

BugX Academy

No bug is allowed.

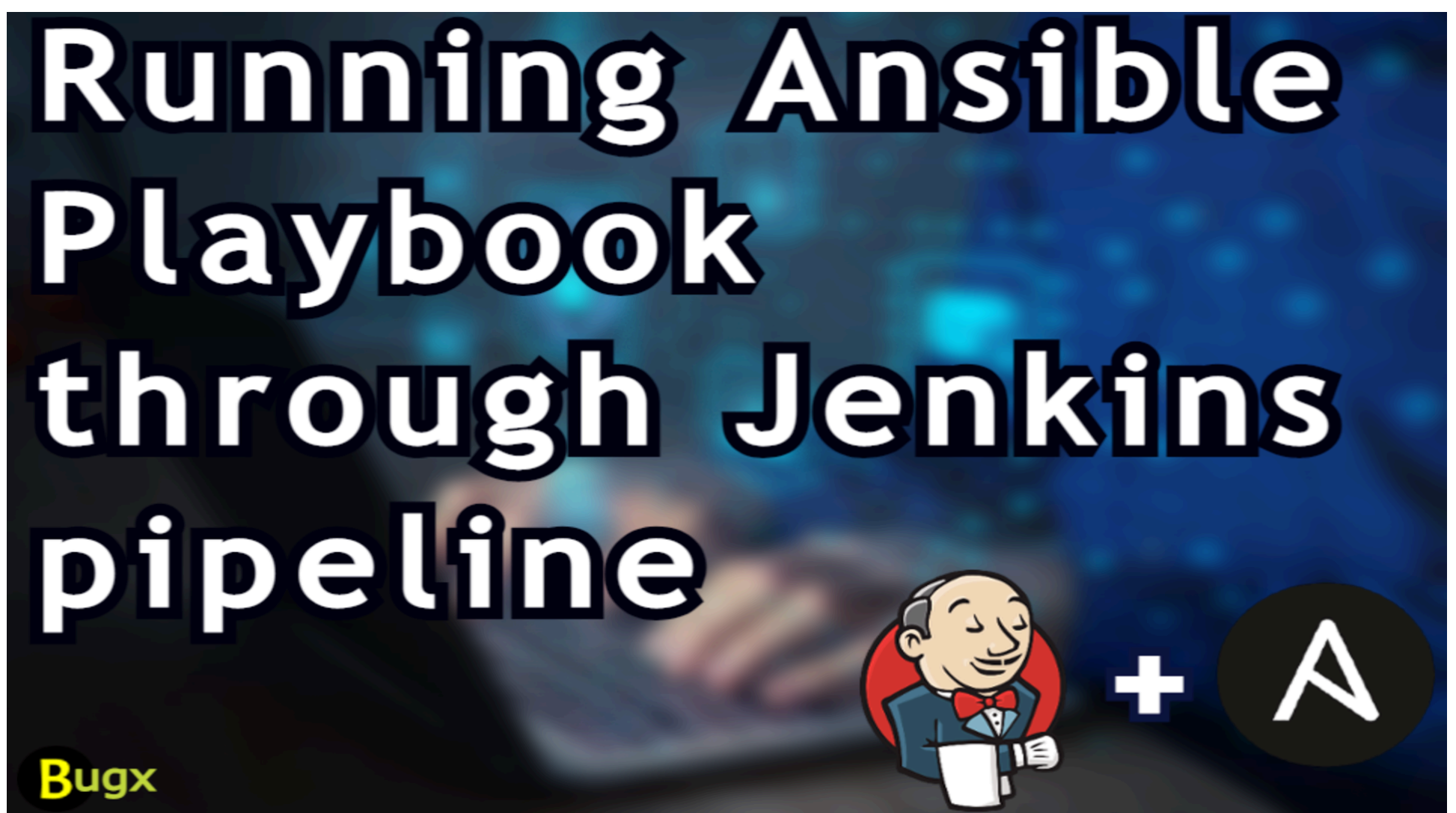


Tahir

Follow

Tahir is a self-taught DevOps Engineer.

