

# BUILDING YOUR FIRST DEVSECOPS PIPELINE

A hands-on workshop

## Abstract

The purpose of this technical workshop is to educate individuals on what DevOps is and provide a hands-on training on how to build their first DevSecOps pipeline.

Fane, Abdel Sy  
Abdel.syfane@cybersecuritynp.org

Revision #	Date	Author
1.0	9/3/2019	Abdel Sy Fane



## Table of Contents

<b><i>Introduction</i></b> .....	<b>3</b>
<b>Background</b> .....	<b>3</b>
<b><i>What is Agile?</i></b> .....	<b>3</b>
<b><i>What is Github?</i></b> .....	<b>4</b>
<b>How will we use Github for our workshop?</b> .....	<b>4</b>
<b><i>What is Jenkins?</i></b> .....	<b>4</b>
<b>How will we use Jenkins for our workshop?</b> .....	<b>4</b>
<b><i>What is Docker?</i></b> .....	<b>5</b>
<b>How will we use Docker for our workshop?</b> .....	<b>5</b>
<b><i>What is SonarQube?</i></b> .....	<b>5</b>
<b>How will we use SonarQube for our workshop?</b> .....	<b>5</b>
<b><i>What is Artifactory?</i></b> .....	<b>6</b>
<b>How will we use Artifactory for our workshop?</b> .....	<b>6</b>
<b><i>What is Pivotal Web Services (PWS)?</i></b> .....	<b>6</b>
<b>How will we use Pivotal Web Services (PWS) for our workshop?</b> .....	<b>6</b>
<b><i>Lab</i></b> .....	<b>7</b>
<b>Requirements:</b> .....	<b>7</b>
<b><i>Instructions</i></b> .....	<b>8</b>
<b>How to register for a Pivotal Web Services (PWS/PCF) Account:</b> .....	<b>8</b>
<b>How to register for a Github Account</b> .....	<b>9</b>
<b>How to register for a Jenkins Account</b> .....	<b>10</b>
<b>How to create a Jenkins Pipeline Job</b> .....	<b>11</b>
<b>Configuring your Jenkins Pipeline</b> .....	<b>12</b>
<b>Create a Jenkins Credential ID</b> .....	<b>13</b>
<b>Run Your Jenkins Job</b> .....	<b>13</b>
<b>Congratulations!</b> .....	<b>14</b>

# Introduction

**Background:** As an agile practitioner and a curious individual who likes learning what and how other organizations are delivering secured code, after giving couple of talks on the Allstate DevSecOps model, I have come to the realization that a lot of organizations are having challenges delivering secured code and that DevSecOps is still a fairly new concept to most individuals so I decided that it would be helpful to do a technical workshop where interested individuals can attend a live hands-on workshop and learn some basics techniques.

In this technical workshop, we will use some common open-source tools to build our DevOps pipeline and we will add a security scan step (I will add more DevOps and Security tools over time).

**Tools:** Github, Jenkins, Docker, SonarQube, Artifactory & Pivotal Web Services.

**Bio:** Abdel Sy Fane

Abdel is an Application Security Manager at Allstate and president of the CyberSecurity NP (CSNP) organization. With over five years of experience in security and 10 years in the IT industry, Abdel is passionate about a wide range of security topics, including Threat Intelligence, DevSecOps, and Artificial Intelligence and security integration. He received his master's in Cyber Forensics & Security from Illinois Institute of Technology in 2014, and since then he has consulted for the Veteran's Administration, PayNet and Allstate. As president of CCS, Abdel is dedicated to unifying the security community and promoting security education.

