



Achieving Continuous DevOps – CI/CD

NAGP – Batch'20

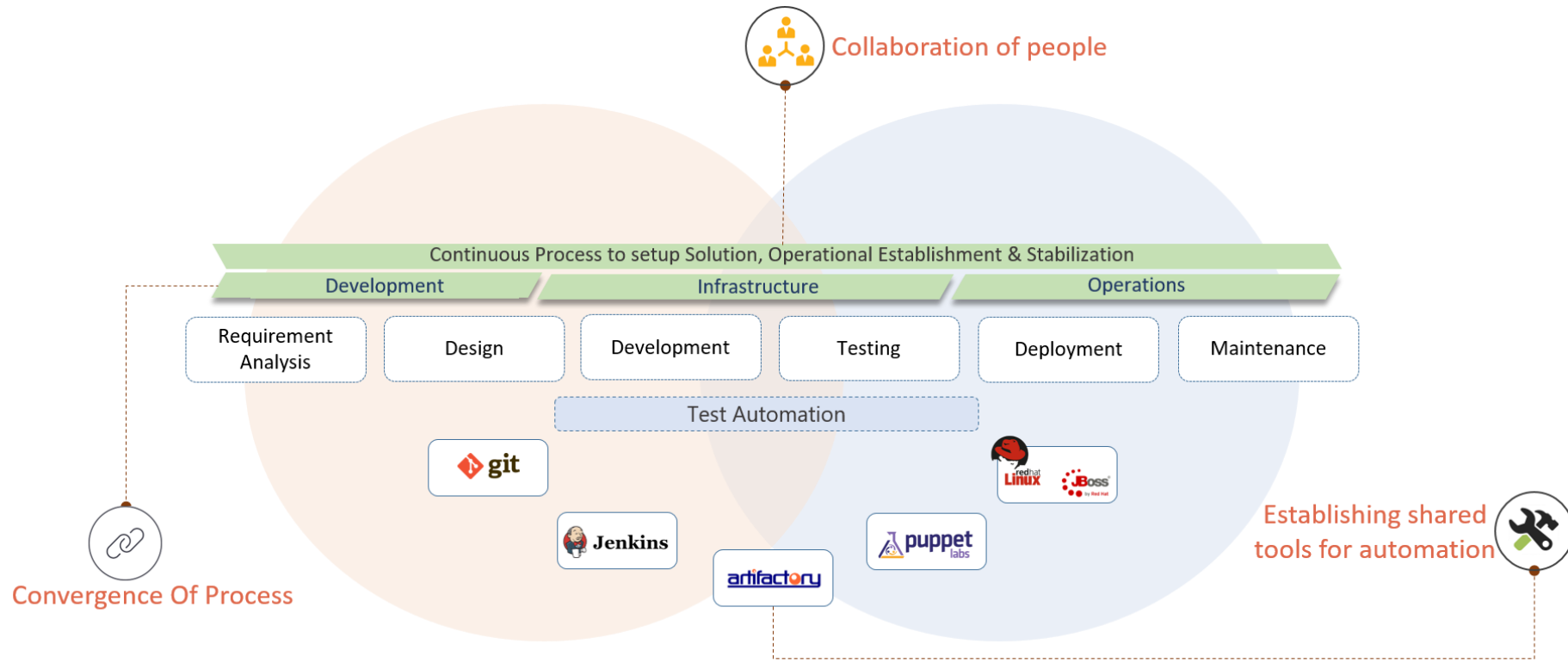


THINKING
BREAKTHROUGHS

DevOps Introduction

Bridging the gap between Development and Operations

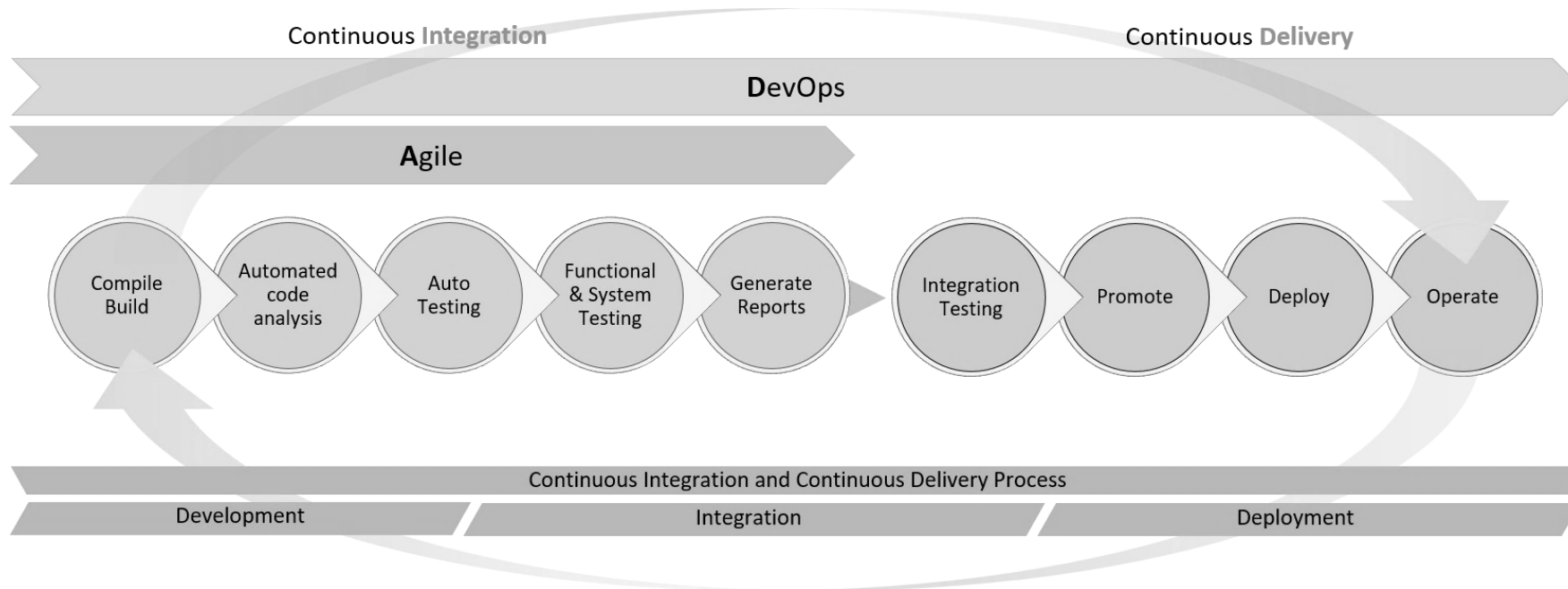
DevOps is a **culture and set of practices** that emphasize **close collaboration** between software Development and IT Operations staff through all phases of software delivery process to **enhance productivity and quality of the overall system**



