Project

Simulation

Group Name: Team TAE (Trial and Error)

Team Leader: Cris Gievid M. Narca

CI/CD: Anjhillian S. Cabrera

Mari Cris P. Ayes

Developer: Joey Ann O. Robles

Mari Cris P. Ayes

Documentation: Pauleen S. Pineda

**CHAPTER 1 – INTRODUCTION**

**1.1 BACKGROUND OF THE STUDY**

As innovation accelerates and customer’s needs rapidly evolve, businesses must become increasingly agile. But unlike the previous system, development always have issues in collaborating with production which causes slow growth of production, more risks to errors, cost, rework which is very time consuming and mostly likely inefficient. Thus, business goals and demands were not achieved.

In order to meet the demands of an agile business, IT operations need to deploy applications in a consistent, repeatable, and reliable manner. This can only be fully achieved with the adoption of automation. DevOps is a new term that focuses on improved collaboration, communication, and integration between software developers and IT operations. It is simply the collaboration of development and operation that produce a continuous integration and continuous deployment of codes to production. It basically automates the whole lifecycle of the system, from build, test and to the deploy process of the system.

**1.2 OBJECTIVE**

**1.3.1 General Objectives**

The main objective of the study is to create a system that will automate development, testing and deployment of codes to production.

**1.3.2 Specific Objectives**

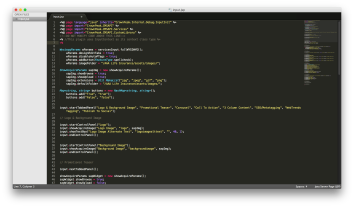
* To create a java based application that will be used to test the system
* To integrate different tools such as GitLab, Maven, Sonarqube, Ansible, Selenium and Tomcat to Jenkins
* To produce continuous integration and continuous deployment
* To maximize the efficiency, security, and maintainability of the whole project

**CHAPTER 2 – RESEARCH METHODOLOGY**

**2.1 Conceptual Framework**

**Developers**



****

****

****

**CI/CD**

****

****

****

****



**CLIENT**

**Figure 1.0.** *Conceptual Diagram of the System*

**2.2 Software Diagram**

**PARANG GANITO!**

**Figure (). Software Design Flowchart**

**EXPLAIN…….**

**2.3 Software Considerations**

**2.3.1 GitLab**

GitLab is an open-source code collaboration platform that enables developers to create, review, and deploy code bases. It includes Git repository management, code reviews, and issue tracking. GitLab was launched in 2014 and it was operated from San Franciso, California

One of the major benefits of the system is the version control that enables you to go back or recover certain file from the previous commit. It is also a convenient user interface that enables users to access everything from one screen to another, anytime and anywhere.

**2.3.2 Jenkins**

Jenkins is a free application that allows continuous integration and continuous delivery of projects, regardless of the platform used. It can handle any kind of build, push files to various artifact repositories, integrate tools for testing and deployment. Jenkins is a java based continuous build system.

Jenkins is the primary platform for plugins, whereas it is supported by over 400 plugins. It is also cloud-enabled.

**2.3.3 Maven**

Maven is a project management and comprehension tool. It provides developers a complete build lifecycle framework whereas the development team can automate the project's build infrastructure in almost no time as Maven uses a standard directory layout and a default build lifecycle. Maven makes life of developer easy while creating reports, checks, build and testing automation setups.

One advantage of Maven is that it has better dependency management. It can add new dependencies quickly, identify unused and transitive dependencies, and it can create reports showing all dependencies used on a project. Second, Maven’s default plugins and life cycle allow a project to perform common build actions with touching build configuration file. Third, it has better debugging and collaboration. Maven repositories allow an artifact’s source code and Javadoc to be published along with the artifact’s JAR. Fourth, it has a consistent project structure which makes it easier to understand each project. And lastly, Maven projects can be use POM (project object model) hierarchy to reduce the duplication.

**2.3.4 Sonarqube**

Sonarqube is an open platform for code quality management where developers can manage, track and improve the quality of the source code. It is written in java but it can analyze code in 20 different programming languages.

Sonarqube platform enables automatic detection of bugs and provides an opportunity to fix them before rolling software out to production. Thus, it avoids risks of software development within a short period of time, improves quality and productivity.

**2.3.5 Ansible**

Ansible is a configuration management, deployment and orchestration tool that provides an automated infrastructure for managing systems devices and applications. It is written in Python and provides an agentless approach that focuses on management of the destination device and application over SSH. Ansible implements playbooks and modules. Playbooks are series of tasks to be executed on a host and provide the fundamental workflow. On the other hand, Modules perform the operations based on the directives of the tasks and playbooks.

**2.3.6 Apache Tomcat**

Apache Tomcat is open source software that implements java Servlet and Java Server pages, enabling sites to run Java servlets and Java-based dynamic content. It allows developers to effortlessly build and check today’s connected Web applications. The advantages of apache tomcat are flexibility and stability.

**2.3.7 Selenium**

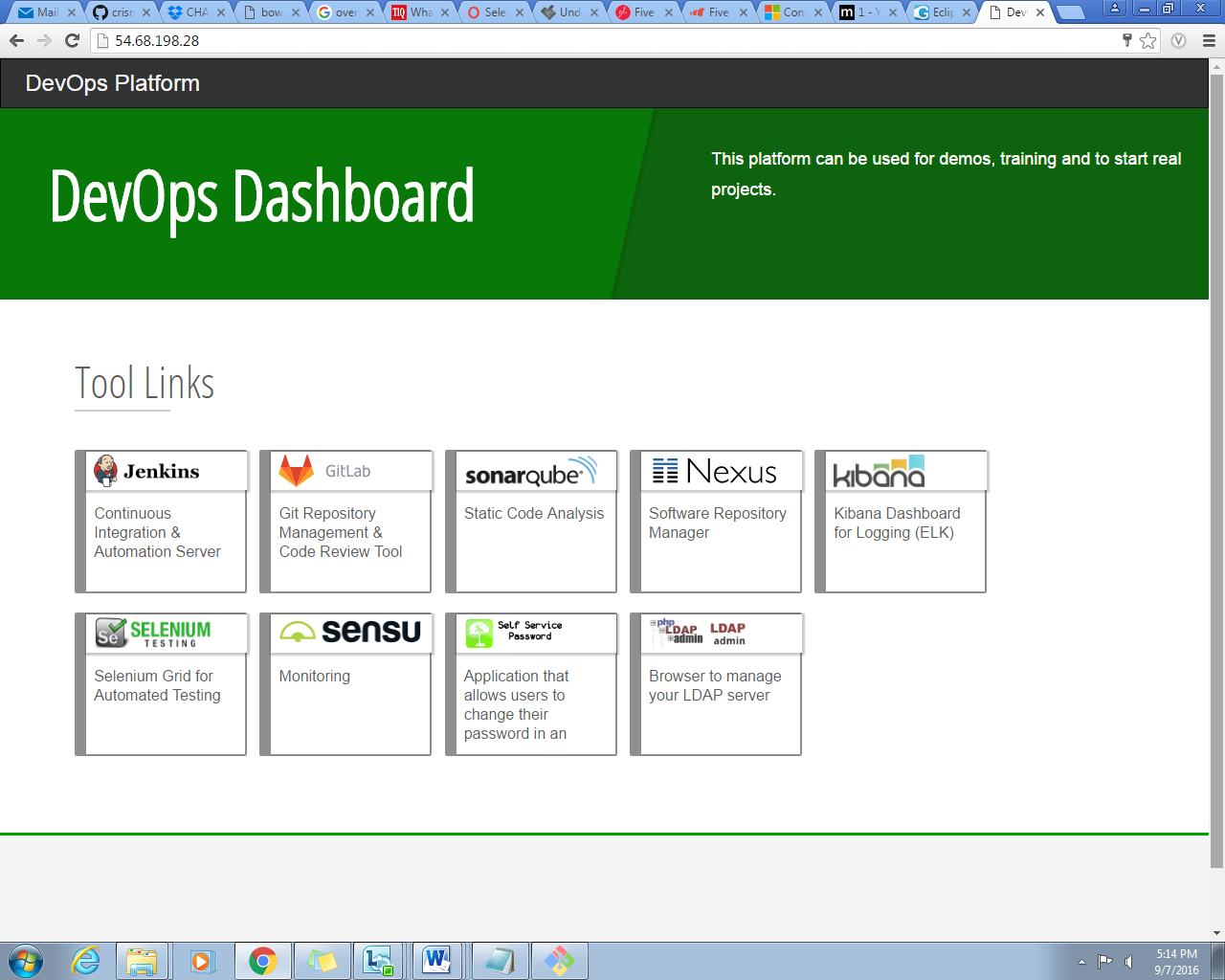
Selenium is a portable software testing framework for web applications. It can be deployed on Windows and Linux. The tests are written as HTML tables or coded in a number of popular programming languages. It can be run directly in most modern web browser.

Some of the advantages of Selenium are frequent regression testing, rapid feedback to developers, virtually unlimited iterations of test case execution and can easily find defects missed by manual testing.