How to run Ansible Playbook through Jenkins pipeline?



In this tutorial, we are gonna learn how to run ansible playbook through Jenkins CI/CD pipeline. (Simple DevOps project for beginners).

Before switch to the steps, If you're interested in pursuing a career in Linux Administration, consider to enroll in our course [Linux Mastery: From Beginner to Confident Linux User](#) on Udemy.

What is Jenkins?

Jenkins is an open source continuous automation software DevOps tool written in the Java programming language. It helps automate the parts of software development related to building, testing, and deploying, facilitating continuous integration (CI) and continuous delivery (CD). It is a server-based system that runs in servlet containers such as Apache Tomcat.

What is Ansible?

Ansible is an open source, command-line IT automation software application written in Python and developed by **Red Hat**. It can configure systems, deploy software, and orchestrate advanced workflows to support application deployment, system updates, and more. Let's say we have hundreds of servers, and we want add some files to those servers. Instead of log in every server and running some commands, we just create simple Ansible playbook where it goes to servers and add files at the same time which is more efficient in terms of time consuming.

Creating new user

STEP 1. Before installation it is better to add new user and install Jenkins, Ansible and run commands with that user.

```
1 useradd jenkins
2 # make password for new user
3 passwd jenkins
```

- Make created user as part of sudoers which can easily run commands with privileges of root user.

```
1 vim /etc/sudoers/
2 # type following line under "root    ALL=(ALL)        ALL"
3 jenkins    ALL=(ALL)        NOPASSWD:  ALL
```

Installing ansible in our machine where we will also install Jenkins.

STEP 2 Make sure that python is installed in your machine.

```
1 python3 -m pip -V
```

- If all is well, you should see something like the following: `pip 23.0.1 from /home/jenkins/.local/lib/python3.9/site-packages/pip (python 3.9)`.
- Use pip to install ansible package.

```
1 python3 -m pip install --user ansible
```

- Confirm that ansible is installed successfully. by this command you check ansible-core package that has been installed.

```
1 ansible --version
```

- To check the version of the ansible package that has been installed:

```
1 python3 -m pip show ansible
```

Installing Jenkins in our machine where we also installed Ansible before.

STEP 3 To install Jenkins run below commands in your machine.

```
1 sudo wget -O /etc/yum.repos.d/jenkins.repo \  
2     https://pkg.jenkins.io/redhat-stable/jenkins.repo  
3 #If wget command is not installed in your machine, try to install it wit 'sudo yum install wget -y'  
4 sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io.key  
5 sudo yum upgrade -y  
6 # Add required dependencies for the jenkins package  
7 sudo yum install java-11-openjdk  
8 sudo yum install jenkins  
9 sudo systemctl daemon-reload
```

- If Jenkins succesfully installed then start it.

```
1 sudo systemctl enable jenkins  
2 sudo systemctl start jenkins  
3 # make sure jenkins is started succesfully, active nad running by following command  
4 sudo systemctl status jenkins
```

- As a default Jenkins running on port number **8080**, so if you are using firewall make sure that port is added to your firewall.

```
1 sudo firewall-cmd --zone=public --add-port=8080/tcp  
2 # do not forget to reload  
3 sudo firewall-cmd --reload
```

NOTE : If you have application which is already running on port number **8080** then you need to forward Jenkins to work on another port number.

Configuring Jenkins

STEP 4 Now, you can go to your browser and type **http://<ip address of your machine>:8080** , then you will see following screen below:

All you need to do is go to your machine terminal and run **cat <path-to-initial-password>** , then copy resulted password and paste it to the **Administrator password** part.

Getting Started

Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log (**not sure where to find it?**) and this file on the server:

```
/var/lib/jenkins/secrets/initialAdminPassword
```

Please copy the password from either location and paste it below.

Administrator password

Continue

After typing password you will see below screen. Click metioned section **Install suggested plugins** or you can also install specific plugins as you want.