[www.linkedin.com/in/abdelsyfane](http://www.linkedin.com/in/abdelsyfane)  
[abdel.syfane@cybersecuritynp.org](mailto:abdel.syfane@cybersecuritynp.org)

## What is Agile?

- People who practice agile methodology can't seem to agree on a clear definition but for the purpose of this workshop, agile is both a culture and a process to effectively and continuously deliver/deploy applications. This process involve key stakeholders like the customer, product/project managers, scrum masters and developers. The key here is that all stakeholders are on the same page when developers are developing a product unlike the waterfall methodology where developers are told what the customer "Wants" and they proceed to build the application and the customer has no idea what the developers are building until the product is complete/in production. Waterfall can be especially painful if the customer does not like the product or if there are security bugs but the product is already in production. The whole concept of DevOps was born from Agile.



## What is **Github**?

- Github is a Source Control Management (SCM) tool, essentially a code repository. Github has many features but we'll keep it simple here. Github allows multiple developers to work on the same code and create different version of the code through branches (development, staging, production, etc.). Github keeps track of all of your code changes that were made over time, who made those changes and even allows you to revert back to a code change anytime in the past.

## How will we use **Github** for our workshop?

- The code we will write for this workshop will be stored in Github



## What is **Jenkins**?

- Jenkins is a Continuous Integration / Continuous Deployment/Delivery tool that allows you to define your application build process in a repeatable manner. This basically means that you only ever need to define/build your application development/deployment process only once and over time, you can iterate by adding features without re-defining the entire process and each time you build your application, the build will work as long there's no dependency changes.

## How will we use **Jenkins** for our workshop?

- We will create a Jenkins job with instructions for Jenkins to build and deploy our sample application.
- When we run Jenkins jobs, a random Jenkins slave will be chosen to run our job. A Jenkins Node is a collection of slaves.



## What is **Docker**?

- Docker allows you to create a container (or a very light virtual machine) with its own runtime environment. Unlike traditional virtual machines which could take hours/days to configure/build and configure again, a docker container only takes seconds to create and configure. Docker allows us to create an isolated virtual environment where we practically do/run anything we want without affecting other running containers.

## How will we use **Docker** for our workshop?

- We're going to use docker in Jenkins to create containers so that whatever we are testing or building in our Jenkins environment would not affect someone else who is using the same Jenkins slave.



## What is **SonarQube**?

- SonarQube is a Quality Assurance (QA) tool that is used by testers and developers to ensure that their code does not have bugs and that they are following software development best practices. SonarQube also has a slight security feature that scans your code for known vulnerabilities or “bad” coding that could introduce vulnerabilities.

## How will we use **SonarQube** for our workshop?

- We're going to use SonarQube to focus on security related issues in our code.
- Note that in the real world, you can set security policies that would stop a developer from deploying their application into production or any environment if there are security bugs in their code but for this lab, we will only review the security flaws.



## What is **Artifactory**?

- Artifactory is a code repository mainly used for both application version control and as well for storing binaries and other 3<sup>rd</sup> party libraries.

## How will we use **Artifactory** for our workshop?

- We will make an API call to Artifactory to download our sample java application and once we build our java application, a jar file will be generated, we will store this jar file back on Artifactory. This is the same jar file that we will deploy on Pivotal Web Services.



Pivotal **Web Services**

## What is **Pivotal Web Services (PWS)**?

- To simplify Pivotal Web Services, it is a Platform as a Service tool that allows you to deploy any application on top of without worry about the underlying operating system, what kernel version, what language is supported and so much more. PWS abstracts this underlying layer for the developer so that they can focus more on building their application while PWS handles the operations of the application.

## How will we use **Pivotal Web Services (PWS)** for our workshop?

\*\*\*PCF is the paid version while PWS is free\*\*\*

- We're simply going to take our sample application and deploy it on PWS.

