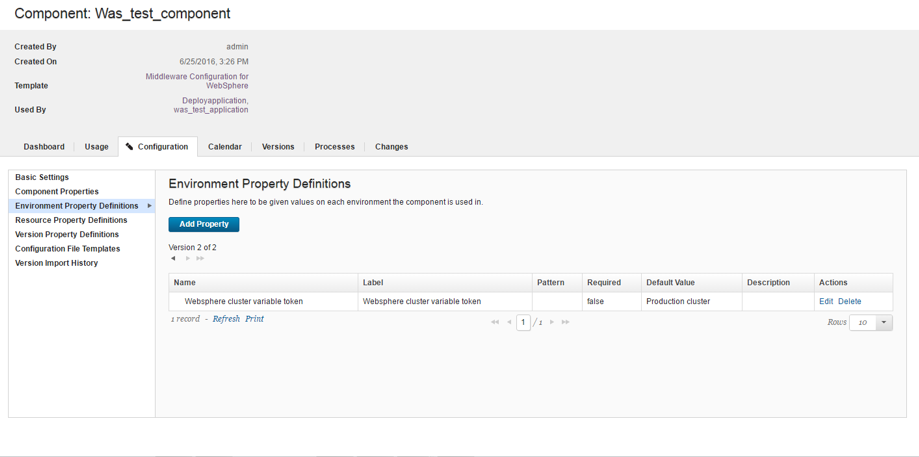
While creating that component fil following fields –

1. Name of component.
2. Template type - middleware configuration for WebSphere.
3. Choose the Use the system default version import agent/tag.

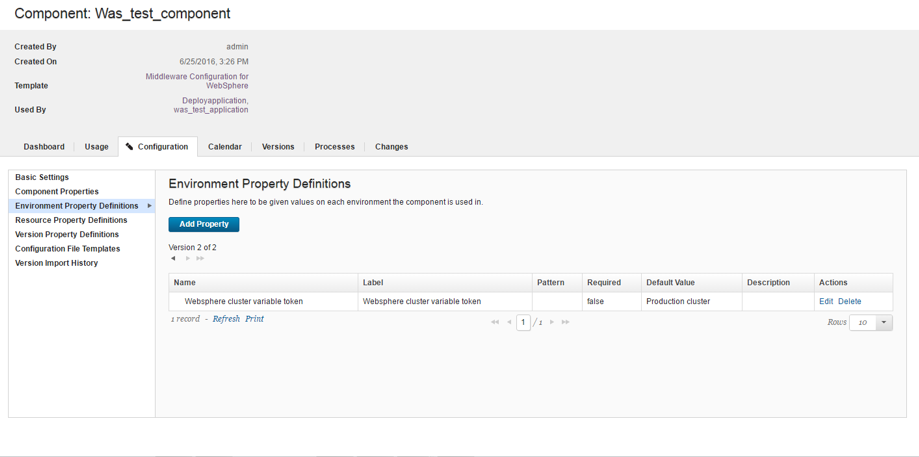


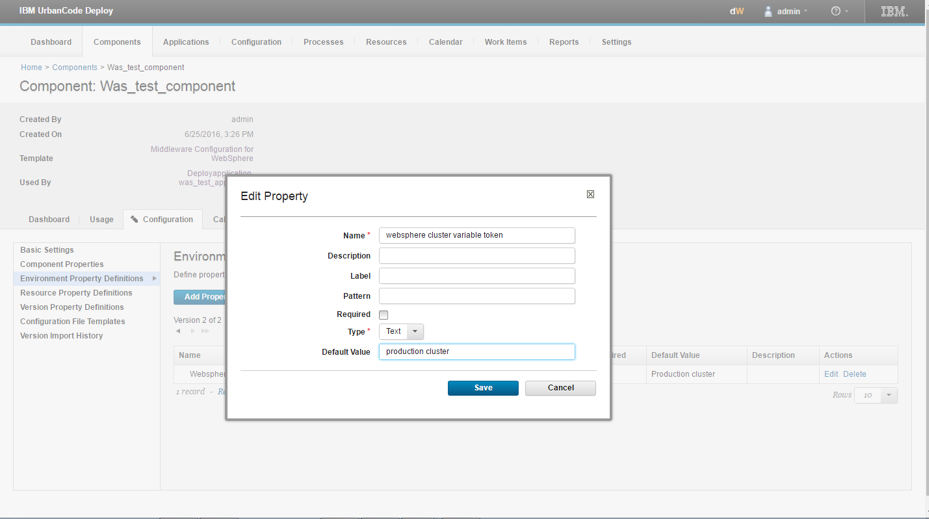


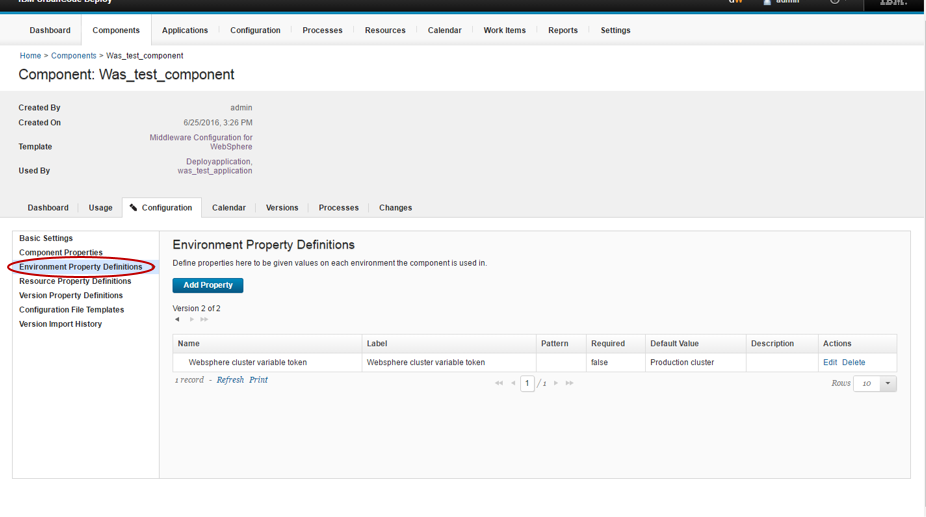
Go to component 🡪 Click on created component 🡪Click on configuration



Set Environment property definition.

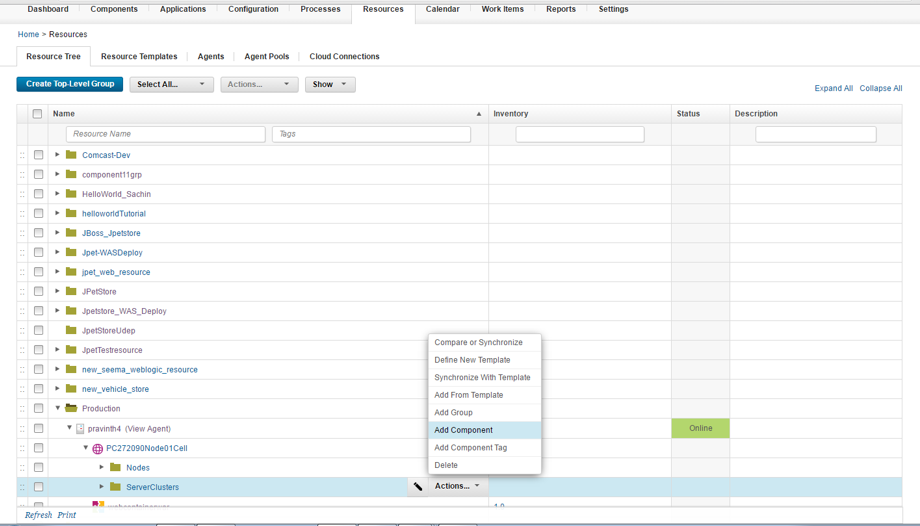


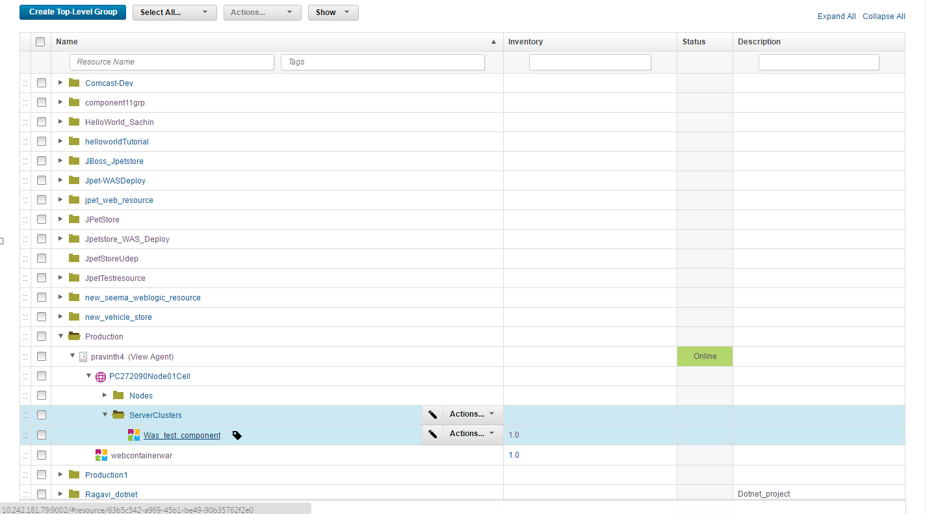




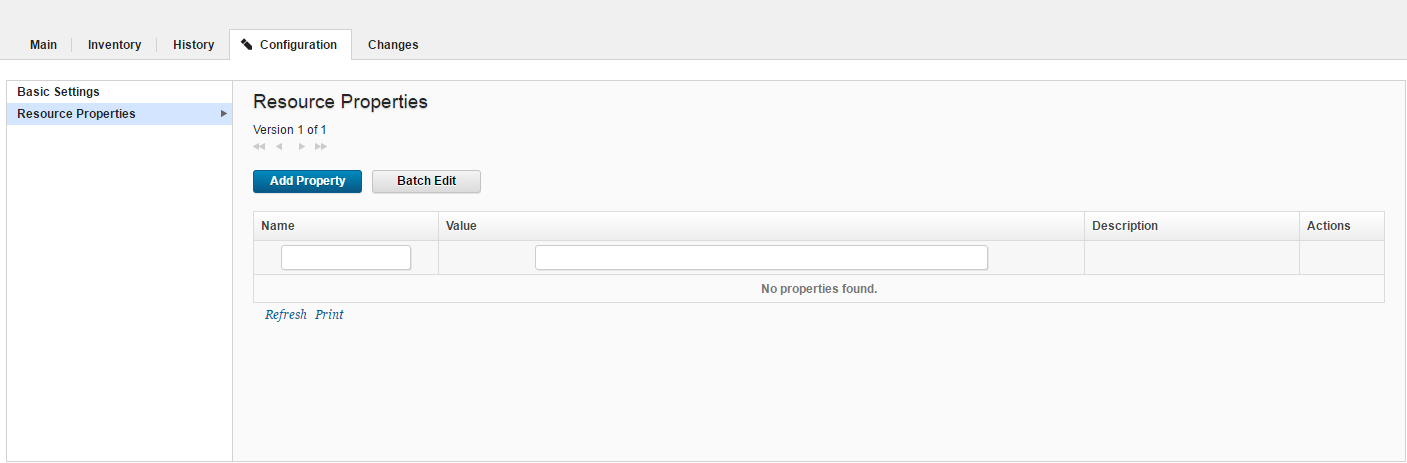
## Map component

Go to resource 🡪 Click on created resource





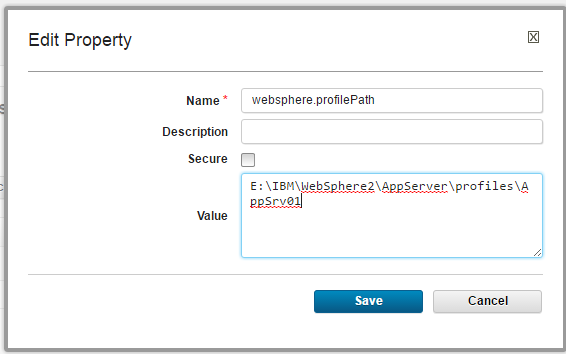
Go to the resource tab 🡪 Select resource 🡪 Go to configuration 🡪 Add resource property

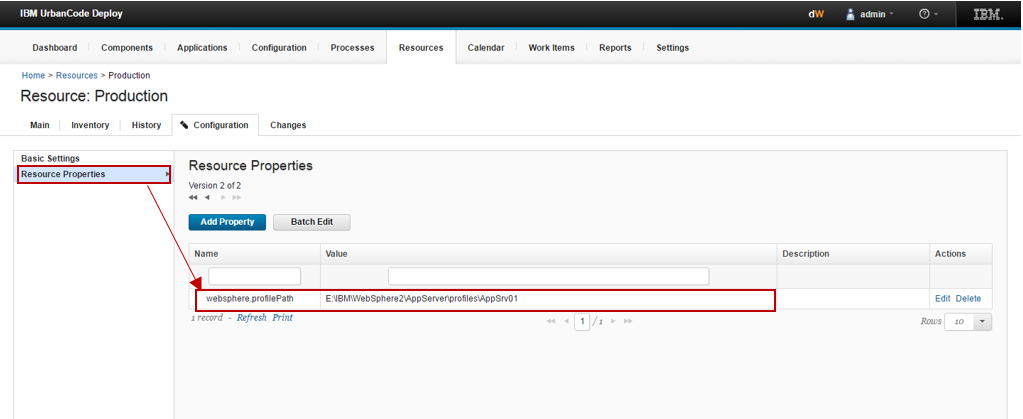


**Edit property**

Set the name – websphere.profile.path

Value – The path of location where WebSphere server is present.