Getting Started

Customize Jenkins

Plugins extend Jenkins with additional features to support many different needs.

Install suggested plugins

Install plugins the Jenkins community finds most useful.

Select plugins to install

Select and install plugins most suitable for your needs.

When installation was completed, below screen will seem. You have two options here. You can skip creating user (you can create it later) and continue as a admin or write down details of user and continue.

Getting Started

Create First Admin User

Username

Password

Confirm password

Full name

Jenkins 2.375.3

Skip and continue as admin

Save and Continue

In following window, **save and continue** for Jenkins url where you will able to access your Jenkins environment.

Getting Started

Instance Configuration

Jenkins URL:

http://3.64.178.191:8080/

The Jenkins URL is used to provide the root URL for absolute links to various Jenkins resources. That means this value is required for proper operation of many Jenkins features including email notifications, PR status updates, and the `BUILD_URL` environment variable provided to build steps.

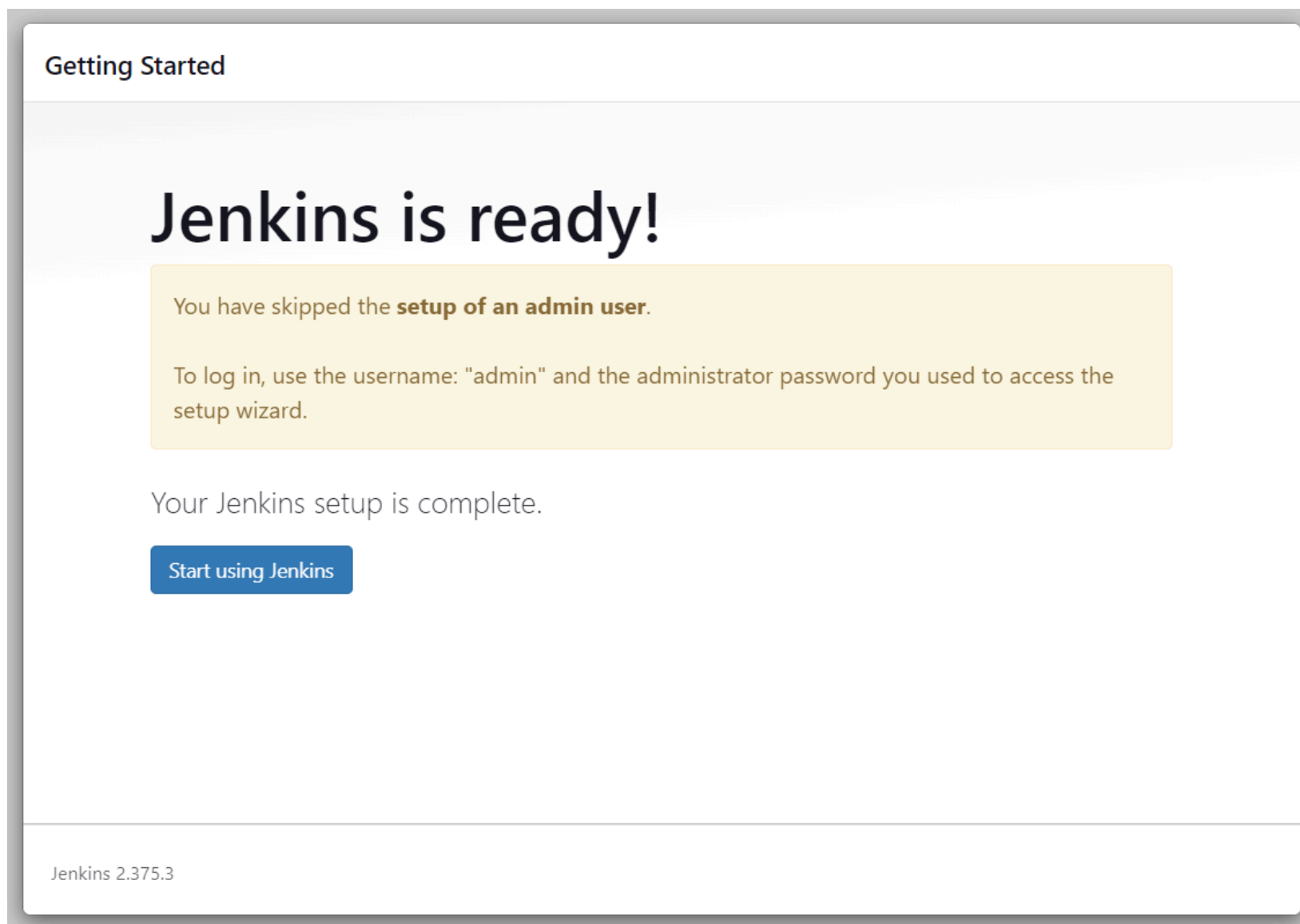
The proposed default value shown is **not saved yet** and is generated from the current request, if possible. The best practice is to set this value to the URL that users are expected to use. This will avoid confusion when sharing or viewing links.

