MIRACLE
SOFTWARE SYSTEMS

AUTOMATION OF DATAPOWER WITH DCM USING ANTSCRIPT

API Management Team
Miracle Software Systems

16-06-2017

MIRACLE SOFTWARE SYSTEMS

INDEX

1.DataPower Configuration manager

1.1 Introduction:

The purpose of this document to describe the implementaion of continuous Delivery attempt with WebSphere DataPower through Jenkins; GITHub serves as the source control in this implementaion.Since Automation,Continuos Delivery and Dev Ops saves you time and lets you incorporate your project into the Continuos Integration(CI) process,many organizations are keen on the implementation in their development and delivery process.

- Advantages of Continuos Integration Testing
- When tests fail or bugs emerge,developers can revert the codebase to a bug-free state without wasting time for debugging.
- Developers detect and fix integration problems continuosly – and thus avoid last-minute chaos at release dates.
- Early warning of broken/incompatible code.
- Early warning of conflicting changes.
- Immediate testing of all changes.
- Constant availability of a "current" build for testing,demo,or release purposes.
- Immediate feedback to developers on the quality,functionality,or system-wide impact of their writen code.
- Frequent code check-ins push developers to create modular,less complex code.

DCM is core for the implementation of Continuous Integration and Deployment through Jenkins.DCM (**DataPower Configuration Manager**) is an open source tool published by IBM for automating and simplifying the configuration and management of IBM DataPower appliances (with the exception of the XC10).

.

DCM uses DataPower's XML Management Interface (XMI) to interact and manipulate appliance's management tasks in automated fashion.

DCM is a package for dealing with IBM DataPower configuration management. It provides an Ant-based command line tool.

1.1.Apache-Ant:

- ➔ Ant is a Java-based build tool with the full portability of pure Java code.
- ➔ The name is an acronym for "**Another Neat Tool**".
- ➔ Initially, Ant was part of the Tomcat code base, when it was donated to the Apache Software Foundation. It was created by James Duncan Davidson, who is also the original author of Tomcat. Ant was there to build Tomcat, nothing else.
- ➔ Soon thereafter, several open source Java projects realized that Ant could solve the problems they had with Make files. Starting with the projects hosted at Jakarta and the old Java Apache project, Ant spread like a virus and is now the build tool of choice for a lot of projects.
- ➔ In January 2000, Ant was moved to a separate CVS module and was promoted to a project of its own, independent of Tomcat, and became Apache Ant.

1.2.Targets:

A target is a **set of tasks** you want to be executed. A target can depend on other targets. For Example, You might have a target for compiling, and a target for creating a distributable. You can only build a distributable when you have compiled first, so the distribute target depends on the compile target. Ant resolves these dependencies.

1.3.Properties:

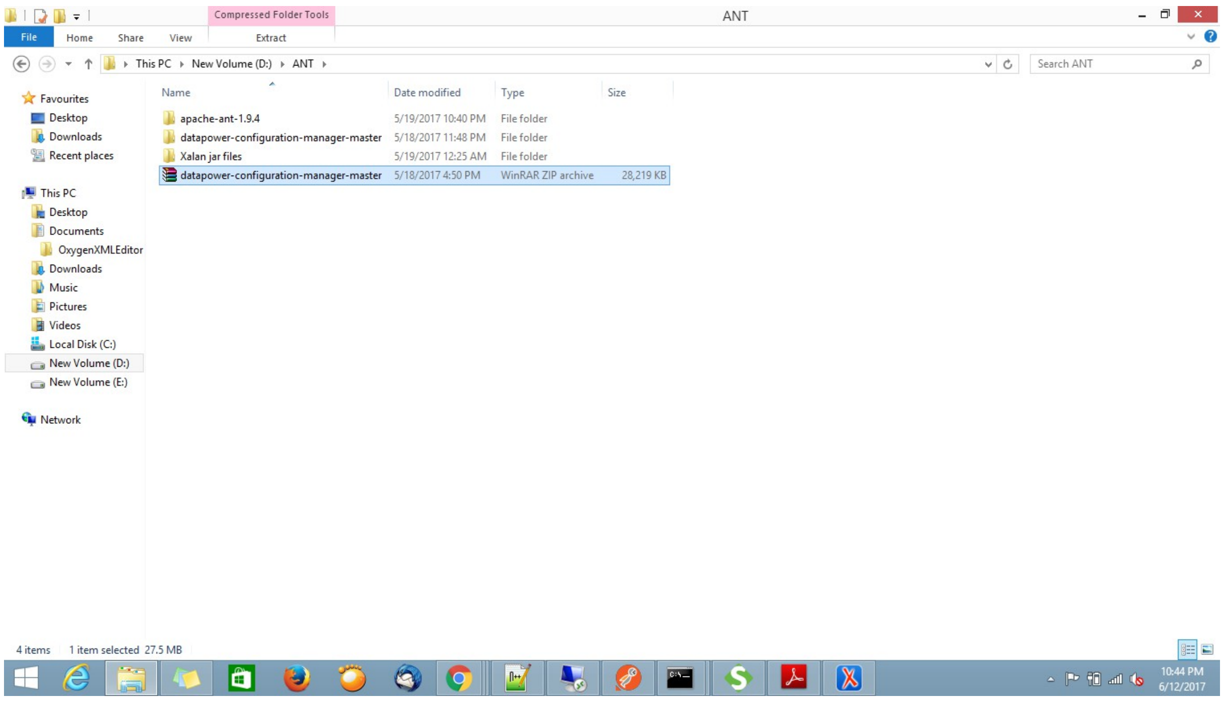
Properties are an important way to customize a build process or to just provide shortcuts for strings that are used repeatedly inside a build file.

2.Process of using DCM with ANT:

2.1:Download DCM Package:

Download DCM package from the below link and extract it .

https://blogs.perficient.com/ibm/2015/01/12/datapower-configuration-management-tool-part-i/



Name	Date modified	Type	Size
apache-ant-1.9.4	5/19/2017 10:40 PM	File folder	
datapower-configuration-manager-master	5/18/2017 11:48 PM	File folder	
Xalan jar files	5/19/2017 12:25 AM	File folder	
datapower-configuration-manager-master	5/18/2017 4:50 PM	WinRAR ZIP archive	28,219 KB

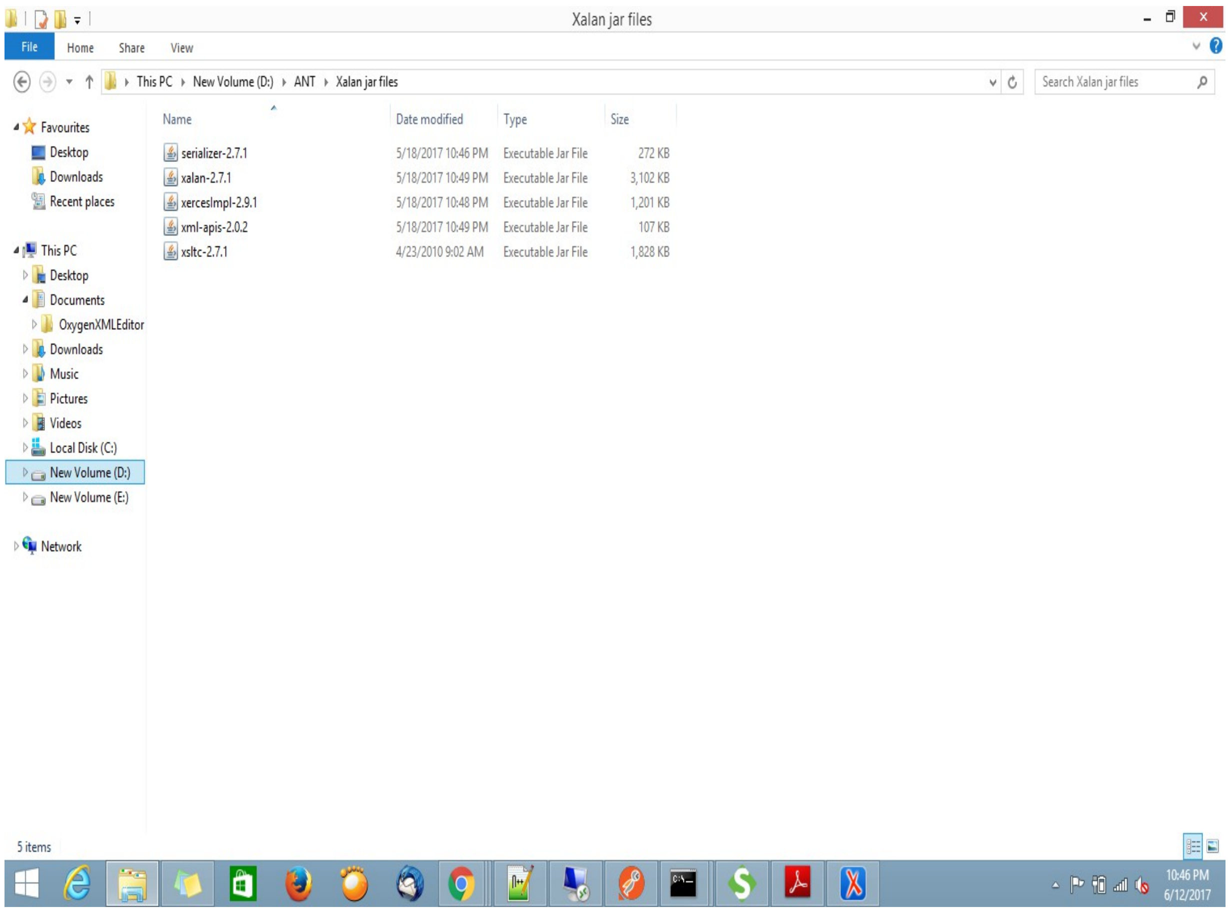
2.3:Download the jar files:

Download the jar files:

serializer-2.7.1.jar
xalan-2.7.1.jar,
xerceslmpl-2.9.1.jar,
xml-apis-2.0.2.jar,
xsltc-2.7.1.jar

You will get these jars from the below link

http://java2s.com/Code/Jar/x/Downloadxalanjar.htm

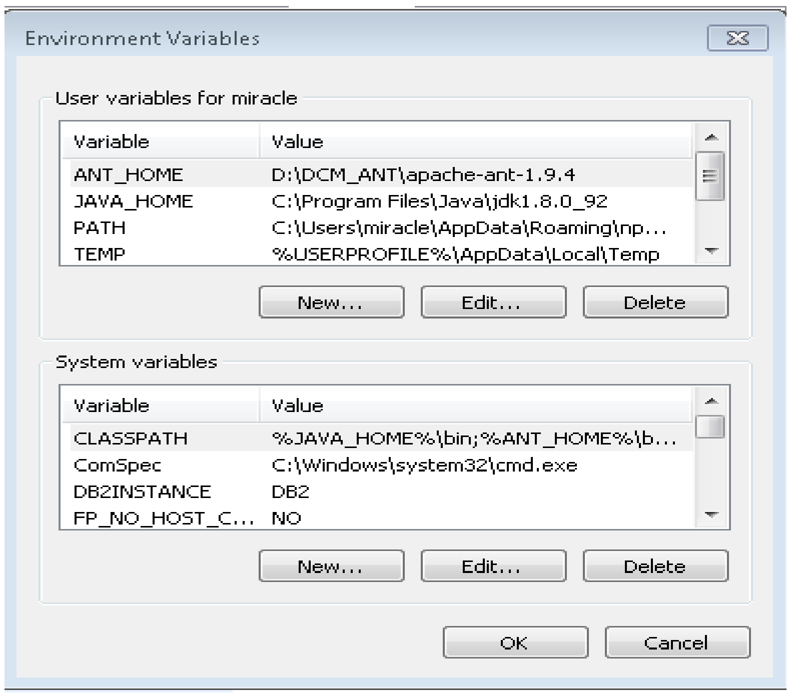


Name	Date modified	Type	Size
serializer-2.7.1	5/18/2017 10:46 PM	Executable Jar File	272 KB
xalan-2.7.1	5/18/2017 10:49 PM	Executable Jar File	3,102 KB
xercesImpl-2.9.1	5/18/2017 10:48 PM	Executable Jar File	1,201 KB
xml-apis-2.0.2	5/18/2017 10:49 PM	Executable Jar File	107 KB
xsltc-2.7.1	4/23/2010 9:02 AM	Executable Jar File	1,828 KB

2.4:Setting the Path:

Set the path for **java**, **Ant** and **Jars** as shown below :

My computer - > Properties - >Advanced settings - >Environment variables


Environment Variables

User variables for miracle

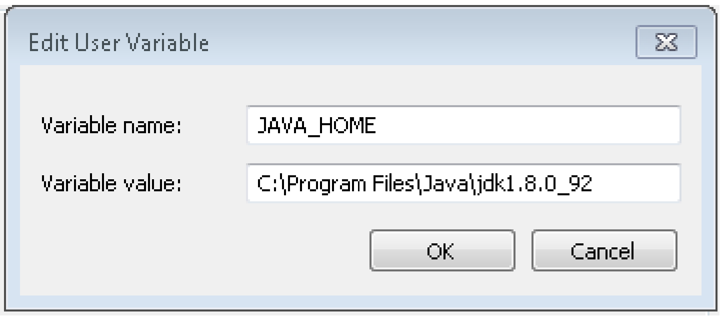
Variable	Value
ANT_HOME	D:\DCM_ANT\apache-ant-1.9.4
JAVA_HOME	C:\Program Files\Java\jdk1.8.0_92
PATH	C:\Users\miracle\AppData\Roaming\np...
TEMP	%USERPROFILE%\AppData\Local\Temp

System variables

Variable	Value
CLASSPATH	%JAVA_HOME%\bin;%ANT_HOME%\b...
ComSpec	C:\Windows\system32\cmd.exe
DB2INSTANCE	DB2
FP_NO_HOST_C...	NO

2.4.1:Path for Java:

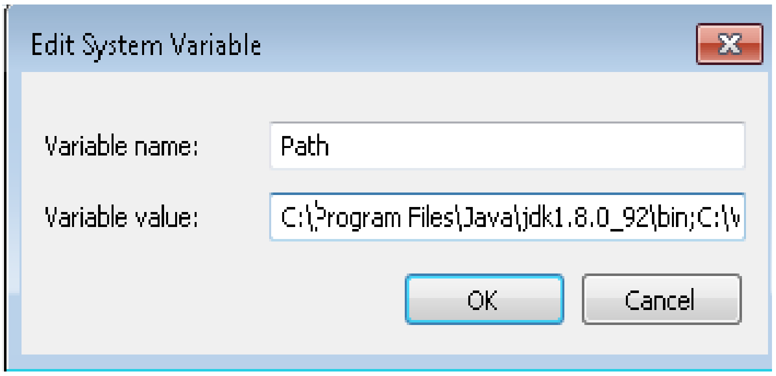
Select C:\Program Files\Java\jdk1.8.0_92 and paste in JAVA_HOME.


Edit User Variable

Variable name: JAVA_HOME

Variable value: C:\Program Files\Java\jdk1.8.0_92

Select C:\Program Files\Java\jdk1.8.0_92\bin and paste in Path.


Edit System Variable

Variable name: Path

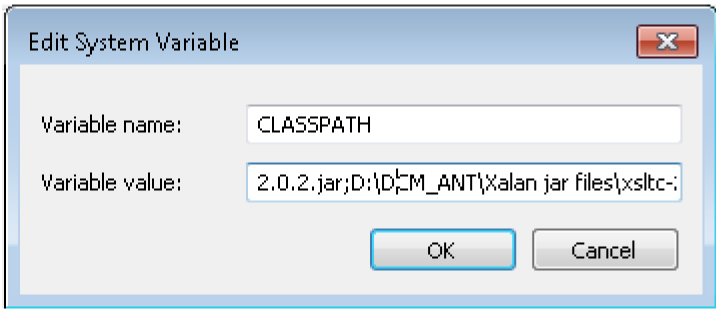
Variable value: C:\Program Files\Java\jdk1.8.0_92\bin;C:\V

2.4.2:Path for Jars:

Select D:\DCM_ANT\Xalan jar files paste it in CLASSPATH.

Note:specify all the 5 jar files path **individually** in CLASSPATH such as

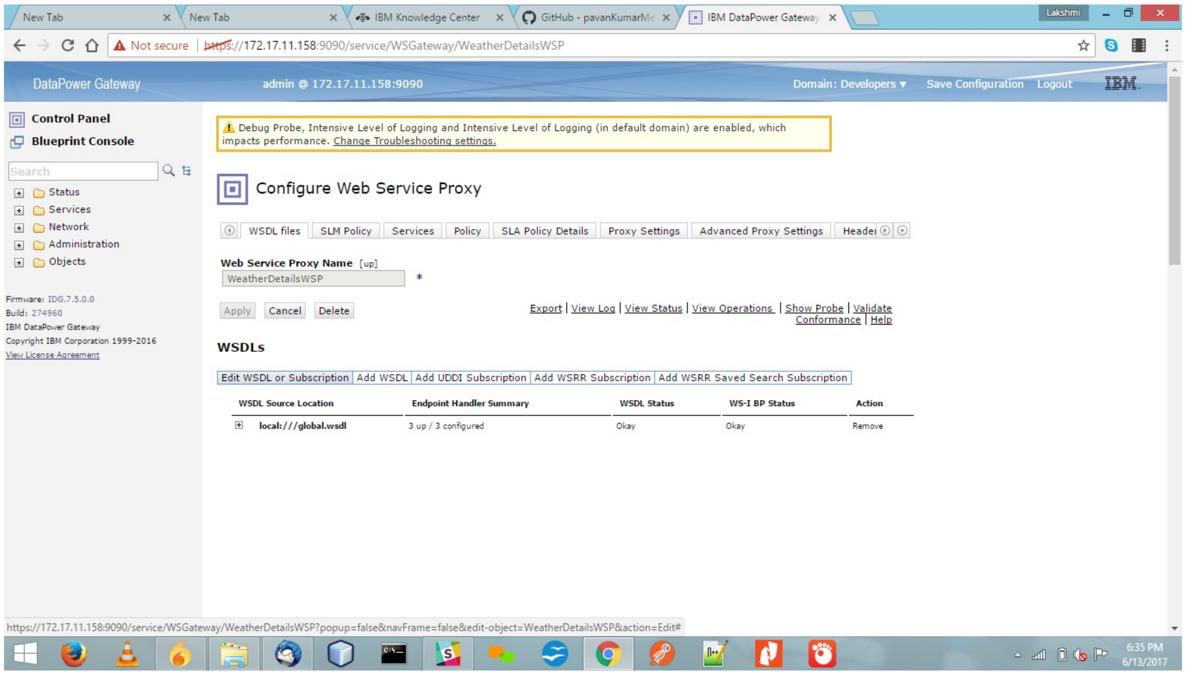
D:\DCM_ANT\Xalanjarfiles\serializer-2.7.1.jar;D:\DCM_ANT\Xalanjarfiles\xalan-2.7.1.jar;D:\DCM_ANT\Xalan jar files\xercesImpl-2.9.1.jar;D:\DCM_ANT\Xalan jar files\xml-apis-2.0.2.jar;D:\DCM_ANT\Xalan jar files\xsltc-2.7.1.jar;


Edit System Variable

Variable name: CLASSPATH

Variable value: 2.0.2.jar;D:\DCM_ANT\Xalan jar files\xsltc-:

Export the service from datapower


https://172.17.11.158:9090/service/WSGateway/WeatherDetailsWSP

DataPower Gateway admin @ 172.17.11.158:9090 Domain: Developers Save Configuration Logout

Debug Probe, Intensive Level of Logging and Intensive Level of Logging (in default domain) are enabled, which impacts performance. Change Troubleshooting settings.