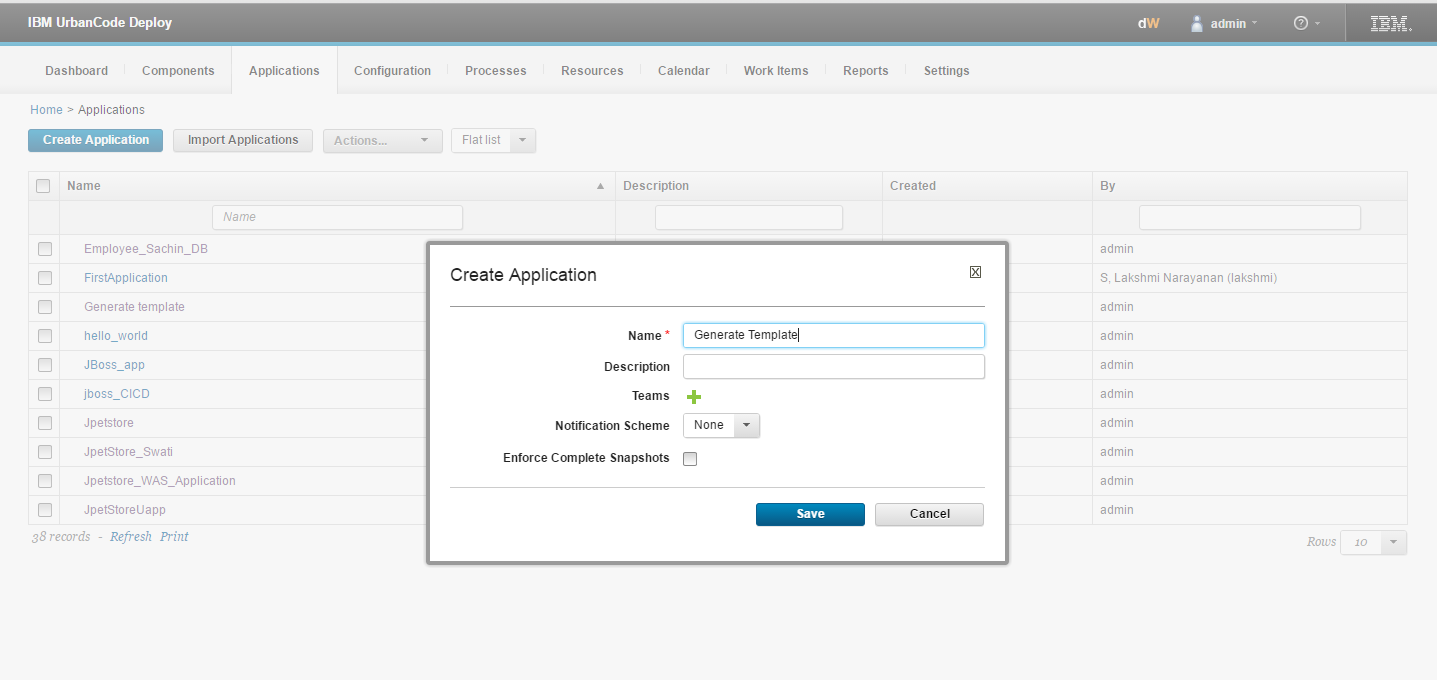




# Application creation (For Template)

## Create application

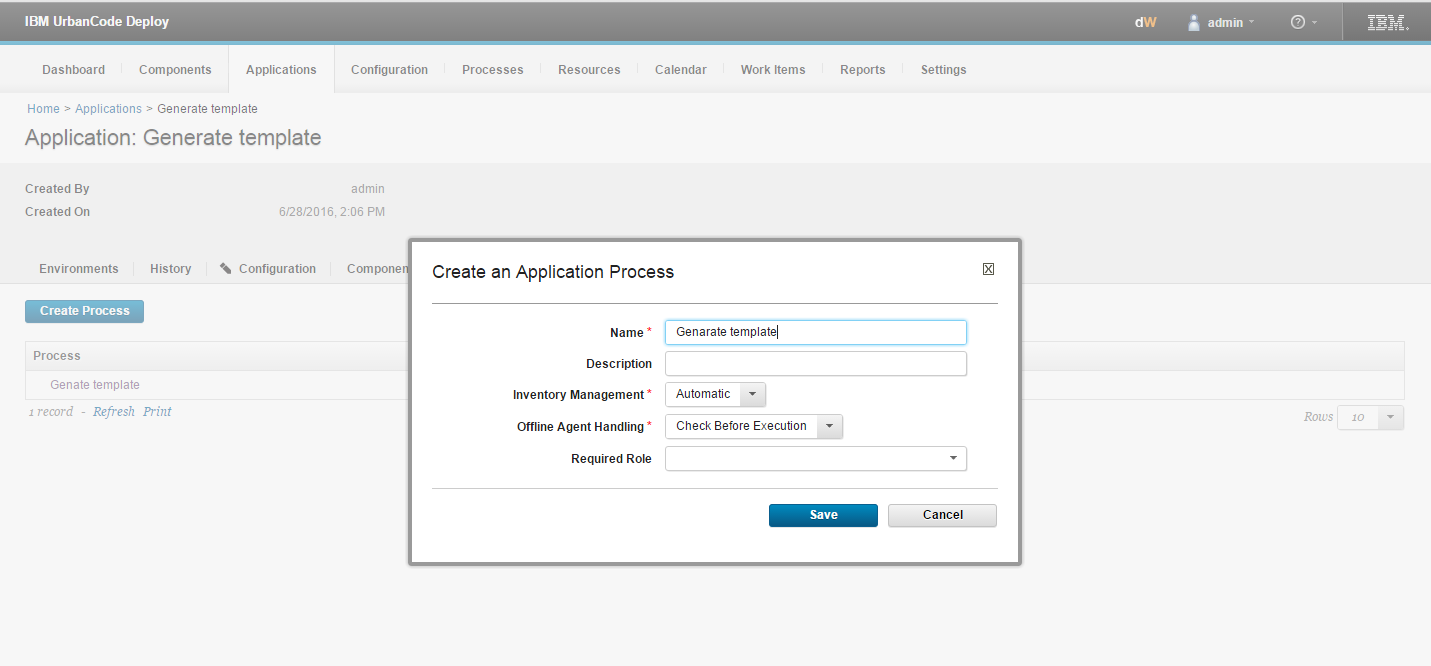
Go to application🡪 Create application

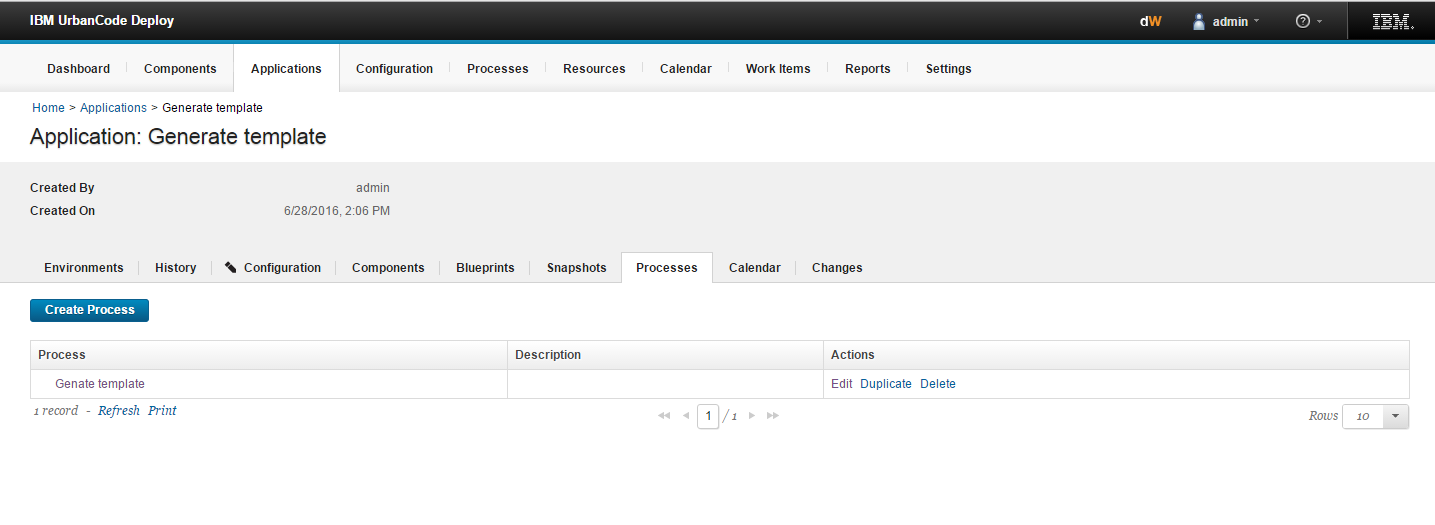


Give the name of application and save it.