**2.4 Working Theories**

**2.4.1 DevOps Platform**

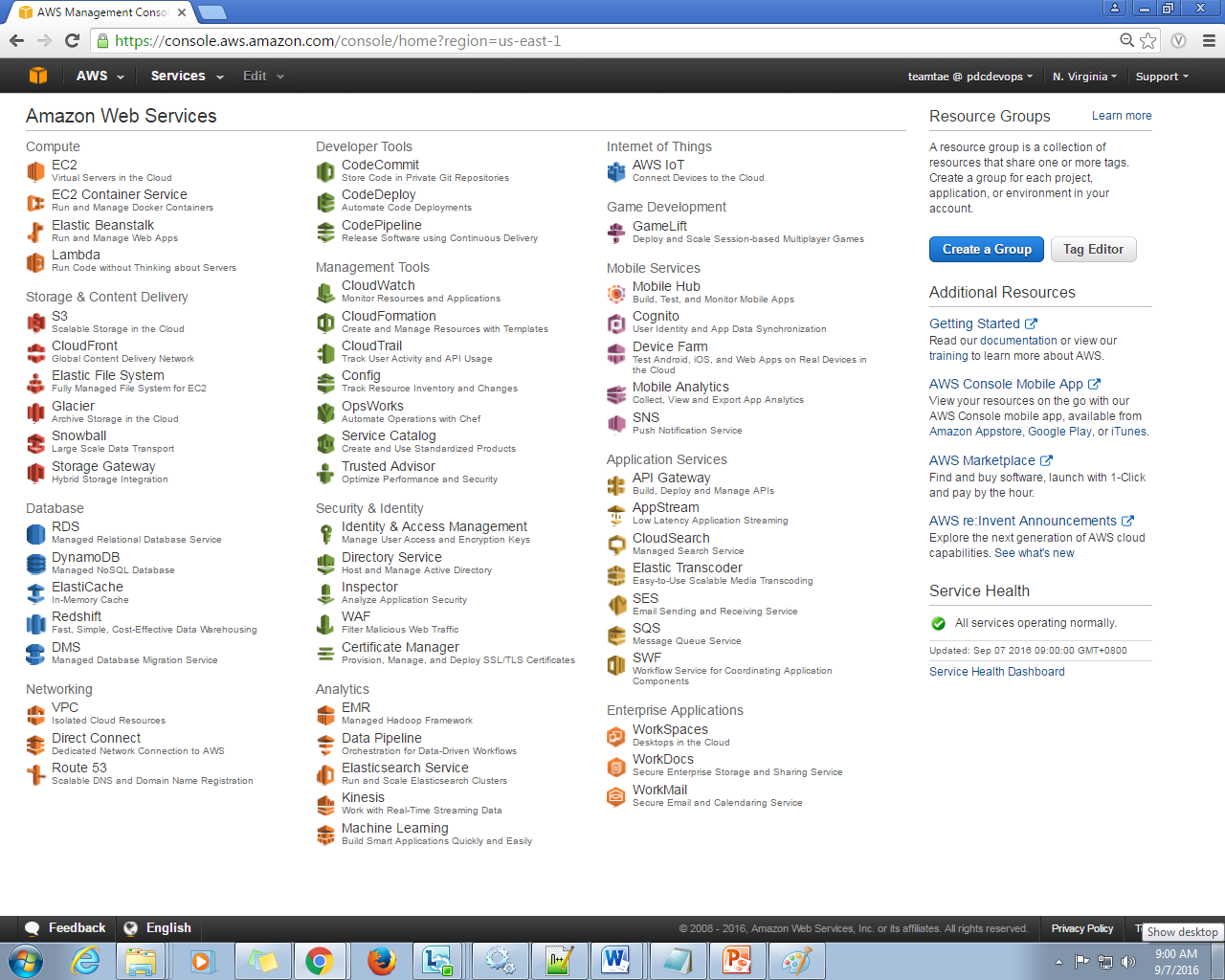
[**http://54.68.198.28/**](http://54.68.198.28/)Screenshot/ Description blah blah…..



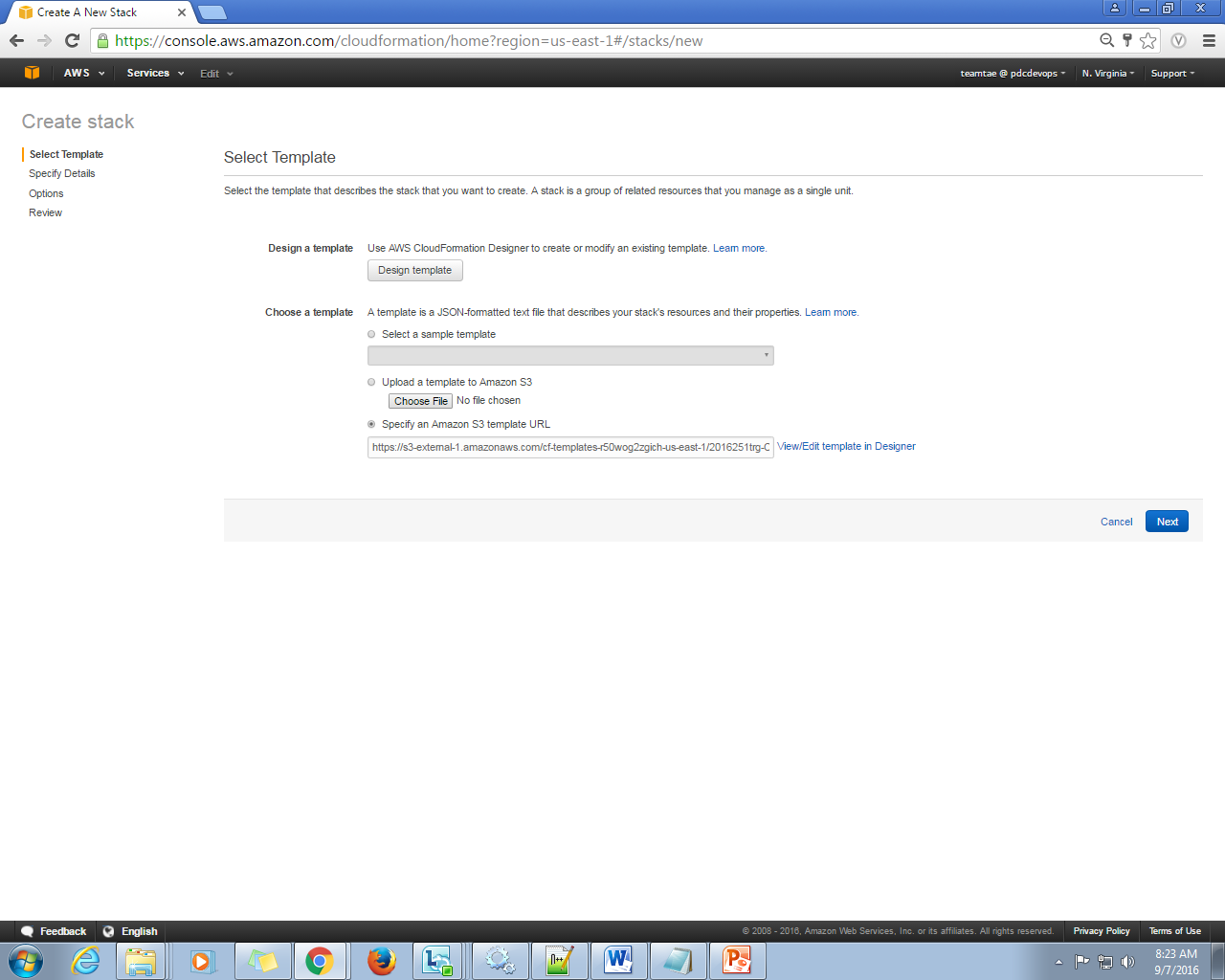
**Figure ().**

**2.4.2 Creating Stacks in AWS**

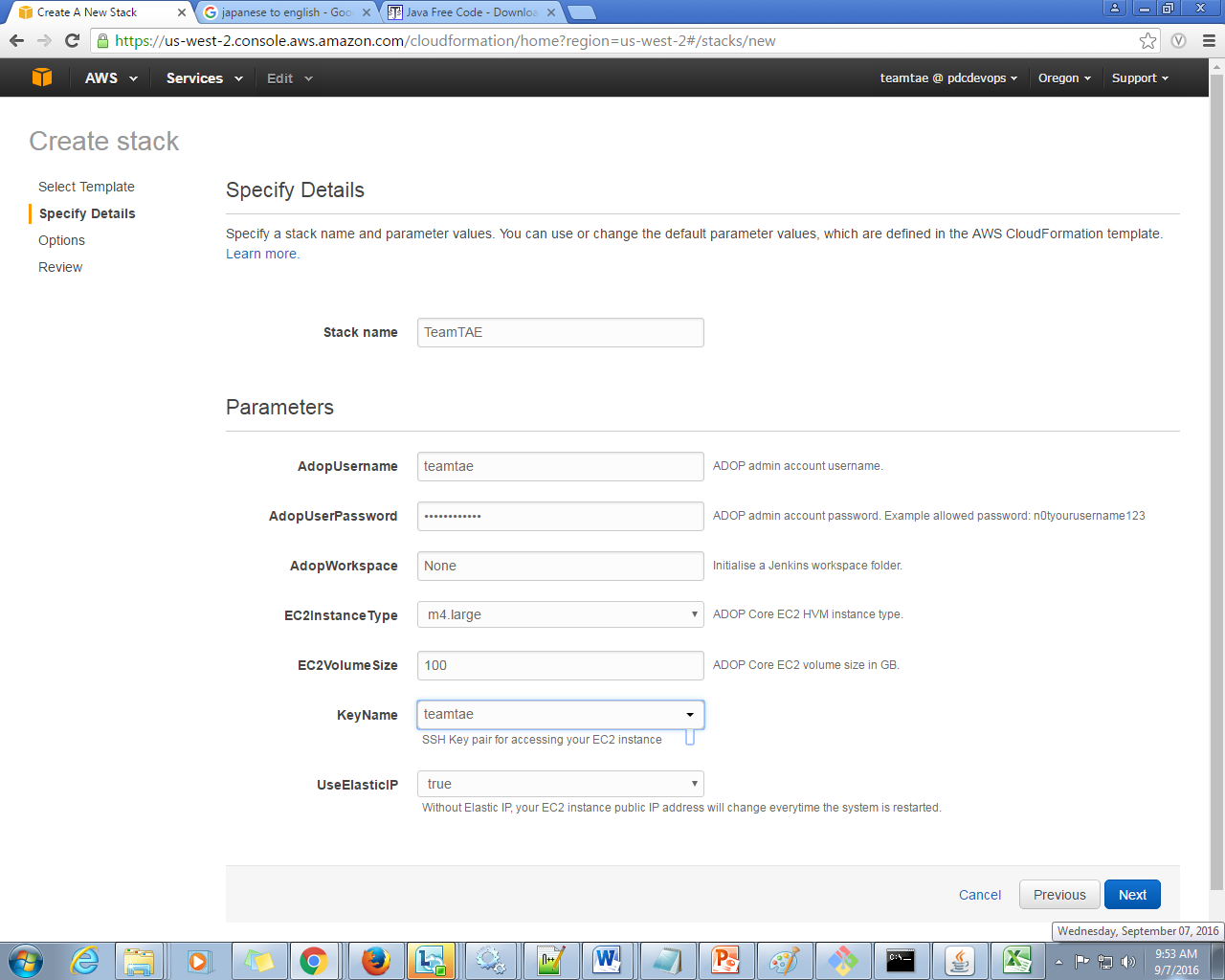
Screenshot/ Description blah blah…..