Understanding Agile with DevOps

It's an extension to agile development methodology.

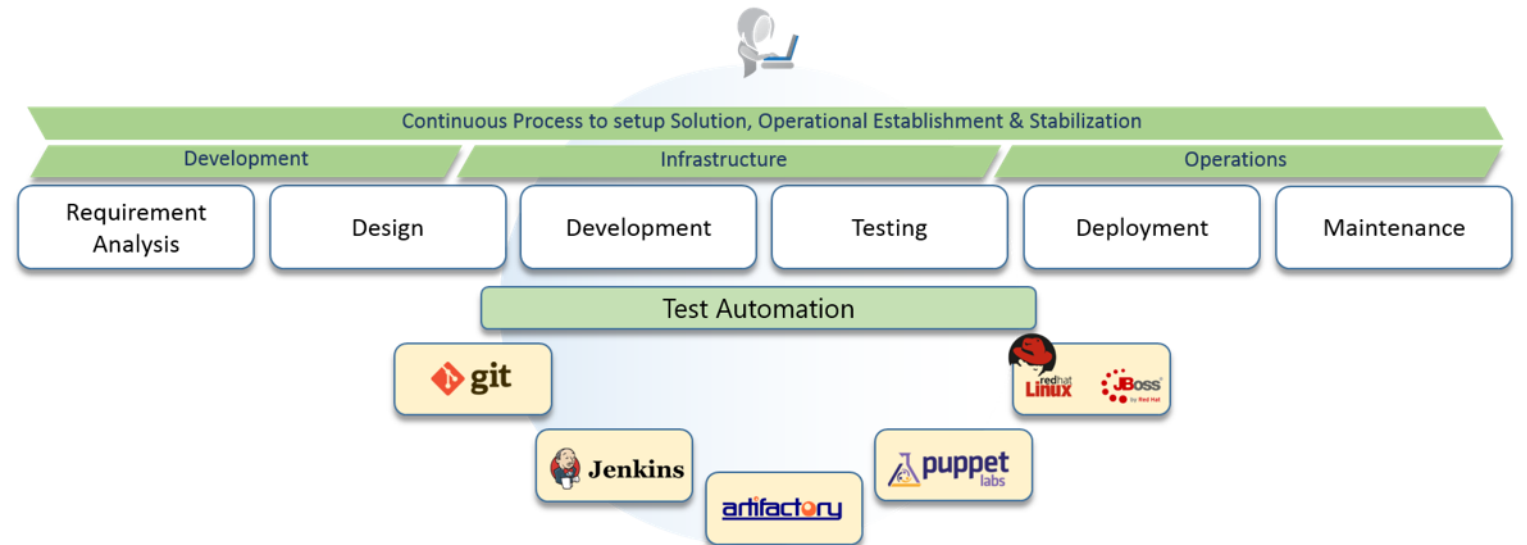
Agile software development methodology describes a set of principles that focuses on the development of software with *business* and *development* team, but **DevOps** on the other hand is responsible for **development** as well as **deployment** of the software via bringing *development*, *business* and *operations* team together.