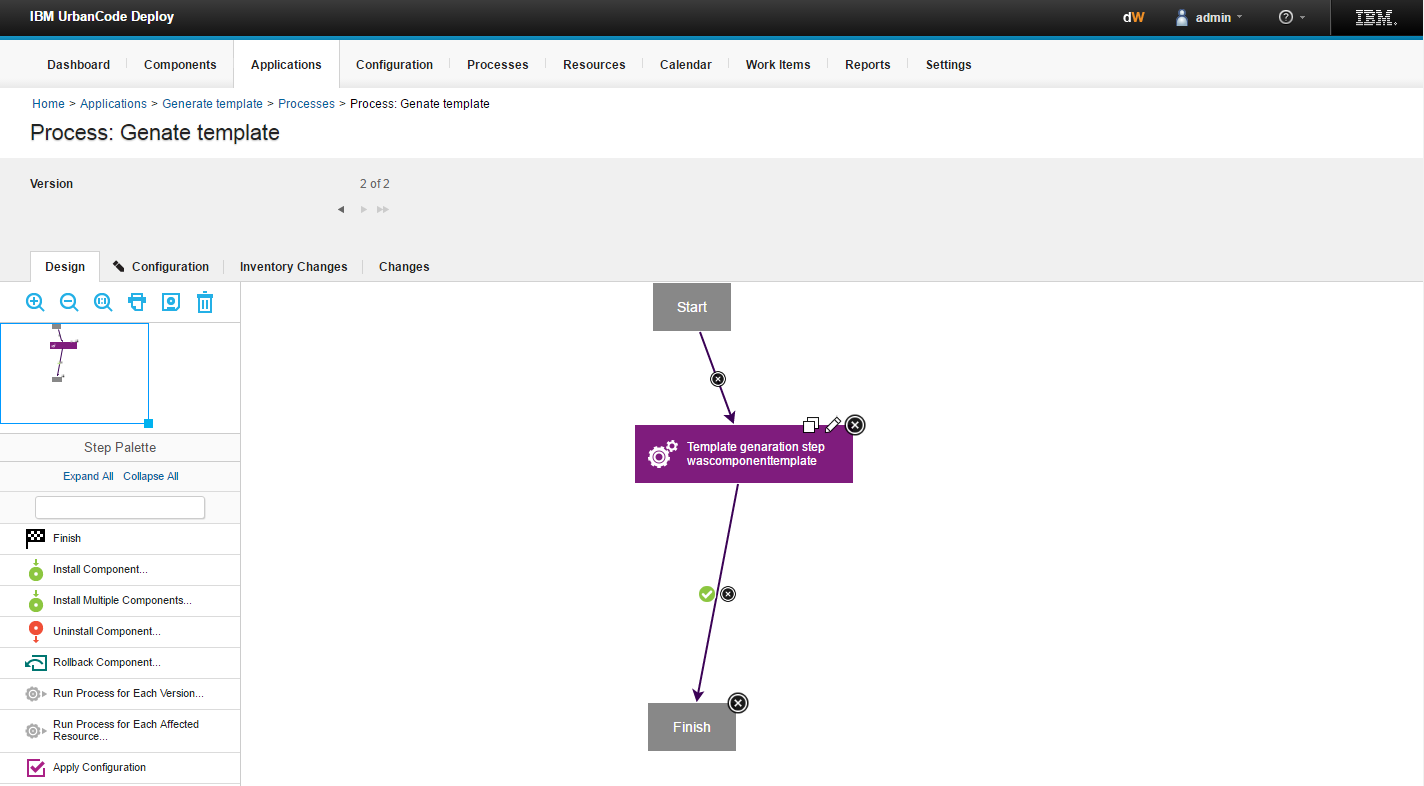
## Create process

Now click on application 🡪 Process 🡪 Create process





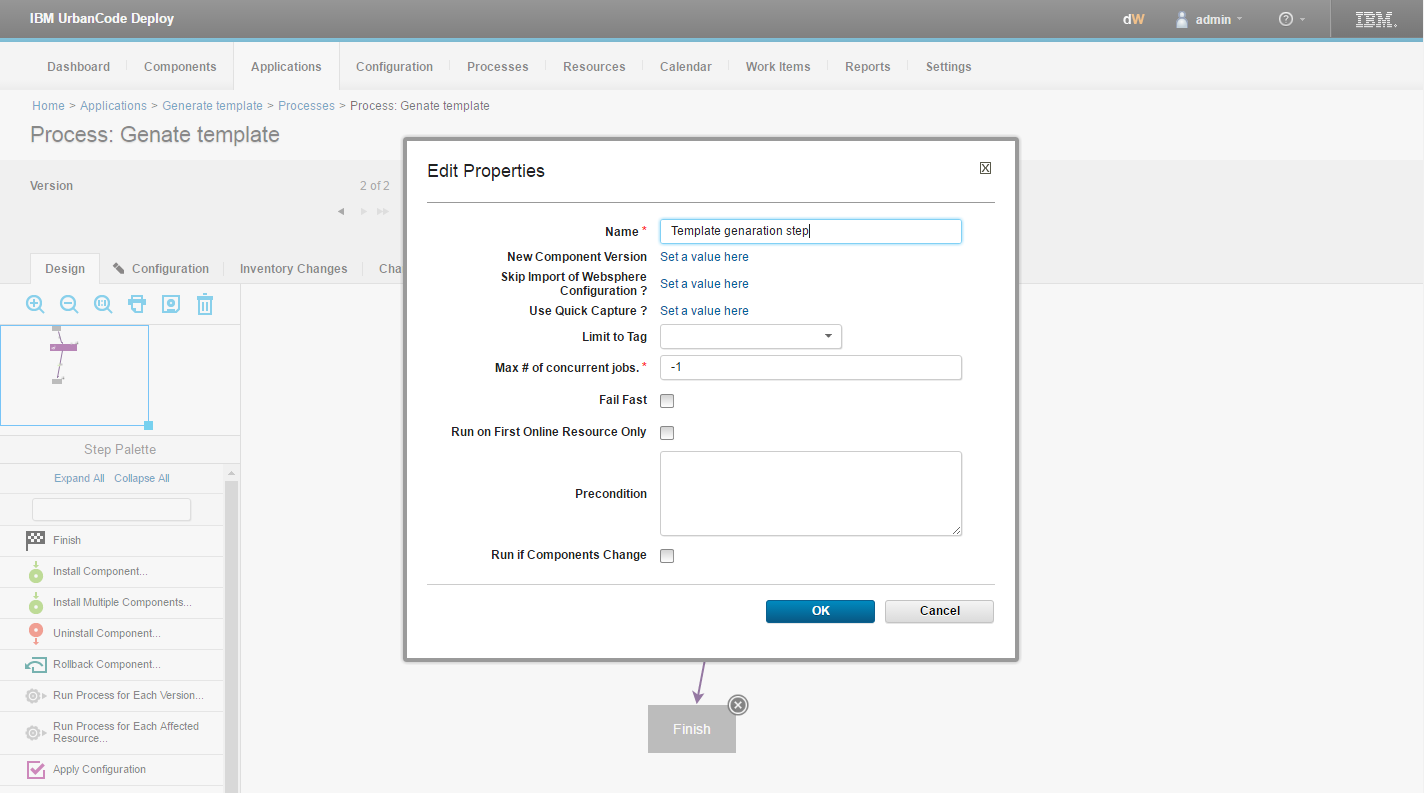
Click on process 🡪 Select your component from step palette 🡪under that select generate template step



**Edit properties –**

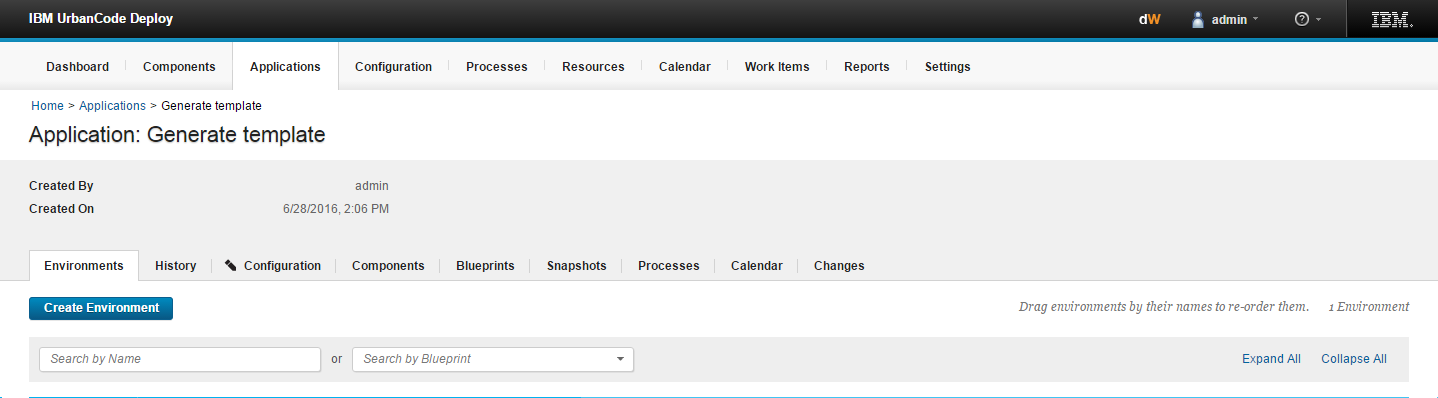
1. Name – Give name of step

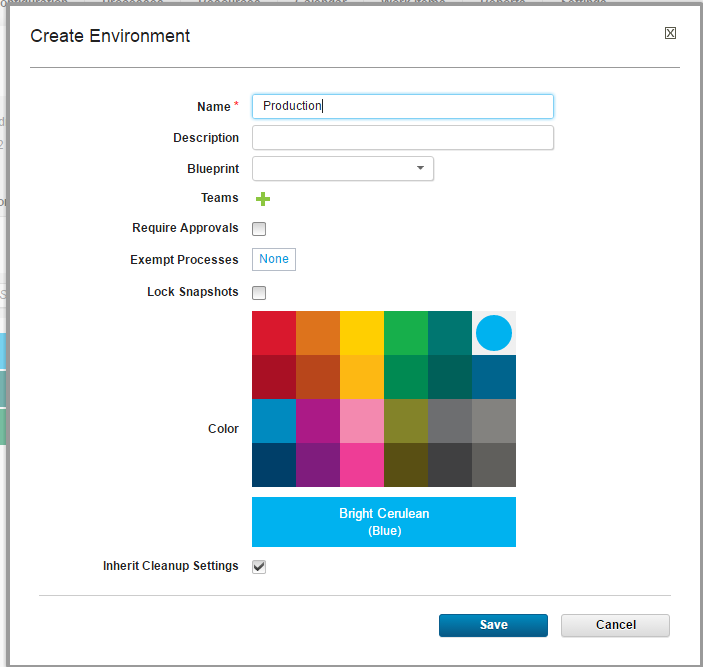
Save it.



## Environment Creation

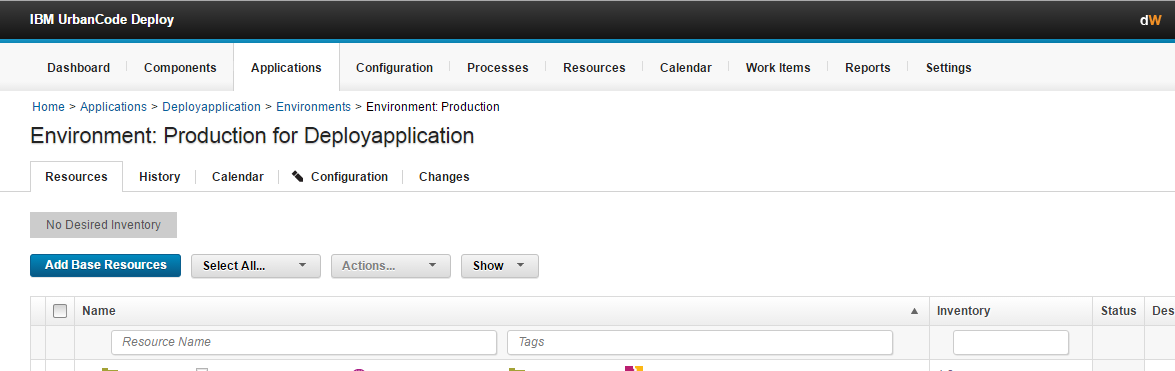
Go to application tab 🡪 Select your application 🡪 Create environment

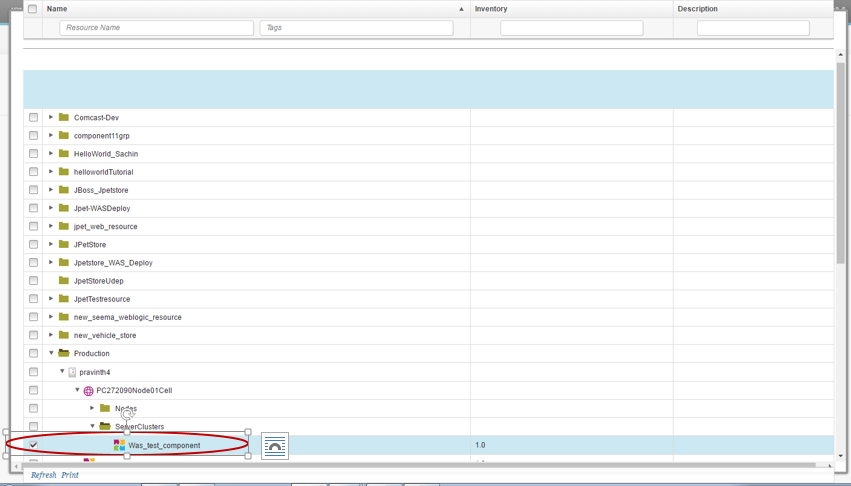


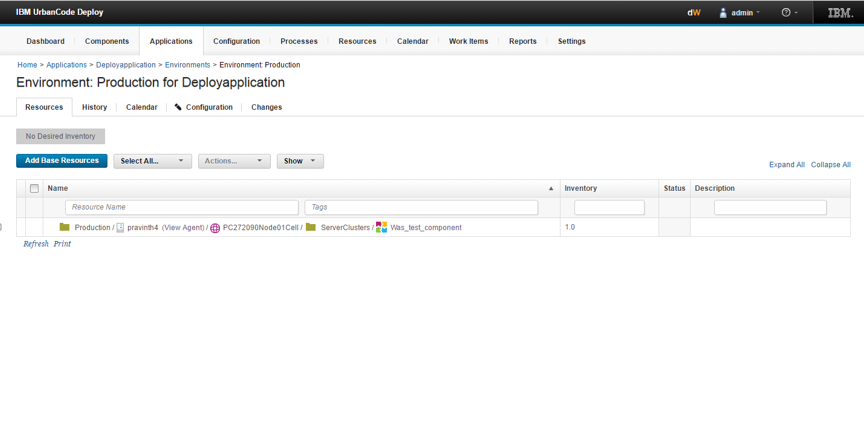


## Map resources

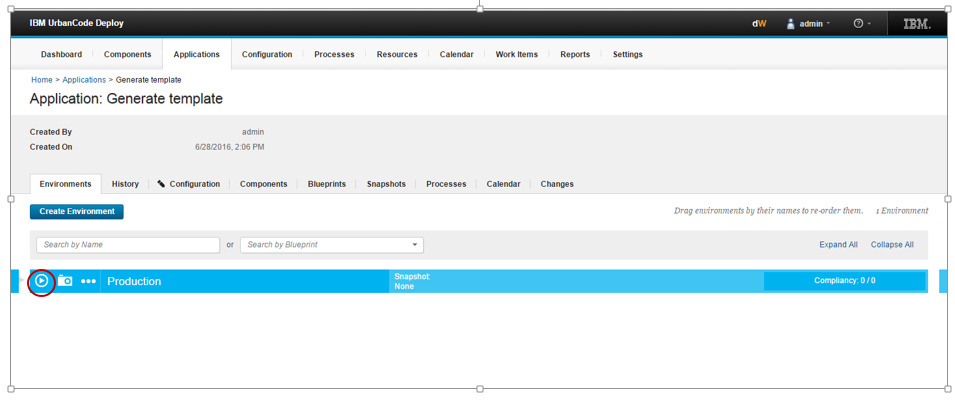
Click on the environment 🡪 Click on resource 🡪Add base resource