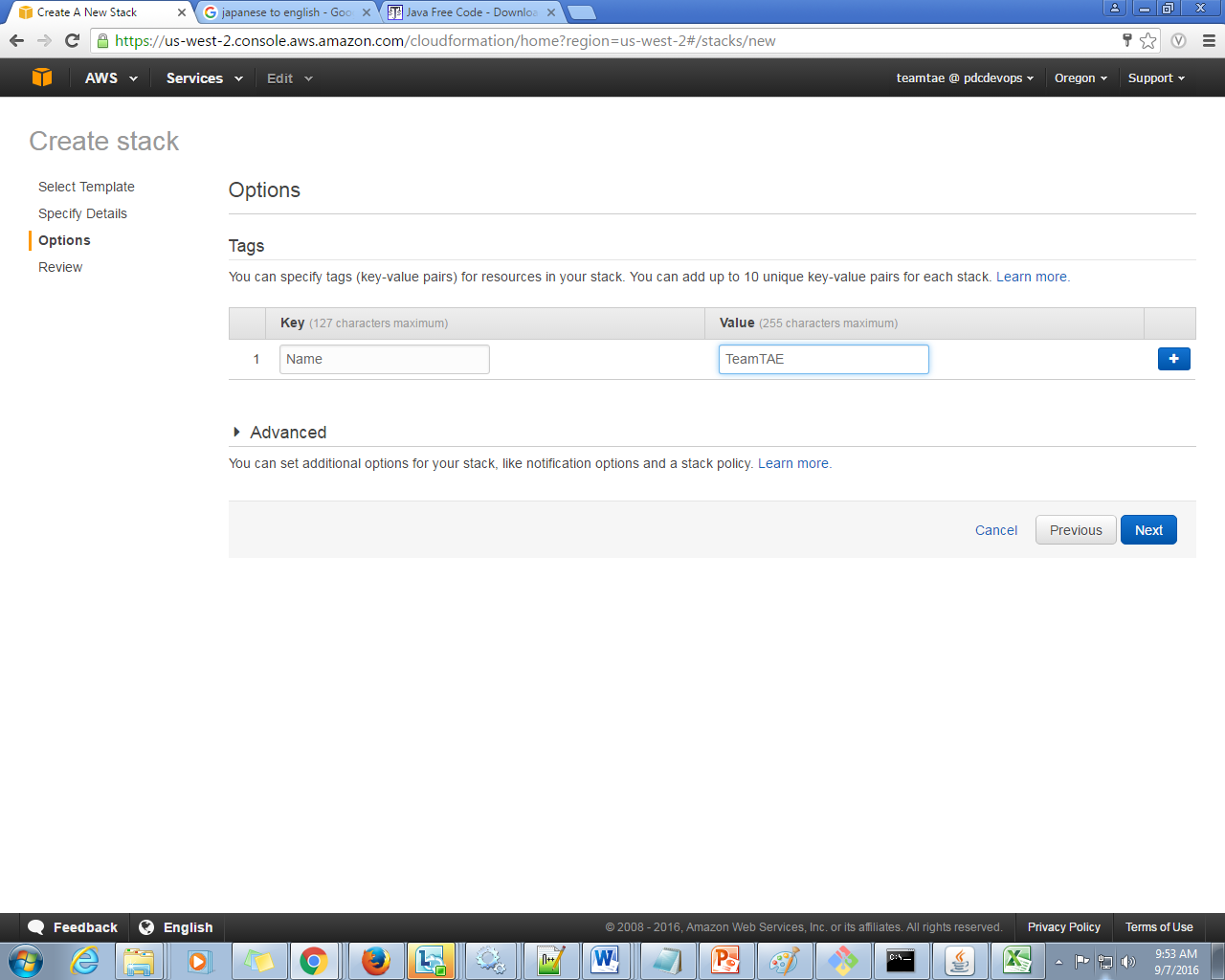
**Figure (). Amazon Web Services home page**