# Lab

## Requirements:

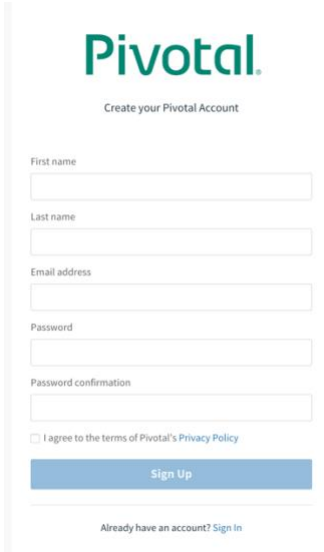
1. Bring a **Laptop** (Windows or Mac)
2. Password Manager - <https://www.dashlane.com/> (Optional)
3. Create a Github Account <https://github.com/join> (Instructions below) **\*\*\*Please Use a Unique Password**
4. Install Git <https://git-scm.com/downloads>
5. Create a PWS Account <https://account.run.pivotal.io/z/uaa/sign-up> (Instructions below) **\*\*\*Please Use a Unique Password**
6. Create a Jenkins Account in the CSNP Lab [URL](#) (Instructions below)
7. Any Code Editor like Notepad++ or Sublime/Vs Code (Optional)



# Instructions

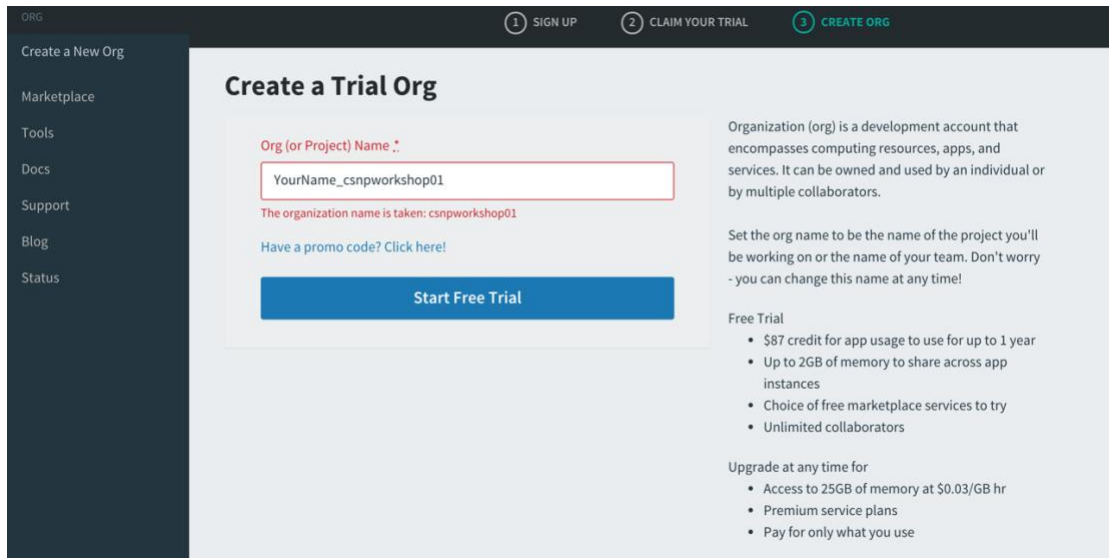
## How to register for a Pivotal Web Services (PWS/PCF) Account:

1. Follow this [URL](#) and complete the form



The image shows the Pivotal account creation form. At the top is the Pivotal logo and the text "Create your Pivotal Account". Below this are input fields for "First name", "Last name", "Email address", "Password", and "Password confirmation". There is a checkbox for "I agree to the terms of Pivotal's Privacy Policy" and a blue "Sign Up" button. At the bottom, it says "Already have an account? Sign In".

2. Do Not Enter Any Billing Info or you will be charged after a period!
3. Once you've created your free account, create an organization.