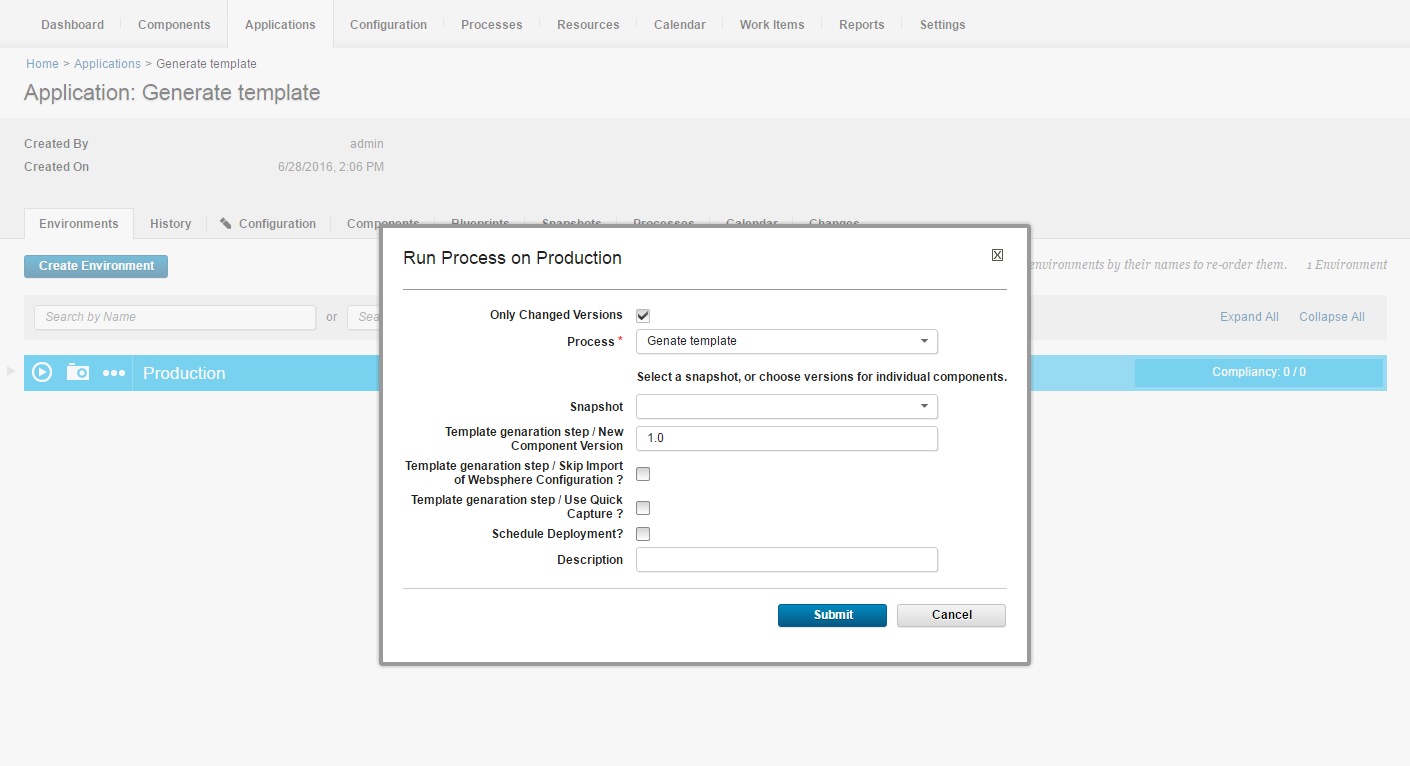


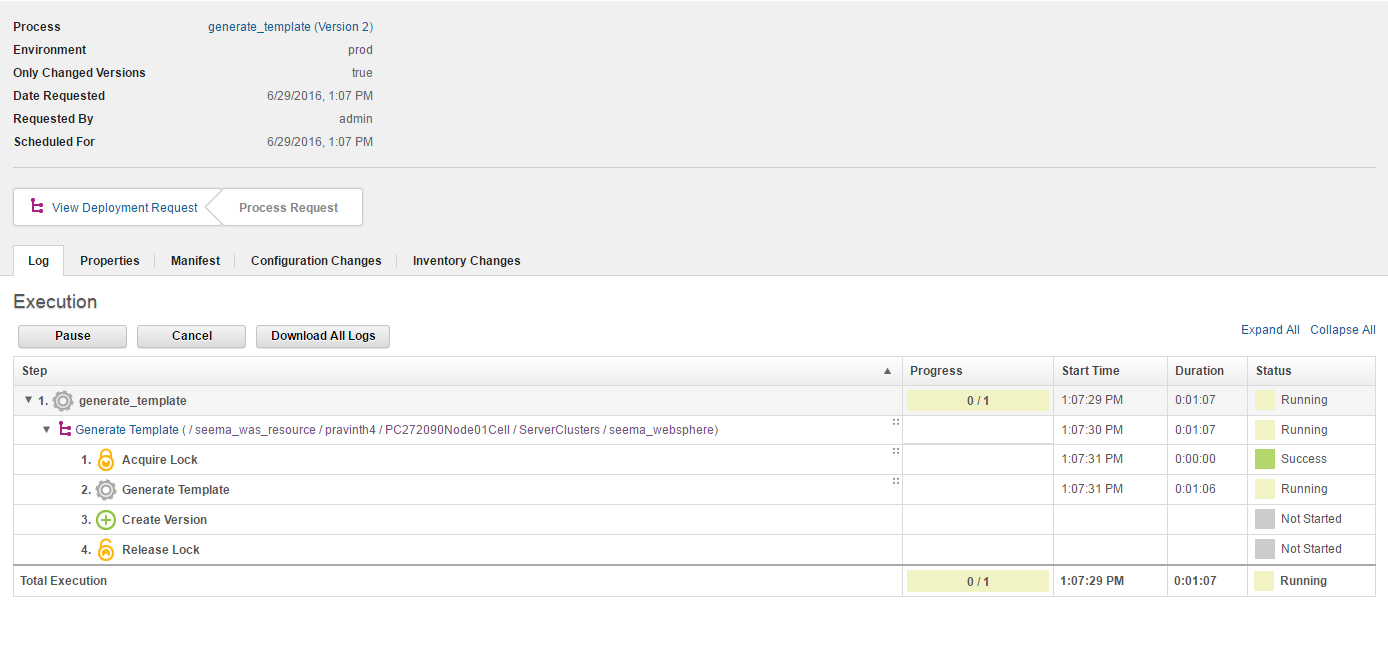


Again go to the application and run that



Uncheck the Template generation step / Use Quick Capture

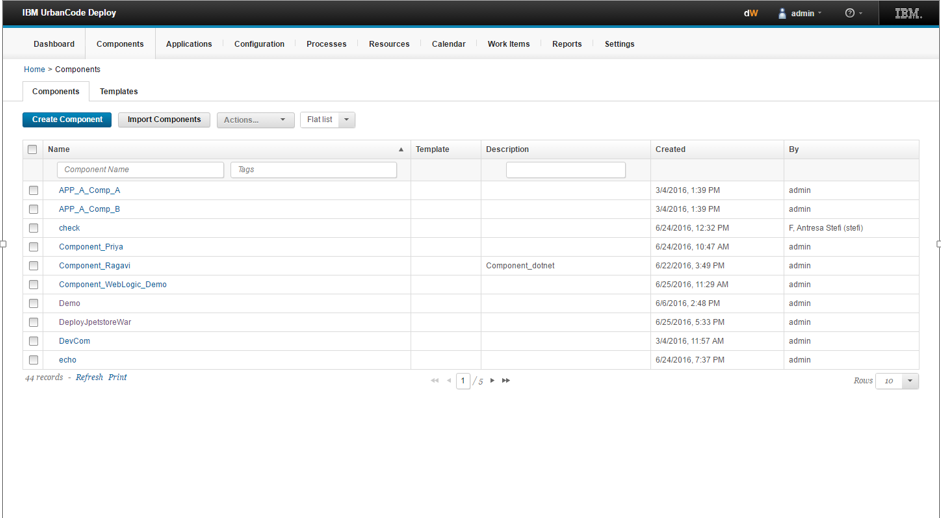




# Deployment

## Create component

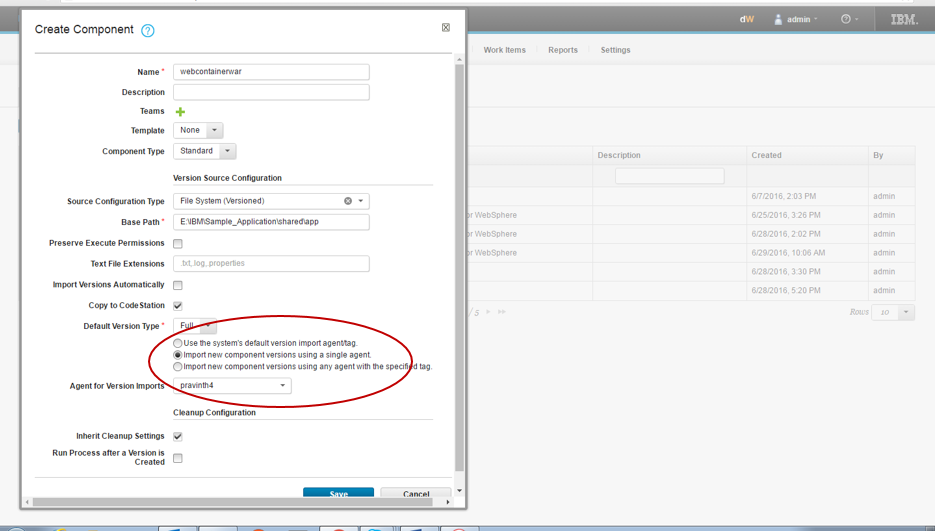
Create the component which represent the War file.

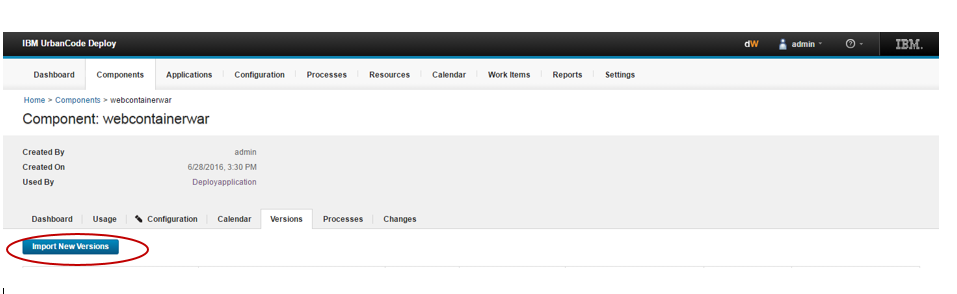


In Base path pass the path of location where war file of application is present.

In create component click on import new component versions using single agent.

In agent for import version select the agent which is present in WebSphere server.



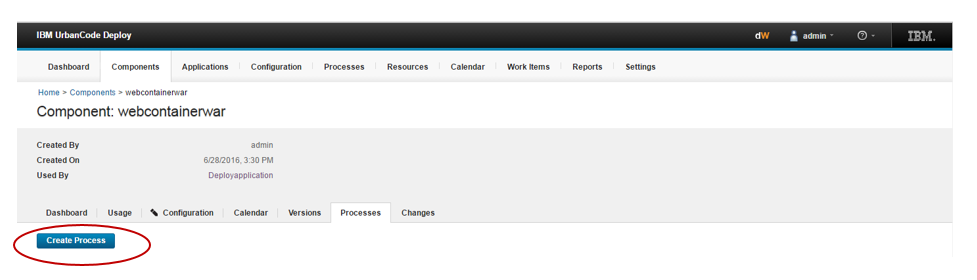


Click on component 🡪 version🡪 Import version 🡪 you will get the version here.

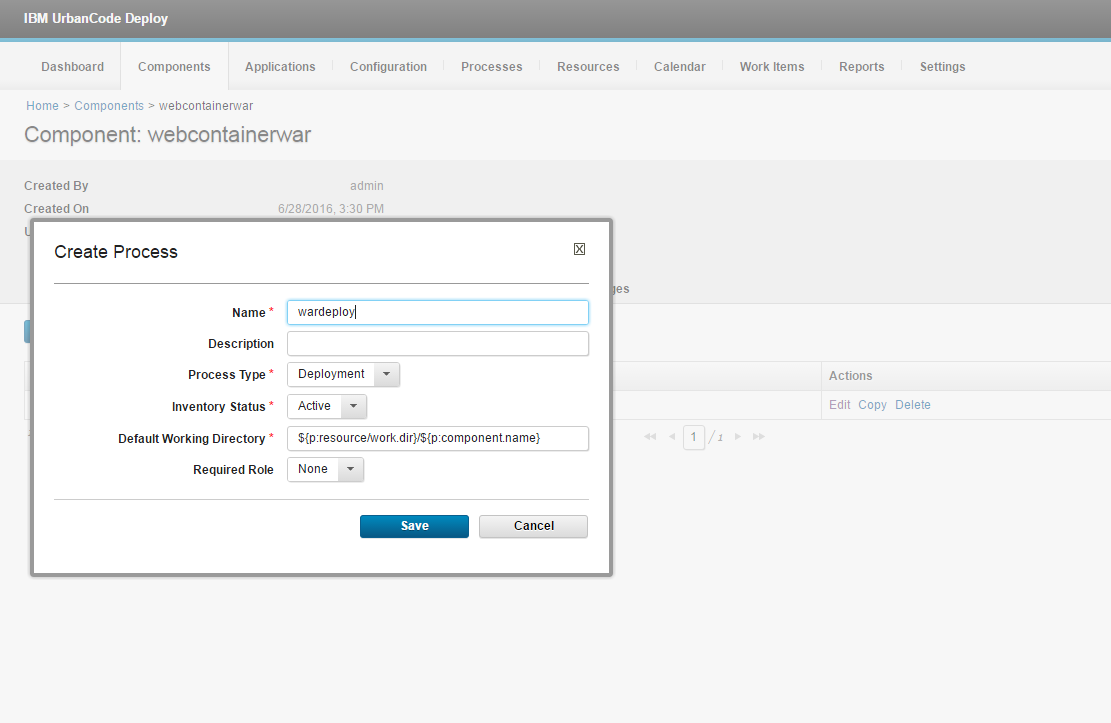


## Process creation on component level

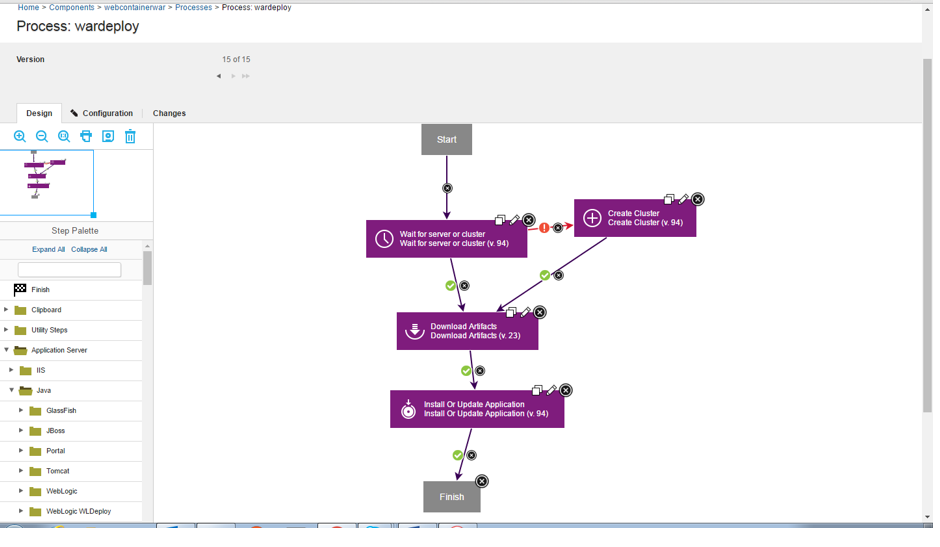
Click on process 🡪 Create process



In create process add the name of process and save it.