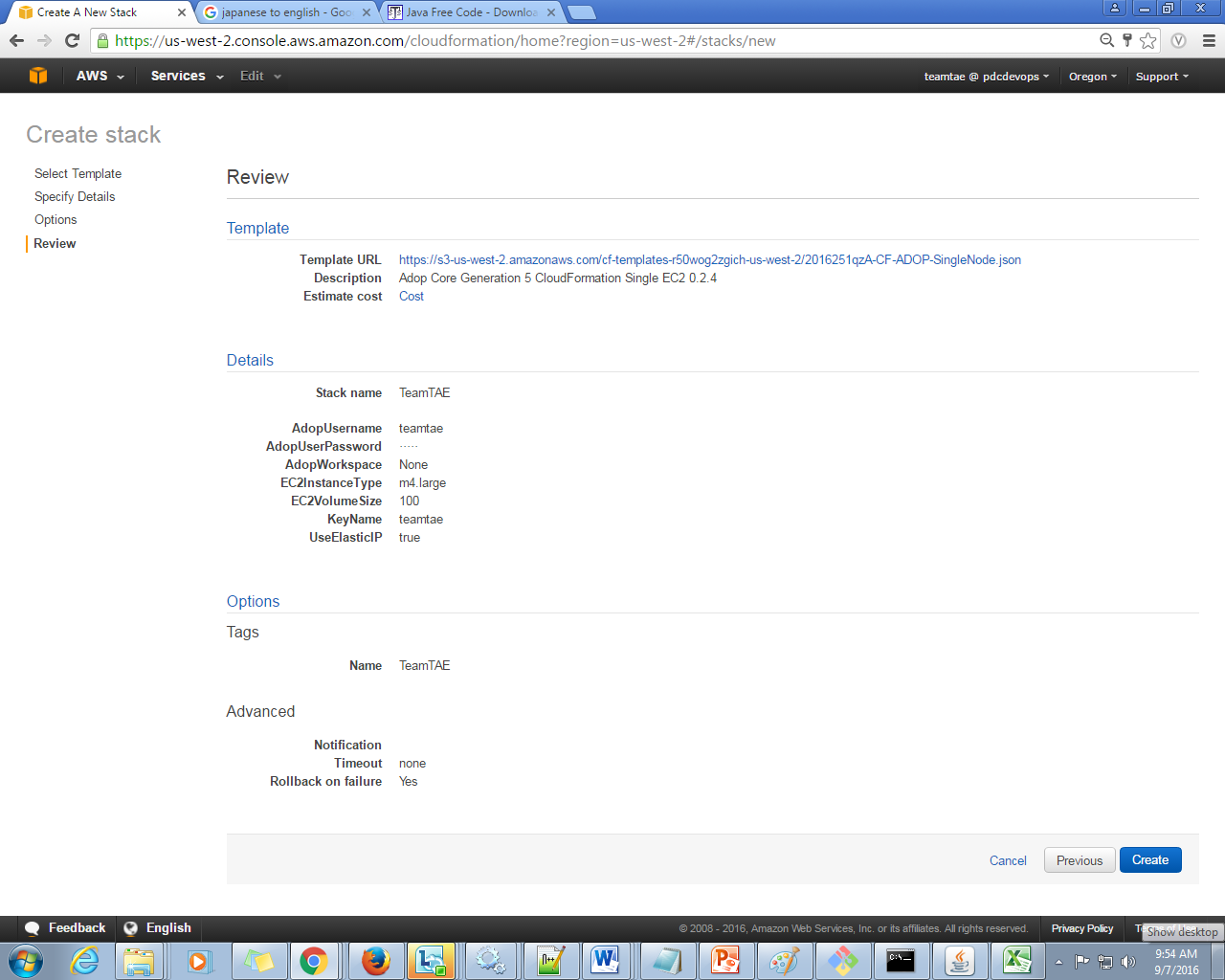
**Figure (). Template URL**



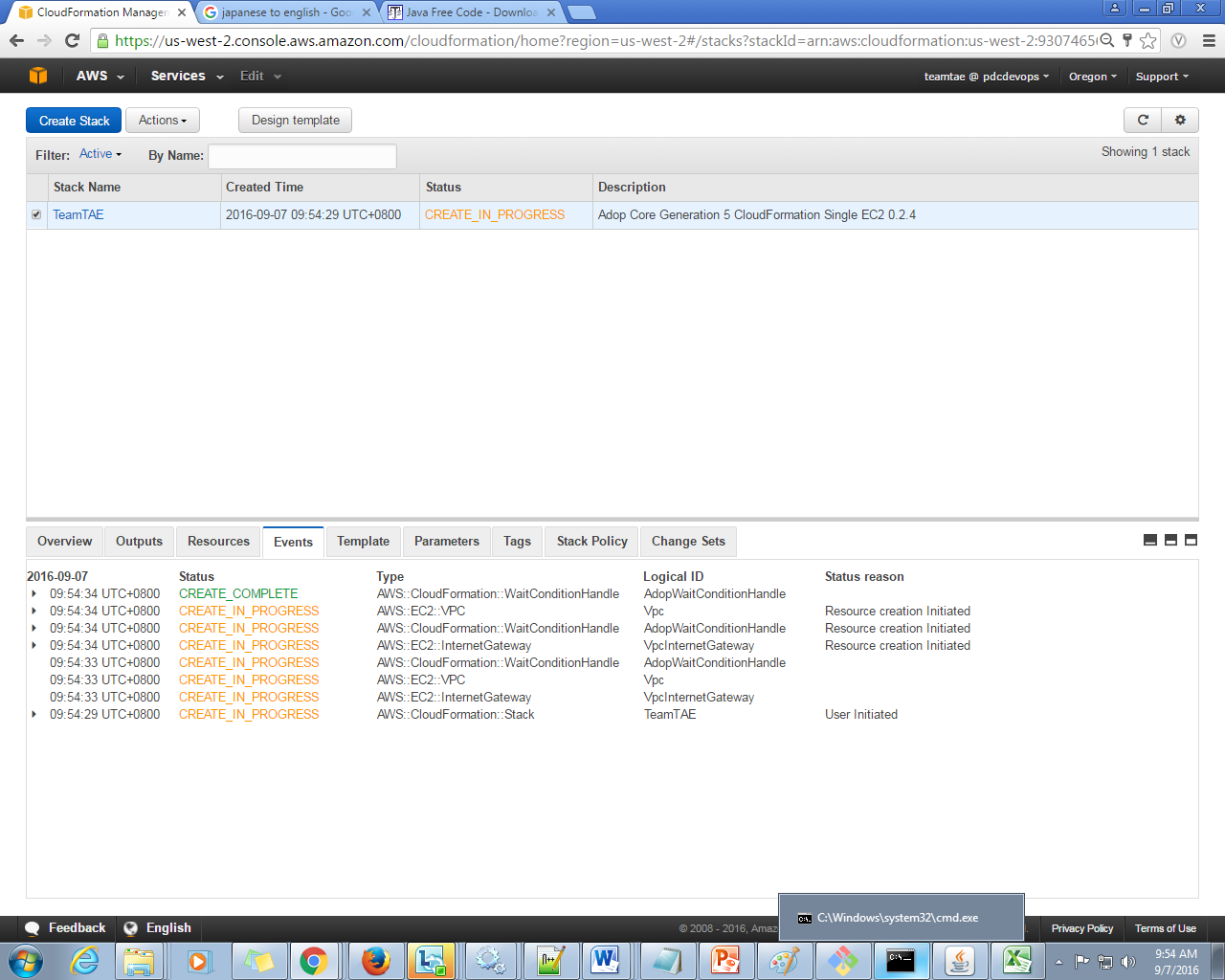
**Figure (). Details and Parameters**



**Figure (). Tags**

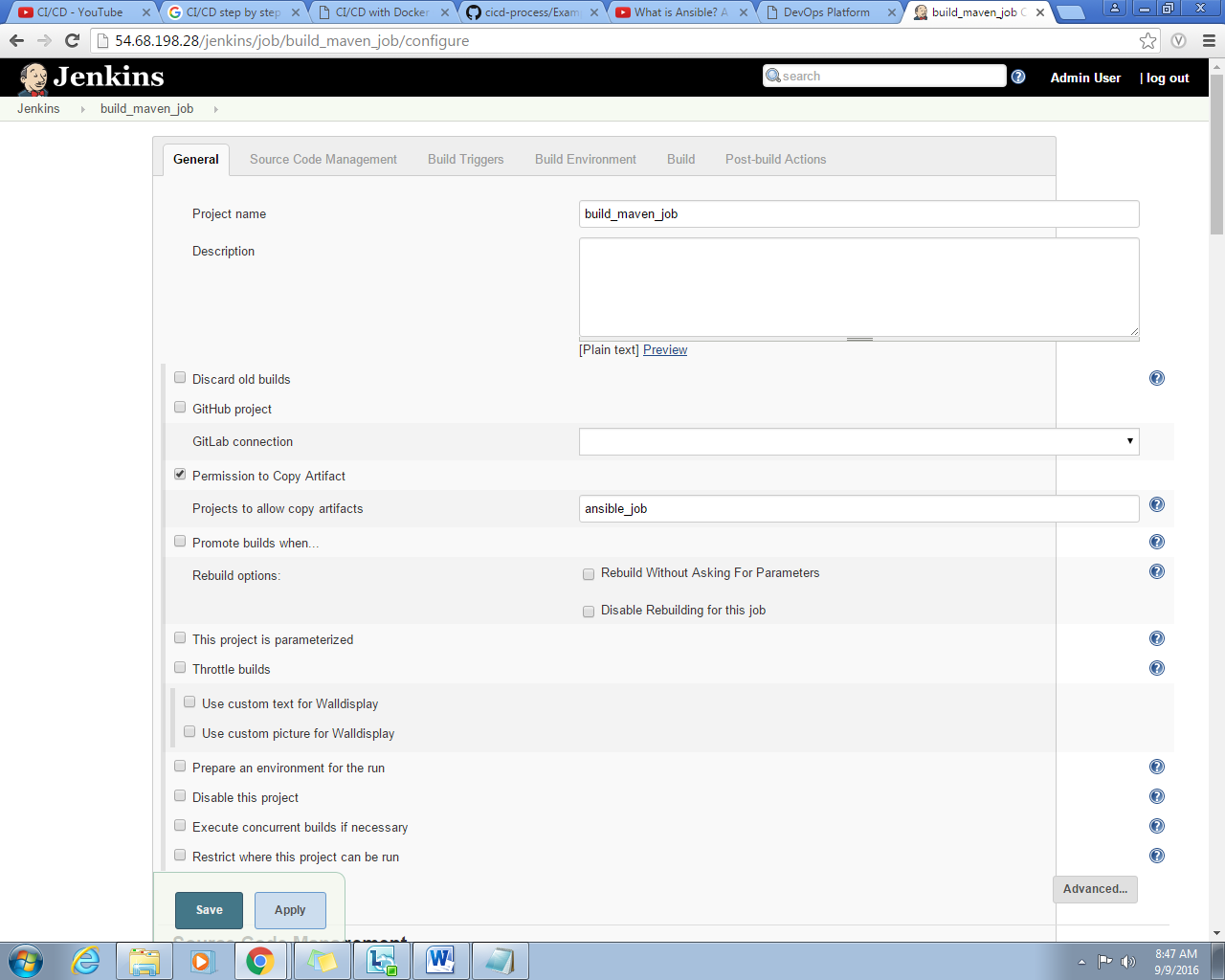