Configure Web Service Proxy

Web Service Proxy Name [up]
WeatherDetailsWSP *

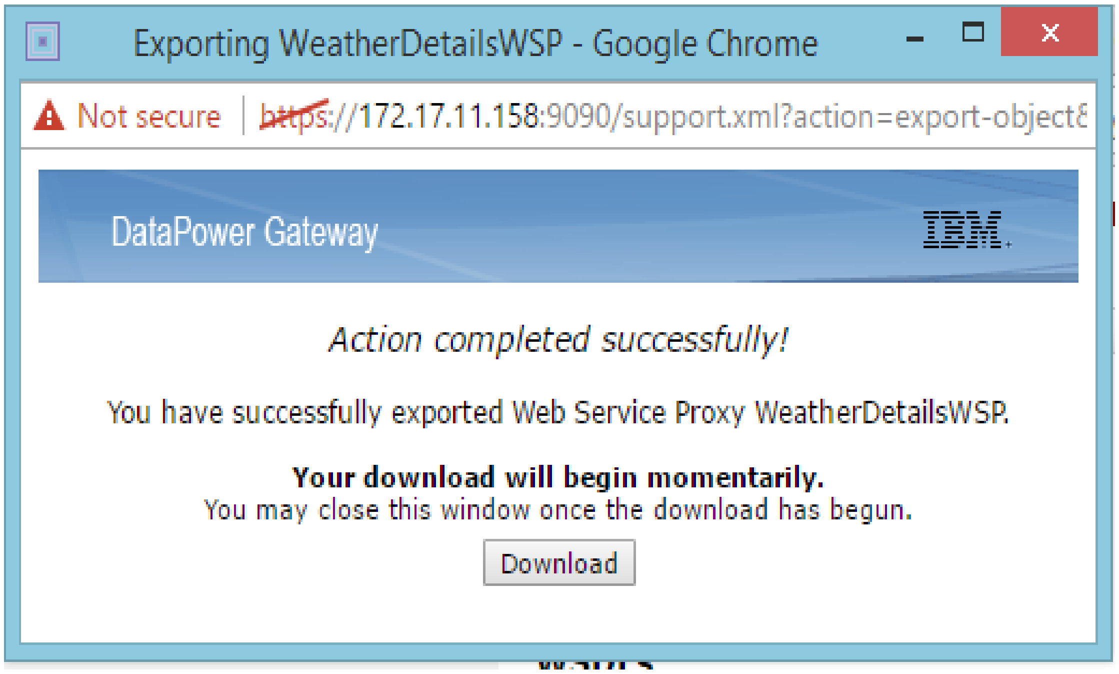
Firmware: IDG.7.5.0.0
Build: 274960
IBM DataPower Gateway
Copyright IBM Corporation 1999-2016
View License Agreement

WSDLs

WSDL Source Location	Endpoint Handler Summary	WSDL Status	WS-I BP Status	Action
local:///global.wsdl	3 up / 3 configured	Okay	Okay	Remove

Control Panel
Blueprint Console
Status
Services
Network
Administration
Objects
WSDL files
SLM Policy
Services
Policy
SLA Policy Details
Proxy Settings
Advanced Proxy Settings
Apply
Cancel
Delete
Export | View Log | View Status | View Operations | Show Probe | Validate Conformance | Help
Edit WSDL or Subscription | Add WSDL | Add UDDI Subscription | Add WSRR Subscription | Add WSRR Saved Search Subscription

You can download the service to local machine as below


Exporting WeatherDetailsWSP - Google Chrome

Not secure | https://172.17.11.158:9090/support.xml?action=export-object&

DataPower Gateway

Action completed successfully!

You have successfully exported Web Service Proxy WeatherDetailsWSP.

Your download will begin momentarily.
You may close this window once the download has begun.

Service exported from datapower is pushed into Git .

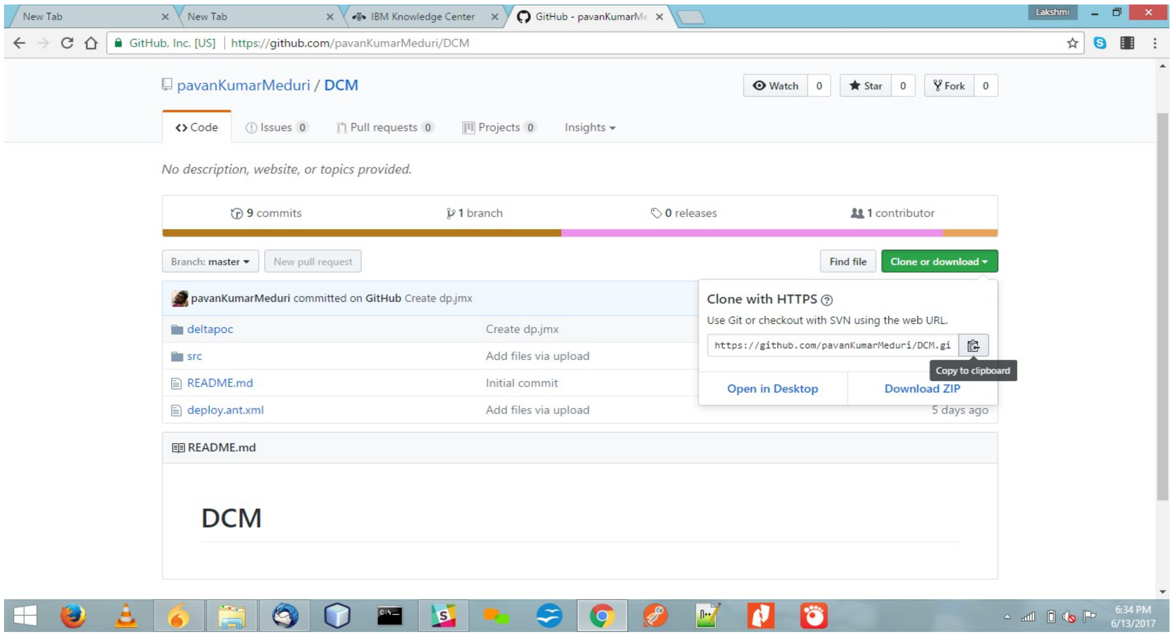
3.GitHub:

The GitHub plugin for Jenkins is the most basic plugin for integrating Jenkins with GitHub projects. If you are a GitHub user, this plugin enables you to schedule your build, pull your code and data files from your GitHub repository to your Jenkins machine, and automatically trigger each build on the Jenkins server after each Commit on your Git repository.

This saves you time and lets you incorporate your project into the Continuous Integration (CI) process.

Why we use something like Git? Say you and a coworker are both updating pages on the same website. You make your changes, save them, and upload them back to the website. So far, so good. The problem comes when your coworker is working on the same page as you at the same time.One is about to have your work overwritten and erased.

A version control application like Git keeps that from happening. You and your coworker can each upload your revisions to the same page, and Git will save two copies. Later, you can merge your changes together without losing any work along the way. You can even revert to an earlier version at any time, because Git keeps a "snapshot" of every change ever made.



No description, website, or topics provided.

pavanKumarMeduri committed on GitHub Create dp.jmx

deltapoc	Create dp.jmx
src	Add files via upload
README.md	Initial commit
deploy.ant.xml	Add files via upload

5 days ago

Clone with HTTPS
Use Git or checkout with SVN using the web URL.
https://github.com/pavanKumarMeduri/DCM.gi

README.md

DCM

4.Jenkins

4.1 About Jenkins :

Jenkins is an open source automation server written in Java. The project was forked from Hudson after a dispute with Oracle

Jenkins helps to automate the **non-human** part of software development process,with continues integration(**CI**) and facilitating technical aspects of continues delivery. It is a sever-based system that runs in servelet containers such as Apache Tomcat. It supports version control tools including CVS,Subversion,Git,Clear-Case and can execute Apache-Ant,Apache Maven as well as arbitrary shell scripts and windows batch commands.

4.2Jenkins Installation:

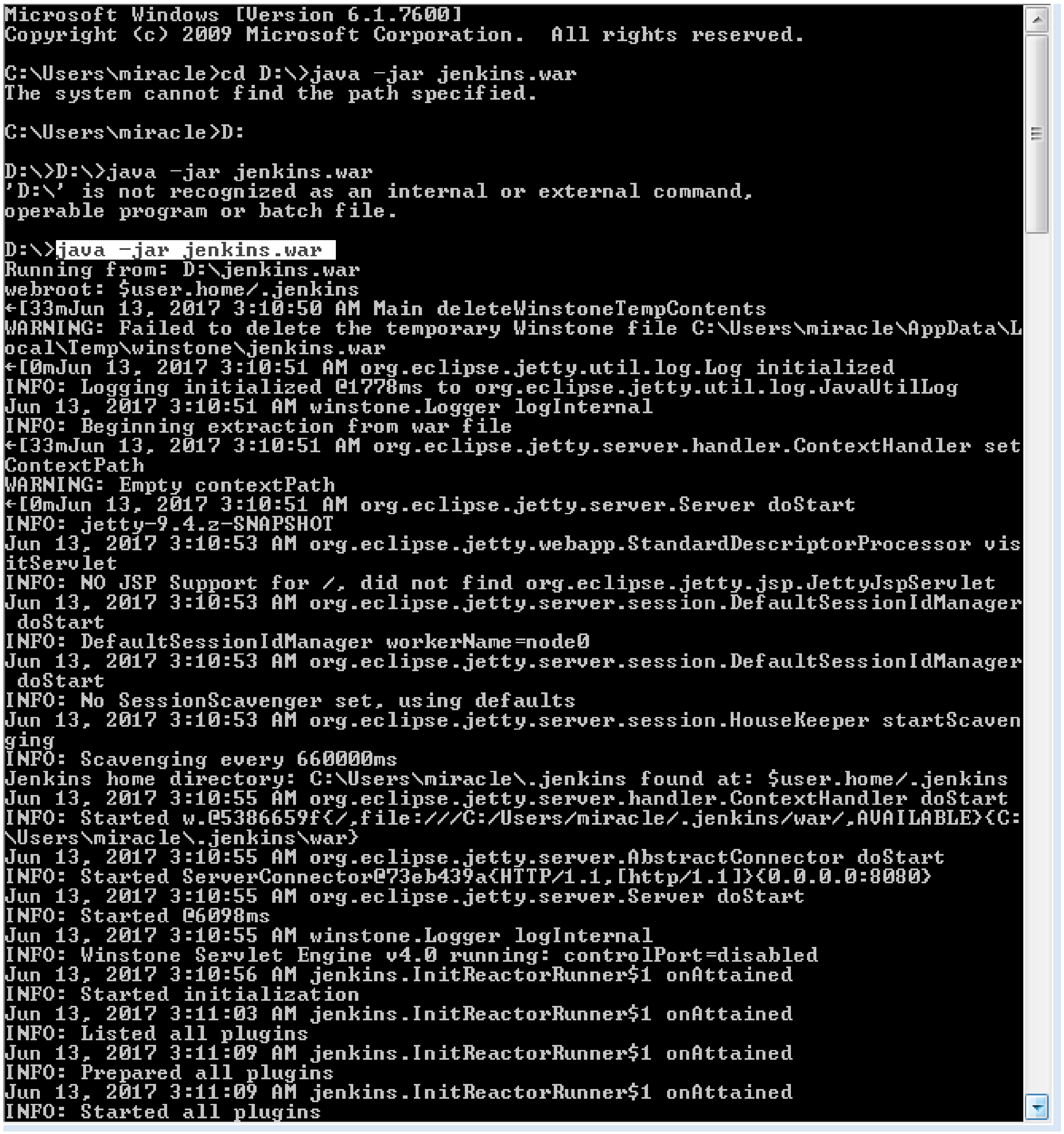
For the Jenkins installation we need to download Jenkins.war file or jenkins.exe(for windows).Here we are using .war file.

Download the Jenkins war file from the below link

https://jenkins.io/download/

cmd: Java -jar jenkins.war

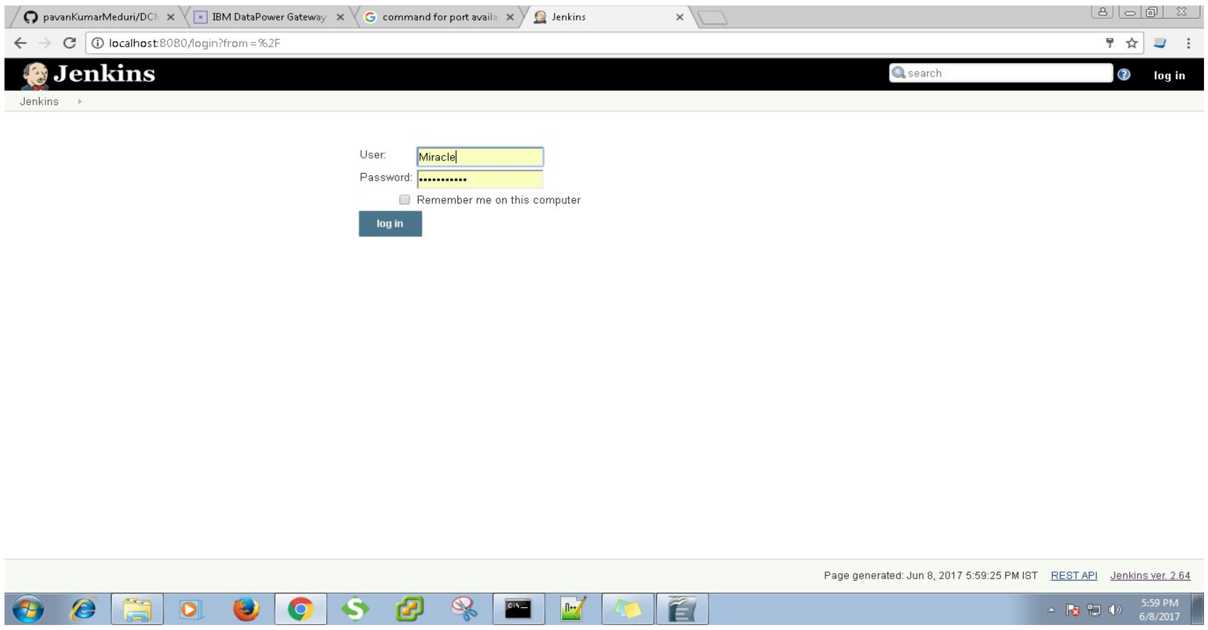
Note: Before running the above command first we need to change the specified directory where Jenkins.war file is placed.



```
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\miracle>cd D:\>java -jar jenkins.war
The system cannot find the path specified.

C:\Users\miracle>D:

D:\>D:\>java -jar jenkins.war
'D:\' is not recognized as an internal or external command,
operable program or batch file.

D:\>java -jar jenkins.war
Running from: D:\jenkins.war
webroot: $user.home/.jenkins
←[33mJun 13, 2017 3:10:50 AM Main deleteWinstoneTempContents
WARNING: Failed to delete the temporary Winstone file C:\Users\miracle\AppData\L
ocal\Temp\winstone\jenkins.war
←[0mJun 13, 2017 3:10:51 AM org.eclipse.jetty.util.log.Log initialized
INFO: Logging initialized @1778ms to org.eclipse.jetty.util.log.JavaUtilLog
Jun 13, 2017 3:10:51 AM winstone.Logger logInternal
INFO: Beginning extraction from war file
←[33mJun 13, 2017 3:10:51 AM org.eclipse.jetty.server.handler.ContextHandler set
ContextPath
WARNING: Empty contextPath
←[0mJun 13, 2017 3:10:51 AM org.eclipse.jetty.server.Server doStart
INFO: jetty-9.4.z-SNAPSHOT
Jun 13, 2017 3:10:53 AM org.eclipse.jetty.webapp.StandardDescriptorProcessor vis
itServlet
INFO: NO JSP Support for /, did not find org.eclipse.jetty.jsp.JettyJspServlet
Jun 13, 2017 3:10:53 AM org.eclipse.jetty.server.session.DefaultSessionIdManager
 doStart
INFO: DefaultSessionIdManager workerName=node0
Jun 13, 2017 3:10:53 AM org.eclipse.jetty.server.session.DefaultSessionIdManager
 doStart
INFO: No SessionScavenger set, using defaults
Jun 13, 2017 3:10:53 AM org.eclipse.jetty.server.session.HouseKeeper startScaven
ging
INFO: Scavenging every 660000ms
Jenkins home directory: C:\Users\miracle\.jenkins found at: $user.home/.jenkins
Jun 13, 2017 3:10:55 AM org.eclipse.jetty.server.handler.ContextHandler doStart
INFO: Started w.@5386659f{/,file:///C:/Users/miracle/.jenkins/war/,AVAILABLE}{C:
\Users\miracle\.jenkins\war}
Jun 13, 2017 3:10:55 AM org.eclipse.jetty.server.AbstractConnector doStart
INFO: Started ServerConnector@73eb439a{HTTP/1.1,[http/1.1]}{0.0.0.0:8080}
Jun 13, 2017 3:10:55 AM org.eclipse.jetty.server.Server doStart
INFO: Started @6098ms
Jun 13, 2017 3:10:55 AM winstone.Logger logInternal
INFO: Winstone Servlet Engine v4.0 running: controlPort=disabled
Jun 13, 2017 3:10:56 AM jenkins.InitReactorRunner$1 onAttained
INFO: Started initialization
Jun 13, 2017 3:11:03 AM jenkins.InitReactorRunner$1 onAttained
INFO: Listed all plugins
Jun 13, 2017 3:11:09 AM jenkins.InitReactorRunner$1 onAttained
INFO: Prepared all plugins
Jun 13, 2017 3:11:09 AM jenkins.InitReactorRunner$1 onAttained
INFO: Started all plugins
```

Run the Jenkins through cmd and it opens on localhost in browser. Then we need to create an account in jenkins.