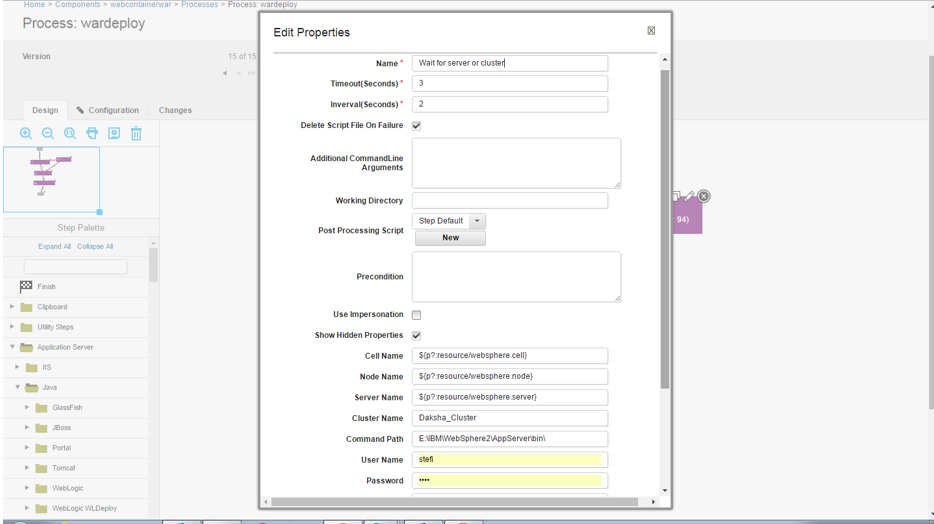


Click on process name you will get the design of process.



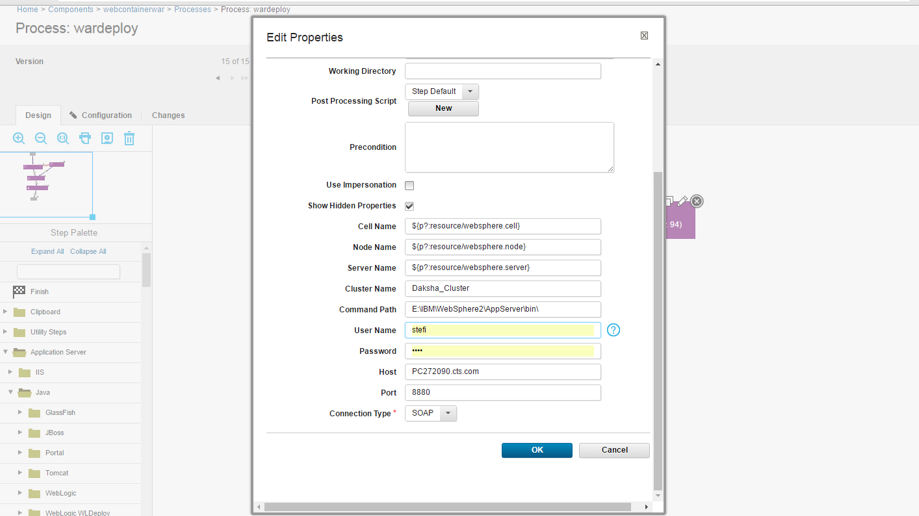
**Edit properties**

1. Wait for server or cluster

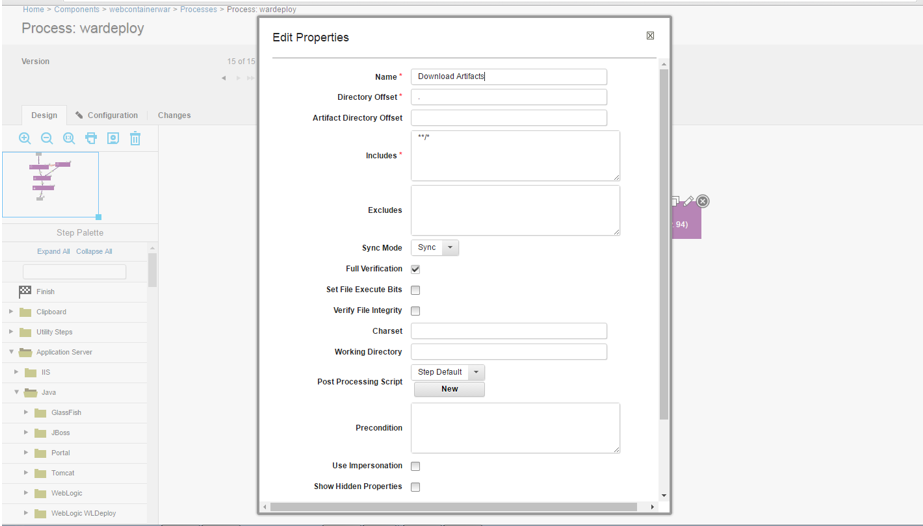


1. Name – add the name of step.
2. Time out – pass the time out in second.
3. Interval - pass the time out in second.

**Set hidden properties**

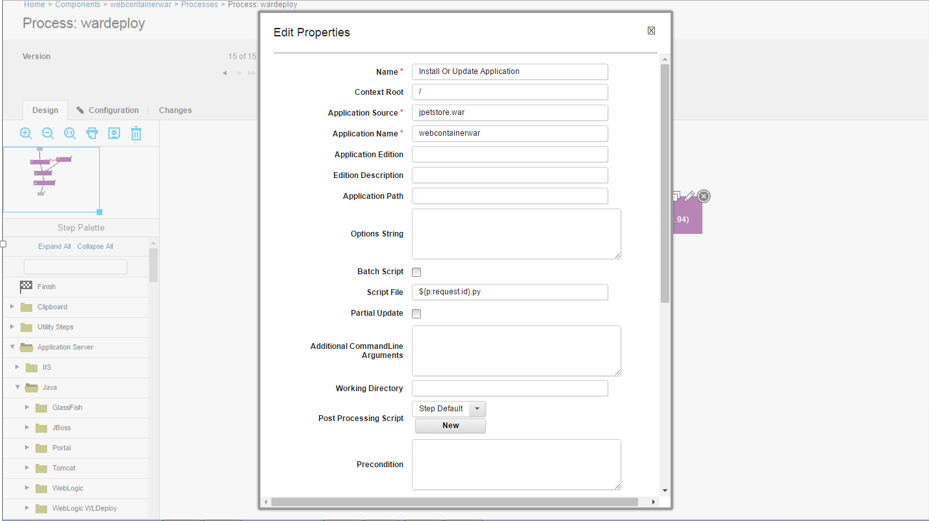


1. Cluster name - pass the name of cluster.
2. Command path – the path of root where server is installed.(upto bin folder)
3. User name - pass user name.
4. Password - pass the password.
5. Host - Pass the host name.
6. Port - pass the port number.
7. Download artifact



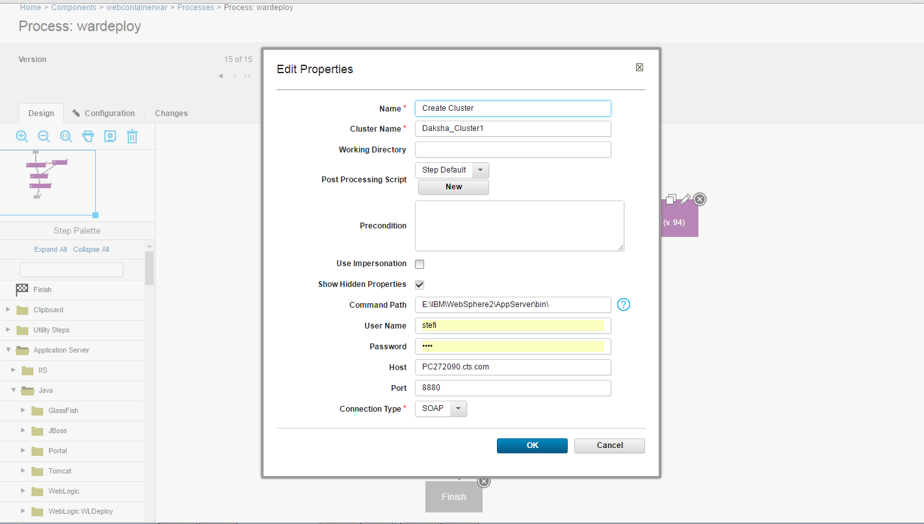
Only give the Name here and keep rest of properties as it is and save it.

1. Install or update application



The properties -

1. Name - Pass the name of step.
2. Context root - It is where you want to store your application.
3. Application source - Name of war file.
4. Application name - Pass the name of application.
5. Create cluster

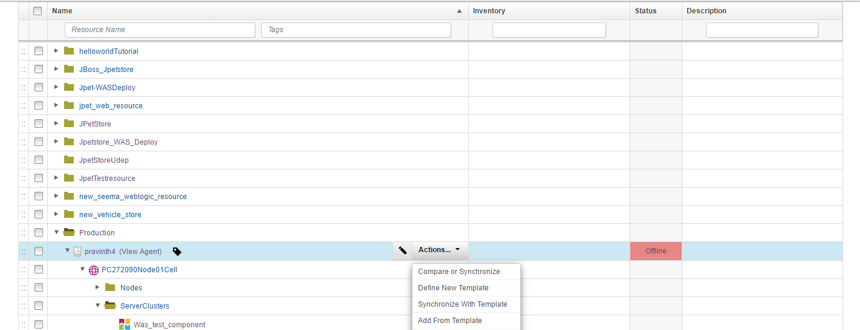


Properties -

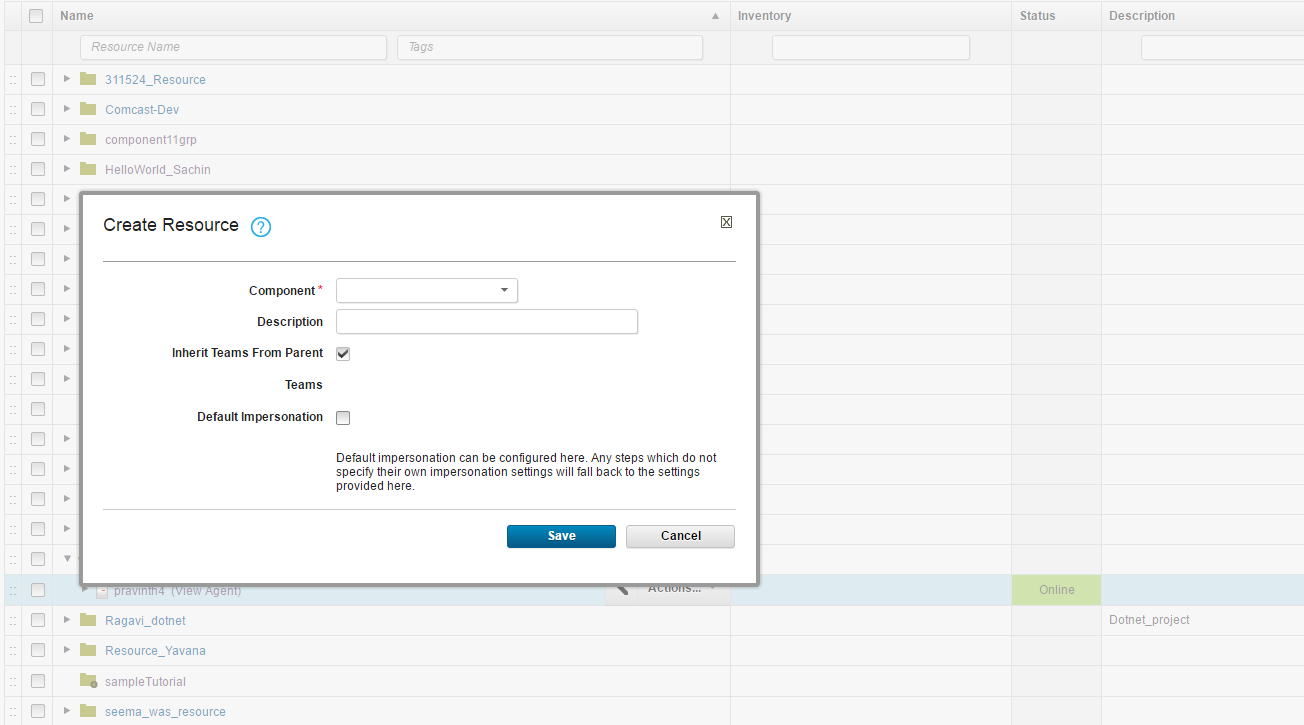
1. Name - Pass the name of step.
2. Cluster Name - Pass the name of cluster.
3. Command path - Pass the path of root location of websphere server.
4. User name - Pass the user name.
5. Password - Pass the password.
6. Pass the host and port .
7. Save it.

## Map component

Go to resource 🡪 Select your resource 🡪 Click on agent 🡪Action



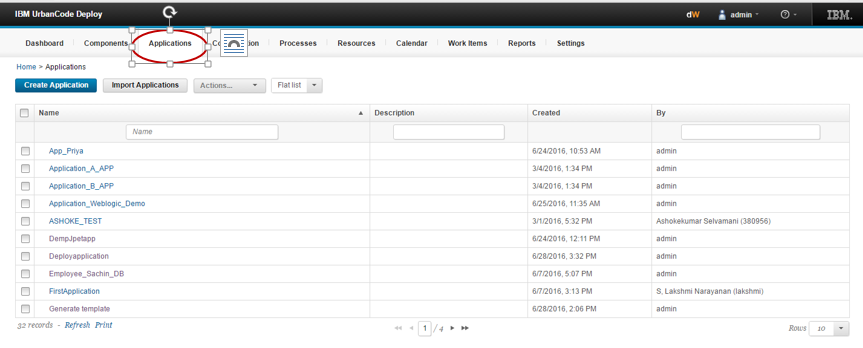
Add component 🡪 add your second component here



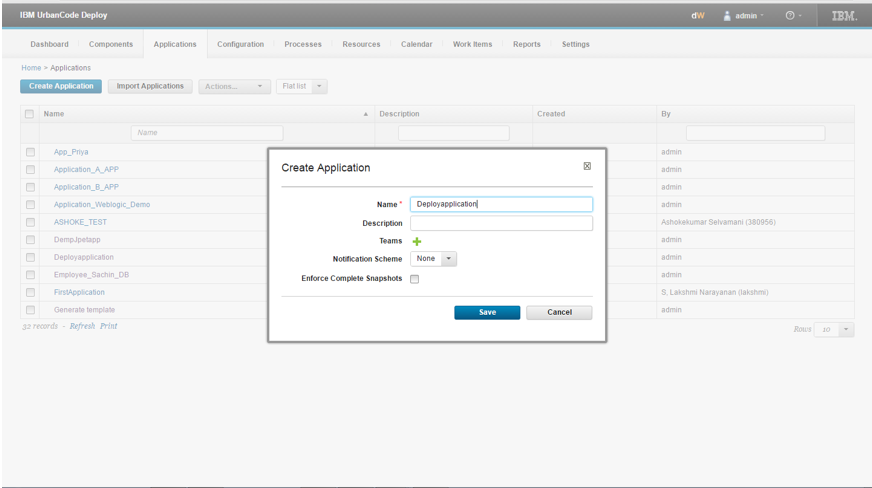
## Application Creation

### Create Application

Create the application for deployment.