**Figure (). Summary/Review**

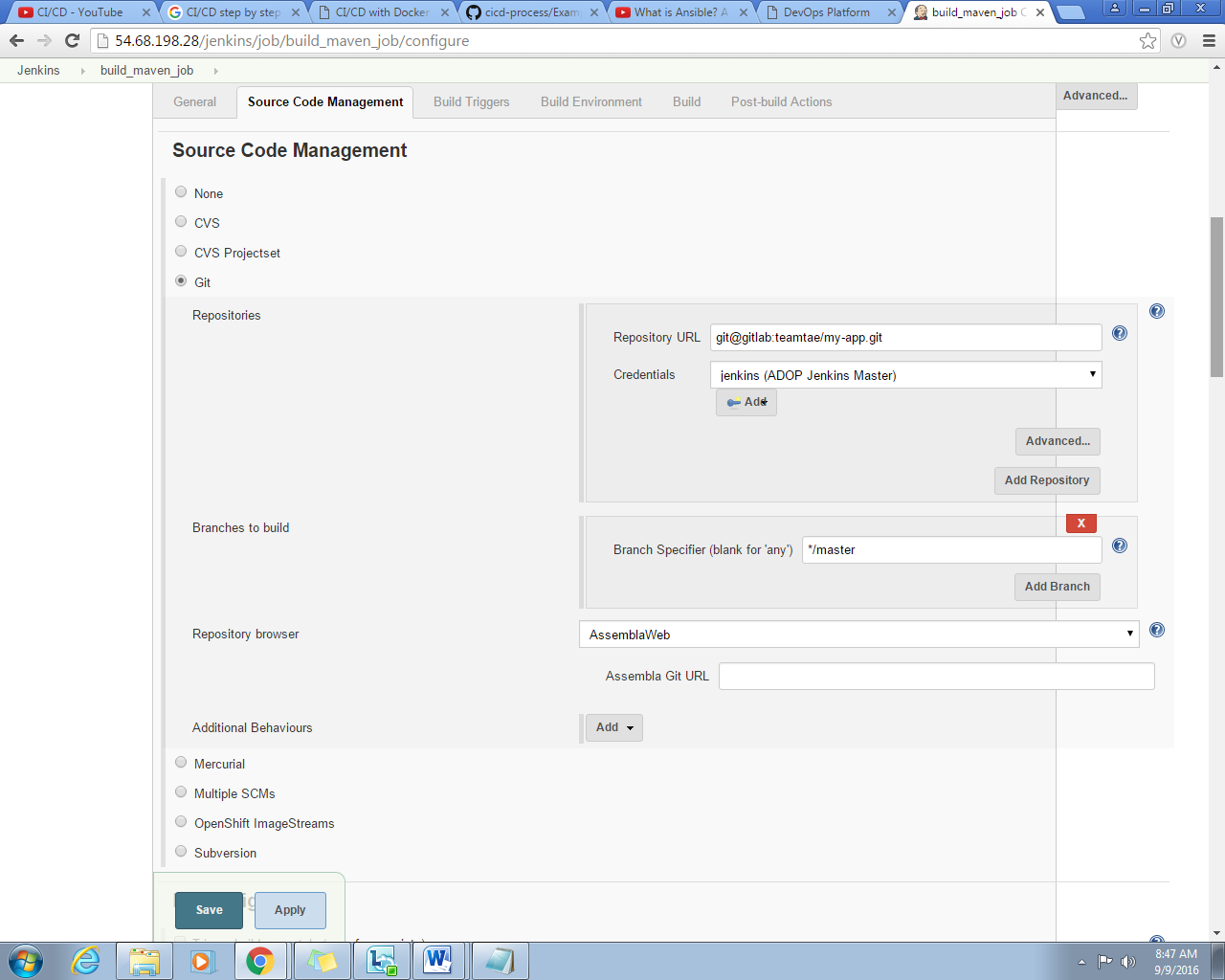


**Figure (). Stack Status**

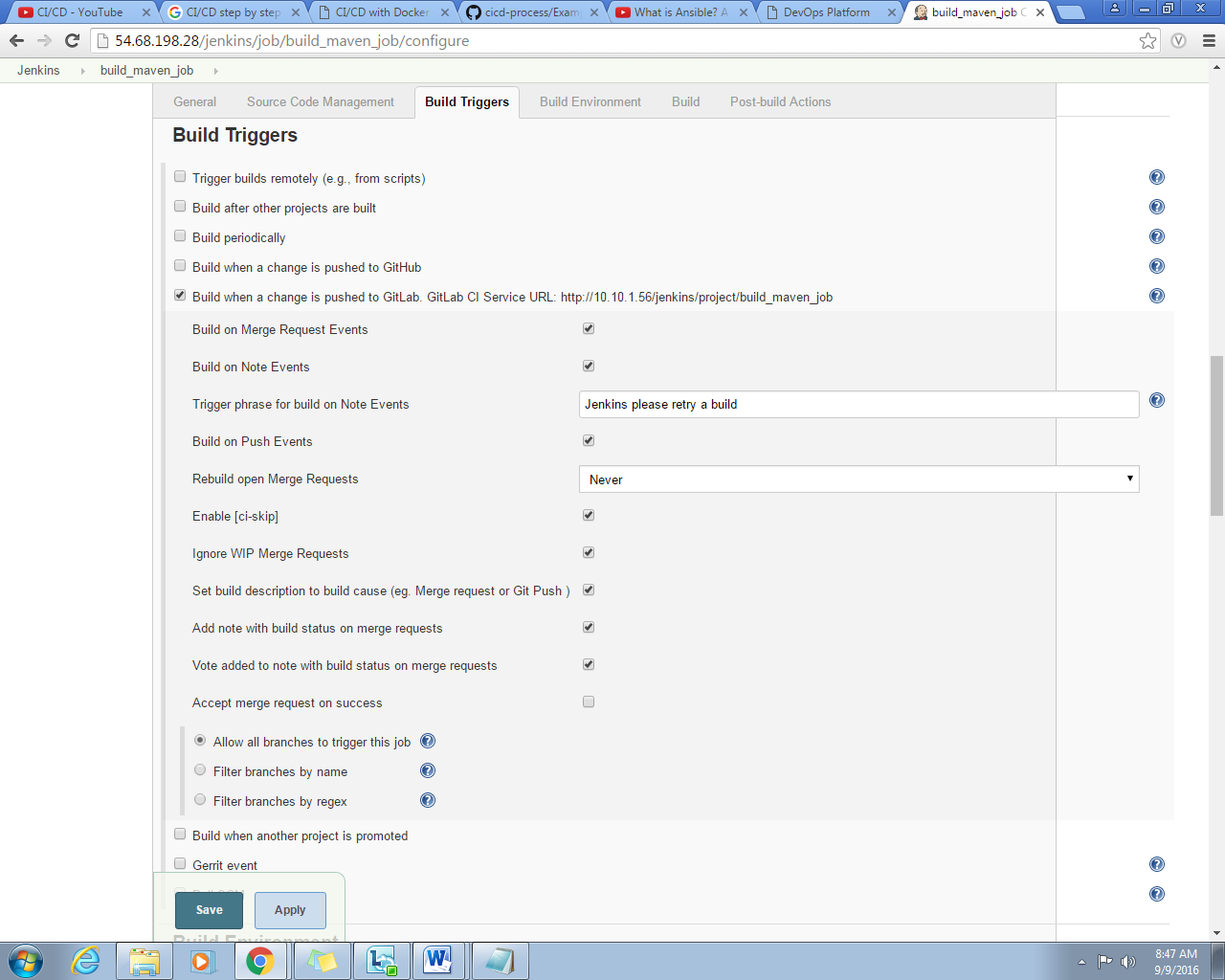
**2.4.3 Maven Configuration**



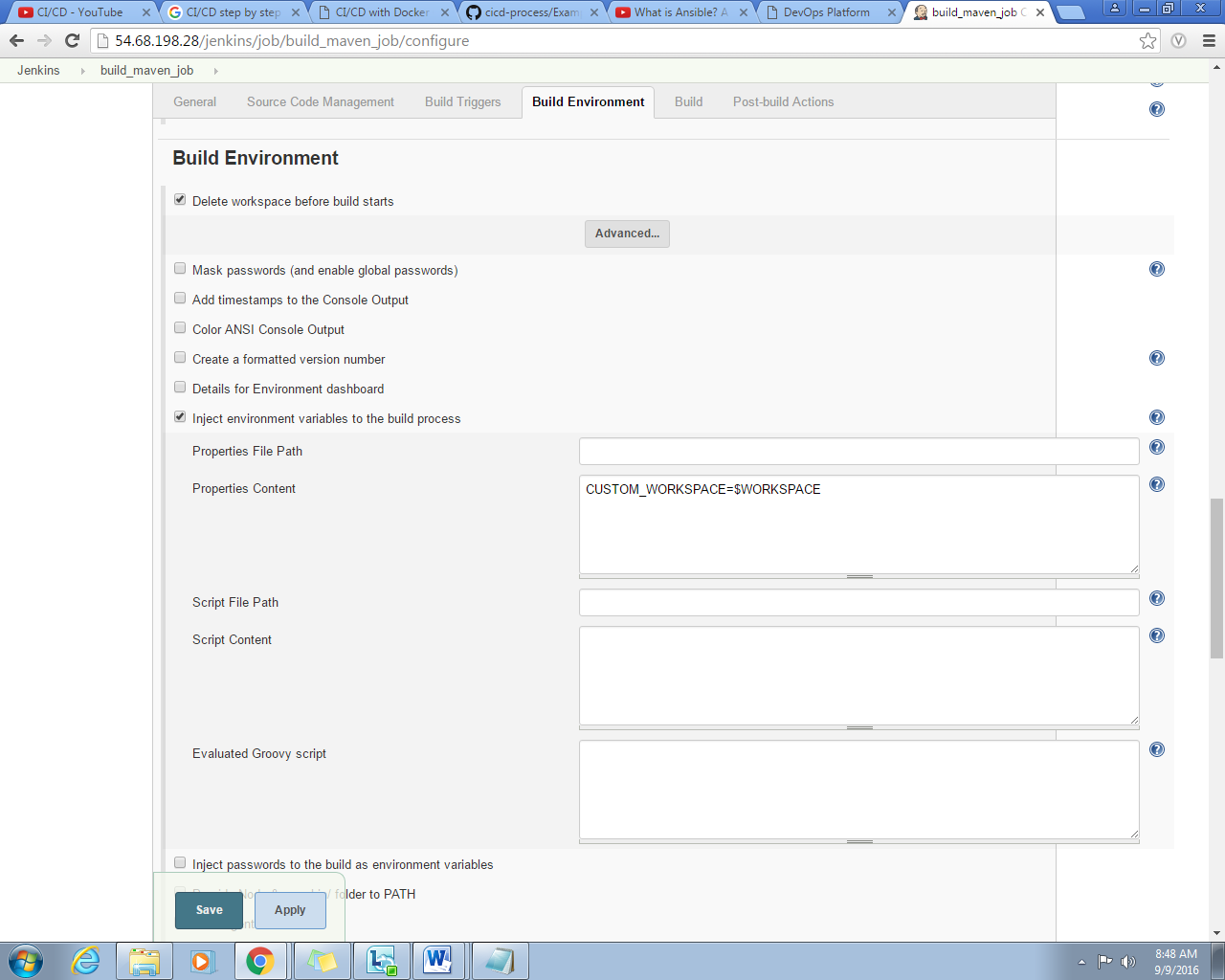