Login to Jenkins with the credentials.


localhost:8080/login?from=%2F

Jenkins

User: Miracle

Password: ••••••••••

Remember me on this computer

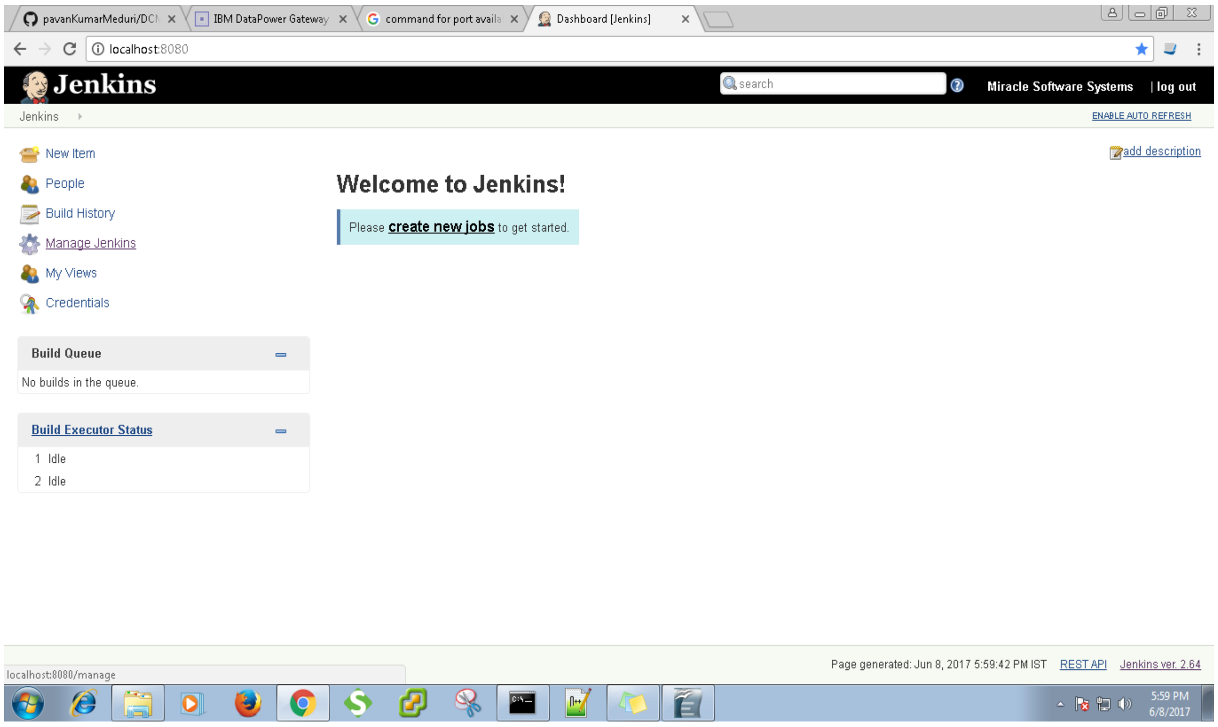
log in

Page generated: Jun 8, 2017 5:59:25 PM IST REST API Jenkins ver. 2.64

Welcome page will be displayed and you can create new jobs here

Click on **Manage jenkins → Manage Plugins**

Here we you can install various plugins required externally as well internally.


localhost:8080

Jenkins

Miracle Software Systems | log out

ENABLE AUTO REFRESH

- New Item
- People
- Build History
- Manage Jenkins
- My Views
- Credentials

Build Queue

No builds in the queue.

Build Executor Status

1 Idle

2 Idle

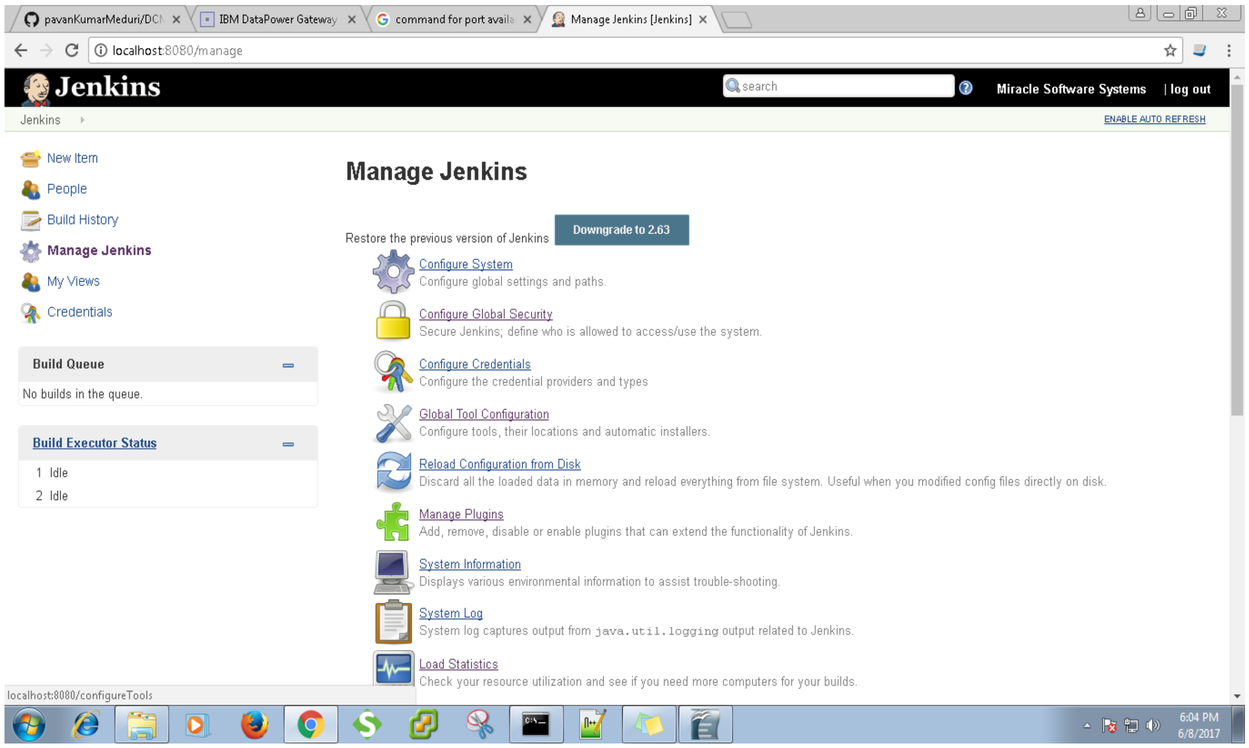
add description

Welcome to Jenkins!

Please **create new jobs** to get started.

localhost:8080/manage

Page generated: Jun 8, 2017 5:59:42 PM IST REST API Jenkins ver. 2.64


localhost:8080/manage

Manage Jenkins

Restore the previous version of Jenkins

Configure System
Configure global settings and paths.

Configure Global Security
Secure Jenkins; define who is allowed to access/use the system.

Configure Credentials
Configure the credential providers and types

Global Tool Configuration
Configure tools, their locations and automatic installers.

Reload Configuration from Disk
Discard all the loaded data in memory and reload everything from file system. Useful when you modified config files directly on disk.

Manage Plugins
Add, remove, disable or enable plugins that can extend the functionality of Jenkins.

System Information
Displays various environmental information to assist trouble-shooting.

System Log
System log captures output from `java.util.logging` output related to Jenkins.

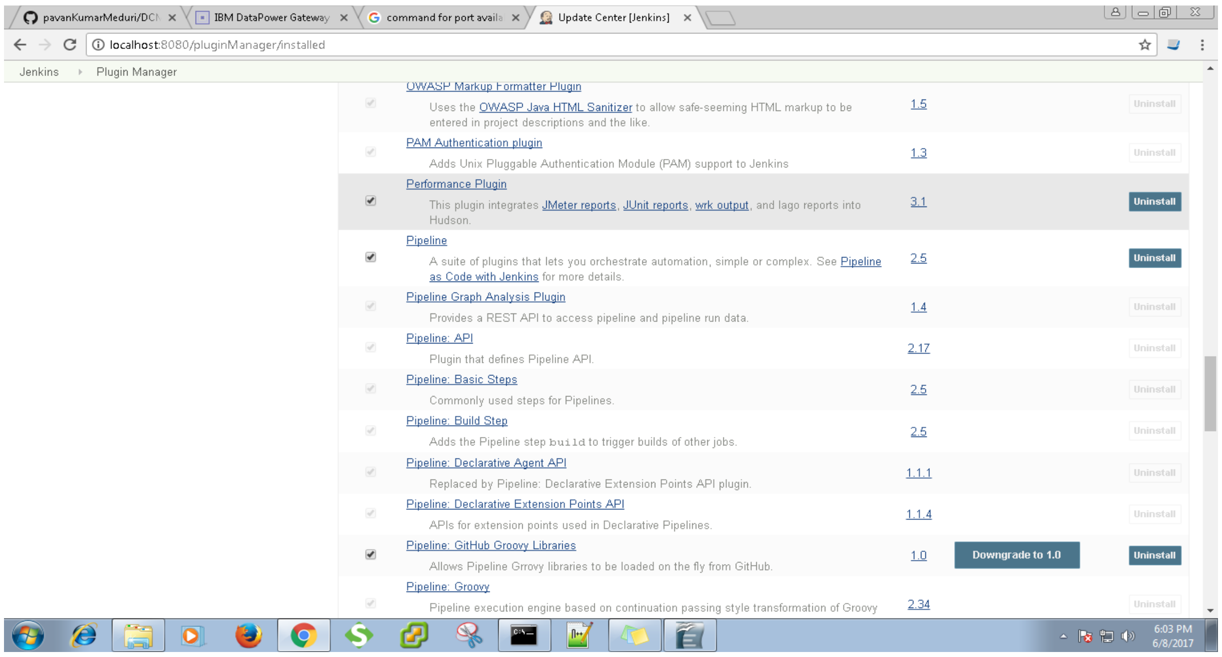
Load Statistics
Check your resource utilization and see if you need more computers for your builds.

Build Queue
No builds in the queue.

Build Executor Status
1 Idle
2 Idle

As per our requirement we installed **GIT Plugin,Perfomance Plugin,Ant plugin.**

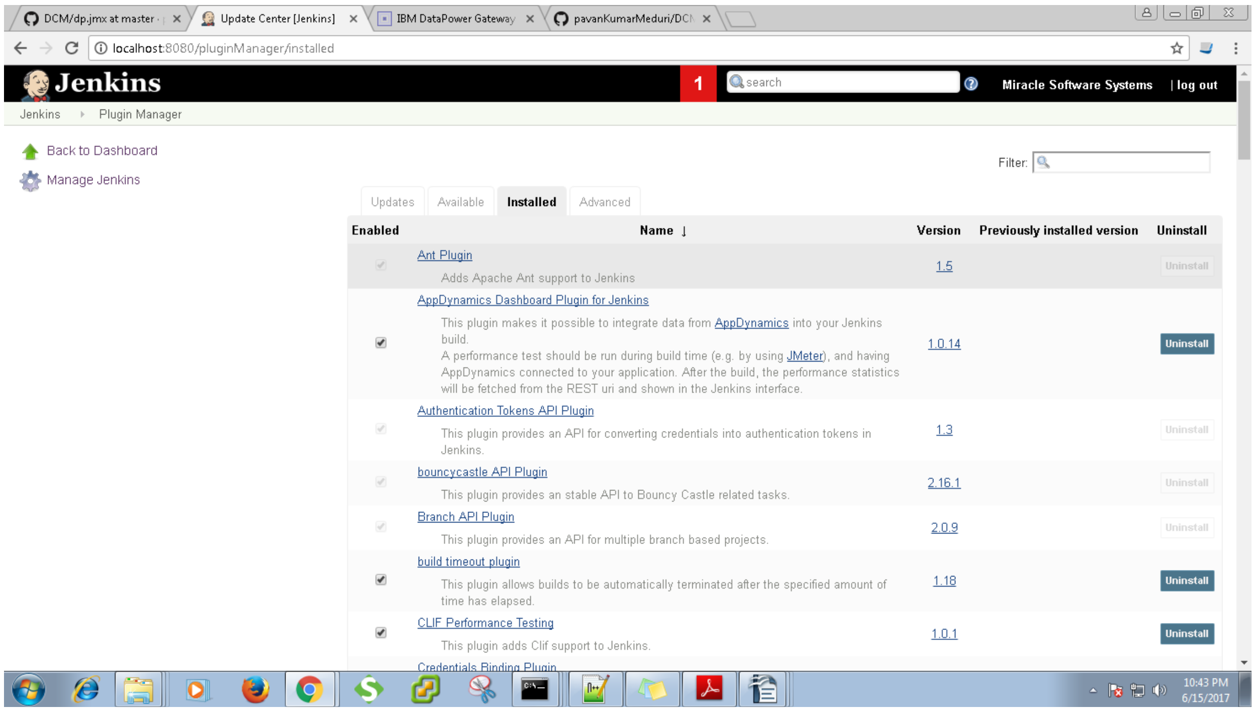
After installing all required plug-ins we need configure tools .for that we need move Jenkins_Home page.


localhost:8080/pluginManager/installed

Jenkins › Plugin Manager

✓	OWASP Markup Formatter Plugin Uses the OWASP Java HTML Sanitizer to allow safe-seeming HTML markup to be entered in project descriptions and the like.	1.5		Uninstall
✓	PAM Authentication plugin Adds Unix Pluggable Authentication Module (PAM) support to Jenkins	1.3		Uninstall
✓	Performance Plugin This plugin integrates JMeter reports, JUnit reports, wrk output, and Iago reports into Hudson.	3.1		Uninstall
✓	Pipeline A suite of plugins that lets you orchestrate automation, simple or complex. See Pipeline as Code with Jenkins for more details.	2.5		Uninstall
✓	Pipeline Graph Analysis Plugin Provides a REST API to access pipeline and pipeline run data.	1.4		Uninstall
✓	Pipeline: API Plugin that defines Pipeline API.	2.17		Uninstall
✓	Pipeline: Basic Steps Commonly used steps for Pipelines.	2.5		Uninstall
✓	Pipeline: Build Step Adds the Pipeline step `build` to trigger builds of other jobs.	2.5		Uninstall
✓	Pipeline: Declarative Agent API Replaced by Pipeline: Declarative Extension Points API plugin.	1.1.1		Uninstall
✓	Pipeline: Declarative Extension Points API APIs for extension points used in Declarative Pipelines.	1.1.4		Uninstall
✓	Pipeline: GitHub Groovy Libraries Allows Pipeline Grrovy libraries to be loaded on the fly from GitHub.	1.0	Downgrade to 1.0	Uninstall
✓	Pipeline: Groovy Pipeline execution engine based on continuation passing style transformation of Groovy	2.34		Uninstall

6:03 PM 6/8/2017


localhost:8080/pluginManager/installed

Jenkins — Miracle Software Systems | log out

Jenkins › Plugin Manager

Back to Dashboard

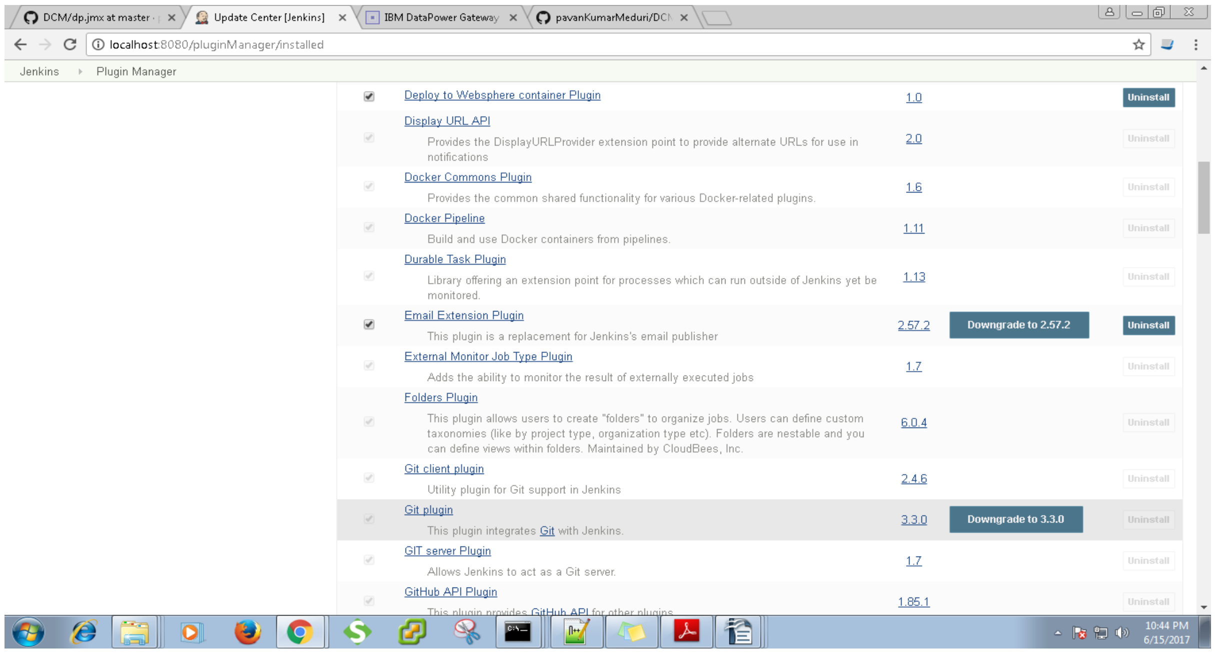
Manage Jenkins

Filter:

Updates | Available | **Installed** | Advanced

Enabled	Name ↓	Version	Previously installed version	Uninstall
✓	Ant Plugin Adds Apache Ant support to Jenkins	1.5		Uninstall
✓	AppDynamics Dashboard Plugin for Jenkins This plugin makes it possible to integrate data from AppDynamics into your Jenkins build. A performance test should be run during build time (e.g. by using JMeter), and having AppDynamics connected to your application. After the build, the performance statistics will be fetched from the REST uri and shown in the Jenkins interface.	1.0.14		Uninstall
✓	Authentication Tokens API Plugin This plugin provides an API for converting credentials into authentication tokens in Jenkins.	1.3		Uninstall
✓	bouncycastle API Plugin This plugin provides an stable API to Bouncy Castle related tasks.	2.16.1		Uninstall
✓	Branch API Plugin This plugin provides an API for multiple branch based projects.	2.0.9		Uninstall
✓	build timeout plugin This plugin allows builds to be automatically terminated after the specified amount of time has elapsed.	1.18		Uninstall
✓	CLIF Performance Testing This plugin adds Clif support to Jenkins.	1.0.1		Uninstall
	Credentials Binding Plugin			