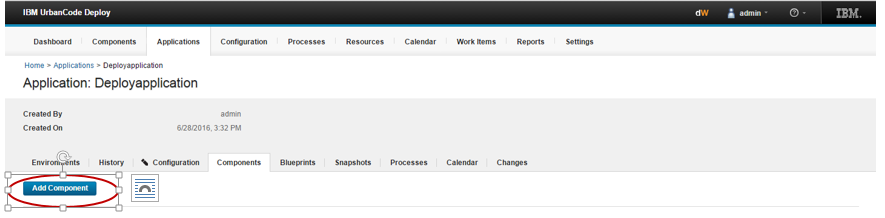


Give the name of application and save it.

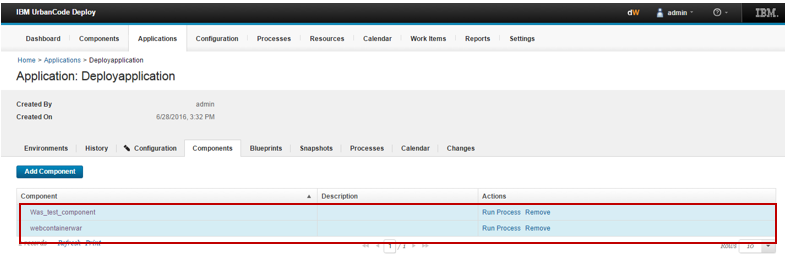


### Map Component

Click on application name 🡪 Component tab 🡪 Add component

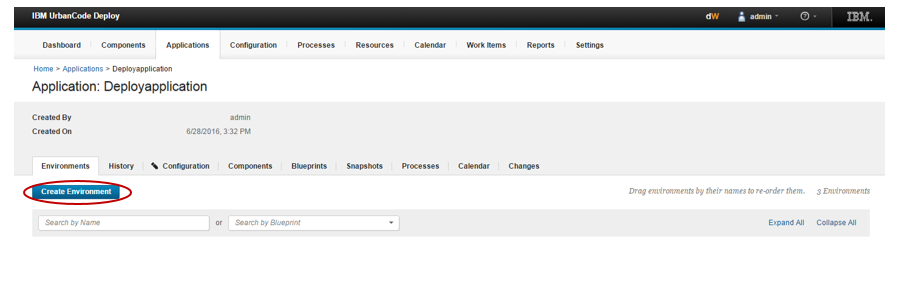


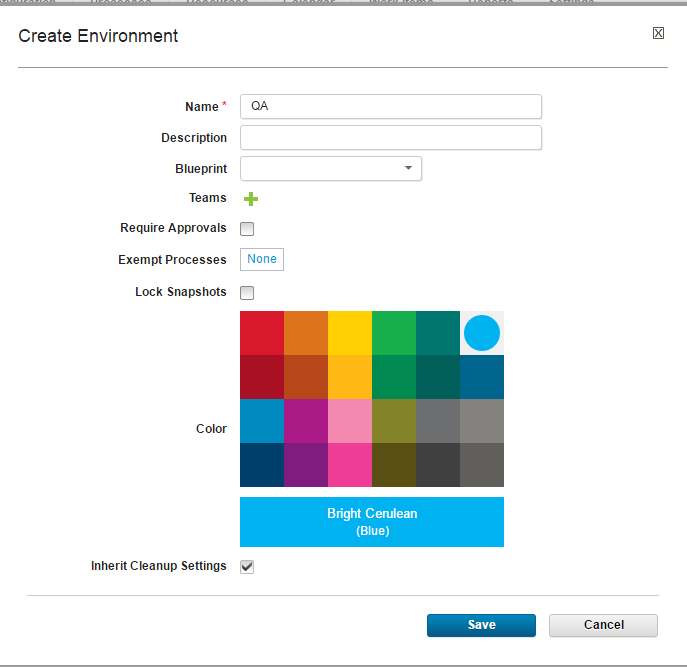
Select the created component from drop down and save it.

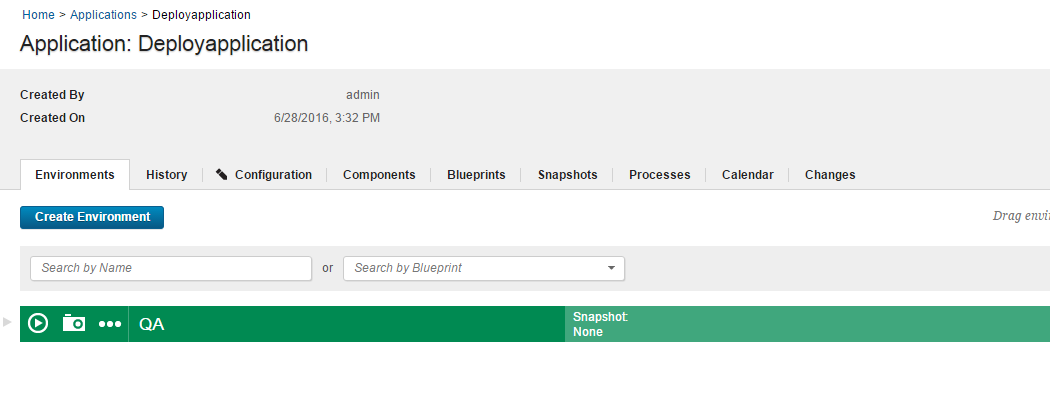


### Create environment

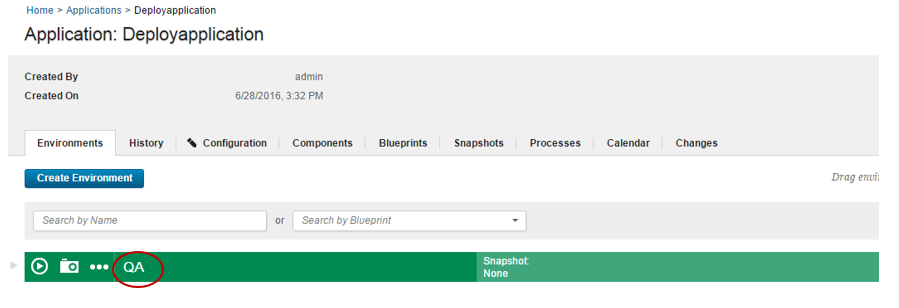
Click on create environment tab





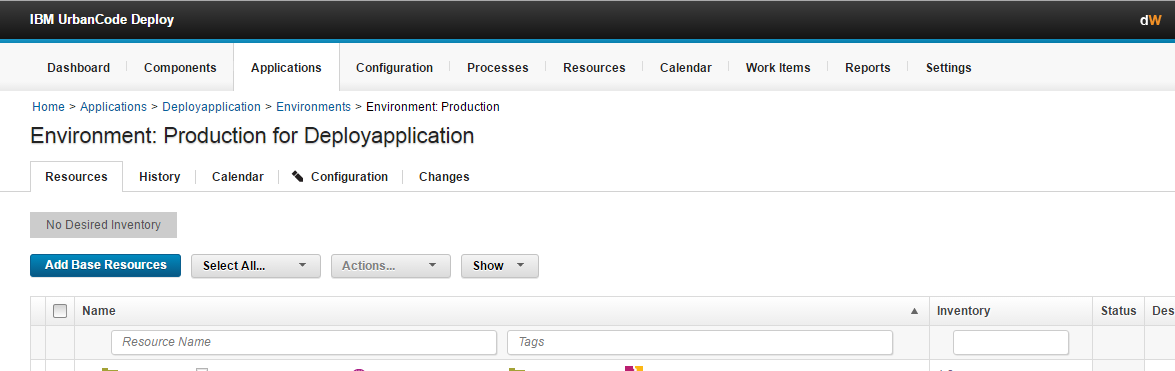


Click on environment

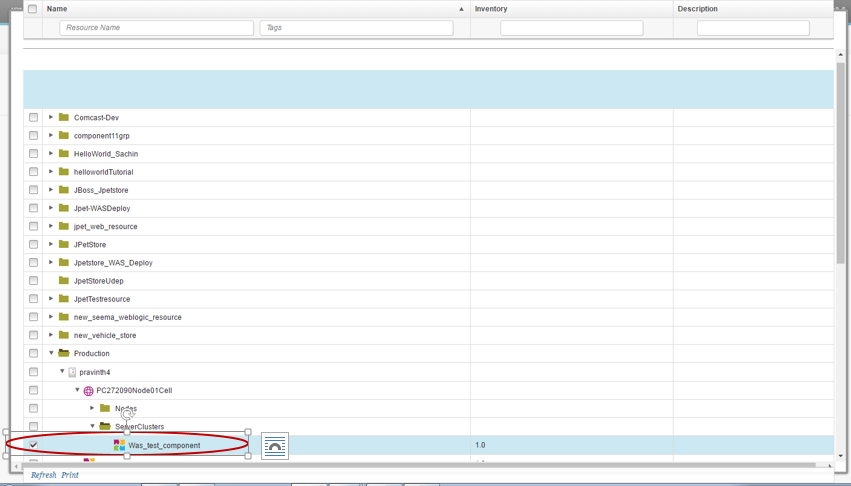


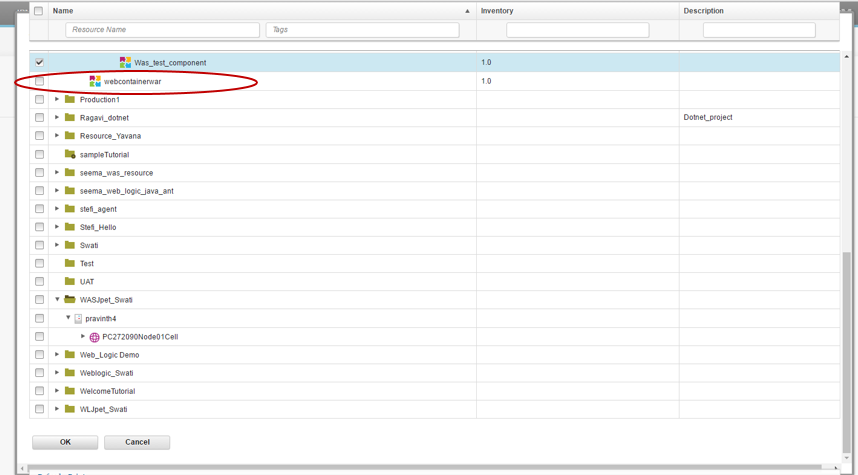
### Add resources

Click on add base resource:-

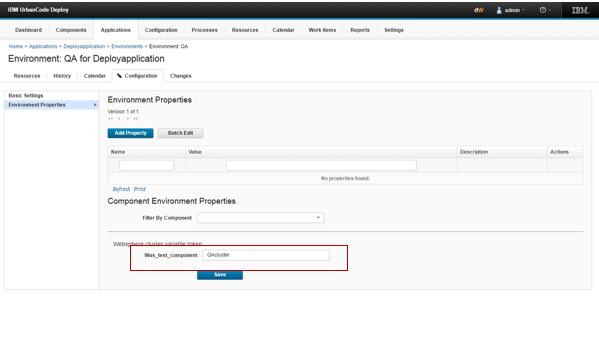


Select the component from list and save it.



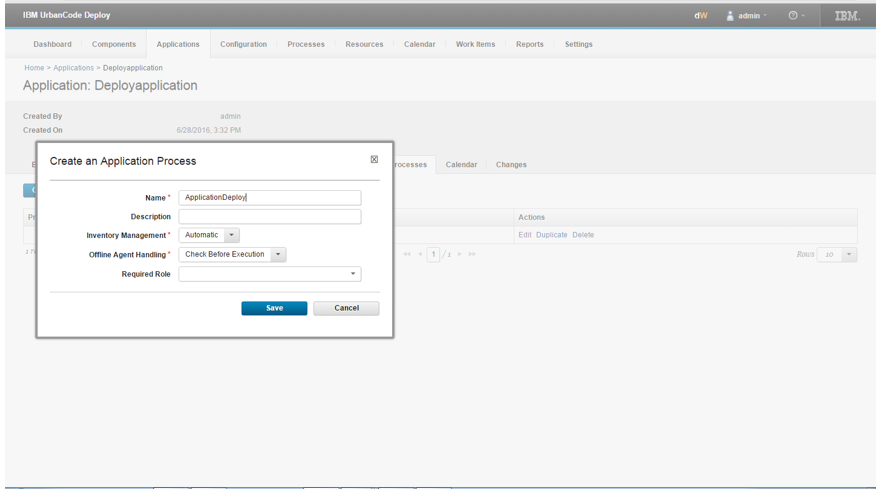


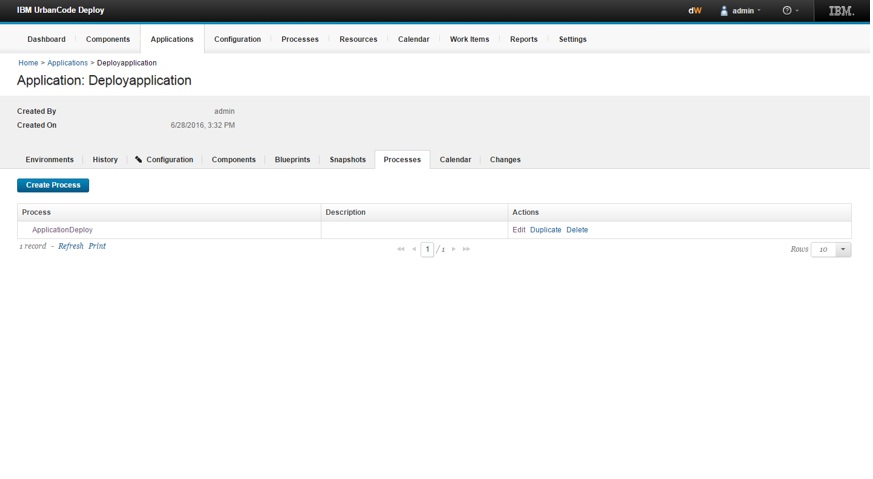
Go to environment 🡪 configuration 🡪 Environment properties 🡪 change the cluster variable token.