Jenkins 2.375.3

Not now

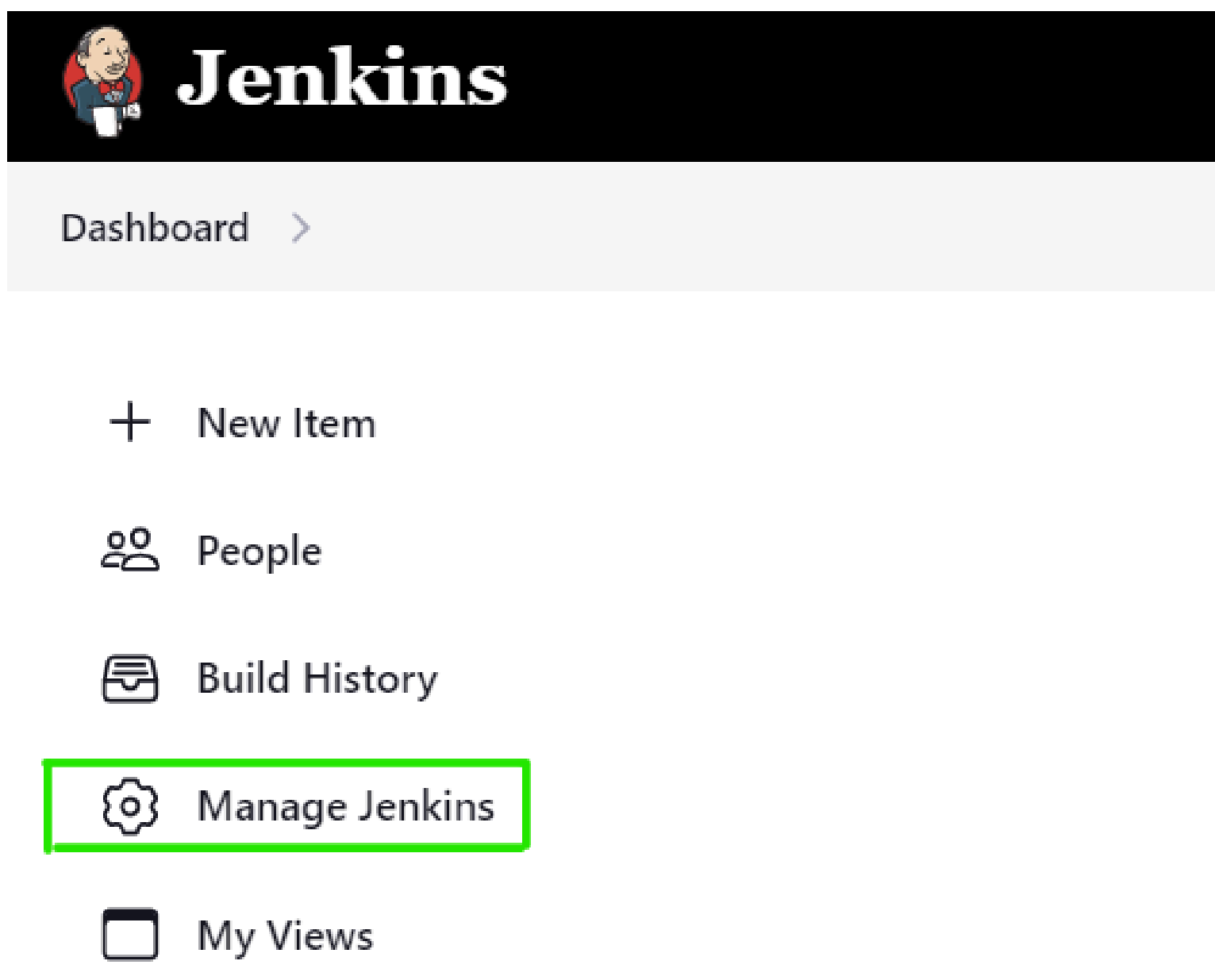
Save and Finish

Now our Jenkins is ready, so press **Start using Jenkins** for going further steps.