10:43 PM 6/15/2017

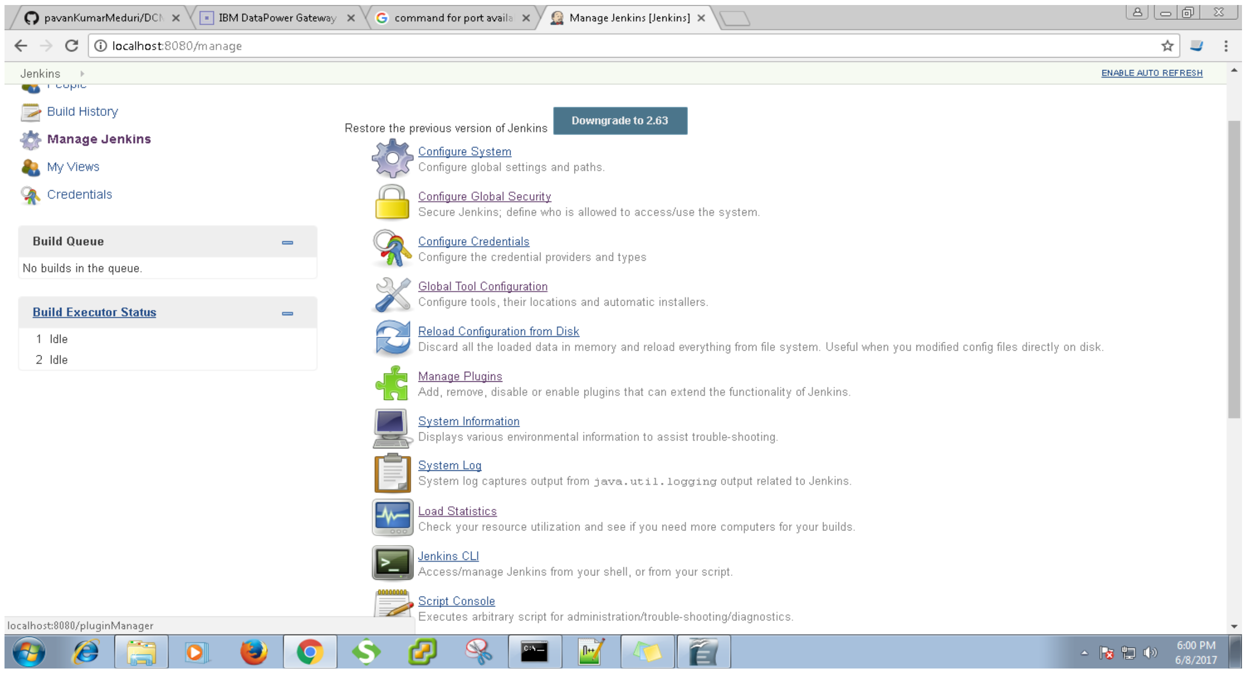

localhost:8080/pluginManager/installed

Jenkins › Plugin Manager

	Name	Version		
✓	Deploy to Websphere container Plugin	1.0		Uninstall
✓	Display URL API Provides the DisplayURLProvider extension point to provide alternate URLs for use in notifications	2.0		Uninstall
✓	Docker Commons Plugin Provides the common shared functionality for various Docker-related plugins.	1.6		Uninstall
✓	Docker Pipeline Build and use Docker containers from pipelines.	1.11		Uninstall
✓	Durable Task Plugin Library offering an extension point for processes which can run outside of Jenkins yet be monitored.	1.13		Uninstall
✓	Email Extension Plugin This plugin is a replacement for Jenkins's email publisher	2.57.2	Downgrade to 2.57.2	Uninstall
✓	External Monitor Job Type Plugin Adds the ability to monitor the result of externally executed jobs	1.7		Uninstall
✓	Folders Plugin This plugin allows users to create "folders" to organize jobs. Users can define custom taxonomies (like by project type, organization type etc). Folders are nestable and you can define views within folders. Maintained by CloudBees, Inc.	6.0.4		Uninstall
✓	Git client plugin Utility plugin for Git support in Jenkins	2.4.6		Uninstall
✓	Git plugin This plugin integrates Git with Jenkins.	3.3.0	Downgrade to 3.3.0	Uninstall
✓	GIT server Plugin Allows Jenkins to act as a Git server.	1.7		Uninstall
✓	GitHub API Plugin This plugin provides GitHub API for other plugins.	1.85.1		Uninstall

jenkins_Home → Manage Jenkins → Global Tool Configurations

As per our requirement we need to configure JAVA,Git-Hub,Ant.


localhost:8080/manage

Jenkins › ENABLE AUTO REFRESH

People
Build History
Manage Jenkins
My Views
Credentials

Build Queue
No builds in the queue.

Build Executor Status
1 Idle
2 Idle

Restore the previous version of Jenkins Downgrade to 2.63

Configure System
Configure global settings and paths.

Configure Global Security
Secure Jenkins; define who is allowed to access/use the system.

Configure Credentials
Configure the credential providers and types

Global Tool Configuration
Configure tools, their locations and automatic installers.

Reload Configuration from Disk
Discard all the loaded data in memory and reload everything from file system. Useful when you modified config files directly on disk.

Manage Plugins
Add, remove, disable or enable plugins that can extend the functionality of Jenkins.

System Information
Displays various environmental information to assist trouble-shooting.

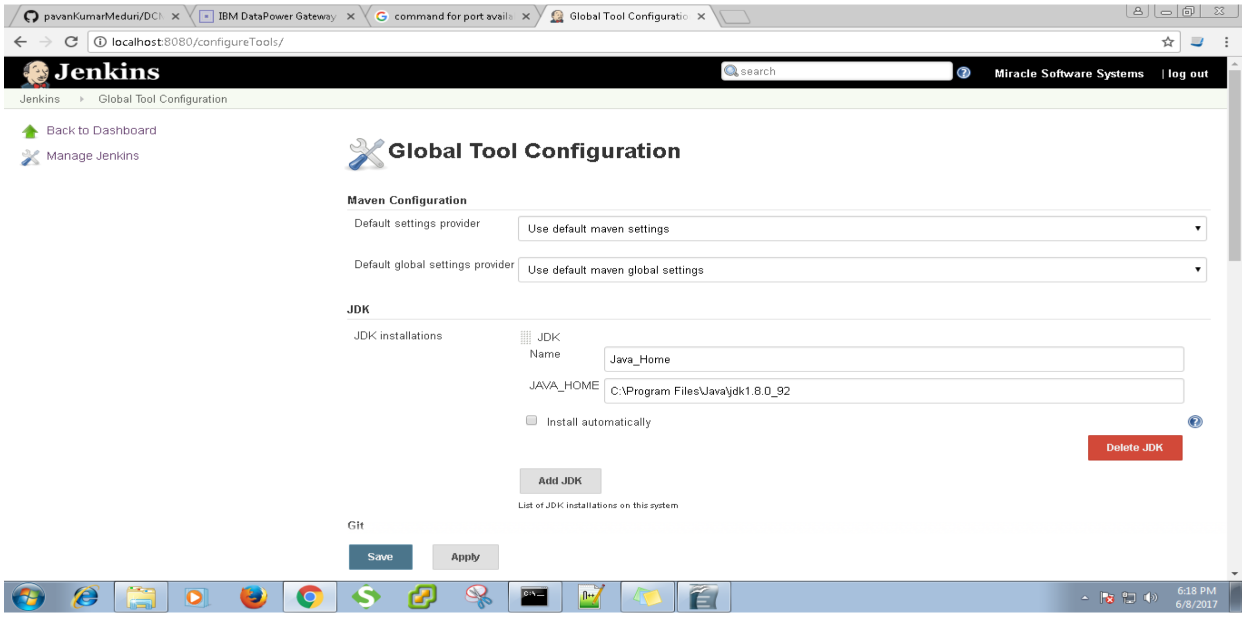
System Log
System log captures output from `java.util.logging` output related to Jenkins.

Load Statistics
Check your resource utilization and see if you need more computers for your builds.

Jenkins CLI
Access/manage Jenkins from your shell, or from your script.

Script Console
Executes arbitrary script for administration/trouble-shooting/diagnostics.

localhost:8080/pluginManager



localhost:8080/configureTools/

Jenkins

Jenkins › Global Tool Configuration

Back to Dashboard
Manage Jenkins

Global Tool Configuration

Maven Configuration

Default settings provider: Use default maven settings

Default global settings provider: Use default maven global settings

JDK

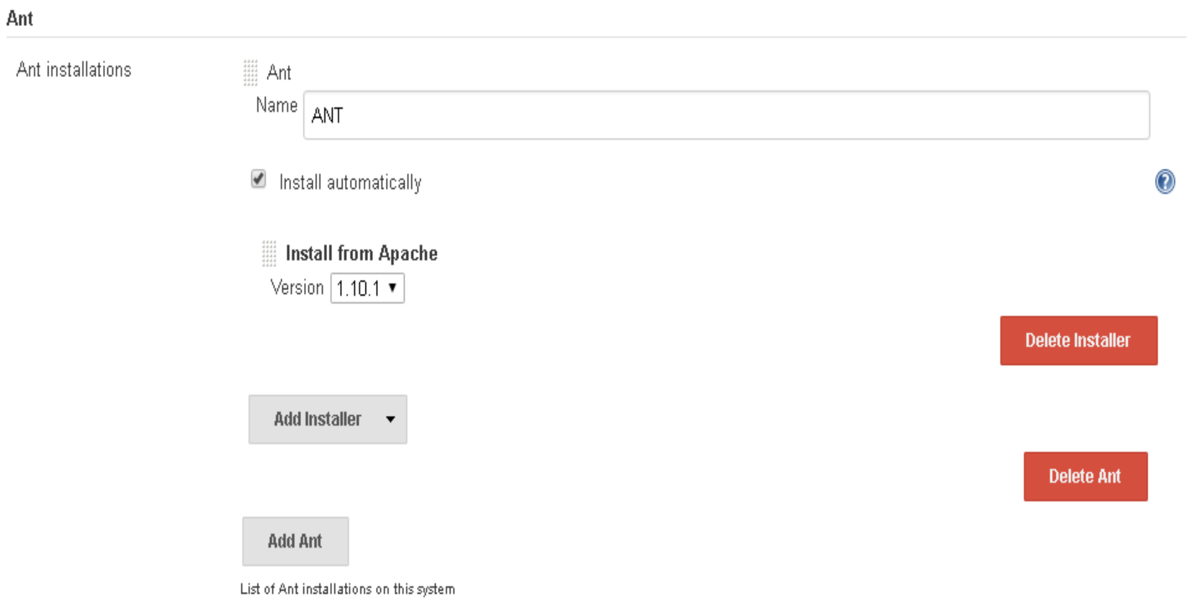
JDK installations

JDK
Name: Java_Home
JAVA_HOME: C:\Program Files\Java\jdk1.8.0_92
Install automatically

Delete JDK
Add JDK
List of JDK installations on this system

Git

Save Apply



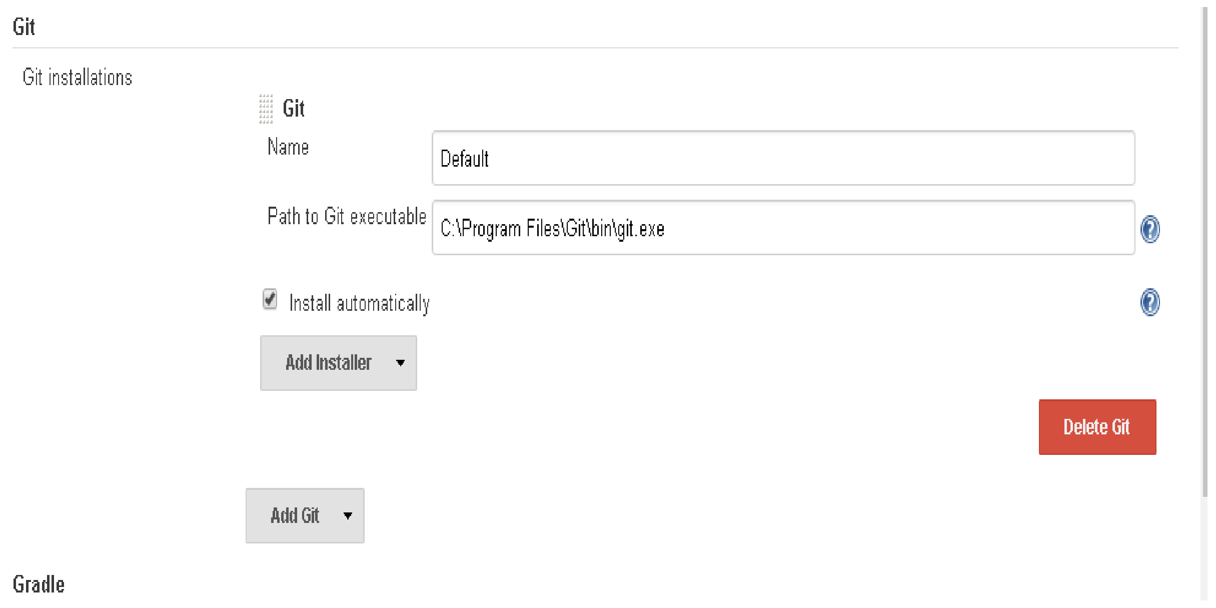
Ant

Ant installations

Ant
Name: ANT
[✓] Install automatically

Install from Apache
Version: 1.10.1

Delete Installer
Add Installer
Delete Ant
Add Ant
List of Ant installations on this system



Git

Git installations

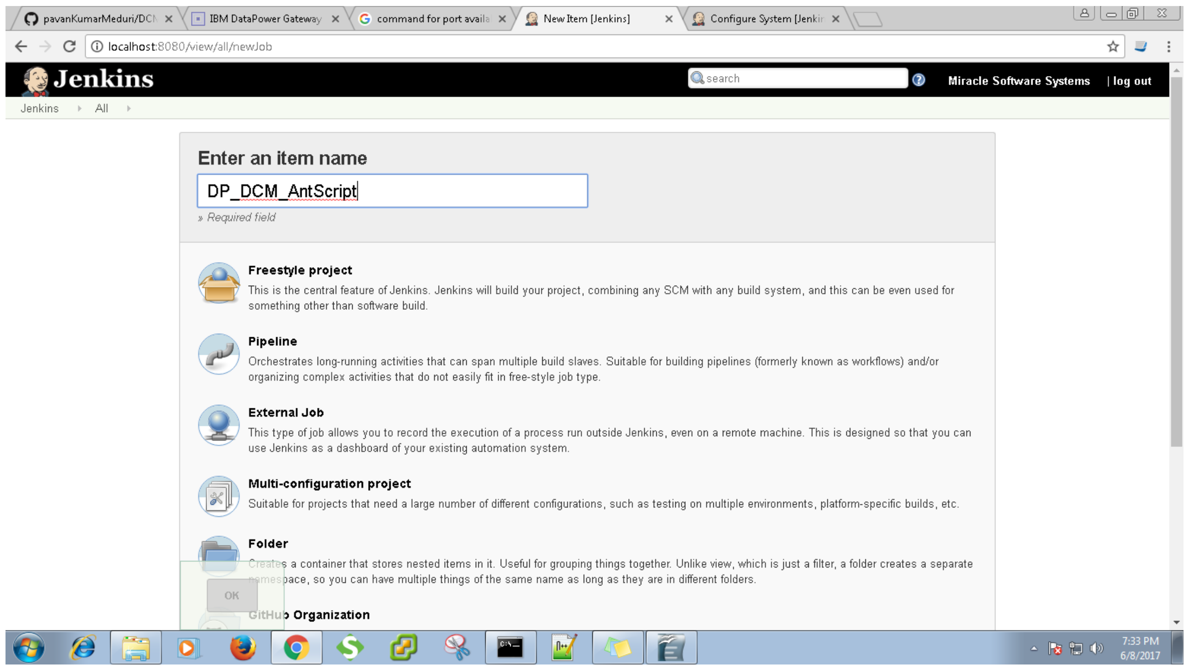
Git

Name: Default

Path to Git executable: C:\Program Files\Git\bin\git.exe

☑ Install automatically

To create a work space in jenkins
 Jenkin's Home → click on New Item



Enter an item name

DP_DCM_AntScript

» Required field

Freestyle project
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

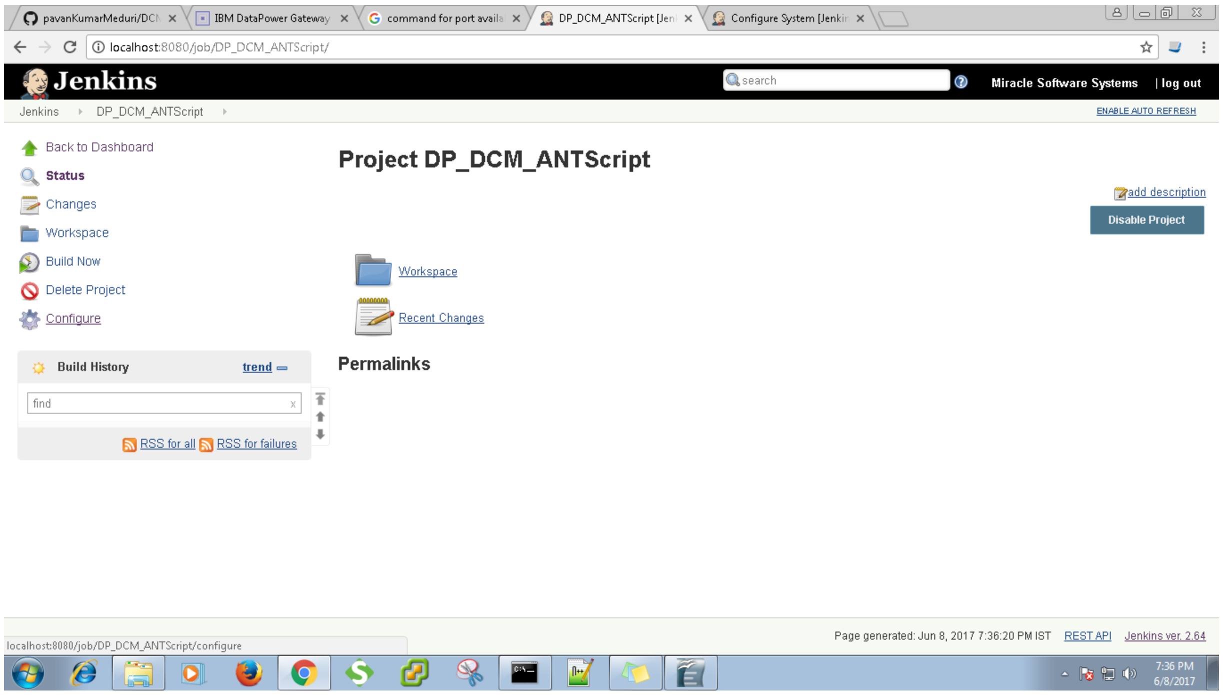
Pipeline
Orchestrates long-running activities that can span multiple build slaves. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

External Job
This type of job allows you to record the execution of a process run outside Jenkins, even on a remote machine. This is designed so that you can use Jenkins as a dashboard of your existing automation system.

Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

Folder
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

GitHub Organization


localhost:8080/job/DP_DCM_ANTScript/

Jenkins > DP_DCM_ANTScript

Project DP_DCM_ANTScript

- Back to Dashboard
- Status
- Changes
- Workspace
- Build Now
- Delete Project
- Configure

Build History trend

RSS for all RSS for failures

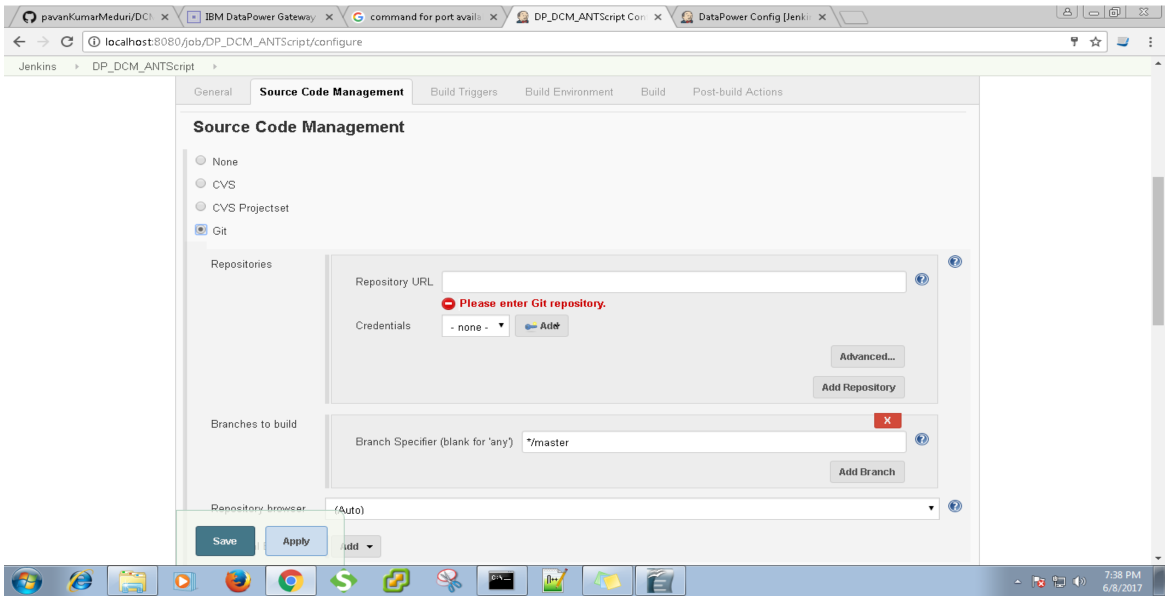
Workspace

Recent Changes

Permalinks

Page generated: Jun 8, 2017 7:36:20 PM IST REST API Jenkins ver. 2.64

After creating the work space we need to configure the work space, for that
Jenkins_Home → Workspace(required) → configure


localhost:8080/job/DP_DCM_ANTScript/configure

General | Source Code Management | Build Triggers | Build Environment | Build | Post-build Actions

Source Code Management

- None
- CVS
- CVS Projectset
- [✓ Git]

Repositories

Repository URL

Please enter Git repository.