**Figure ().**



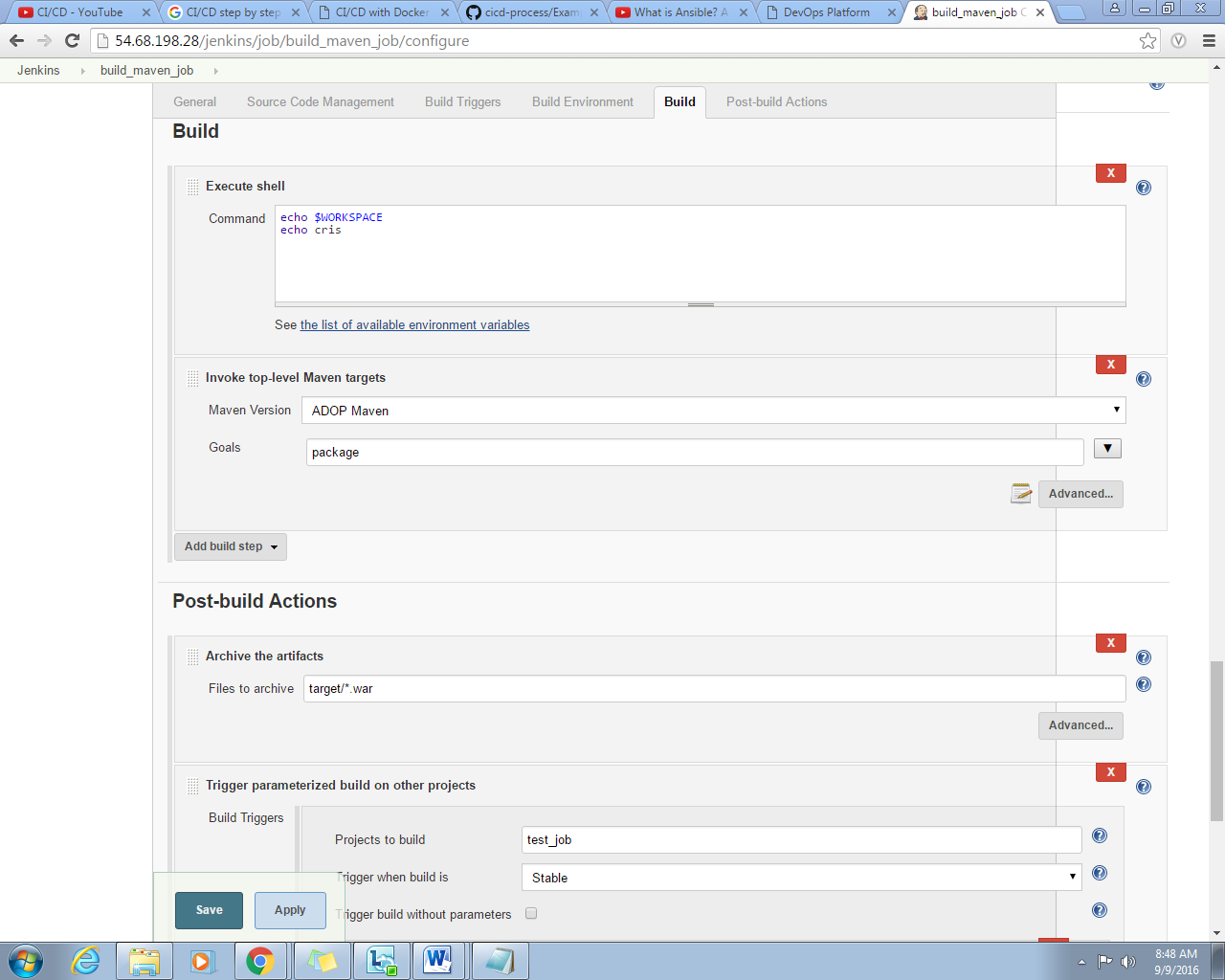
**Figure ().**



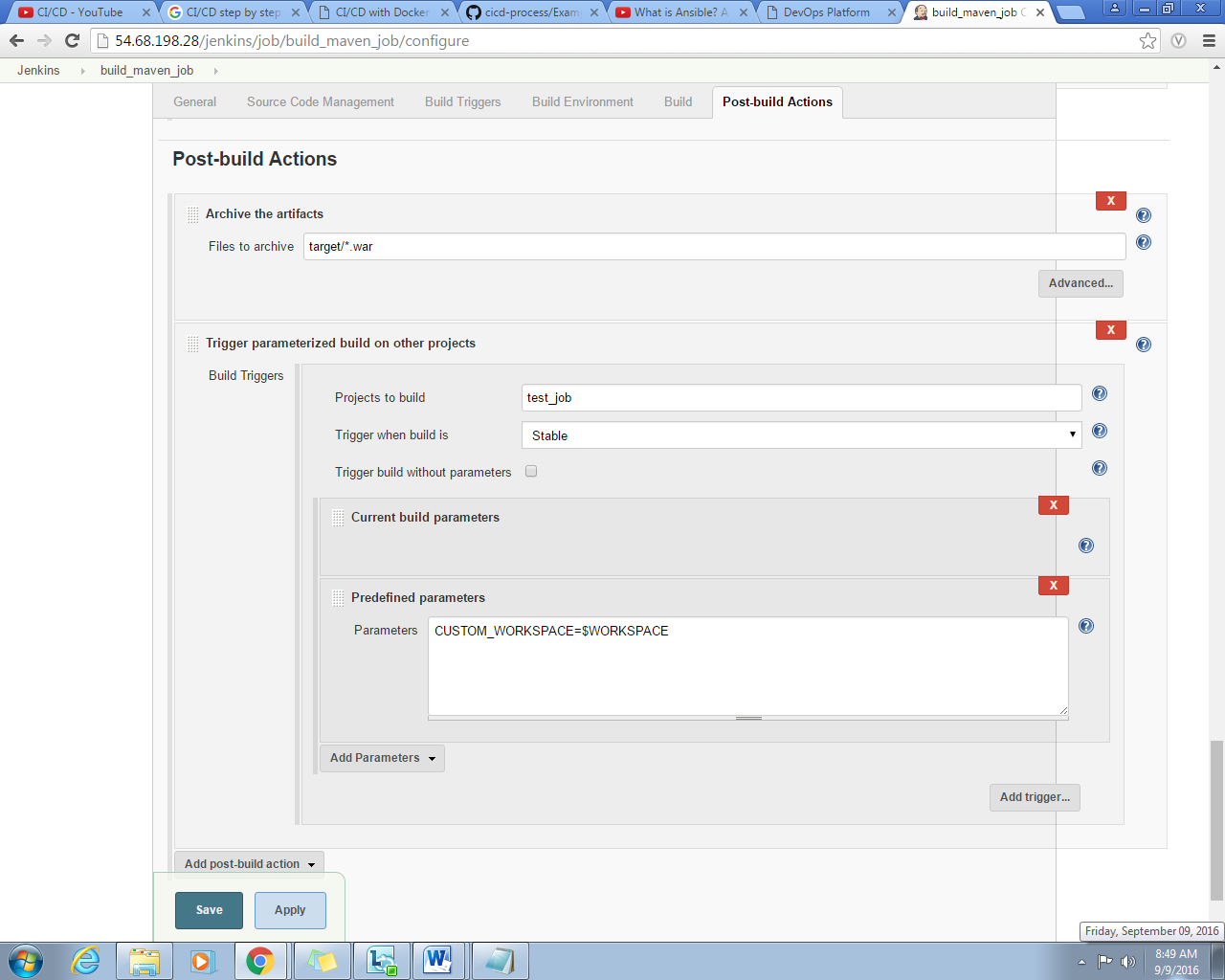
**Figure ().**



**Figure ().**

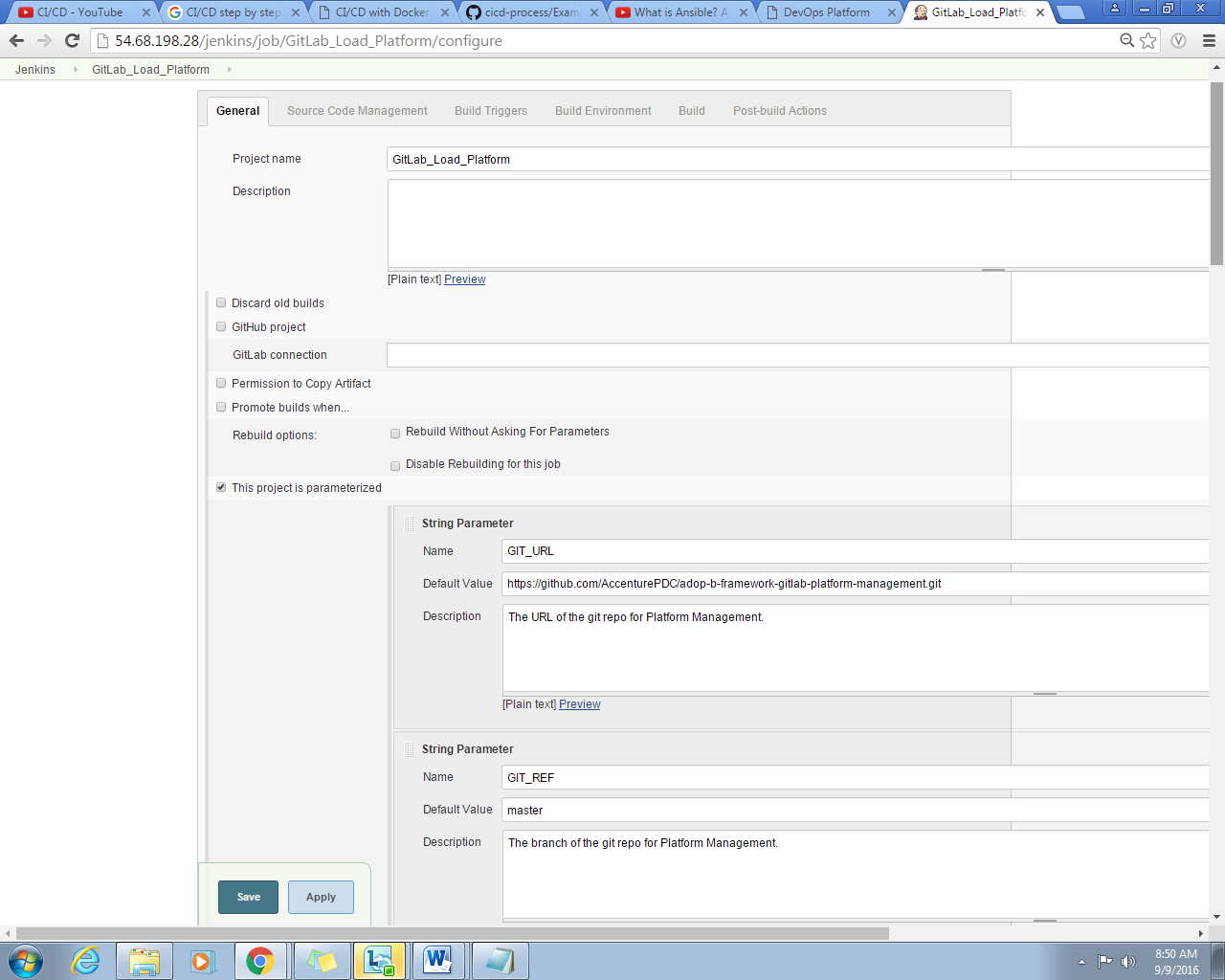


**Figure ().**

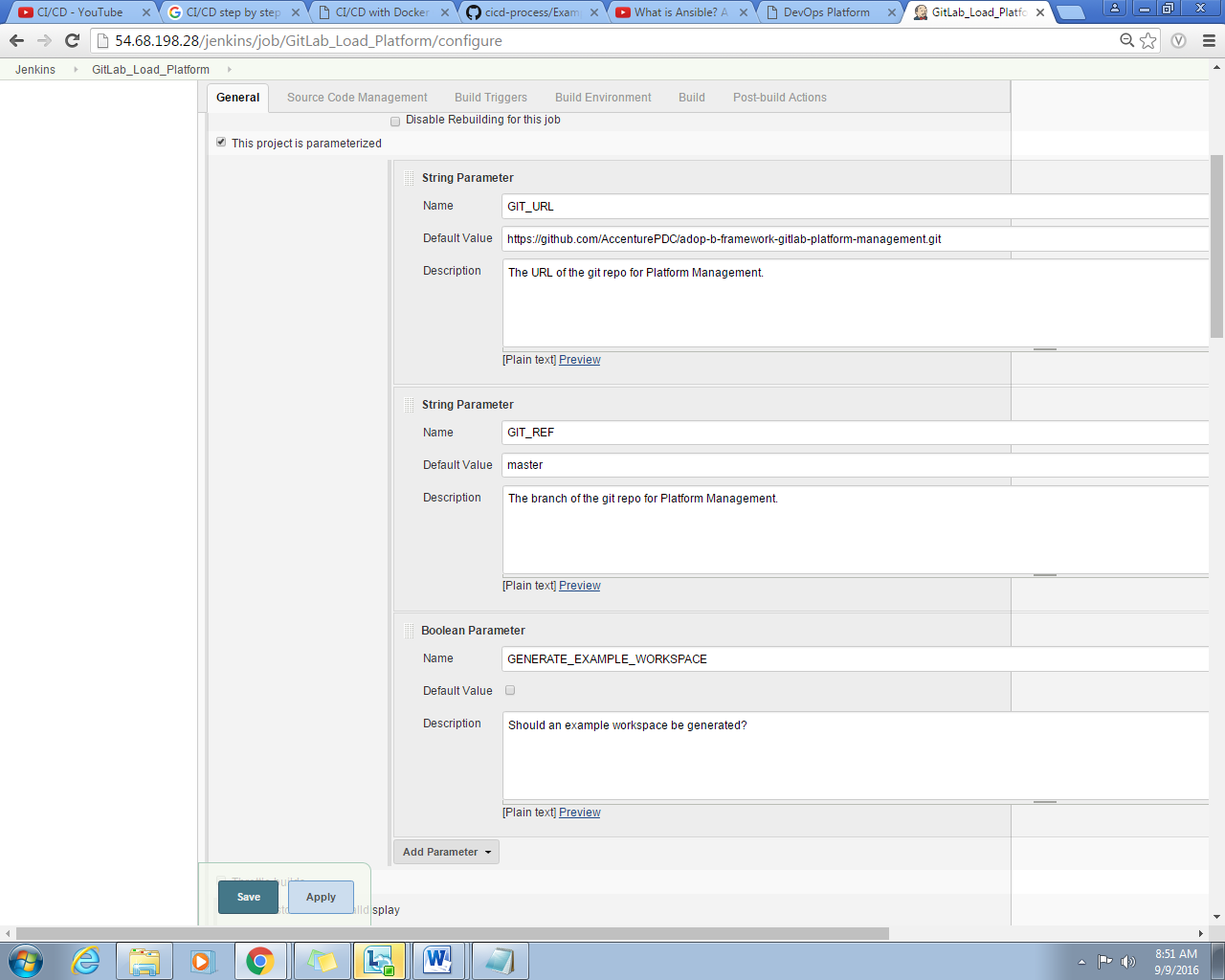