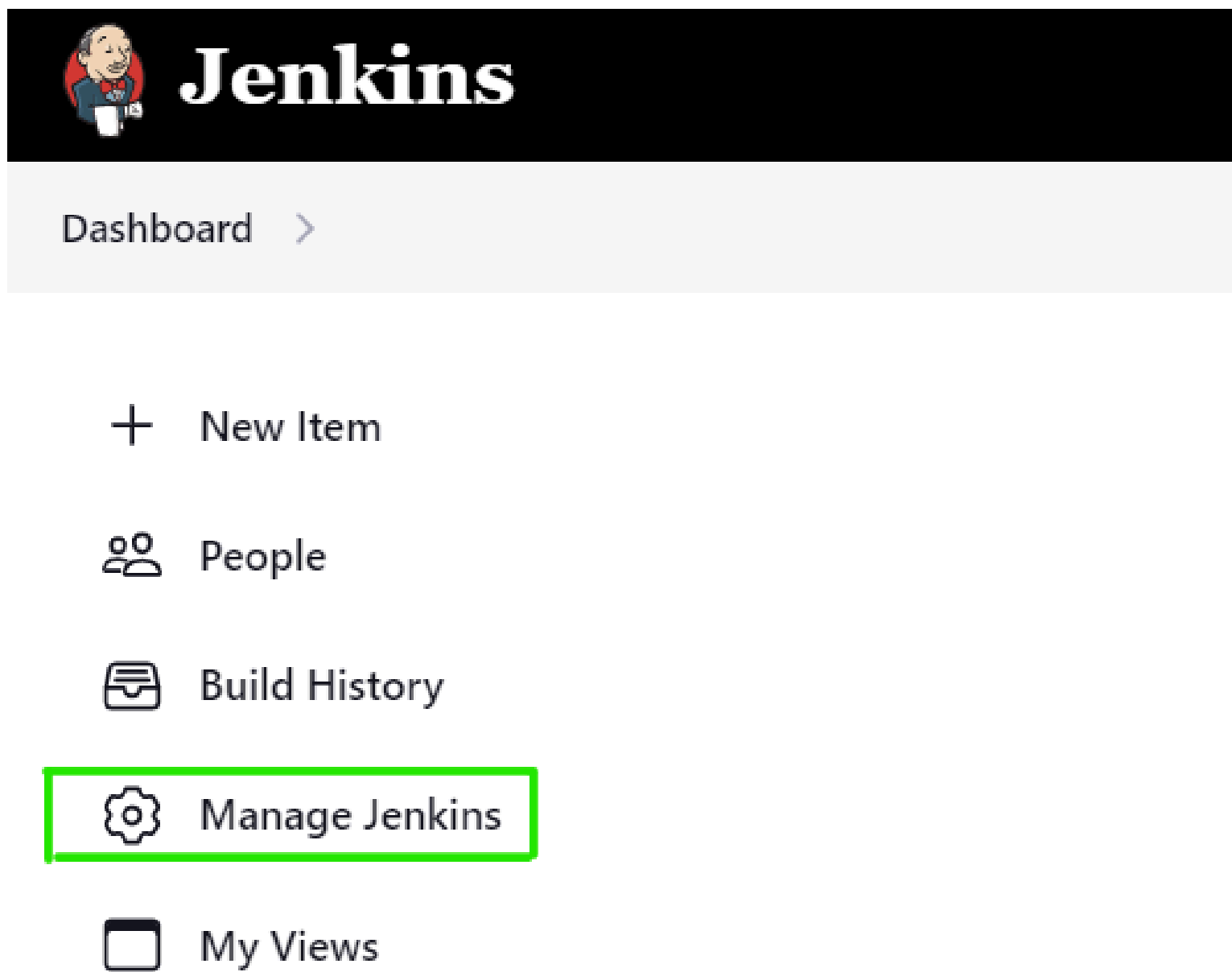


Adding Ansible plugins to Jenkins