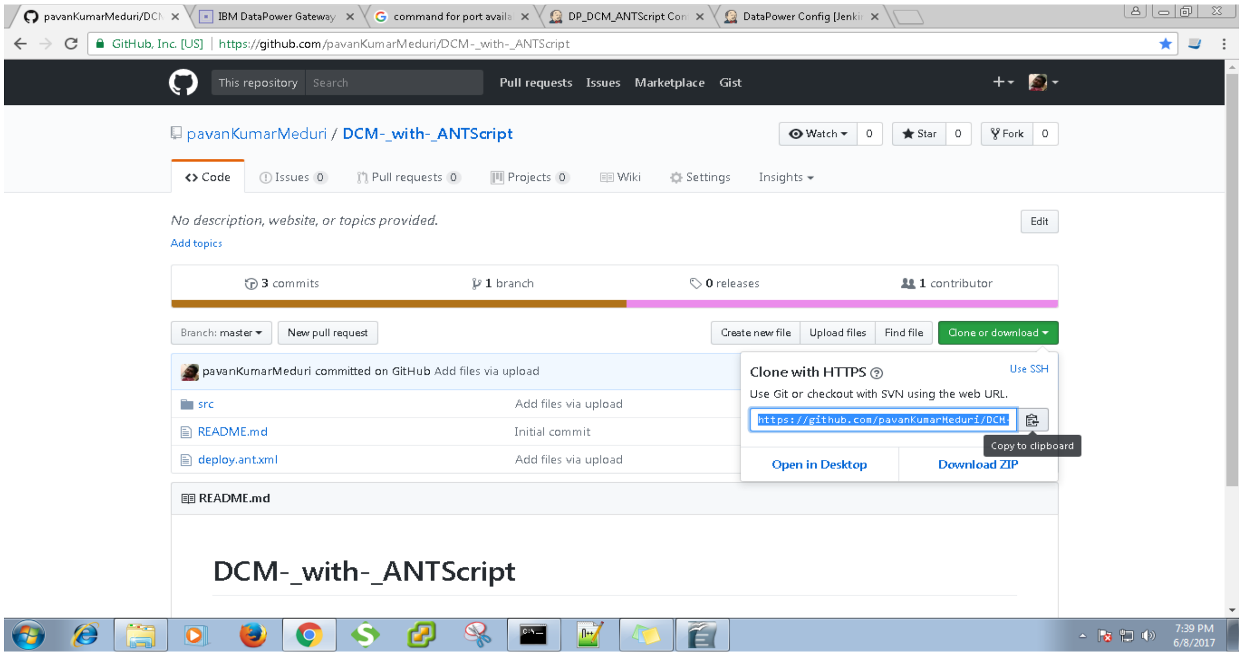
Credentials - none -

Branches to build

Branch Specifier (blank for 'any') */master

Repository browser (Auto)

The Git Repository URL can be taken from the GitHub Repository


pavanKumarMeduri / DCM-_with-_ANTScript

https://github.com/pavanKumarMeduri/DCM-_with-_ANTScript

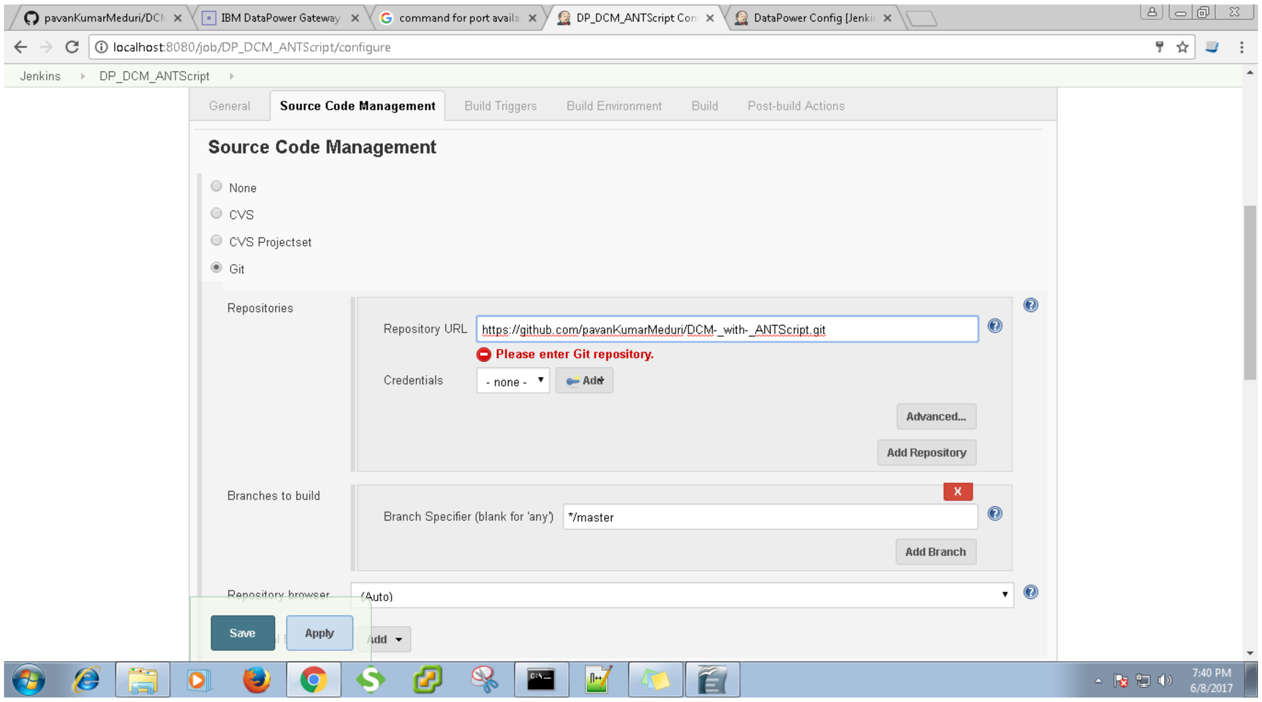
Clone with HTTPS

Use Git or checkout with SVN using the web URL.

https://github.com/pavanKumarMeduri/DCM-

src	Add files via upload
README.md	Initial commit
deploy.ant.xml	Add files via upload

DCM-_with-_ANTScript


localhost:8080/job/DP_DCM_ANTScript/configure

Source Code Management

None

CVS

CVS Projectset

[✓ Git]

Repositories

Repository URL: https://github.com/pavanKumarMeduri/DCM-_with-_ANTScript.git

Please enter Git repository.

Credentials: - none -

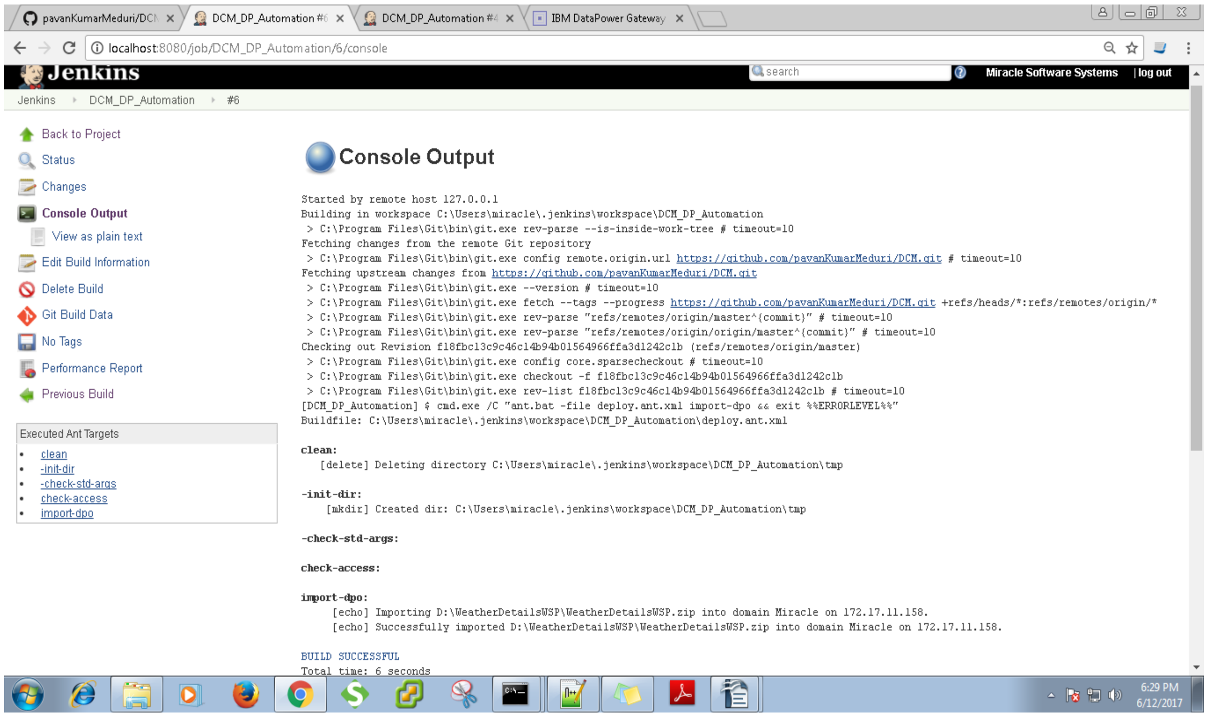
Branches to build

Branch Specifier (blank for 'any'): */master

Repository browser: (Auto)

In the Configure section click on BUILD option,in that we need specify the target name which we want to execute.

Next move to Jenkins_Home → Workspace → click on **Build Now** option. Then the console output should be like this.



Console Output

```
Started by remote host 127.0.0.1
Building in workspace C:\Users\miracle\.jenkins\workspace\DCM_DP_Automation
 > C:\Program Files\Git\bin\git.exe rev-parse --is-inside-work-tree # timeout=10
Fetching changes from the remote Git repository
 > C:\Program Files\Git\bin\git.exe config remote.origin.url https://github.com/pavanKumarMeduri/DCM.git # timeout=10
Fetching upstream changes from https://github.com/pavanKumarMeduri/DCM.git
 > C:\Program Files\Git\bin\git.exe --version # timeout=10
 > C:\Program Files\Git\bin\git.exe fetch --tags --progress https://github.com/pavanKumarMeduri/DCM.git +refs/heads/*:refs/remotes/origin/*
 > C:\Program Files\Git\bin\git.exe rev-parse "refs/remotes/origin/master^{commit}" # timeout=10
 > C:\Program Files\Git\bin\git.exe rev-parse "refs/remotes/origin/origin/master^{commit}" # timeout=10
Checking out Revision f18fbc13c9c46c14b94b01564966ffa3d1242c1b (refs/remotes/origin/master)
 > C:\Program Files\Git\bin\git.exe config core.sparsecheckout # timeout=10
 > C:\Program Files\Git\bin\git.exe checkout -f f18fbc13c9c46c14b94b01564966ffa3d1242c1b
 > C:\Program Files\Git\bin\git.exe rev-list f18fbc13c9c46c14b94b01564966ffa3d1242c1b # timeout=10
[DCM_DP_Automation] $ cmd.exe /C "ant.bat -file deploy.ant.xml import-dpo && exit %%ERRORLEVEL%%"
Buildfile: C:\Users\miracle\.jenkins\workspace\DCM_DP_Automation\deploy.ant.xml

clean:
   [delete] Deleting directory C:\Users\miracle\.jenkins\workspace\DCM_DP_Automation\tmp

-init-dir:
    [mkdir] Created dir: C:\Users\miracle\.jenkins\workspace\DCM_DP_Automation\tmp

-check-std-args:

check-access:

import-dpo:
     [echo] Importing D:\WeatherDetailsWSP\WeatherDetailsWSP.zip into domain Miracle on 172.17.11.158.
     [echo] Successfully imported D:\WeatherDetailsWSP\WeatherDetailsWSP.zip into domain Miracle on 172.17.11.158.

BUILD SUCCESSFUL
Total time: 6 seconds
```

Till now we deployed a service from the one Domain to another domain in the Data Power by manually.

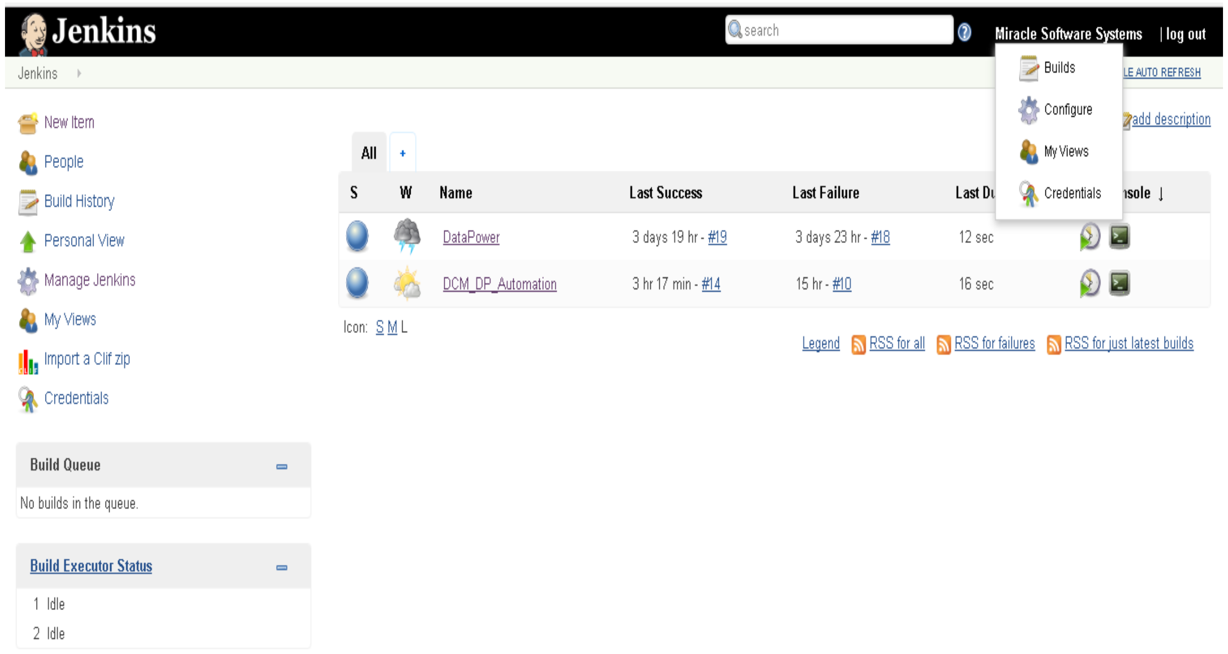
As of our flow we need to automate the entire process for that we need to configure the Jenkins and give the command through CURL. To do this we need **Release Management Process** .

5.Release Management Process

Release management is the process of **managing**, planning, scheduling and controlling software build through different stages and environments; including testing and deploying software releases

for that we need to collect the API Token from the jenkins.To get the API Token

Jenkins_Home → Click on Name of jenkins → Configure



S	W	Name	Last Success	Last Failure	Last Du	nsole ↓
●		DataPower	3 days 19 hr - #19	3 days 23 hr - #18	12 sec	
●		DCM_DP_Automation	3 hr 17 min - #14	15 hr - #10	16 sec	

Icon: S M L

Legend RSS for all RSS for failures RSS for just latest builds

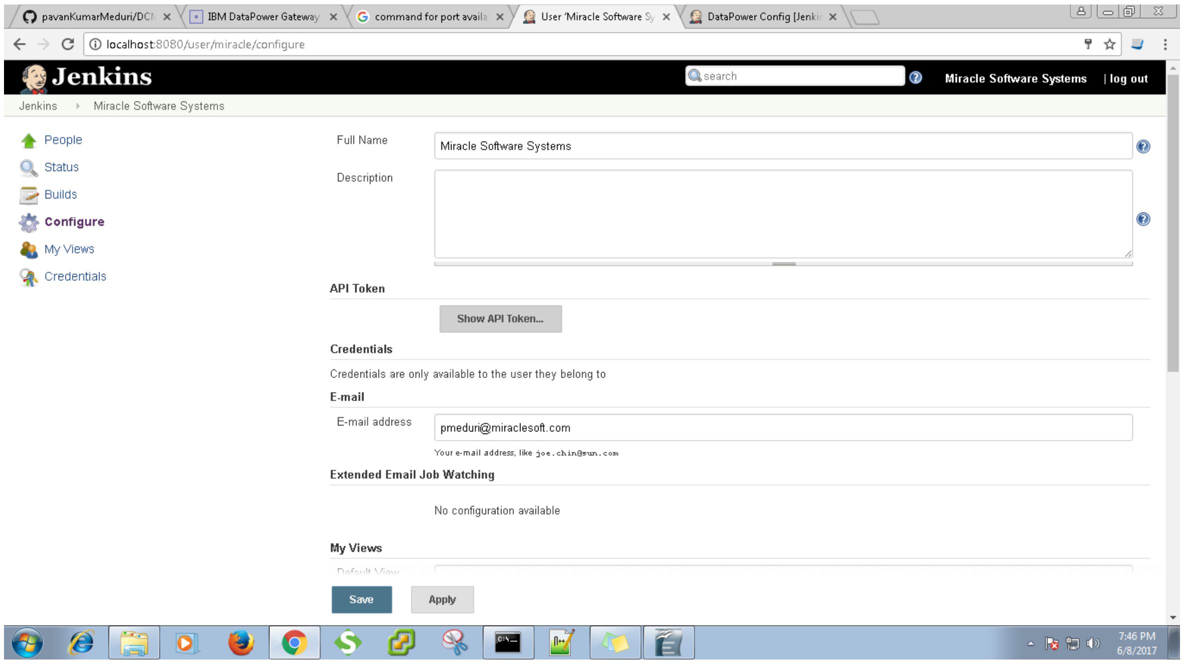
Build Queue

No builds in the queue.

Build Executor Status

1 Idle

2 Idle


localhost:8080/user/miracle/configure

Jenkins > Miracle Software Systems

Full Name: Miracle Software Systems

Description:

API Token

Show API Token...