How to achieve DevOps

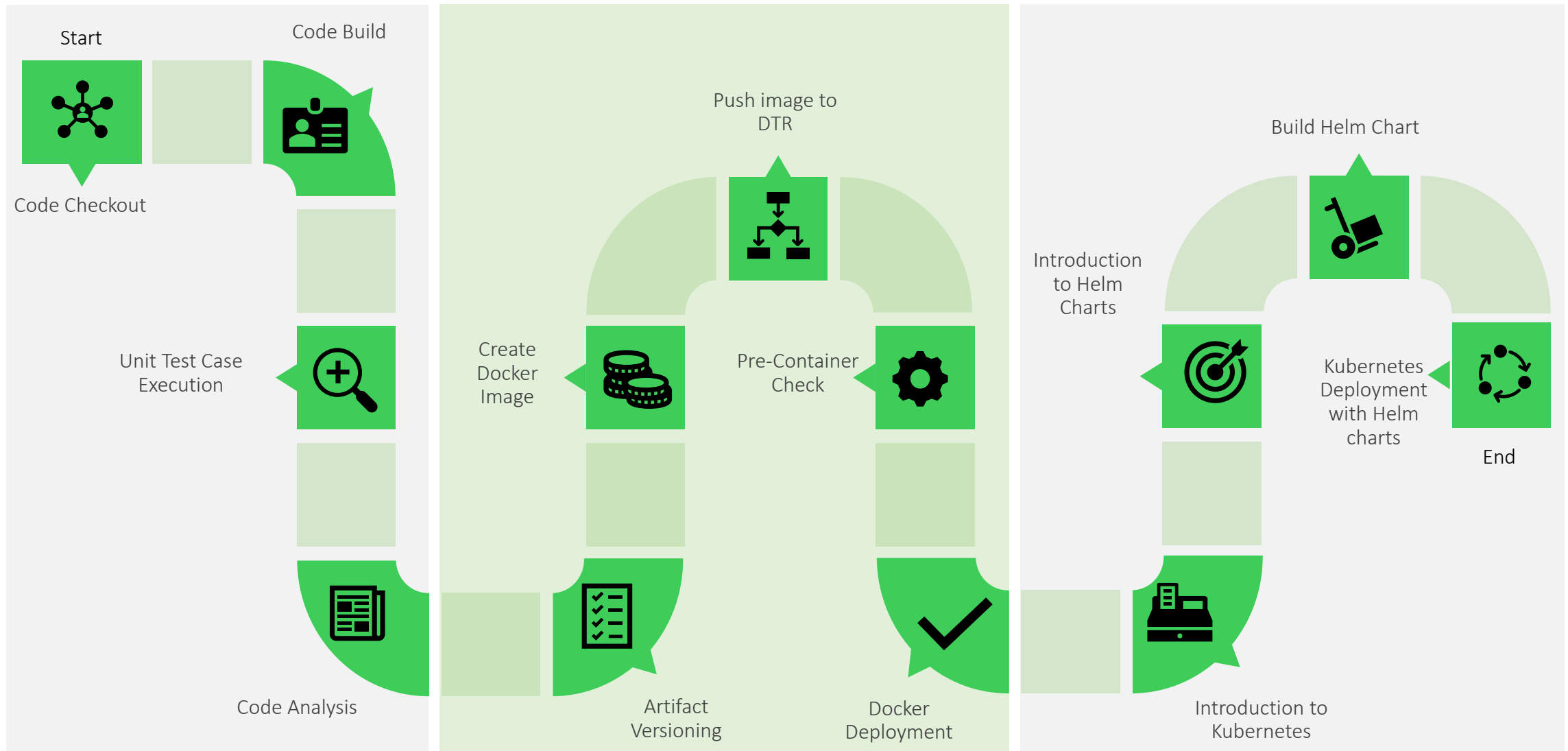
DevOps as process

1. Version control management
2. Continuous build
3. Continuous integration
4. Continuous delivery
5. Continuous deployment
6. Automating entire chain – tool chain