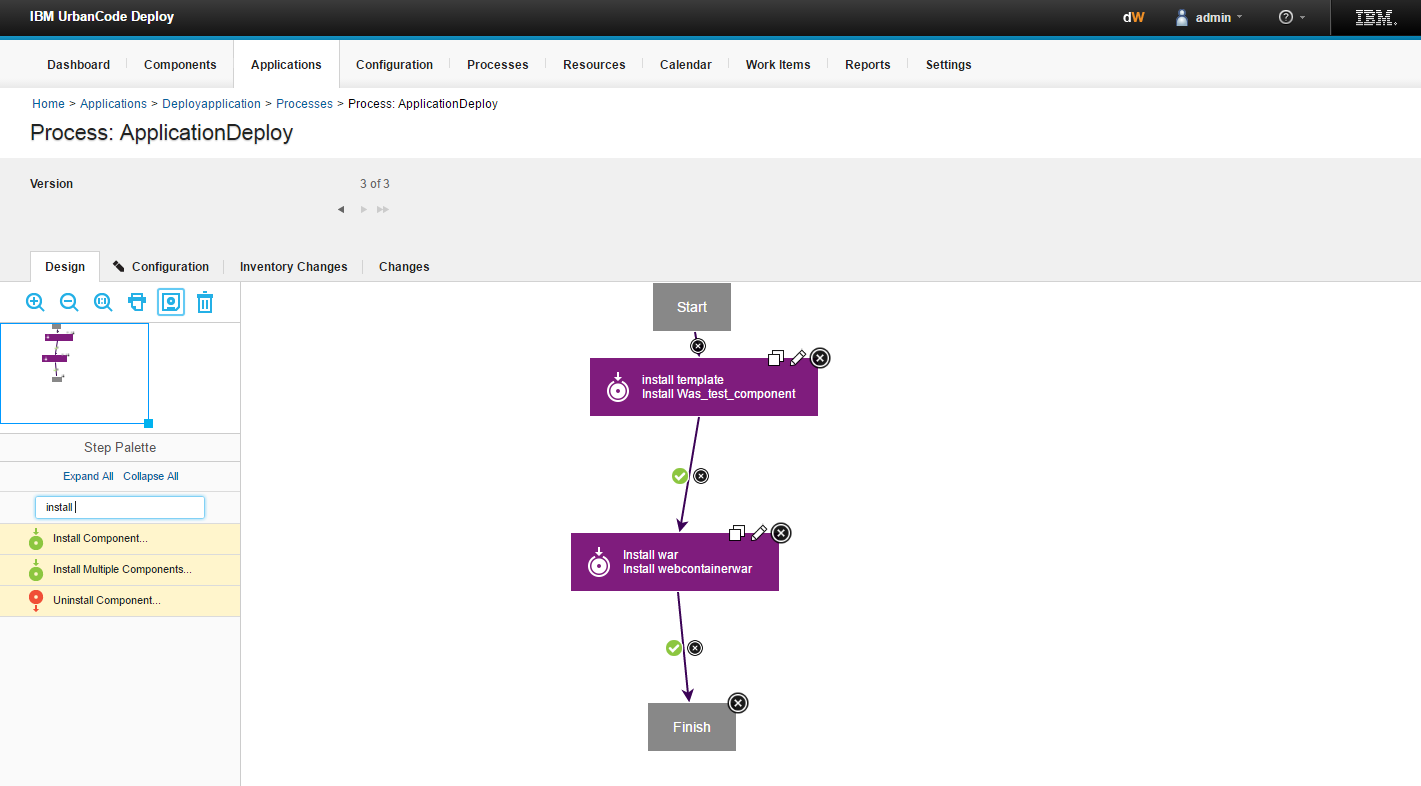
### Create process (Application level)

Click on application 🡪 Click on process 🡪 Create process

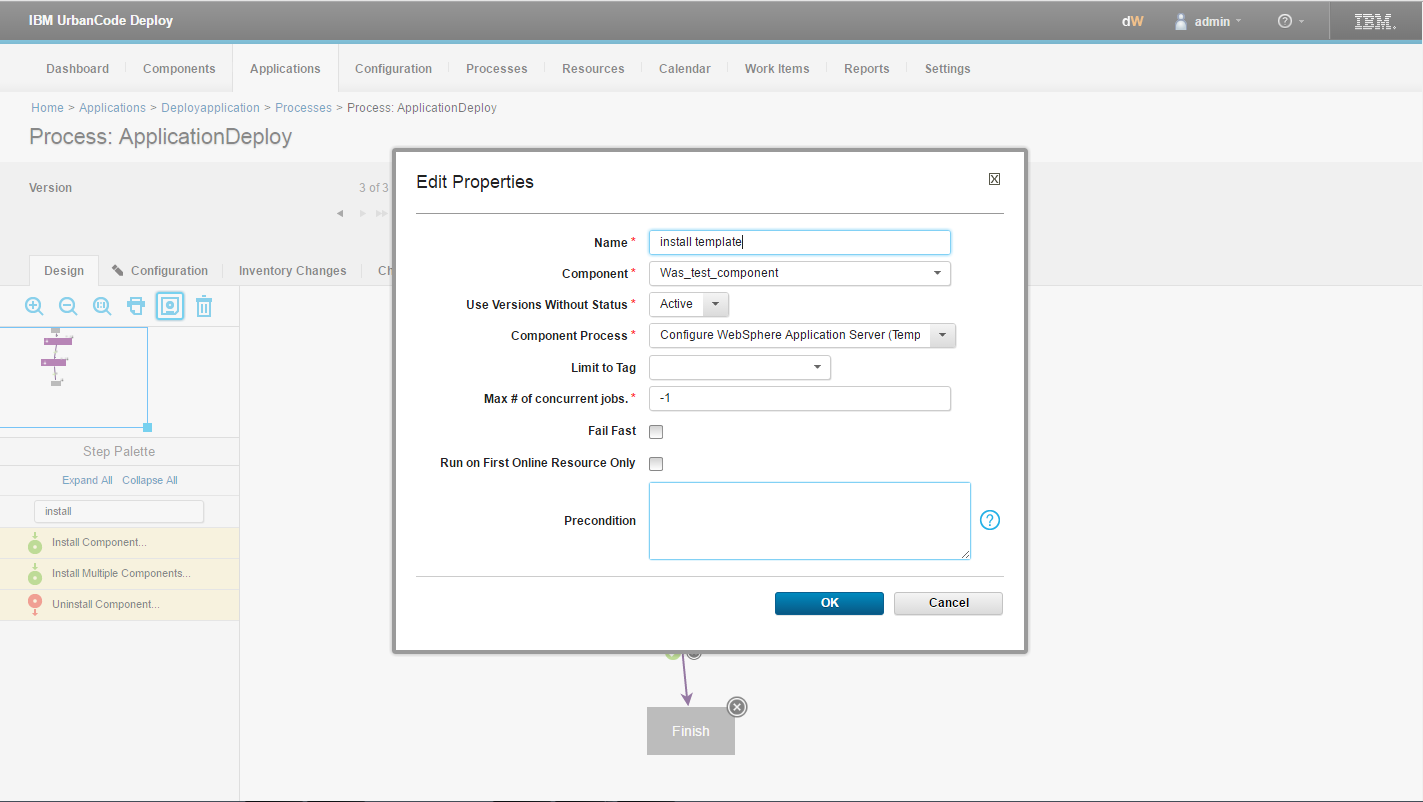




Click on process design the steps and click on  icon to save it.