The image shows the "Create a Trial Org" page. At the top, there is a navigation bar with three steps: 1 SIGN UP, 2 CLAIM YOUR TRIAL, and 3 CREATE ORG (highlighted in green). On the left is a dark sidebar with links: "Create a New Org", "Marketplace", "Tools", "Docs", "Support", "Blog", and "Status". The main content area has the heading "Create a Trial Org". Below this is a form with the label "Org (or Project) Name :". The input field contains "YourName\_csnpworkshop01". Below the input field, a red error message says "The organization name is taken: csnpworkshop01". There is a link "Have a promo code? Click here!". Below the form is a blue "Start Free Trial" button. To the right of the form, there is text explaining that an organization (org) is a development account that encompasses computing resources, apps, and services. It can be owned and used by an individual or by multiple collaborators. Below this, it says "Set the org name to be the name of the project you'll be working on or the name of your team. Don't worry - you can change this name at any time!". Further down, it lists the "Free Trial" benefits: \$87 credit for app usage to use for up to 1 year, Up to 2GB of memory to share across app instances, Choice of free marketplace services to try, and Unlimited collaborators. At the bottom, it says "Upgrade at any time for" and lists the upgrade benefits: Access to 25GB of memory at \$0.03/GB hr, Premium service plans, and Pay for only what you use.

# How to register for a Github Account

1. Go to this [URL](#) and create your account

The screenshot shows the GitHub registration page. At the top, it says 'Join GitHub' and 'The best way to design, build, and ship software.' Below this are three steps: Step 1: Set up your account, Step 2: Choose your subscription, and Step 3: Tailor your experience. The main section is 'Create your personal account' with fields for Username, Email address, and Password. There are instructions for each field. To the right, a box titled 'You'll love GitHub' lists benefits: Unlimited public repositories, Unlimited private repositories, Limitless collaboration, Frictionless development, and Open source community. Below the registration fields is a 'Verify account' section with a puzzle image and a 'Verify' button. At the bottom, there is a 'Create an account' button and a link to the Terms of Service and Privacy Statement.

2. Once you've logged in to Github, please Fork this [repo](#) by clicking on “Fork” on the top right corner of the page. This will create your own copy of this code.


The screenshot shows the GitHub repository page for 'cybersecuritynp / spring-music-app'. The repository is forked from 'abdelsfane/spring-music-app'. At the top right, there are buttons for Watch (0), Star (0), and Fork (1). Below these are tabs for Code, Pull requests (0), Projects (0), Wiki, Security, Insights, and Settings. The main content area shows 'No description, website, or topics provided.' and a 'Manage topics' link. Below this, there are statistics: 12 commits, 2 branches, 0 releases, 1 contributor, and GPL-3.0 license. At the bottom, there are buttons for 'Branch: master', 'New pull request', 'Create new file', 'Upload files', 'Find File', and 'Clone or download'.

3. Using your terminal/command line, create a copy of your repo to your local machine for editing by typing “git clone” and URL of your repo. (i.e. git clone <https://github.com/yourgithubusername/spring-music-app.git>)

4. Change directory into your cloned folder and edit the “Jenkinsfile”. In your version of the code, we’re going to change the following variables:
  - a. PCF\_ORG on **line 17** to **“your\_org\_name”**
  - b. abdel\_pcf\_user on **line 33** to **“yourname\_pcf\_user”**
5. Once you’ve made the above changes, now go ahead and add, commit your changes and push it by typing “git add .”, “git commit -m “updating environment variables” and then “git push”

## How to register for a Jenkins Account

1. Go to this [URL](#) and create your account



### Welcome to Jenkins!

Please sign in below or [create an account](#).

Sign in

☐ Keep me signed in

### Create an account!


If you already have a Jenkins account, [please sign in](#).

☐ Show

A strong password is a long password that's unique for every site. Try using a phrase with 5-6 words for the best security.