Achieving Continuous DevOps

Agenda

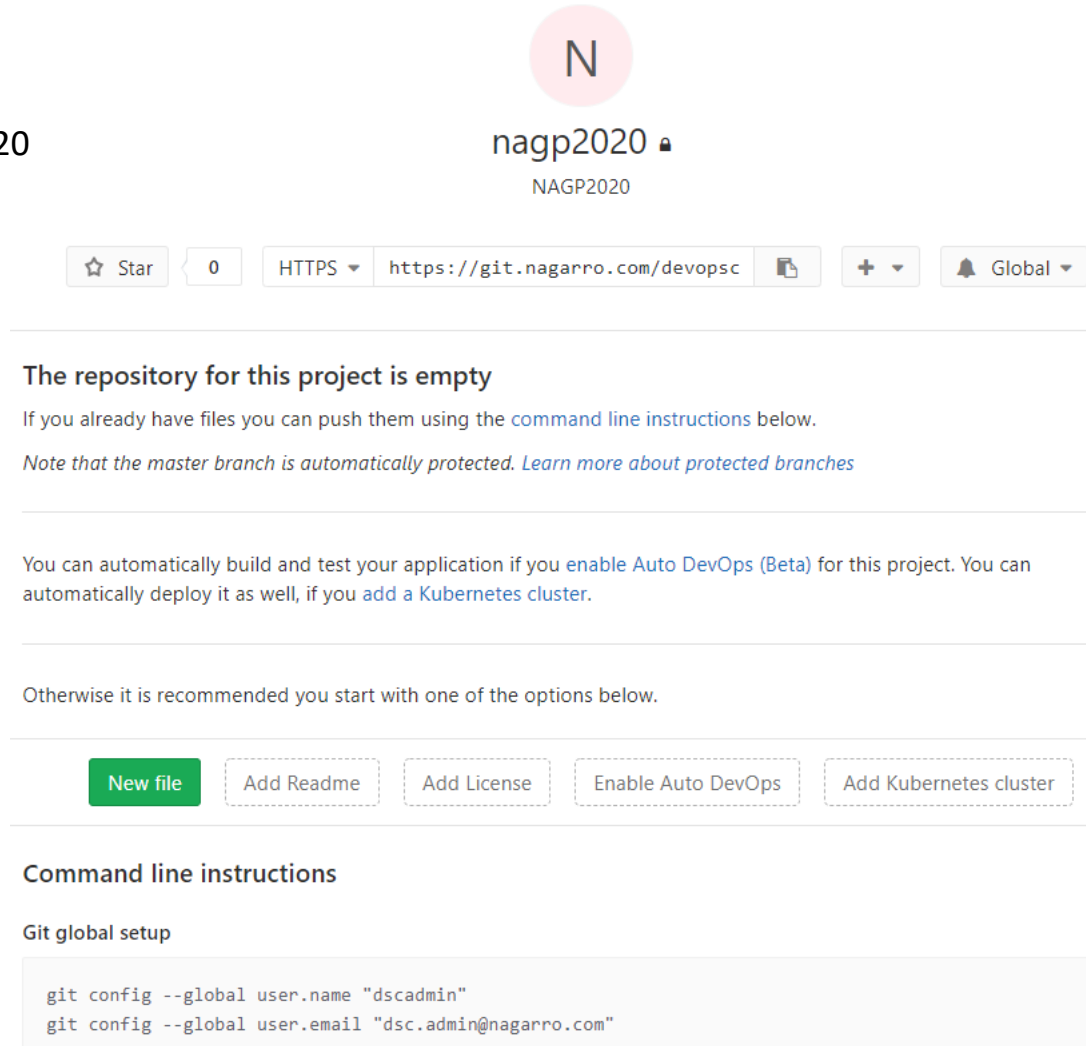


Git Repository

Validate if it exist or not

Navigation steps:

1. Navigate to git.nagarro.com
2. Look for devopscoe/nagp2020



The screenshot shows the interface for creating a new Git repository. At the top, the user's profile 'nagp2020' is displayed with a pink circular avatar containing the letter 'N'. Below the profile, there are buttons for 'Star' (0), 'HTTPS' (with a dropdown arrow), the repository URL 'https://git.nagarro.com/devopsc', a file icon, a plus sign with a dropdown arrow, and a 'Global' notification bell. The main content area states 'The repository for this project is empty' and provides instructions on how to push files using command line instructions. It also mentions that the master branch is automatically protected and provides a link to learn more about protected branches. Below this, there is a section for enabling Auto DevOps (Beta) and adding a Kubernetes cluster. At the bottom, there are buttons for 'New file', 'Add Readme', 'Add License', 'Enable Auto DevOps', and 'Add Kubernetes cluster'. The 'Command line instructions' section shows the following commands for global setup:

```
git config --global user.name "dscadmin"
git config --global user.email "dsc.admin@nagarro.com"
```

Make sure, it's a blank repository




Sample Code

Available here – Java/Dotnet

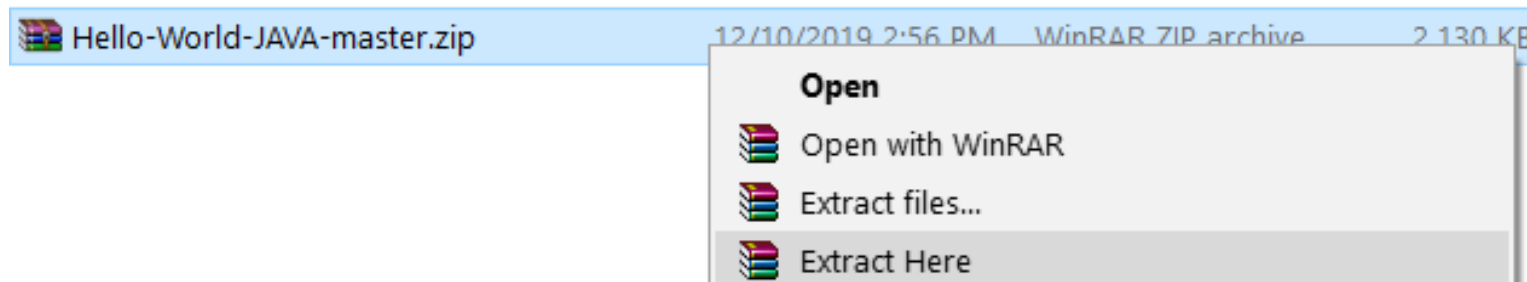
Navigation steps:

- On your laptops, press Window+R, and navigate to <\\SHREY314699701\Shared>



> Network > SHREY314699701 > Shared

	Name	Date modified	Type	Size
	Delete_TEMP.bat	6/22/2018 11:19 AM	Windows Batch File	1 KB
	Hello-World-DotNet-master.zip	12/10/2019 2:56 PM	WinRAR ZIP archive	553 KB
	Hello-World-JAVA-master.zip	12/10/2019 2:56 PM	WinRAR ZIP archive	2,130 KB

- Attendees with *DotNet background*, copy Hello-World-DotNet-master.zip on your laptop, right click on it and click extract here
- Attendees with *Java background*, copy Hello-World-Java-master.zip on your laptop, right click on it and click extract here; eg:



> This PC > New Volume (E:) > New folder

	Name	Date modified	Type	Size
	Hello-World-JAVA-master	11/27/2019 11:25 ...	File folder	
	Hello-World-JAVA-master.zip	12/10/2019 2:56 PM	WinRAR ZIP archive	2,130 KB

Git Repository

Push extracted code to version control system

Steps to be followed for pushing a sample code (Java/DotNet) to Git:

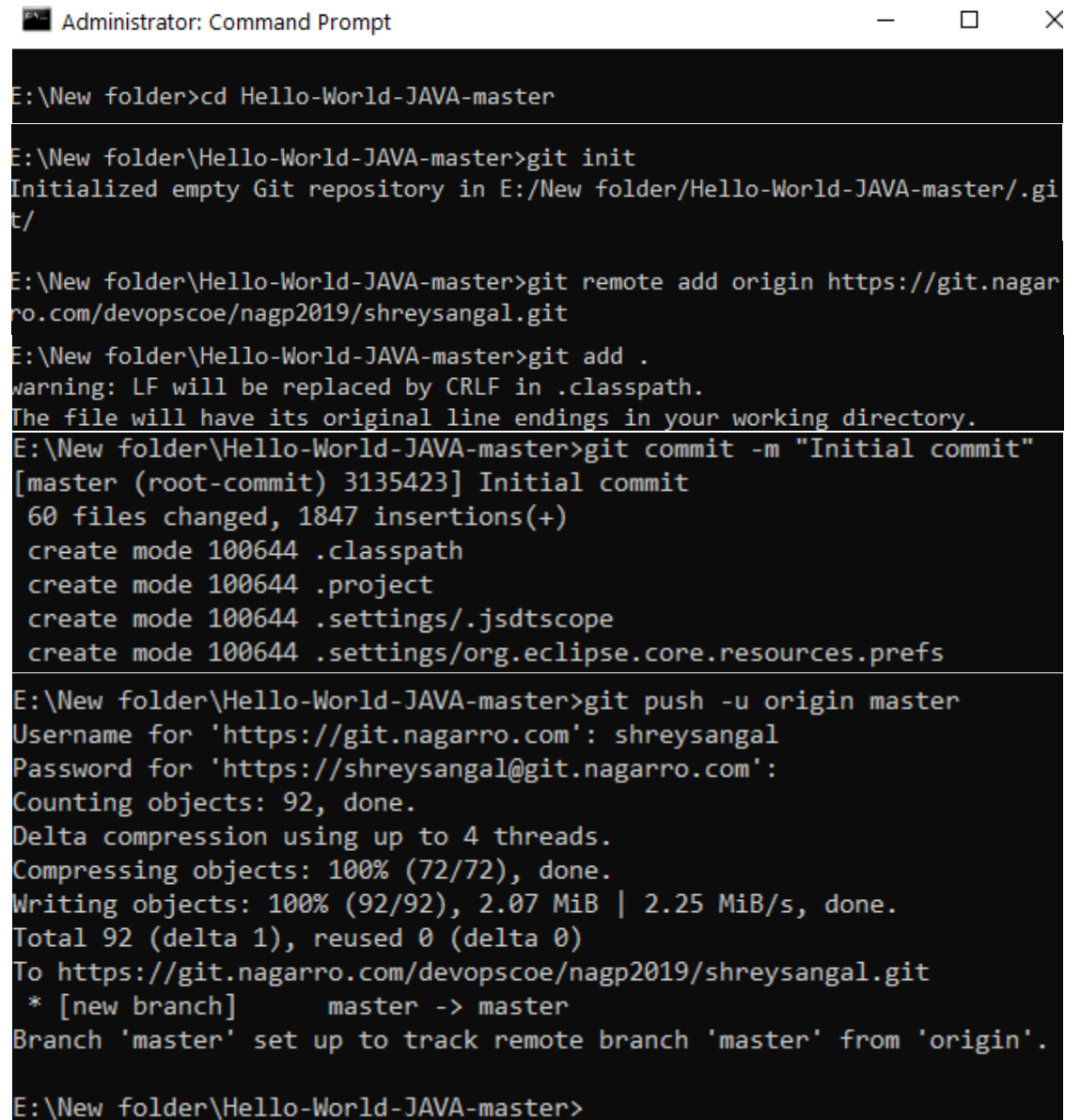
- Open command prompt, and navigate to directory, having extracted sample code and navigate to source code directory
- Since the code already exist, do init, to initialize it as git repository
- Since the repository is already present, add remote to the local repository
- Add the files in your new local repository, this stages them for the first commit; git add .
- Commit the files, that are staged in local repository, with a message; eg: Initial commit
- Push the code to Git repository

FAQ:

- Problem encountered while pushing code:
E:\New folder\Hello-World-JAVA-master>git push -u origin master
remote: HTTP Basic: Access denied
fatal: Authentication failed for
'https://git.nagarro.com/devopscoe/training/shreysangal.git/'

Below solution will help and try pushing again

git config --system --unset credential.helper



```
Administrator: Command Prompt

E:\New folder>cd Hello-World-JAVA-master

E:\New folder\Hello-World-JAVA-master>git init
Initialized empty Git repository in E:/New folder/Hello-World-JAVA-master/.git/

E:\New folder\Hello-World-JAVA-master>git remote add origin https://git.nagarro.com/devopscoe/nagp2019/shreysangal.git

E:\New folder\Hello-World-JAVA-master>git add .
warning: LF will be replaced by CRLF in .classpath.
The file will have its original line endings in your working directory.

E:\New folder\Hello-World-JAVA-master>git commit -m "Initial commit"
[master (root-commit) 3135423] Initial commit
 60 files changed, 1847 insertions(+)
 create mode 100644 .classpath
 create mode 100644 .project
 create mode 100644 .settings/.jsdtscope
 create mode 100644 .settings/org.eclipse.core.resources.prefs

E:\New folder\Hello-World-JAVA-master>git push -u origin master
Username for 'https://git.nagarro.com': shreysangal
Password for 'https://shreysangal@git.nagarro.com':
Counting objects: 92, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (72/72), done.
Writing objects: 100% (92/92), 2.07 MiB | 2.25 MiB/s, done.
Total 92 (delta 1), reused 0 (delta 0)
To https://git.nagarro.com/devopscoe/nagp2019/shreysangal.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.

E:\New folder\Hello-World-JAVA-master>
```