**Figure ().**

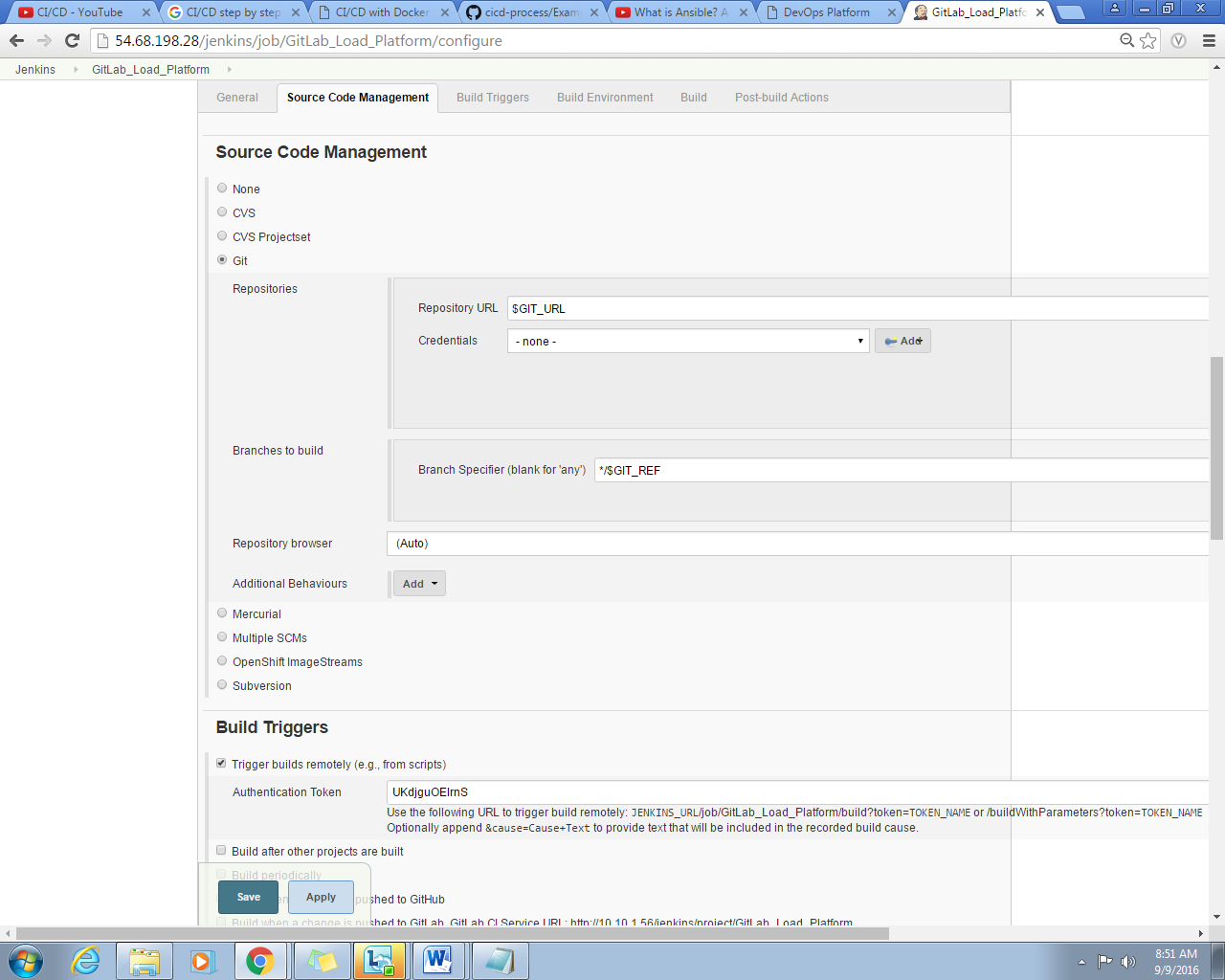
**2.4.4 GitLab Configuration**



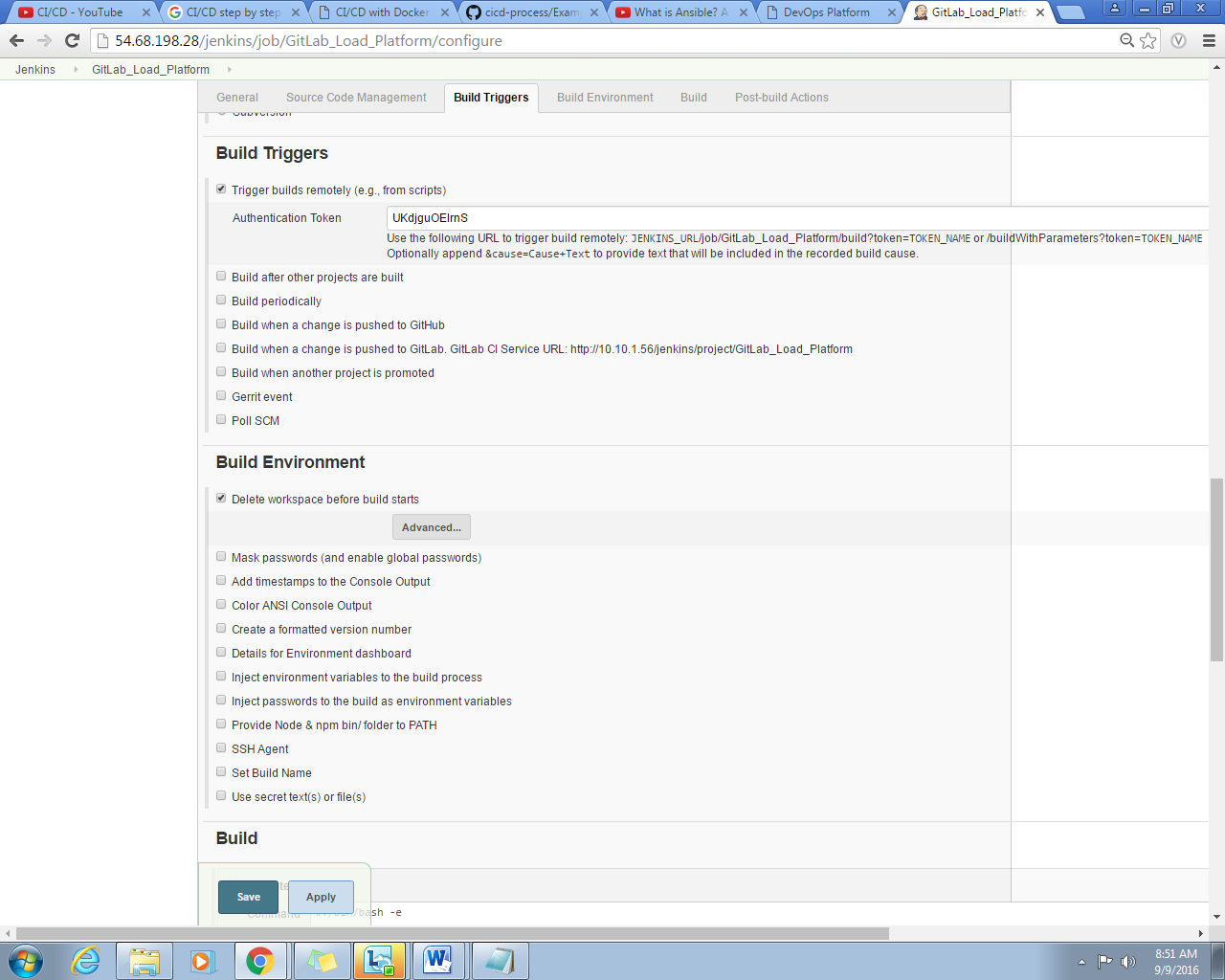
**Figure ().**



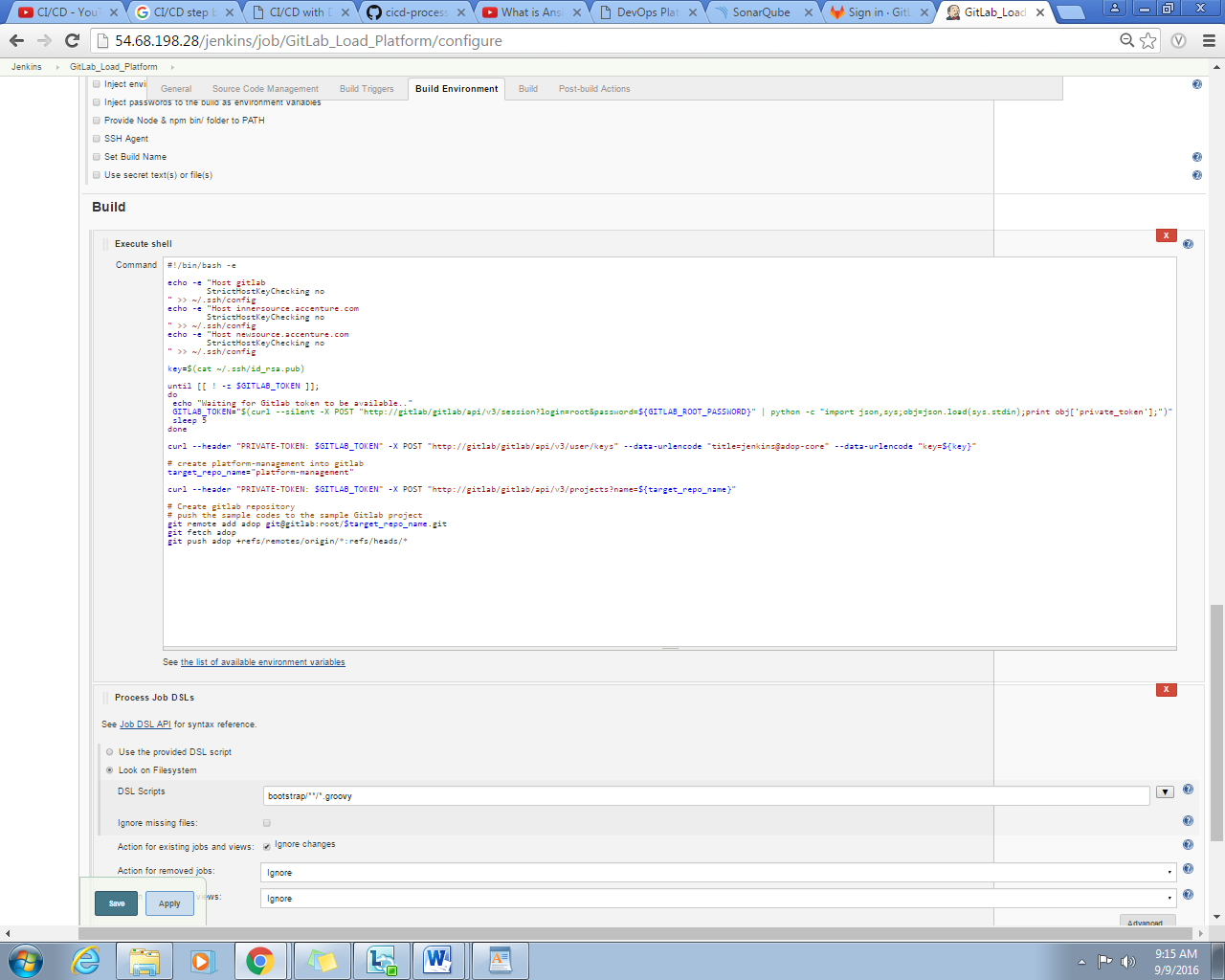