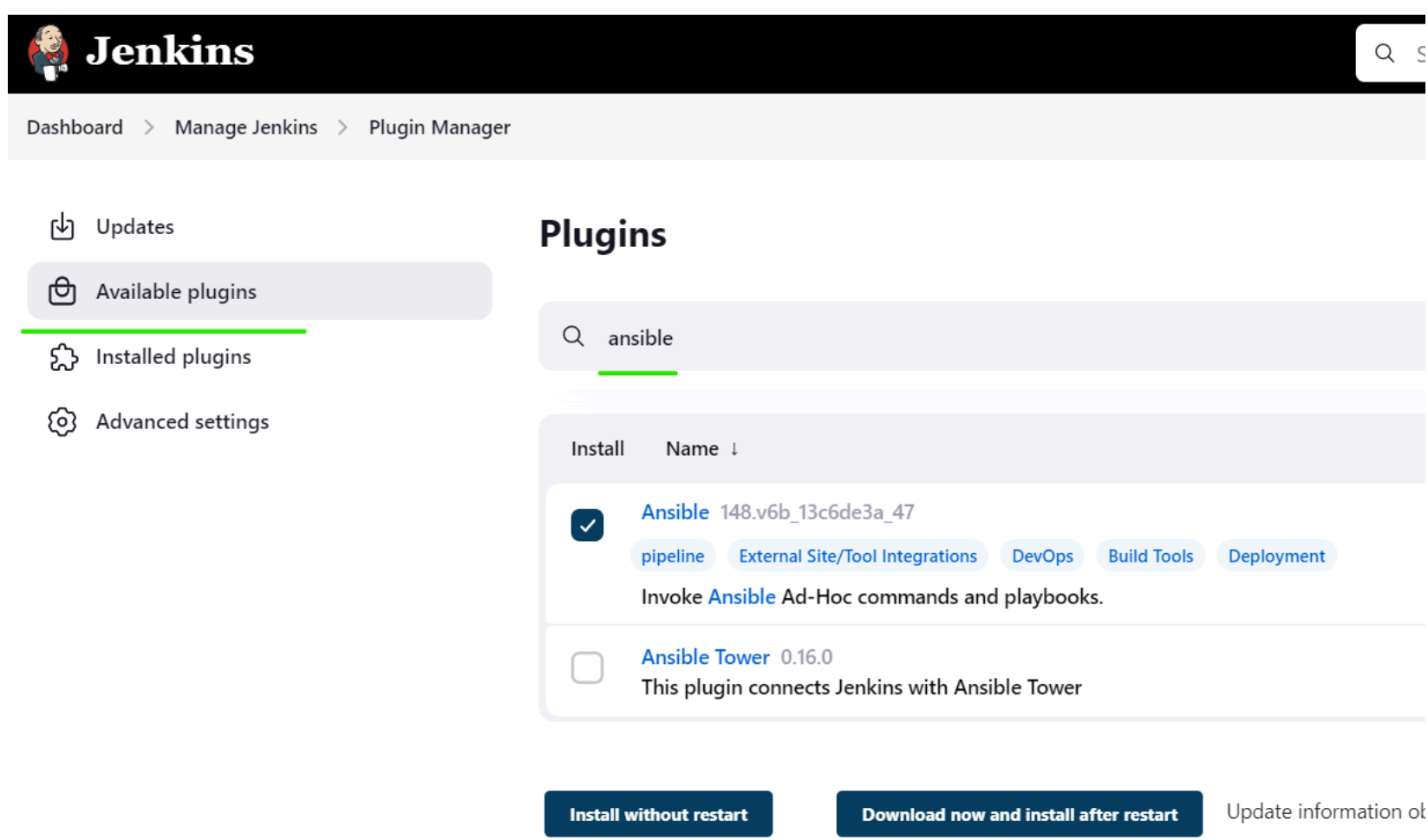
STEP 5 On the left side of screen click the **Manage Jenkins**.