Git Repository (Java)

Verify if the pushed code exist

Navigation steps:

1. Navigate to git.nagarro.com
2. Look for [devopscoe/nagp2020](#)

The screenshot displays the GitLab interface for a repository owned by 'shreysangal'. The left sidebar contains navigation links: Overview, Details, Activity, Cycle Analytics, Repository, Issues (0), Merge Requests (0), CI / CD, Wiki, Snippets, and Settings. The main content area shows the repository's profile, including a 'Star' button (0), a 'Fork' button (0), and the repository URL: <https://git.nagarro.com/devopsc>. Below this, there are buttons for 'Add Changelog', 'Add License', 'Add Contribution guide', 'Add Kubernetes cluster', and 'Set up CI/CD'. The repository is currently on the 'master' branch. A table lists the files in the repository, all of which were committed in the 'Initial commit' by 'shreysangal' 26 minutes ago.

Name	Last commit	Last update
.settings	Initial commit	26 minutes ago
DemoSampleApp_Imeter	Initial commit	26 minutes ago
DemoSampleApp_selenium	Initial commit	26 minutes ago
src	Initial commit	26 minutes ago
.classpath	Initial commit	26 minutes ago
.project	Initial commit	26 minutes ago

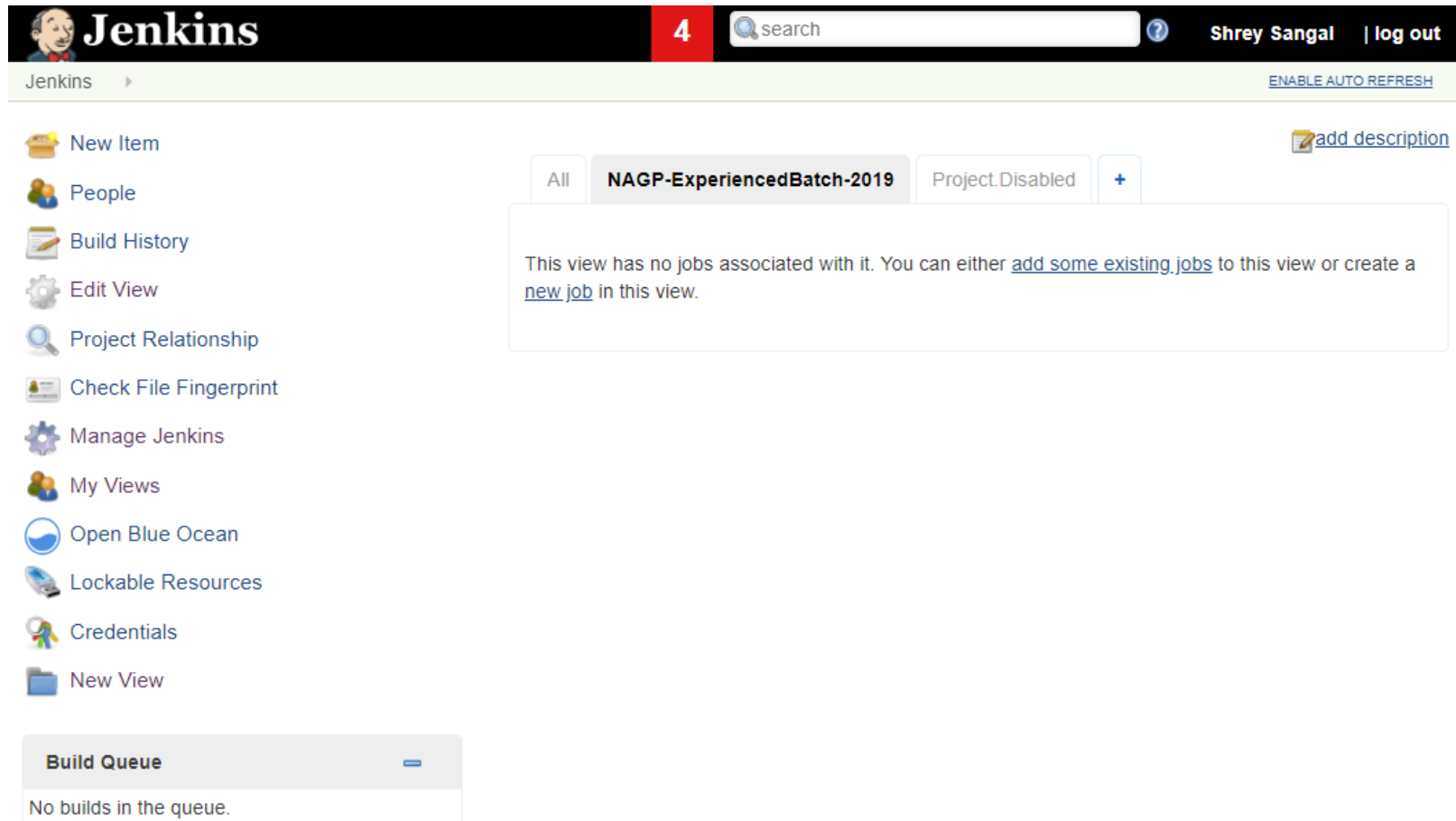
Make sure, it is not a blank repository

Continuous Integration (Jenkins)

Open source Continuous Integration server capable of orchestrating a chain of actions in an automated fashion