Credentials

Credentials are only available to the user they belong to

E-mail

E-mail address: pmeduri@miraclesoft.com

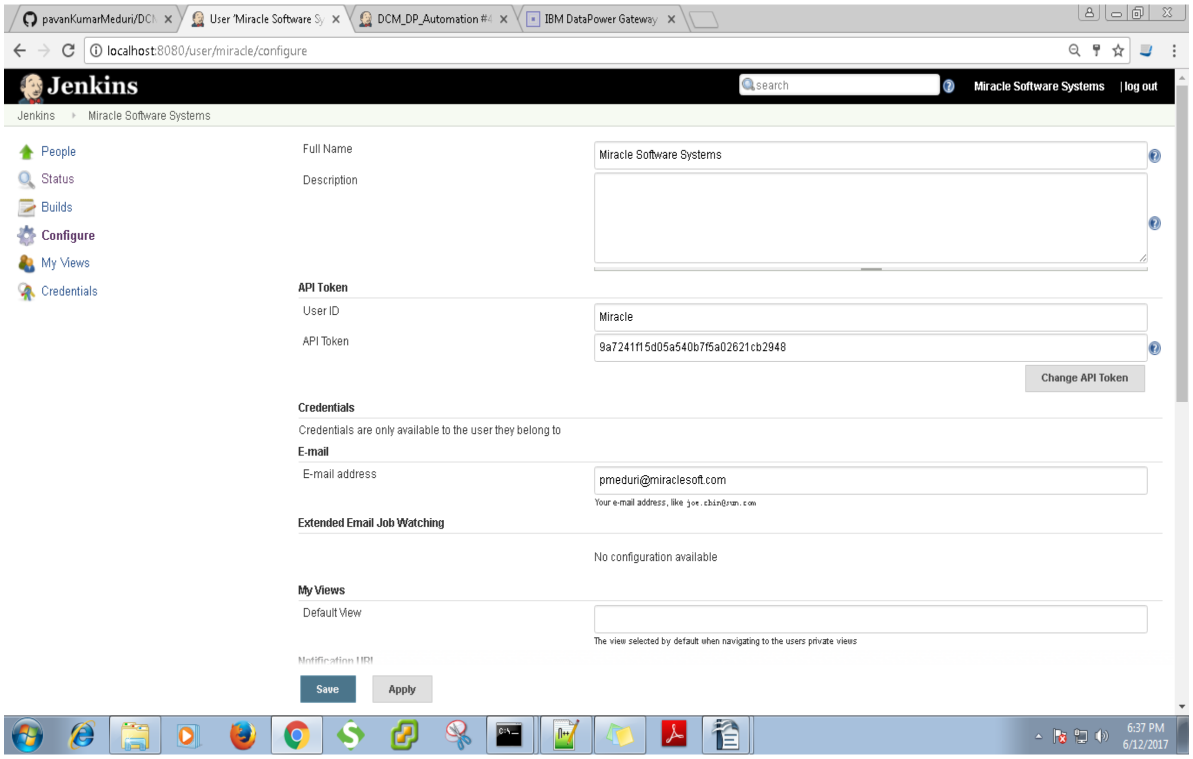
Your e-mail address, like joe.chin@sun.com

Extended Email Job Watching

No configuration available

My Views

Save Apply


localhost:8080/user/miracle/configure

Jenkins > Miracle Software Systems

Full Name: Miracle Software Systems

Description:

API Token

User ID: Miracle

API Token: 9a7241f15d05a540b7f5a02621cb2962

Change API Token

Credentials

Credentials are only available to the user they belong to

E-mail

E-mail address: pmeduri@miraclesoft.com

Your e-mail address, like joe.chin@sun.com

Extended Email Job Watching

No configuration available

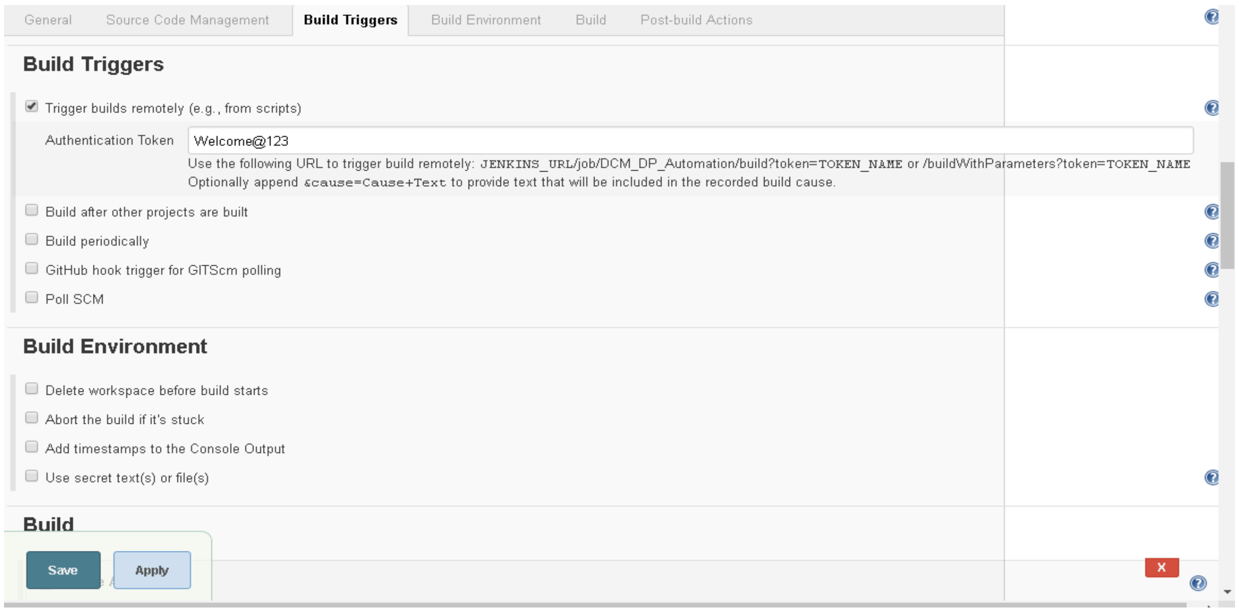
My Views

Default View

The view selected by default when navigating to the users private views

Save Apply

Note: Here we need to take User ID and API Token.
Next move to Jenkins_Home → workspace → click on Build Trigger



Build Triggers

- [✓] Trigger builds remotely (e.g., from scripts)
  - Authentication Token: Welcome@123
  - Use the following URL to trigger build remotely: `JENKINS_URL/job/DCM_DP_Automation/build?token=TOKEN_NAME` or `/buildWithParameters?token=TOKEN_NAME`
  - Optionally append `&cause=Cause+Text` to provide text that will be included in the recorded build cause.
- [ ] Build after other projects are built
- [ ] Build periodically
- [ ] GitHub hook trigger for GITScm polling
- [ ] Poll SCM

Build Environment

- [ ] Delete workspace before build starts
- [ ] Abort the build if it's stuck
- [ ] Add timestamps to the Console Output
- [ ] Use secret text(s) or file(s)

Build

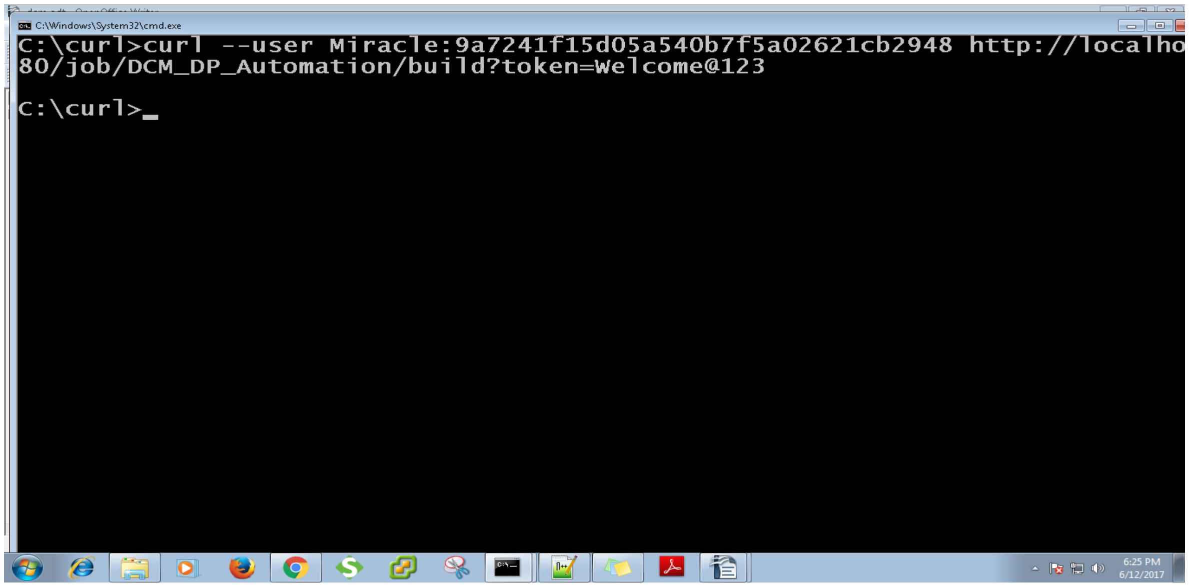
from the above we need to frame the command for the auto triggering the job in Jenkins

e.g:Use the following URL to trigger build

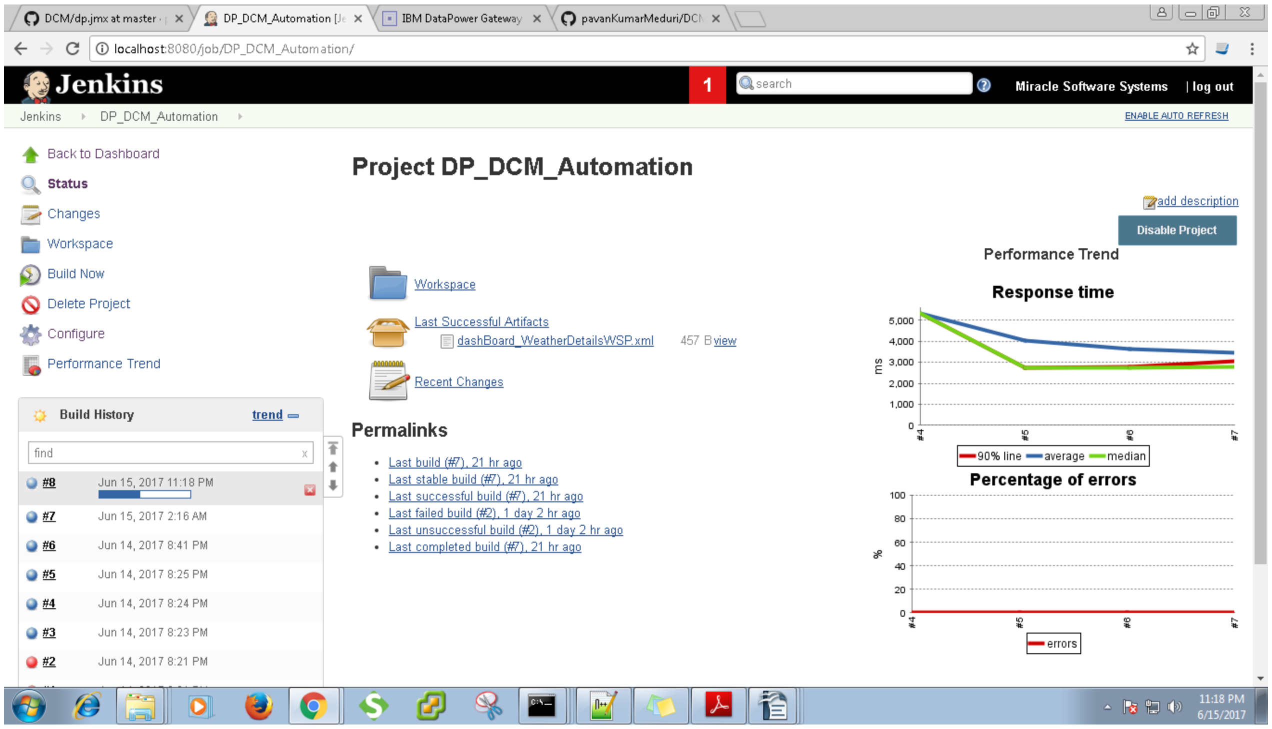
remotely:JENKINS_URL/job/DCM_DP_Automation/build?token=TOKEN_NAME.

Here is the command :

curl –user Miracle: 9a7241f15d05a540b7f5a02621cb2948
http://localhost:8080/job/DCM_DP_Automation/build?token=Welcome@123



```
C:\curl>curl --user Miracle:9a7241f15d05a540b7f5a02621cb2948 http://localho
80/job/DCM_DP_Automation/build?token=Welcome@123

C:\curl>_
```



localhost:8080/job/DP_DCM_Automation/

Jenkins › DP_DCM_Automation

Project DP_DCM_Automation

- Back to Dashboard
- Status
- Changes
- Workspace
- Build Now
- Delete Project
- Configure
- Performance Trend

Build History

#8	Jun 15, 2017 11:18 PM
#7	Jun 15, 2017 2:16 AM
#6	Jun 14, 2017 8:41 PM
#5	Jun 14, 2017 8:25 PM
#4	Jun 14, 2017 8:24 PM
#3	Jun 14, 2017 8:23 PM
#2	Jun 14, 2017 8:21 PM

Workspace

Last Successful Artifacts
dashBoard_WeatherDetailsWSP.xml 457 B view

Recent Changes

Permalinks

- Last build (#7), 21 hr ago
- Last stable build (#7), 21 hr ago
- Last successful build (#7), 21 hr ago
- Last failed build (#2), 1 day 2 hr ago
- Last unsuccessful build (#2), 1 day 2 hr ago
- Last completed build (#7), 21 hr ago

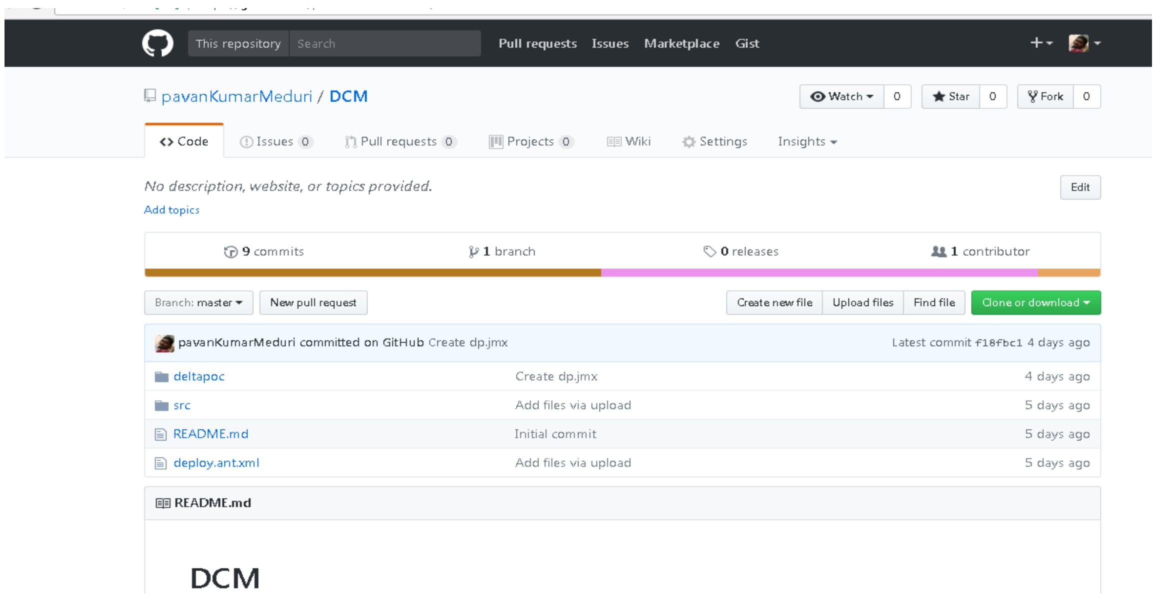
By giving this command we are able to build a job automatically in jenkins

6.Jmeter

Apache Jmeter is an Apache project that can be used as a load testing tool for analyzing and measuring the performance of a variety of services,with a focus on web applications.
Download the Apache Jmeter from the below link

http://jmeter.apache.org/download_jmeter.cgi

when we test the target endpoint in the Jmeter we will get the **.jtl** , **.jml** and **.csv** files these files pushed into the GitHub by the name **deltapoc.**



No description, website, or topics provided.