Create account

2. Once you’ve created your account and login, you’ll see a folder called “csnp\_workshop”, go ahead and click on it.

 Jenkins

2

search

CSNP | log out

Jenkins

New Item

People

Build History

Project Relationship

Check File Fingerprint



Manage Jenkins

My Views

Lockable Resources


Credentials


Allnm+


S	W	Name ↓	Last Success	Last Failure	Last Duration
		csnp_workshop	N/A	N/A	N/A

Icon: S M L

Legend

 RSS for all

 RSS for failures

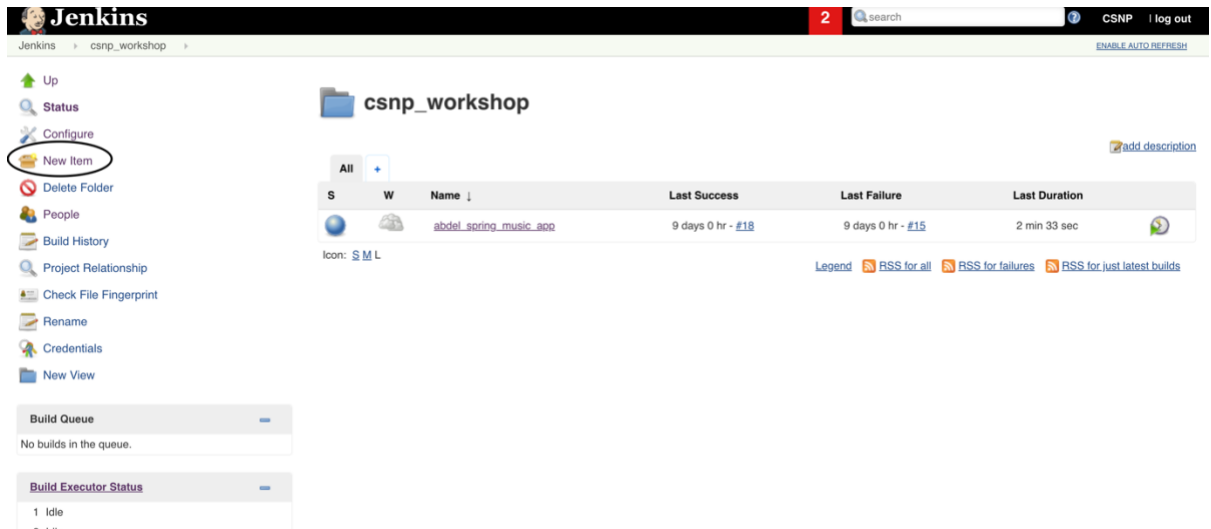
 RSS for just latest builds

add description

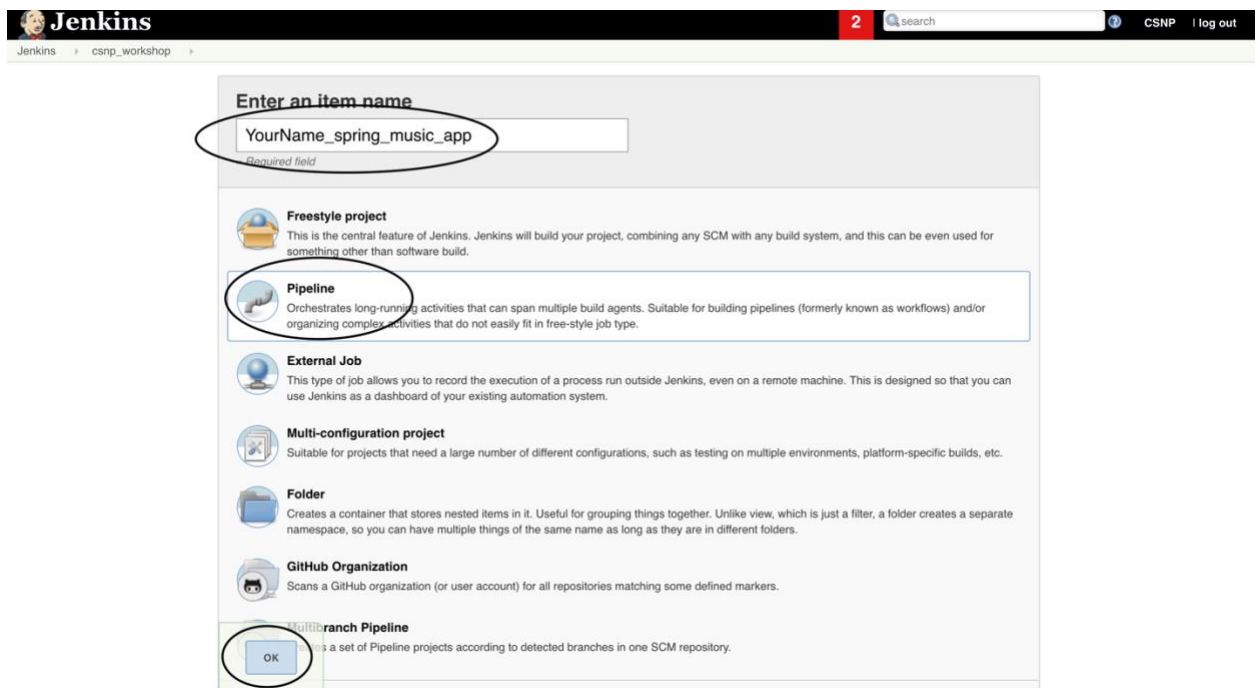
ENABLE AUTO REFRESH

# How to create a Jenkins Pipeline Job

1. On the left pane, click “New Item”



1. Enter a name for your pipeline
2. Click on the “Pipeline” button
3. Click “Ok”



## Configuring your Jenkins Pipeline

1. Click on the “Pipeline” tab
2. Next to “Definition”, drop down to “Pipeline Script from SCM”
3. Next to “SCM”, drop down to “Git”
4. Next to “Repository URL”, enter your Github Repo URL
5. Click “Save”

Jenkins > csnp\_workshop > YourName\_spring\_music\_app >

General Build Triggers Advanced Project Options **Pipeline**

### Pipeline

Definition **Pipeline script from SCM**

SCM **Git**

Repositories

Repository URL **YourGitHub-Repository-URL**  
**Please enter Git repository.**

Credentials **- none -**  
**Add**

**Advanced...**  
**Add Repository**

Branches to build

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

Repository browser **(Auto)**


Additional Behaviours **Add**


Script Path **Jenkinsfile**