Navigation steps:

1. Navigate to <http://10.127.129.114:8080/>
2. Login using your AD credentials
3. NAGP-ExperiencedBatch-2020 view should be visible



The screenshot displays the Jenkins web interface. At the top, there is a black header bar with the Jenkins logo on the left, a red square with the number '4' in the center, a search bar on the right, and the user name 'Shrey Sangal' with a 'log out' link. Below the header is a green bar with the text 'Jenkins' and a right arrow, and a link 'ENABLE AUTO REFRESH' on the far right. The main content area is divided into a left sidebar and a central panel. The sidebar contains a list of navigation links: 'New Item', 'People', 'Build History', 'Edit View', 'Project Relationship', 'Check File Fingerprint', 'Manage Jenkins', 'My Views', 'Open Blue Ocean', 'Lockable Resources', 'Credentials', and 'New View'. The central panel shows a view titled 'NAGP-ExperiencedBatch-2019' with tabs for 'All', 'NAGP-ExperiencedBatch-2019', 'Project.Disabled', and a '+' button. Below the tabs, a message box states: 'This view has no jobs associated with it. You can either [add some existing jobs](#) to this view or create a [new job](#) in this view.' At the bottom left, there is a 'Build Queue' section with a minus sign icon and the text 'No builds in the queue.'

Jenkins and its integrations

Third party tools configuration

Navigation Steps:

1. Go to manage Jenkins
2. Click on Global tool configuration

Third party tools installed on server needs to be configured in jenkins:

- a Java
- b Git
- c Sonarqube scanner
- d Maven
- e Docker

The screenshot displays the 'Global Tool Configuration' page in Jenkins. The page is divided into sections for different tools, each with a list of installations and an 'Add' button. The tools shown are JDK, Git, SonarQube Scanner, and Maven. The 'JDK' section has one installation named 'Java' with 'JAVA_HOME' set to '/usr/java/jdk1.8.0_191-amd64'. The 'Git' section has one installation named 'Default' with 'Path to Git executable' set to 'git'. The 'SonarQube Scanner' section has one installation named 'Sonar' with 'Install automatically' checked and 'Version' set to 'SonarQube Scanner 4.0.0.1744'. The 'Maven' section has one installation named 'M3' with 'MAVEN_HOME' set to '/usr/local/src/apache-maven'. At the bottom, there are 'Save' and 'Apply' buttons.

Jenkins ▾ ▶ Global Tool Configuration

Global Tool Configuration

JDK installations

Add JDK

JDK
Name: Java
JAVA_HOME: /usr/java/jdk1.8.0_191-amd64
☐ Install automatically

Git

Git installations

Git
Name: Default
Path to Git executable: git
☐ Install automatically

SonarQube Scanner

SonarQube Scanner installations

Add SonarQube Scanner

SonarQube Scanner
Name: Sonar
☒ Install automatically
Install from Maven Central
Version: SonarQube Scanner 4.0.0.1744 ▼