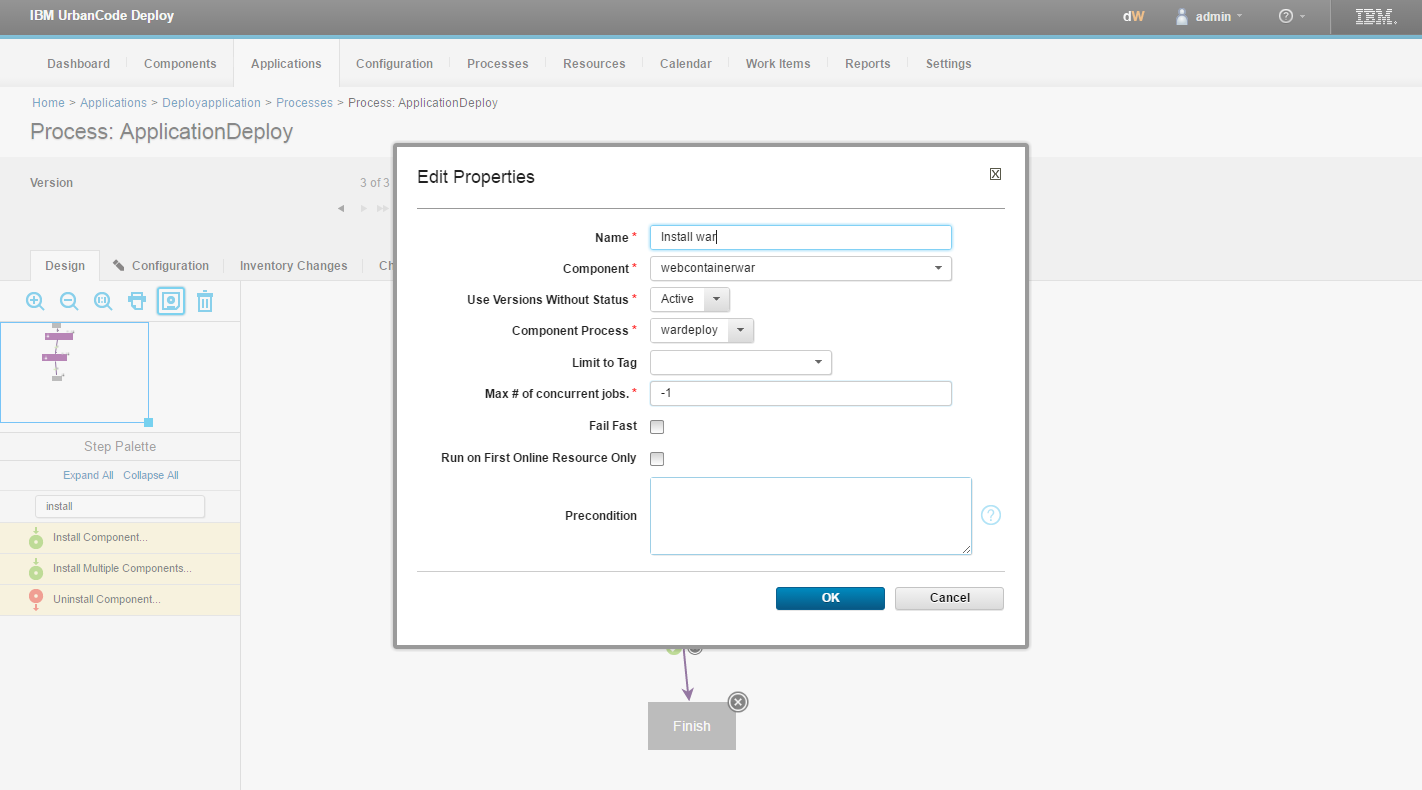


1. Install template -



Properties-

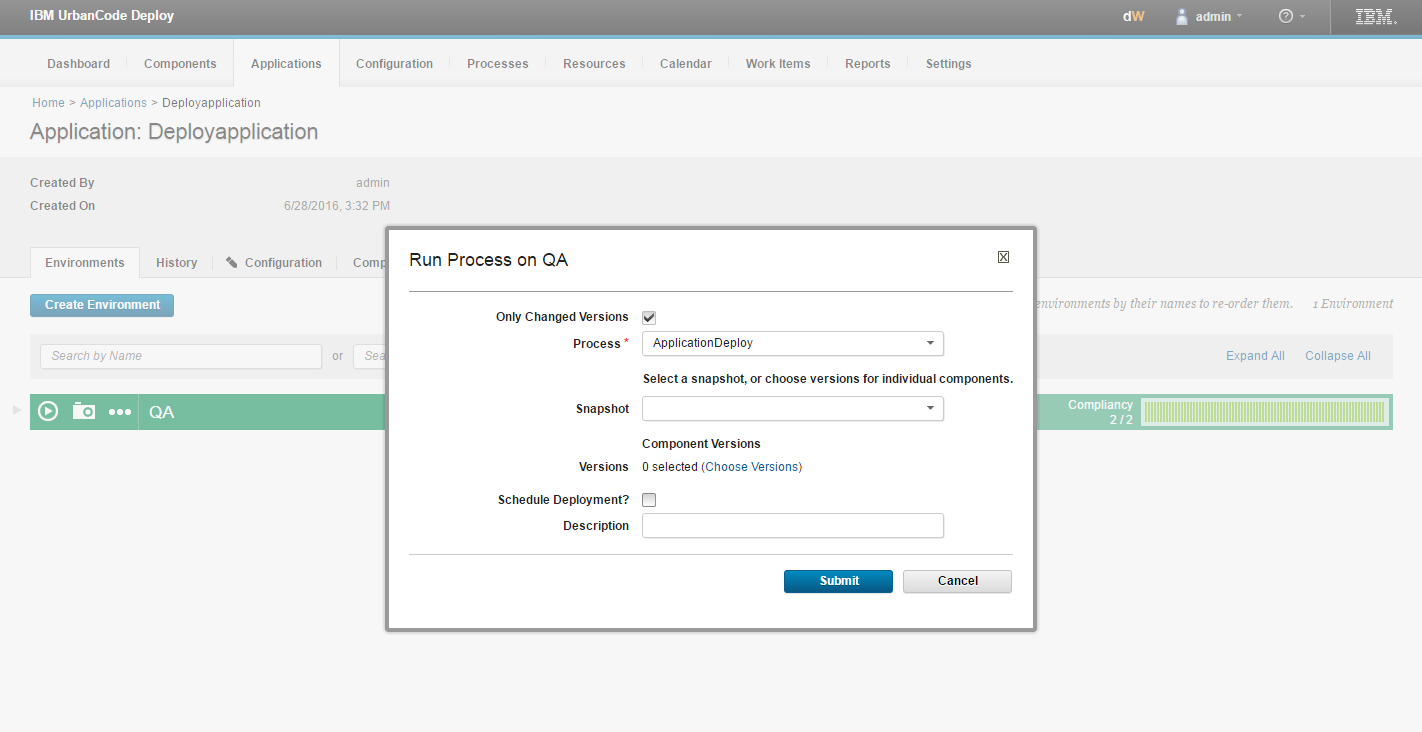
1. Name- Pass name of step.
2. Component – Pass the template.
3. Component process – select the component process as Configure WebSphere application server.
4. Save it.
5. Install war -



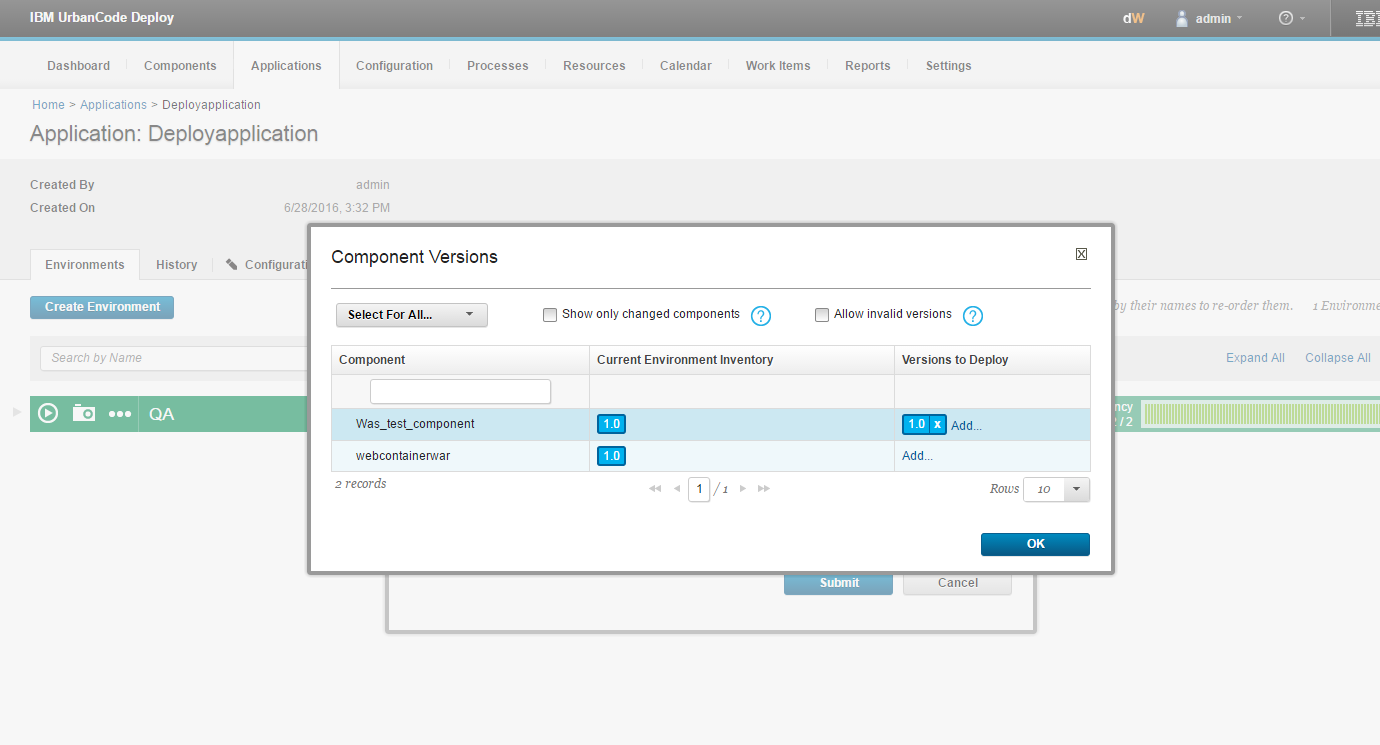
Properties-

1. Name- name of the step.
2. Component - Pass the component which represent the war file.
3. Component process - select the component process as war deploy.
4. Save it.

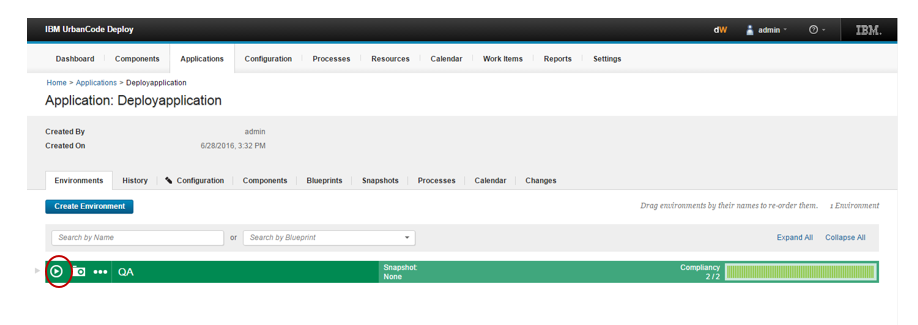
Application 🡪 Click on application 🡪 Run 🡪 add properties

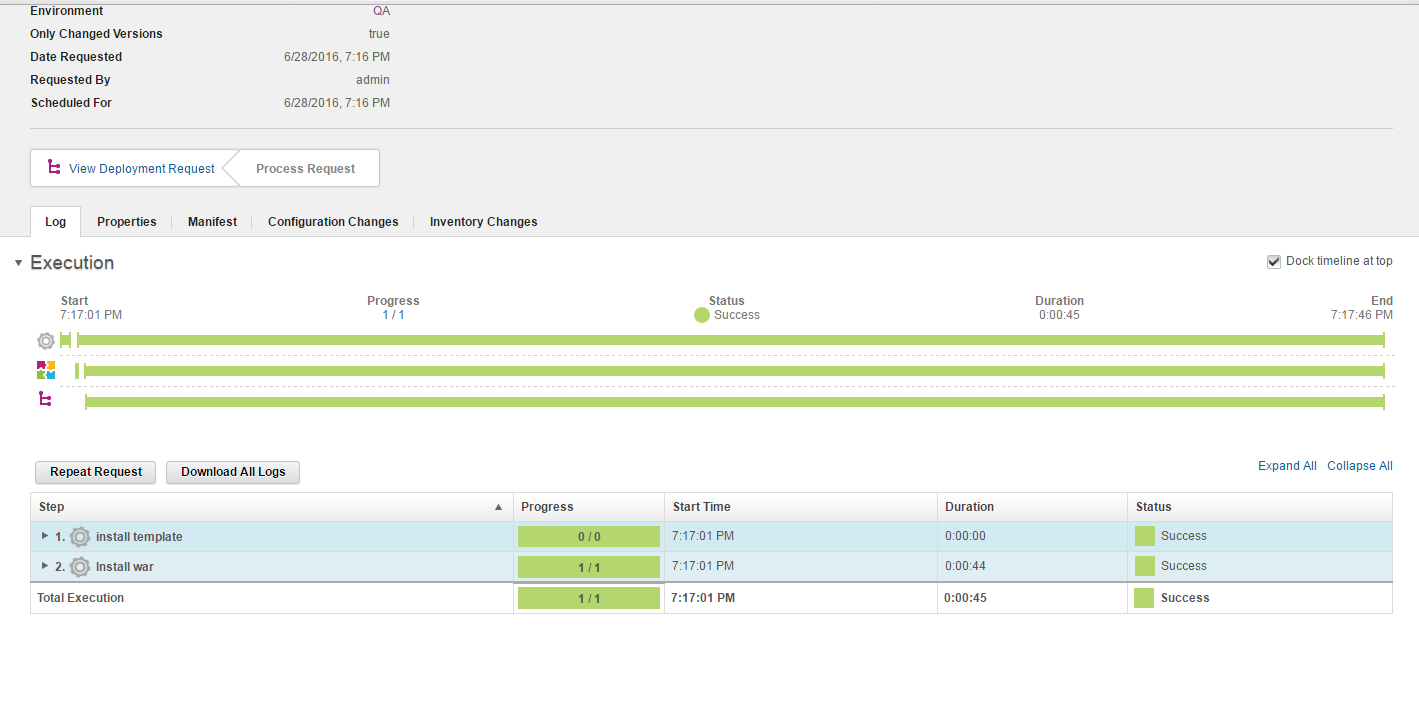


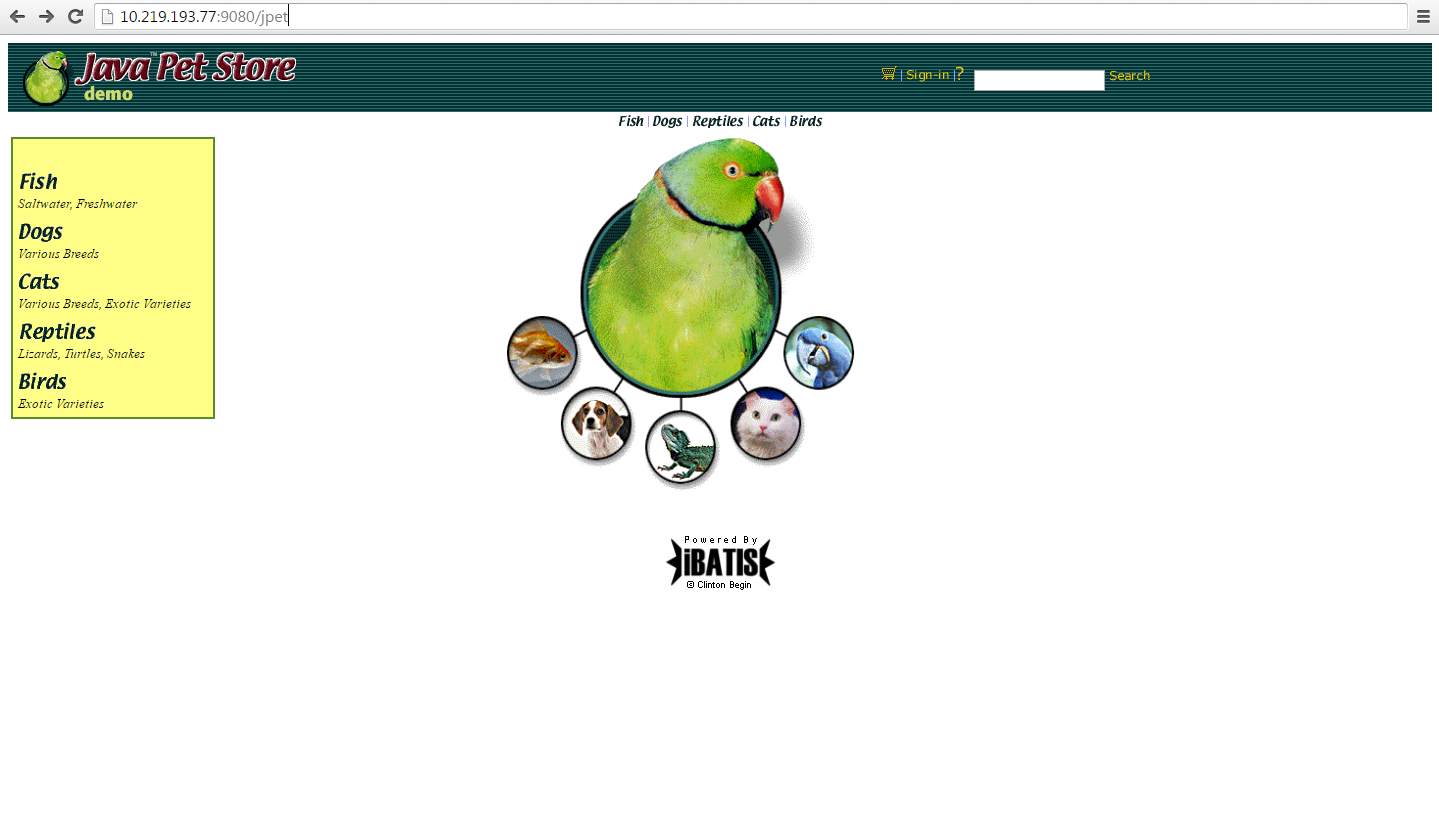
Choose the version



Click on ok and submit.







References- <https://www.youtube.com/watch?v=gowB49bxzr8>