In the opening window go to the next section which is **Manage Plugins**.



After going to the Manage Plugins section, on the left side of the screen click on **available plugins** and just type **ansible** then install it.



When installation was completed, go to the **global tool configuration** from **manage jenkins** section, then scroll down and add ansible. In name part you can give any name as you want , but for the **path to ansible executables directory** go to your machine where ansible was installed type **which ansible** command and copy it paste path to that section.


```
[jenkins@JENKINS ~]$ which ansible
~/.local/bin/ansible
[jenkins@JENKINS ~]$
```

Ansible

Ansible installations

List of Ansible installations on this system

Add Ansible

Ansible

Name

Ansible

Path to ansible executables directory

/home/jenkins/.local/bin/

☐ Install automatically ?

Add Ansible

Save

Apply

NOTE: if you installed ansible with different user, your path to ansible command would not be the same. it will be something like `/home/username/.local/bin/ansible`.


Creating Pipeline Scripts for Ansible and Git.


Pipeline script for git


STEP 6 In the main menu , click **New Item** and in the opened window type the name as you want to call it then click **Pipeline** and click **OK** .


Enter an item name

» 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 agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

**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.

Scroll down and in the pipeline section click **pipeline syntax** where it will helps us to generate pipeline for github repository and ansible.

Pipeline

Definition

Script 

1

☒ **Use Groovy Sandbox** **Pipeline Syntax****Save****Apply**

STEP 7 As shown in the below picture:

- Choose **git:Git** from **Sample Step**

- Paste url of ansible playbook if you have already one. If you do not have any ansible playbook you can use my [Prometheus-Node-Exporter_Ansible-Playbook](#) repository for sample.
- Branch name of your repository
- Click to [Generate Pipeline Script](#) (copy it and save it we will use it after few steps in main pipeline)

Sample Step

git: Git

git ?

Repository URL ?

https://github.com/tahirguluzade/Prometheus-Node-Exporter_Ansible-Playbook.git

Branch ?

master

Credentials ?

- none -

+ Add

☒ Include in polling? ?

☒ Include in changelog? ?

Generate Pipeline Script


Generate Pipeline Script

git 'https://github.com/tahirguluzade/Prometheus-Node-Exporter_Ansible-Playbook.git'

SPECIAL NOTE: If you see following error below while pasting your github repository url, it means **GIT** is not installed in your machine. Therefore, go to your Linux machine and install git with `sudo yum isntall git -y` command.

Repository URL ?

https://github.com/tahirguluzade/Prometheus-Node-Exporter_Ansible-Playbook.git

 Failed to connect to repository : Error performing git command: git ls-remote -h https://github.com/tahirguluzade/Prometheus-Node-Exporter_Ansible-Playbook.git HEAD

Pipeline script for Ansible

STEP 8 Now, we need to generate pipeline for Ansible as well.