9 commits | 1 branch | 0 releases | 1 contributor

pavanKumarMeduri committed on GitHub Create dp.jmx — Latest commit f18fbc1 4 days ago

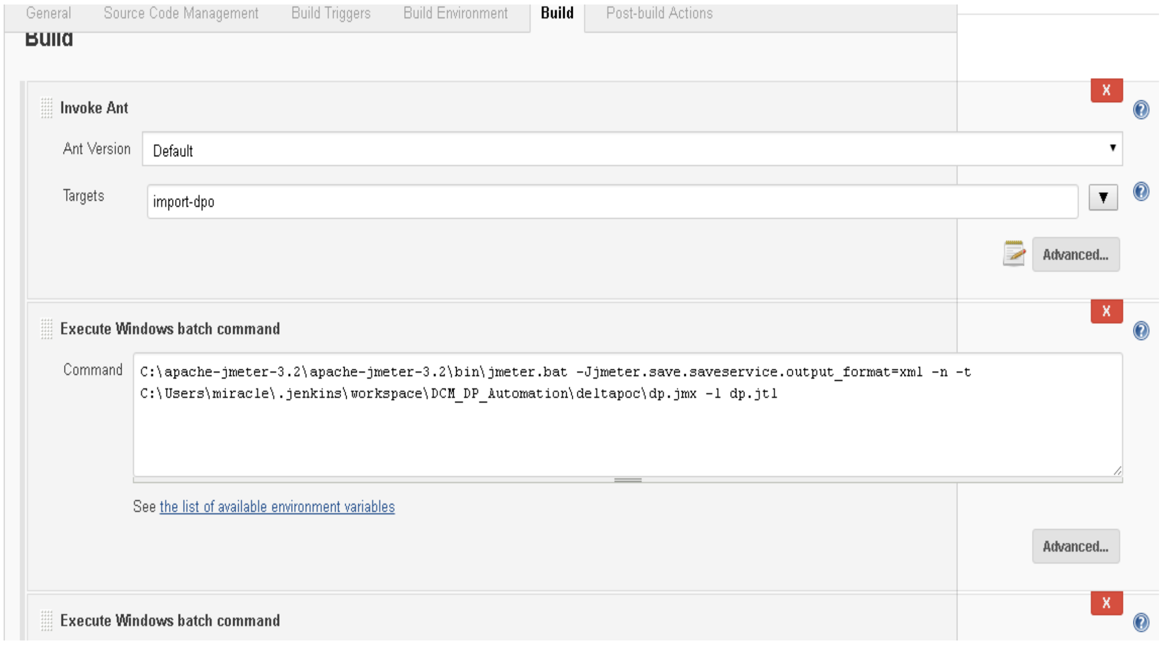
deltapoc	Create dp.jmx	4 days ago
src	Add files via upload	5 days ago
README.md	Initial commit	5 days ago
deploy.ant.xml	Add files via upload	5 days ago

README.md

DCM

For this we need to move
Jenkins_Home → Workspace → Configure → Build → Execute windows batch command

Here we need to give
- ✓ In which path jmeter is working
- ✓ Jmeter output format
- ✓ File location from the .jenkins folder



General | Source Code Management | Build Triggers | Build Environment | **Build** | Post-build Actions

Build

Invoke Ant

Ant Version: Default

Targets: import-dpo

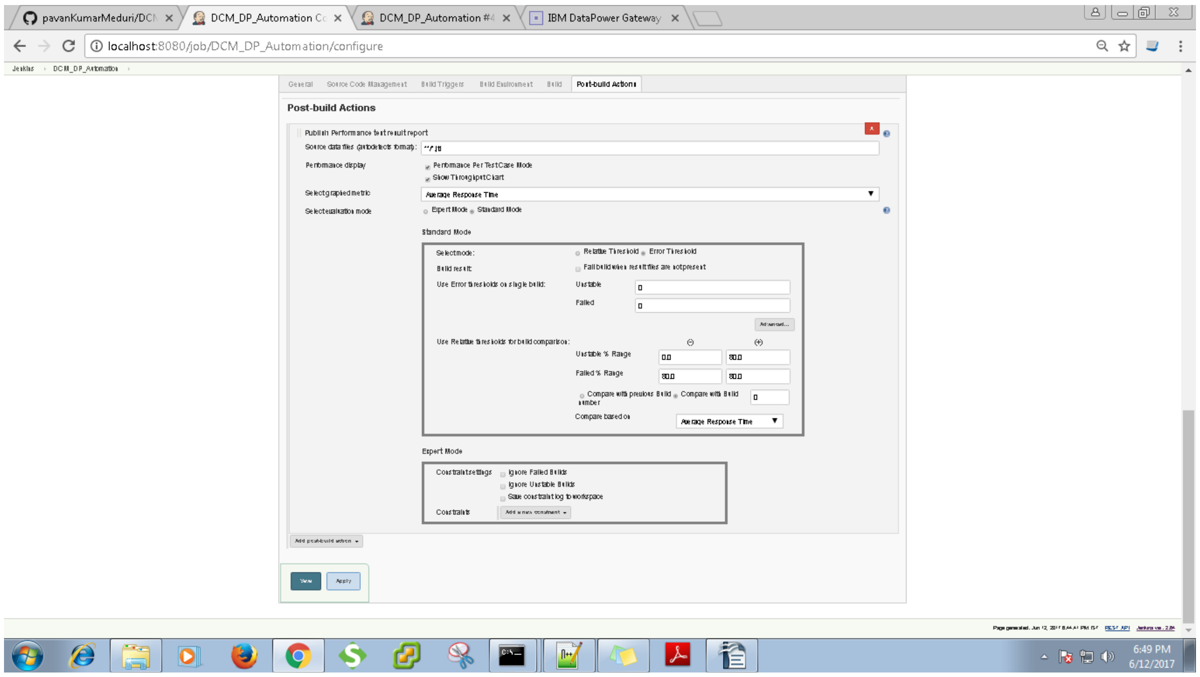
Execute Windows batch command

Command:

```
C:\apache-jmeter-3.2\apache-jmeter-3.2\bin\jmeter.bat -Jjmeter.save.saveservice.output_format=xml -n -t
C:\Users\miracle\.jenkins\workspace\DCM_DP_Automation\deltapoc\dp.jmx -l dp.jtl
```

See the list of available environment variables

Execute Windows batch command



localhost:8080/job/DCM_DP_Automation/configure

General | Source Code Management | Build Triggers | Build Environment | Build | **Post-build Actions**

Post-build Actions

Publish Performance test result report

Source data files (autodetects format): **/*.jtl

Performance display: [✓] Performance Per TestCase Mode; [✓] Show Throughput Chart

Select graphed metric: Average Response Time

Select evaluation mode: Expert Mode [✓ Standard Mode]

Standard Mode

Select mode: Relative Threshold [✓ Error Threshold]

Build result: Fail build when result files are not present

Use Error thresholds on single build: Unstable 0; Failed 0

Use Relative thresholds for build comparison:

	(-)	(+)
Unstable % Range	0.0	80.0
Failed % Range	80.0	80.0

Compare with previous Build; Compare with Build number: 0

Compare based on: Average Response Time

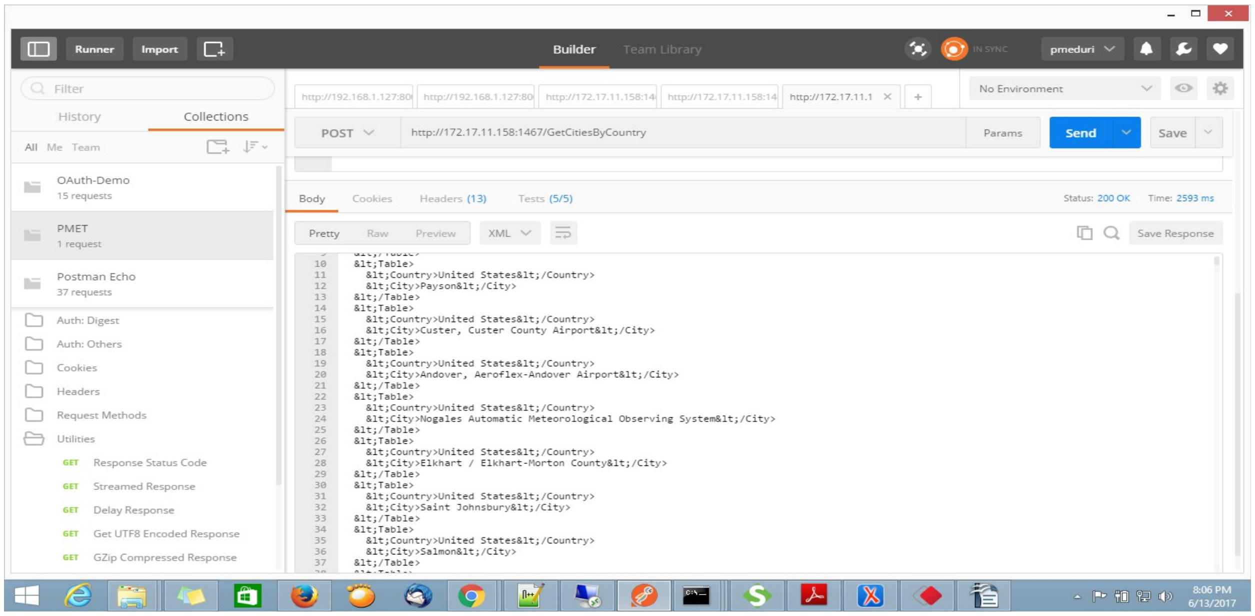
Expert Mode

Constraint settings: Ignore Failed Builds; Ignore Unstable Builds; Save constraint log to workspace

Constraints: Add a new constraint

6.Testing Automation:

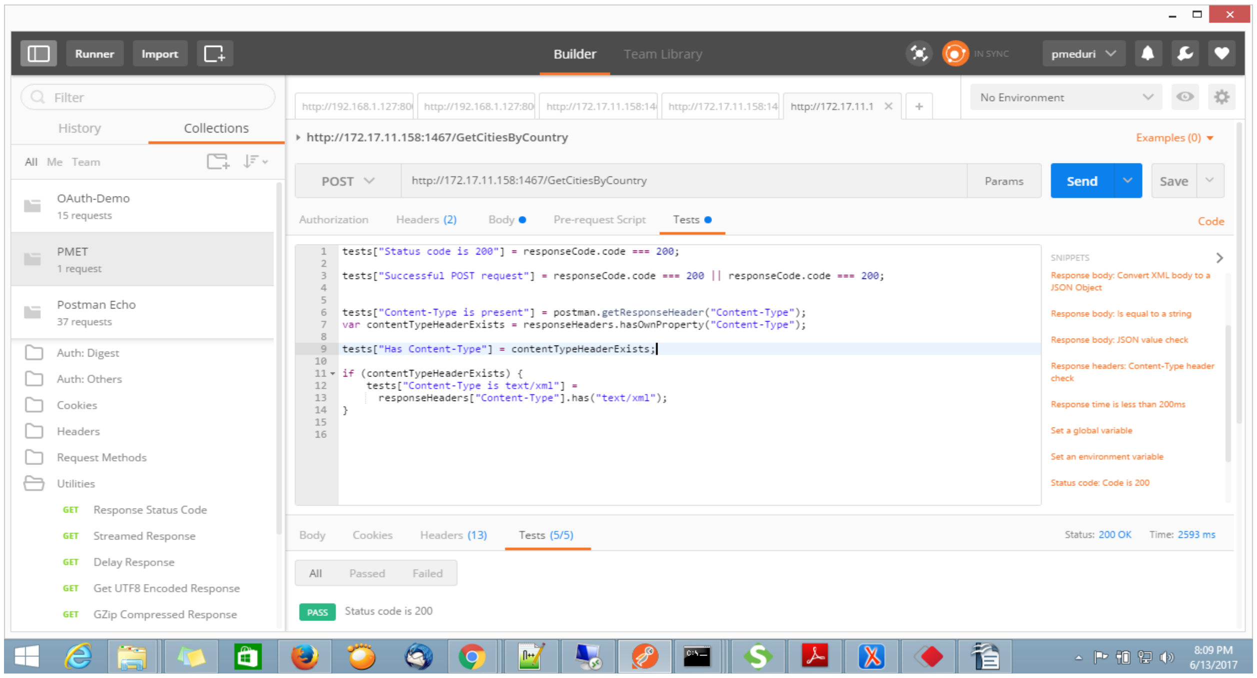
Before going to automation we need write test cases in postman and then we can export the collection.



POST http://172.17.11.158:1467/GetCitiesByCountry

Body · Cookies · Headers (13) · Tests (5/5) · Status: 200 OK · Time: 2593 ms

```
&lt;Table>
  &lt;Country>United States&lt;/Country>
  &lt;City>Payson&lt;/City>
&lt;/Table>
&lt;Table>
  &lt;Country>United States&lt;/Country>
  &lt;City>Custer, Custer County Airport&lt;/City>
&lt;/Table>
&lt;Table>
  &lt;Country>United States&lt;/Country>
  &lt;City>Andover, Aeroflex-Andover Airport&lt;/City>
&lt;/Table>
&lt;Table>
  &lt;Country>United States&lt;/Country>
  &lt;City>Nogales Automatic Meteorological Observing System&lt;/City>
&lt;/Table>
&lt;Table>
  &lt;Country>United States&lt;/Country>
  &lt;City>Elkhart / Elkhart-Morton County&lt;/City>
&lt;/Table>
&lt;Table>
  &lt;Country>United States&lt;/Country>
  &lt;City>Saint Johnsbury&lt;/City>
&lt;/Table>
&lt;Table>
  &lt;Country>United States&lt;/Country>
  &lt;City>Salmon&lt;/City>
&lt;/Table>
```



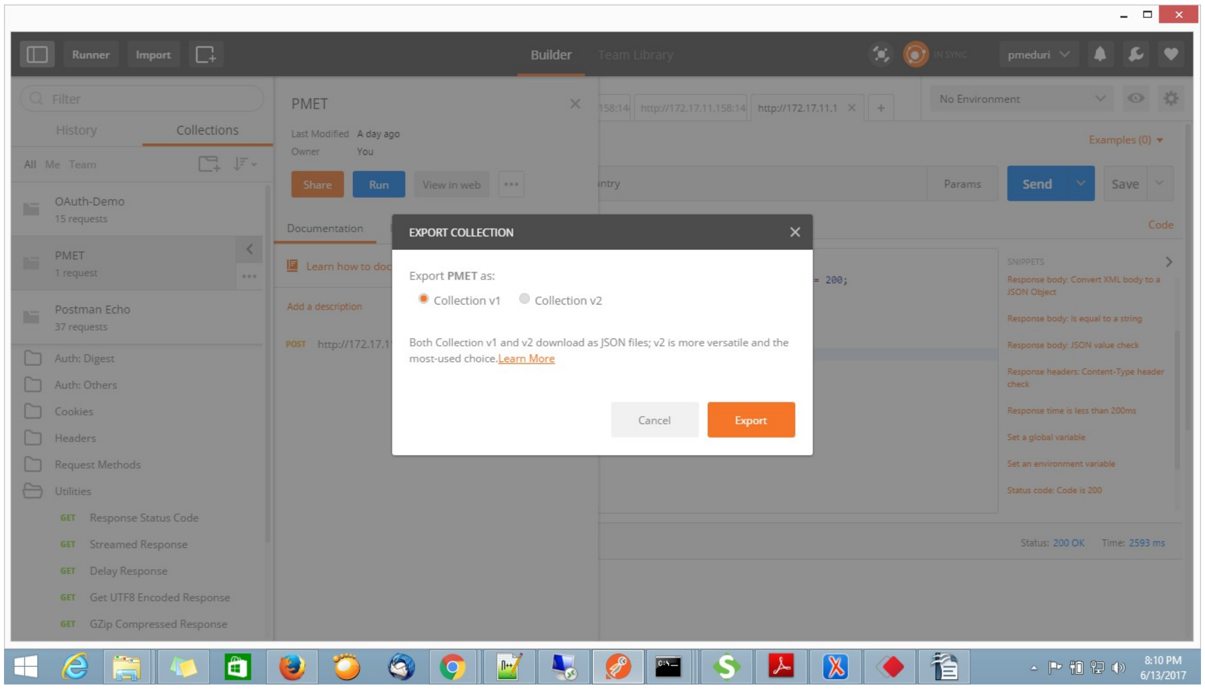
POST http://172.17.11.158:1467/GetCitiesByCountry

Authorization · Headers (2) · Body · Pre-request Script · Tests

```
tests["Status code is 200"] = responseCode.code === 200;

tests["Successful POST request"] = responseCode.code === 200 || responseCode.code === 200;


tests["Content-Type is present"] = postman.getResponseHeader("Content-Type");
var contentTypeHeaderExists = responseHeaders.hasOwnProperty("Content-Type");

tests["Has Content-Type"] = contentTypeHeaderExists;

if (contentTypeHeaderExists) {
    tests["Content-Type is text/xml"] =
      responseHeaders["Content-Type"].has("text/xml");
}
```

Body · Cookies · Headers (13) · Tests (5/5) · Status: 200 OK · Time: 2593 ms

PASS Status code is 200


EXPORT COLLECTION

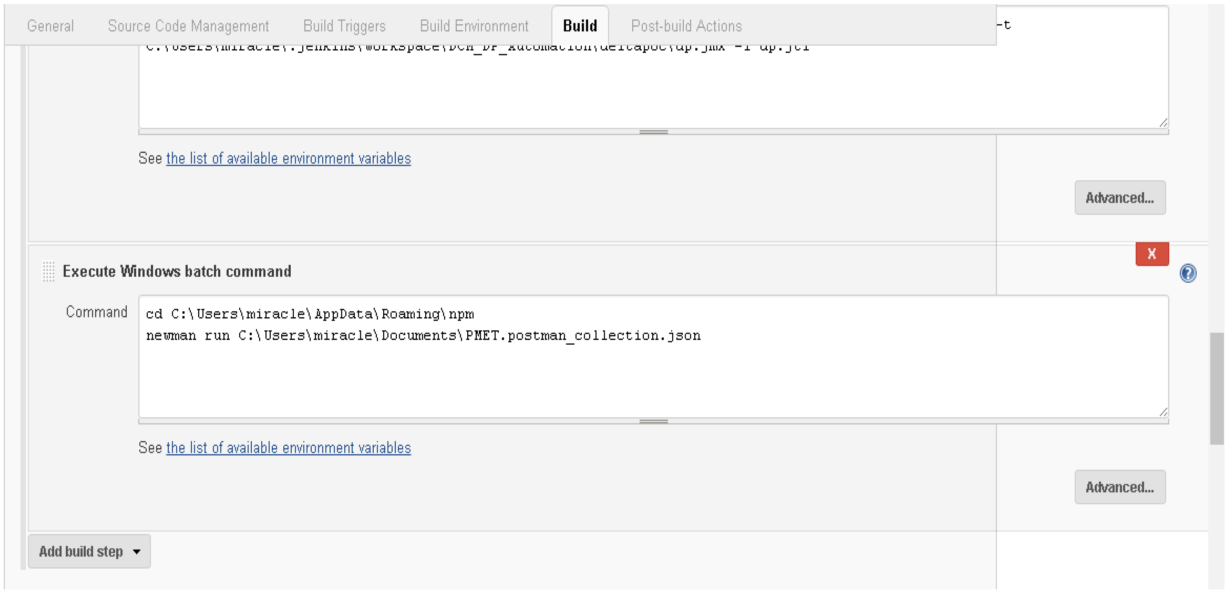
Export PMET as:

[✓ Collection v1] Collection v2

Both Collection v1 and v2 download as JSON files; v2 is more versatile and the most-used choice. Learn More

The collection will automatically saved into Documents folder.