**Save** **Apply** **Lightweight checkout** ☒

## Create a Jenkins Credential ID

1. Click this [link](#) and click “Add Credentials” to add your PWS/PCF username and password
2. Next to “Kind”, select “Username with Password” and type your username/password
3. Next to “ID”, type **“yourname\_pcf\_user”**
4. Set the “Description” to **“something\_you\_recognize pcf user”**

 [Back to credential domains](#)

 **Add Credentials**

Kind

Username with password

Username

myuserID

Password

.....

ID

yourname\_pcf\_user


Description

some\_random\_text pcf user

OK

## Run Your Jenkins Job


1. Now, go back to your Jenkins Job and click “Build Now” and your Jenkins job will start running!


 **Jenkins**


2


CSNP | log out


Jenkins > csnp\_workshop > YourName\_spring\_music\_app > [ENABLE AUTO REFRESH](#)


 Up


 Status


 Changes


 **Build Now**


 Delete Pipeline


 Configure

 Move

 Full Stage View

 Rename

 Pipeline Syntax


 Build History

[trend](#)


[RSS for all](#) [RSS for failures](#)

**Pipeline YourName\_spring\_music\_app**

Full project name: csnp\_workshop/YourName\_spring\_music\_app

 [add description](#)

[Disable Project](#)

 [Recent Changes](#)

**Stage View**

No data available. This Pipeline has not yet run.

**Permalinks**

## Congratulations!

1. [Login](#) to SonarQube to review your results!
  - a. Username: system
  - b. Password: csnpworkshop01
2. [Login](#) to Artifactory to review the new java app you added!
  - a. Username: csnp
  - b. Password: csnpworkshop01
3. [Login](#) to PWS to see your running application!