**Figure ().**



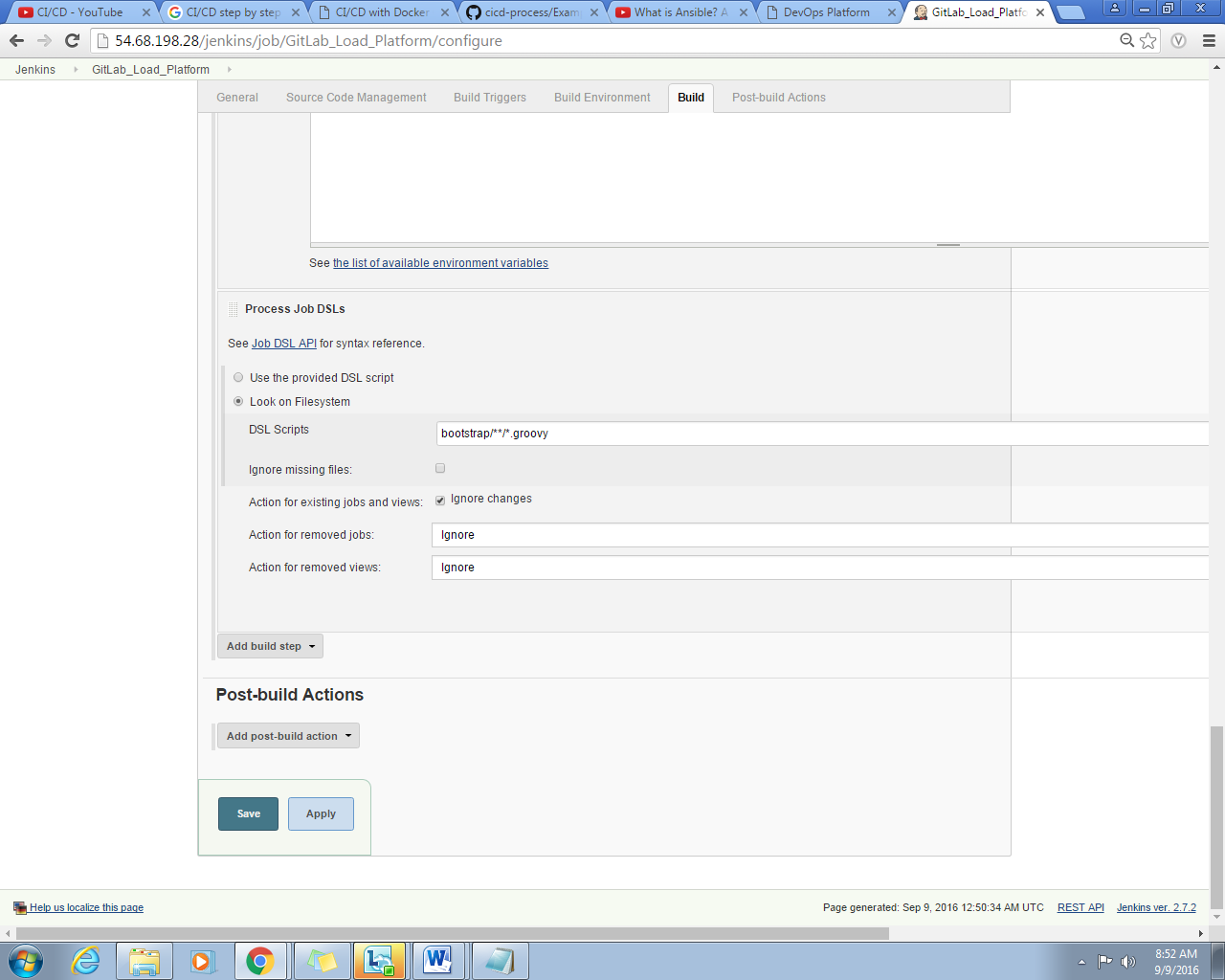
**Figure ().**



**Figure ().**



**Figure ().**



**Figure ().**

**CHAPTER 4 – SUMMARY AND CONCLUSION**

**Blahh… blahh.. blah…**

Display

storage

start

data

Process

decision