- Choose `asnablePlaybook: invoke an Ansible Playbook` from [Sample Step](#).
- For `Playbook file path in workspace` , write down name of your `.yaml` file in the repository.

- Type your inventory file path in repository to **Inventory file path in workspace**.

Sample Step

ansiblePlaybook: Invoke an ansible playbook

ansiblePlaybook ?

Ansible tool

Ansible

Playbook file path in workspace

Prometheus+Node Exporter_Playbook.yml

Inventory file path in workspace

inventory

SSH connection credentials

Jenkins

+ Add

- For **SSH connection credentials**: click to + Add and **Jenkins** then enter details for SSH connection.

Add Credentials

Domain

Global credentials (unrestricted)

Kind

SSH Username with private key

Scope ?

Global (Jenkins, nodes, items, all child items, etc)

ID ?

private-key

Description ?

- For the private key go to your Linux machine and copy private key and directly paste it to **Add** section

☐ Treat username as secret ?

Private Key

Enter directly

Key

No Stored Value

Add

Passphrase

Add

Cancel

After adding your private key then for **SSH connection credentials**, choose your private key which you added.

- For **Become username** and **Sudo username** use an username which we created while installing Jenkins and Ansible.
- Click to **Generate Pipeline Script** and copy script (we will need at at main pipeline).

Generate Pipeline Script

ansiblePlaybook becomeUser: 'jenkins', credentialsId: 'provake-key', installation: 'Ansible', inventory: 'inventory', playbook: 'Prometheus+Node Exporter_Playbook.yml', sudoUser: 'jenkins'

Create Pipeline for building

STEP 9 Now, go back to the pipeline where we created new pipeline item in **Step 6** , and type below script to there also ypu need to paste your git and Ansible pipeline scripts thath we generated before in **Step 7** and **Step 8** to there. Then click save.

```
1 pipeline{
2   agnet any
3   stages{
4     stage{'SCM Checkout'}{
5       steps{
6         <your github pipeline script>
7       }
8     }
9     stage('Execute Ansible'){
10      steps{
11        <your ansible pipeline script>
12      }
13    }
14  }
15
16 }
```

Pipeline

Definition

Pipeline script

Script ?

```

2   agent any
3   stages{
4       stage('SCM Checkout'){
5           steps{
6               git branch: 'main', url: 'https://github.com/tahinguluzade/Prometheus-Node-Exporter_Ansible-Playbook.git'
7           }
8       }
9       stage('Execute Ansible'){
10          steps{
11              ansiblePlaybook becomeUser: 'jenkins', credentialsId: 'provake-key', installation: 'Ansible', inventory: 'inventory', pl
12          }
13      }
14  }
15  }
16  }
17  }

```


☒ Use Groovy Sandbox ?

[Pipeline Syntax](#)

Save

Apply

STEP 10 Now, go to the your pipeline and click **Build Now**, then it will start to build our pipeline. You check status of your pipeline building process as well as looking at logs on the right side. If everything is succesfully completed, then you can check your remote target server for your ansible playbook.


Jenkins

Search (C

Dashboard > ansible-preometheus+node exporter >

Status

Changes

Build Now

Configure

Delete Pipeline

Full Stage View

Rename

Pipeline Syntax

Build History

trend

Pipeline ansible-preometheus+node exporter

Stage View

	SCM Checkout	Execute Ansible
Average stage times: (Average full run time: ~33s)	994ms	34s
#14 Feb 22 19:05 No Changes	1s	34s

Congrats, You have built your first Jenkins project. KEEP GOING!!!

Rating:

24 Feb 2023

[DevOps](#)

[#green](#)

[« Managing Software in Linux](#)

[How to interface photoresistor to Arduino? »](#)

Share



We were unable to load Disqus. If you are a moderator please see our [troubleshooting guide](#).

Explore

[linux \(3\)](#) [devops \(8\)](#) [arduino \(1\)](#) [python \(1\)](#)