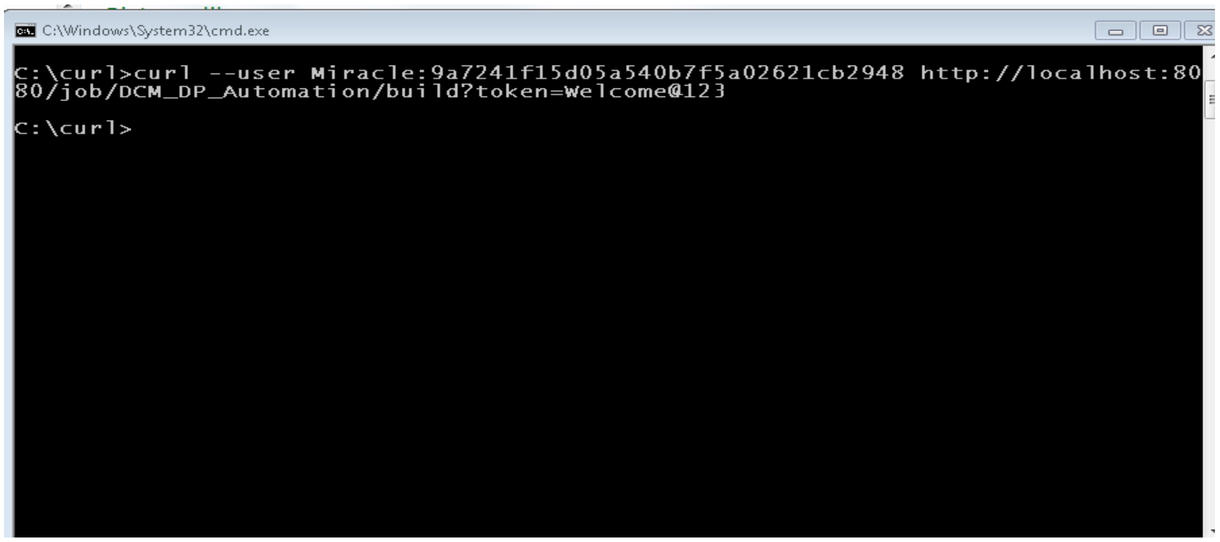
Now we need to configure the jenkins
Jenkins_Home → Workspace → configure → Build → run through Windows batch command


Execute Windows batch command

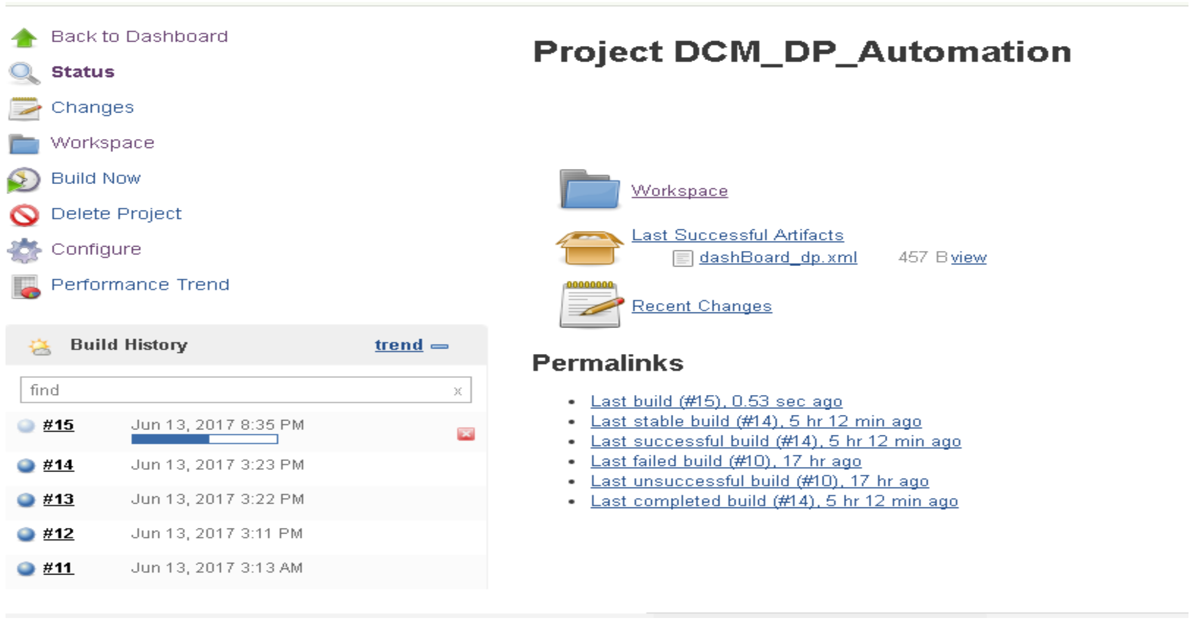
Command

```
cd C:\Users\miracle\AppData\Roaming\npm
newman run C:\Users\miracle\Documents\PMET.postman_collection.json
```

To execute this we need install NEWMAN in our local system using Nodejs.
Here is the command for test the target url through newman
cd C:\Users\miracle\AppData\Roaming\npm
newman run C:\Users\miracle\Documents\PMET.postman_collection.json
After completion these steps we need to give the command in curl command prompt.



```
C:\curl>curl  --user Miracle:9a7241f15d05a540b7f5a02621cb2948 http://localhost:80
80/job/DCM_DP_Automation/build?token=Welcome@123

C:\curl>
```



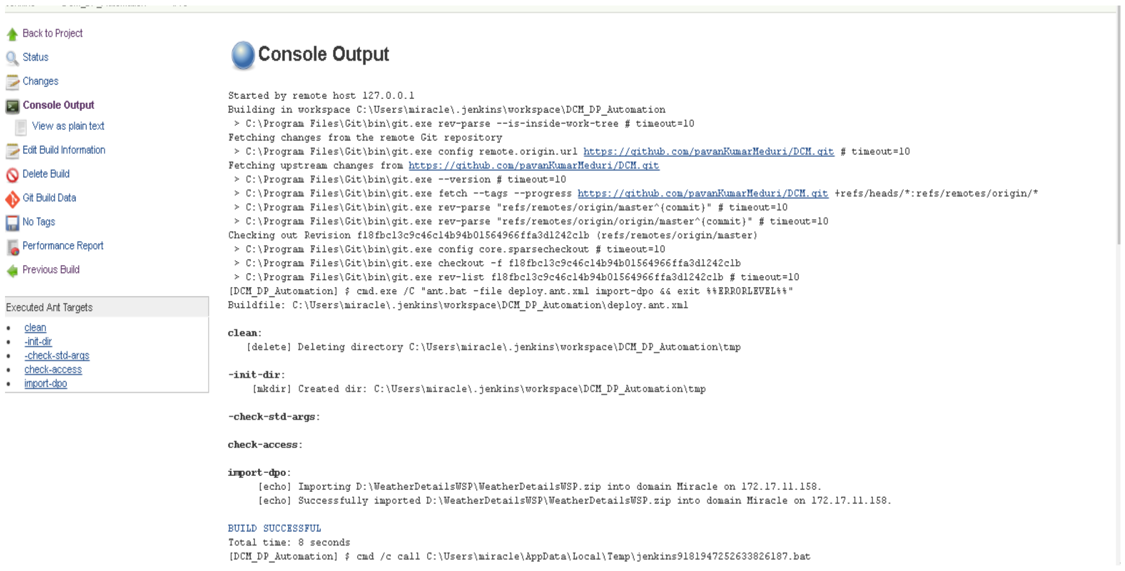
Project DCM_DP_Automation

dashBoard_dp.xml 457 B view

#15	Jun 13, 2017 8:35 PM
#14	Jun 13, 2017 3:23 PM
#13	Jun 13, 2017 3:22 PM
#12	Jun 13, 2017 3:11 PM
#11	Jun 13, 2017 3:13 AM

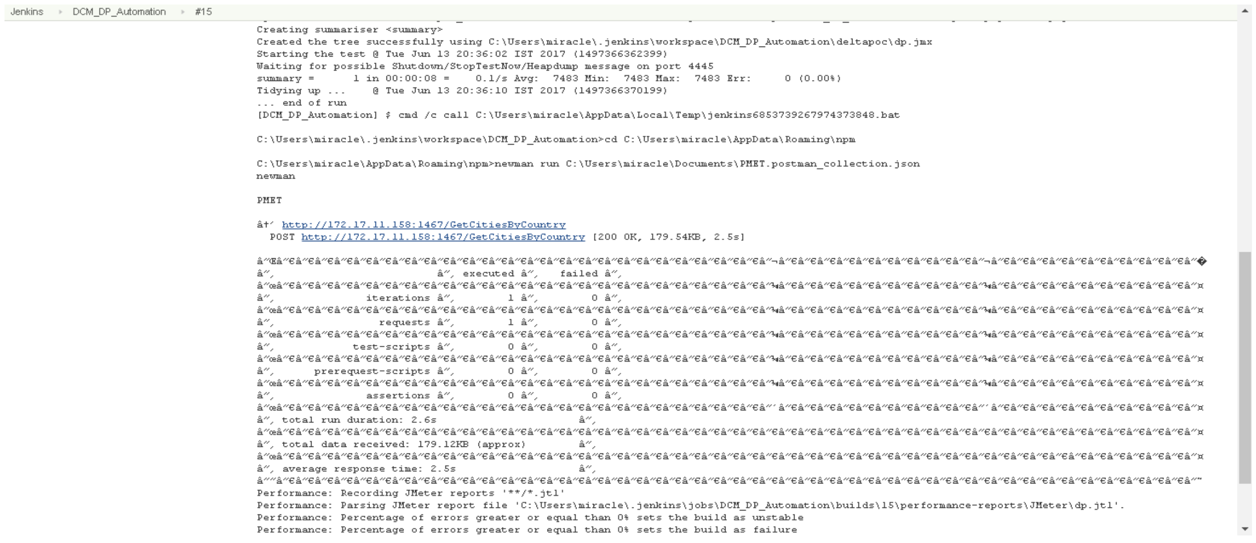
Permalinks

- Last build (#15), 0.53 sec ago
- Last stable build (#14), 5 hr 12 min ago
- Last successful build (#14), 5 hr 12 min ago
- Last failed build (#10), 17 hr ago
- Last unsuccessful build (#10), 17 hr ago
- Last completed build (#14), 5 hr 12 min ago

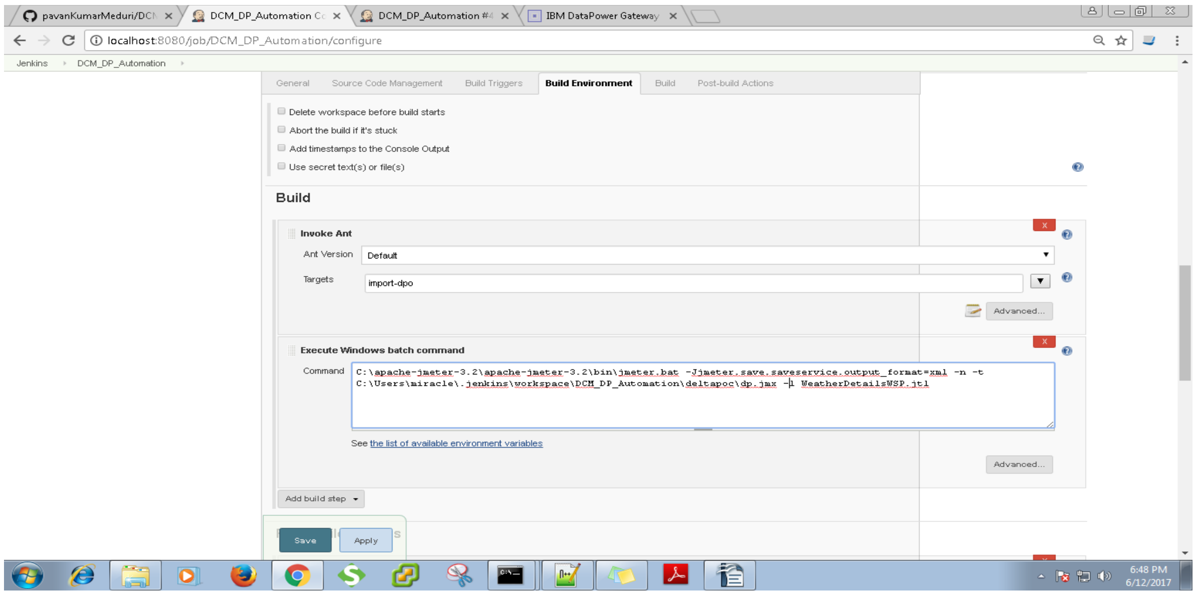


Console Output

```
Started by remote host 127.0.0.1
Building in workspace C:\Users\miracle\.jenkins\workspace\DCM_DP_Automation
 > C:\Program Files\Git\bin\git.exe rev-parse --is-inside-work-tree # timeout=10
Fetching changes from the remote Git repository
 > C:\Program Files\Git\bin\git.exe config remote.origin.url https://github.com/pavanKumarMeduri/DCM.git # timeout=10
Fetching upstream changes from https://github.com/pavanKumarMeduri/DCM.git
 > C:\Program Files\Git\bin\git.exe --version # timeout=10
 > C:\Program Files\Git\bin\git.exe fetch --tags --progress https://github.com/pavanKumarMeduri/DCM.git +refs/heads/*:refs/remotes/origin/*
 > C:\Program Files\Git\bin\git.exe rev-parse "refs/remotes/origin/master^{commit}" # timeout=10
 > C:\Program Files\Git\bin\git.exe rev-parse "refs/remotes/origin/origin/master^{commit}" # timeout=10
Checking out Revision f18fbc13c9c46c14b94b01564966ffa3d1242c1b (refs/remotes/origin/master)
 > C:\Program Files\Git\bin\git.exe config core.sparsecheckout # timeout=10
 > C:\Program Files\Git\bin\git.exe checkout -f f18fbc13c9c46c14b94b01564966ffa3d1242c1b
 > C:\Program Files\Git\bin\git.exe rev-list f18fbc13c9c46c14b94b01564966ffa3d1242c1b # timeout=10
[DCM_DP_Automation] $ cmd.exe /C "ant.bat -file deploy.ant.xml import-dpo && exit %%ERRORLEVEL%%"
Buildfile: C:\Users\miracle\.jenkins\workspace\DCM_DP_Automation\deploy.ant.xml

clean:
   [delete] Deleting directory C:\Users\miracle\.jenkins\workspace\DCM_DP_Automation\tmp

-init-dir:
    [mkdir] Created dir: C:\Users\miracle\.jenkins\workspace\DCM_DP_Automation\tmp

-check-std-args:

check-access:

import-dpo:
     [echo] Importing D:\WeatherDetailsWSP\WeatherDetailsWSP.zip into domain Miracle on 172.17.11.158.
     [echo] Successfully imported D:\WeatherDetailsWSP\WeatherDetailsWSP.zip into domain Miracle on 172.17.11.158.

BUILD SUCCESSFUL
Total time: 8 seconds
[DCM_DP_Automation] $ cmd /c call C:\Users\miracle\AppData\Local\Temp\jenkins9181947252633826187.bat
```



```
Creating summariser <summary>
Created the tree successfully using C:\Users\miracle\.jenkins\workspace\DCM_DP_Automation\deltapoc\dp.jmx
Starting the test @ Tue Jun 13 20:36:02 IST 2017 (1497366362399)
Waiting for possible Shutdown/StopTestNow/Heapdump message on port 4445
summary =      1 in 00:00:08 =    0.1/s Avg:  7483 Min:  7483 Max:  7483 Err:     0 (0.00%)
Tidying up ...    @ Tue Jun 13 20:36:10 IST 2017 (1497366370199)
... end of run
[DCM_DP_Automation] $ cmd /c call C:\Users\miracle\AppData\Local\Temp\jenkins6853739267974373848.bat

C:\Users\miracle\.jenkins\workspace\DCM_DP_Automation>cd C:\Users\miracle\AppData\Roaming\npm

C:\Users\miracle\AppData\Roaming\npm>newman run C:\Users\miracle\Documents\PMET.postman_collection.json
newman

PMET

â†’ http://172.17.11.158:1467/GetCitiesByCountry
  POST http://172.17.11.158:1467/GetCitiesByCountry [200 OK, 179.54KB, 2.5s]

â”Œâ”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”¬â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”¬â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”�
â”‚                         â”‚ executed â”‚   failed â”‚
â”œâ”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”¼â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”¼â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”¤
â”‚              iterations â”‚        1 â”‚        0 â”‚
â”œâ”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”¼â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”¼â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”¤
â”‚                requests â”‚        1 â”‚        0 â”‚
â”œâ”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”¼â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”¼â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”¤
â”‚            test-scripts â”‚        0 â”‚        0 â”‚
â”œâ”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”¼â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”¼â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”¤
â”‚      prerequest-scripts â”‚        0 â”‚        0 â”‚
â”œâ”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”¼â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”¼â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”¤
â”‚              assertions â”‚        0 â”‚        0 â”‚
â”œâ”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”´â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”´â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”¤
â”‚ total run duration: 2.6s                       â”‚
â”œâ”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”¤
â”‚ total data received: 179.12KB (approx)         â”‚
â”œâ”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”¤
â”‚ average response time: 2.5s                    â”‚
â””â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”˜
Performance: Recording JMeter reports '**/*.jtl'
Performance: Parsing JMeter report file 'C:\Users\miracle\.jenkins\jobs\DCM_DP_Automation\builds\15\performance-reports\JMeter\dp.jtl'.
Performance: Percentage of errors greater or equal than 0% sets the build as unstable
Performance: Percentage of errors greater or equal than 0% sets the build as failure
```



localhost:8080/job/DCM_DP_Automation/configure

Jenkins › DCM_DP_Automation ›

General | Source Code Management | Build Triggers | **Build Environment** | Build | Post-build Actions

- [ ] Delete workspace before build starts
- [ ] Abort the build if it's stuck
- [ ] Add timestamps to the Console Output
- [ ] Use secret text(s) or file(s)

Build

Invoke Ant

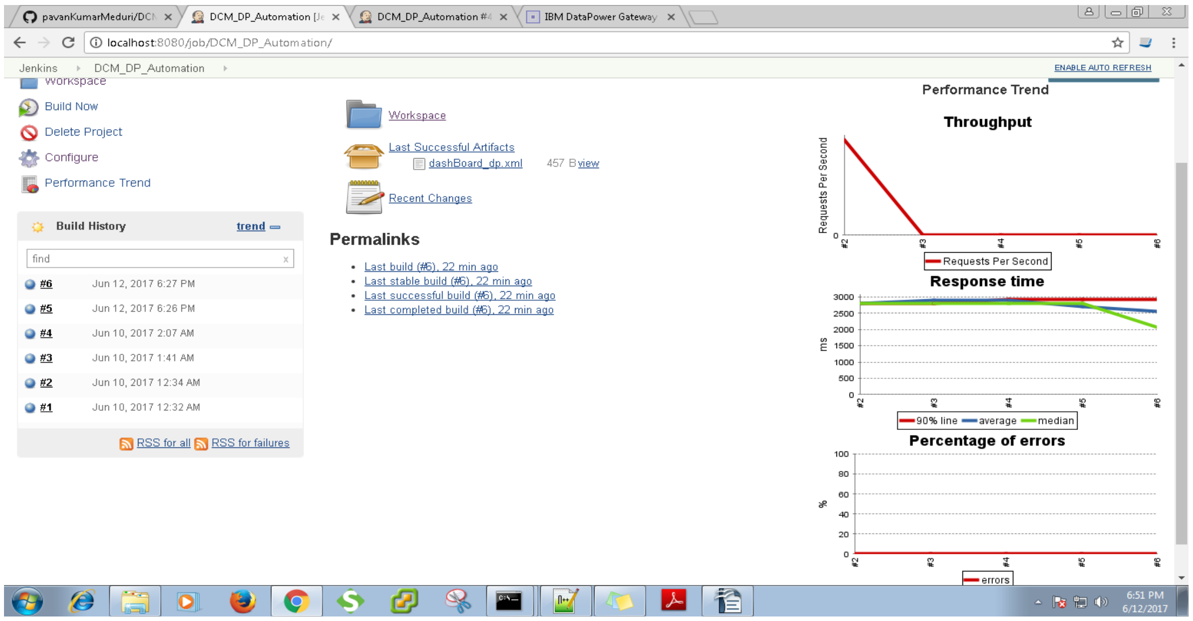
Ant Version	Default
Targets	import-dpo

Execute Windows batch command

Command

```
C:\apache-jmeter-3.2\apache-jmeter-3.2\bin\jmeter.bat -Jjmeter.save.saveservice.output_format=xml -n -t
C:\Users\miracle\.jenkins\workspace\DCM_DP_Automation\deltapoc\dp.jmx -l WeatherDetailsWSP.jtl
```

See the list of available environment variables



localhost:8080/job/DCM_DP_Automation/

Jenkins › DCM_DP_Automation ›

Build History trend

#6	Jun 12, 2017 6:27 PM
#5	Jun 12, 2017 6:26 PM
#4	Jun 10, 2017 2:07 AM
#3	Jun 10, 2017 1:41 AM
#2	Jun 10, 2017 12:34 AM
#1	Jun 10, 2017 12:32 AM

RSS for all RSS for failures

Workspace

Last Successful Artifacts

dashBoard_dp.xml 457 B view

Recent Changes

Permalinks

- Last build (#6), 22 min ago
- Last stable build (#6), 22 min ago
- Last successful build (#6), 22 min ago
- Last completed build (#6), 22 min ago