Maven

Maven installations

Add Maven

Maven
Name: M3
MAVEN_HOME: /usr/local/src/apache-maven

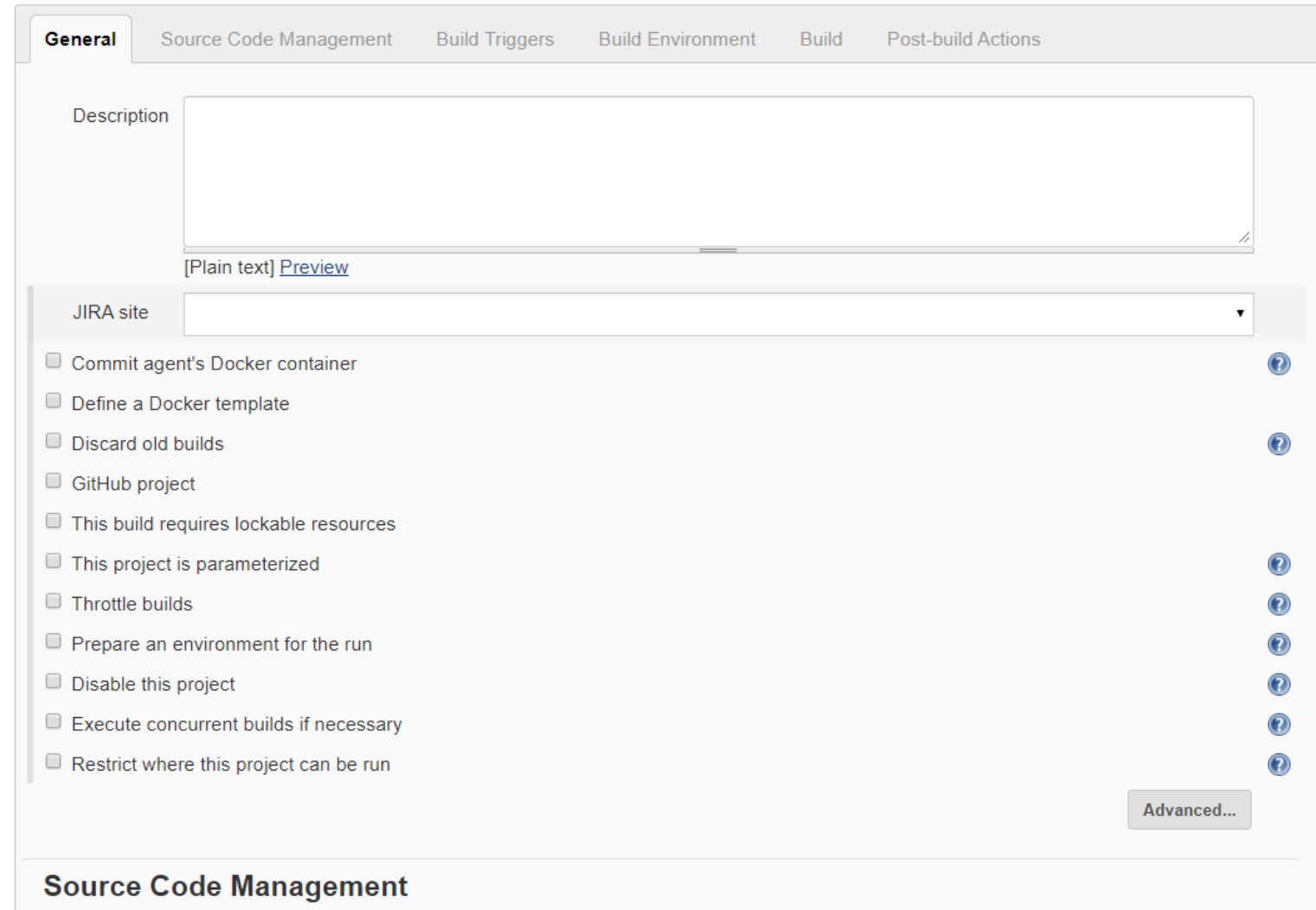
Save **Apply**

Jenkinsfreestyle job

Create a new job

Navigation Steps:

1. Go to manage Jenkins
2. Click on new job



The screenshot shows the Jenkins 'Create new job' configuration page, specifically the 'General' tab. The page has a top navigation bar with tabs: 'General' (selected), 'Source Code Management', 'Build Triggers', 'Build Environment', 'Build', and 'Post-build Actions'. Below the tabs, there is a 'Description' field with a text area and a '[Plain text] [Preview](#)' link. Below the description field is a 'JIRA site' dropdown menu. A list of checkboxes follows, each with a corresponding help icon (question mark in a circle) to its right. The checkboxes are: 'Commit agent's Docker container', 'Define a Docker template', 'Discard old builds', 'GitHub project', 'This build requires lockable resources', 'This project is parameterized', 'Throttle builds', 'Prepare an environment for the run', 'Disable this project', 'Execute concurrent builds if necessary', and 'Restrict where this project can be run'. At the bottom right of the configuration area is an 'Advanced...' button. Below the configuration area is a section header 'Source Code Management'.

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

Description

[Plain text] [Preview](#)

JIRA site

- ☐ Commit agent's Docker container ?
- ☐ Define a Docker template
- ☐ Discard old builds ?
- ☐ GitHub project
- ☐ This build requires lockable resources
- ☐ This project is parameterized ?
- ☐ Throttle builds ?
- ☐ Prepare an environment for the run ?
- ☐ Disable this project ?
- ☐ Execute concurrent builds if necessary ?
- ☐ Restrict where this project can be run ?

Advanced...

Source Code Management

Code Coverage (Sonarqube)

Catch bugs and vulnerabilities, with thousands of automated Static Code Analysis rules

Navigation steps:

1. Navigate to <http://10.127.127.160:9000/>
2. Login using your AD credentials

The screenshot shows the SonarQube web interface. The top navigation bar includes links for Dashboards, Issues, Measures, Rules, Quality Profiles, Quality Gates, Administration, and More. The user is logged in as 'Administrator'. The main content area is divided into two sections. The left section, titled 'Home', contains a 'Welcome to SonarQube Dashboard' message with instructions on how to use the tool. The right section, titled 'PROJECTS', displays a table with columns for QG, Name, Version, LOC, Bugs, Vulnerabilities, and Code Smells. The table is currently empty, showing 'No data'.

sonarqube Dashboards ▾ Issues Measures Rules Quality Profiles Quality Gates Administration More ▾ Administrator ▾ 🔍 ?

Home [Configure widgets](#)

Welcome to SonarQube Dashboard

Since you are able to read this, it means that you have successfully started your SonarQube server. Well done!

If you have not removed this text, it also means that you have not yet played much with SonarQube. So here are a few pointers for your next step:

- » Do you now want to [run analysis](#) on a project?
- » Maybe start [customizing dashboards](#)?
- » Or simply browse the [complete documentation](#)?
- » If you have a question or an issue, please visit the [Get Support](#) page.

MY FAVOURITES

QG	Name ▲	Last Analysis
No data		

PROJECTS

QG	Name ▲	Version	LOC	Bugs	Vulnerabilities	Code Smells
----	--------	---------	-----	------	-----------------	-------------

Jenkins and its integrations

Sonarqube integrated with Jenkins

Navigation steps:

1. Go to manage Jenkins
2. Click on configure system
3. Search for Sonar

sonar

2/9

^

v

X

Jenkins configuration

SonarQube servers

Environment variables

☒ Enable injection of SonarQube server configuration as build environment variables

If checked, job administrators will be able to inject a SonarQube server configuration as environment variables in the build.

SonarQube installations

Name

Test_Sonar

Server URL

http://10.127.127.160:9000

Default is http://localhost:9000

Server authentication token

sonar

Add

SonarQube authentication token. Mandatory when anonymous access is disabled.

Advanced...

Delete SonarQube

Add SonarQube

List of SonarQube installations

