Project-Tetris

Github + Jenkins + Ansible + LAMP CI/CD Pipeline Documentation

Tools Setup

Github SSH Authentication

- Generate ssh key using the following command on host machine:

  `ssh-keygen -t rsa -C "github-access" -f id_rsa`

- Add the containing public key in `id_rsa.pub` file to your github account.

Install and setup 3 VMs with CentOS 7 image One will act as the master/controller node, one is for the Web Server (frontend & backend) and another is for Database. First step after installation is that we need to disable the selinux following the below instructions:

1. Open the selinux config file in the text editor of your choosing. I will use vi for this instance:

   `vi /etc/selinux/config`

2. Edit the line `SELINUX=enforcing` and set it to `SELINUX=disabled`.

3. Reboot the system

Jenkins Installation

1. Need to update packages of all VM by the following command:

   `yum update -y`

2. Jenkins is a java program so firstly we need to install java on the master VM.

   `yum install java-11-openjdk.x86_64 -y`

3. To install the latest stable version of Jenkins , we have to add the official Jenkins repository to the system. Execute the commands below to add the key and repo.

> 🔴IMPORTANT❗🔴 Note: If you want to hit your jenkins URL first you need to stop firewalld on centos or need to enable the port on firewalld. Otherwise you can not access jenkins dashboard on browser.

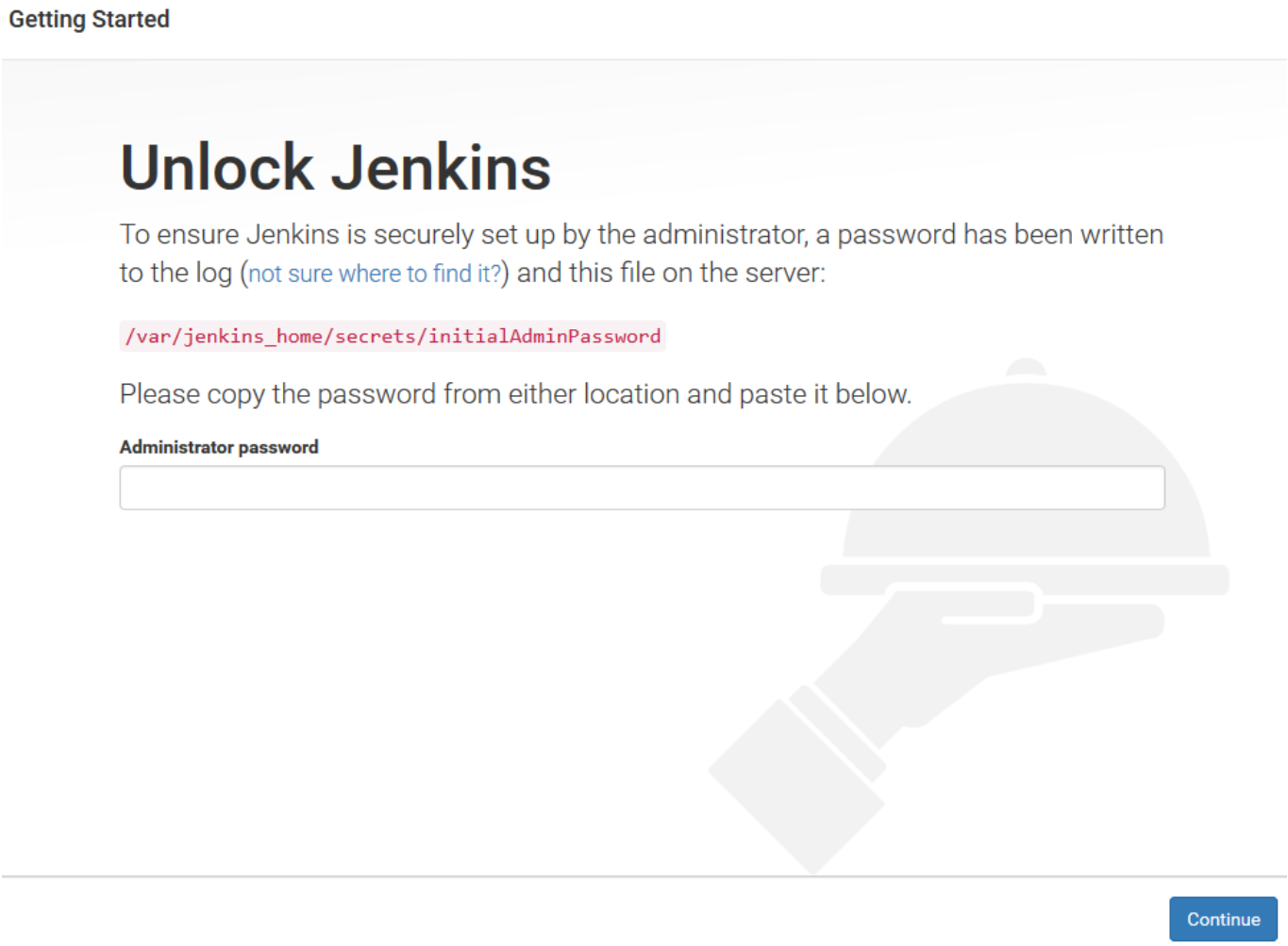
```
yum install wget -y
sudo wget -O /etc/yum.repos.d/jenkins.repo https://pkg.jenkins.io/redhat-
stable/jenkins.repo
sudo rpm --import https://pkg.jenkins.io/redhat/jenkins.io-2023.key
```

Jenkins Setup

After installation go to the browser of your host machine and input the following url in the address bar:

`{your_master_vm_ip}:8080`

You should be greeted with this:

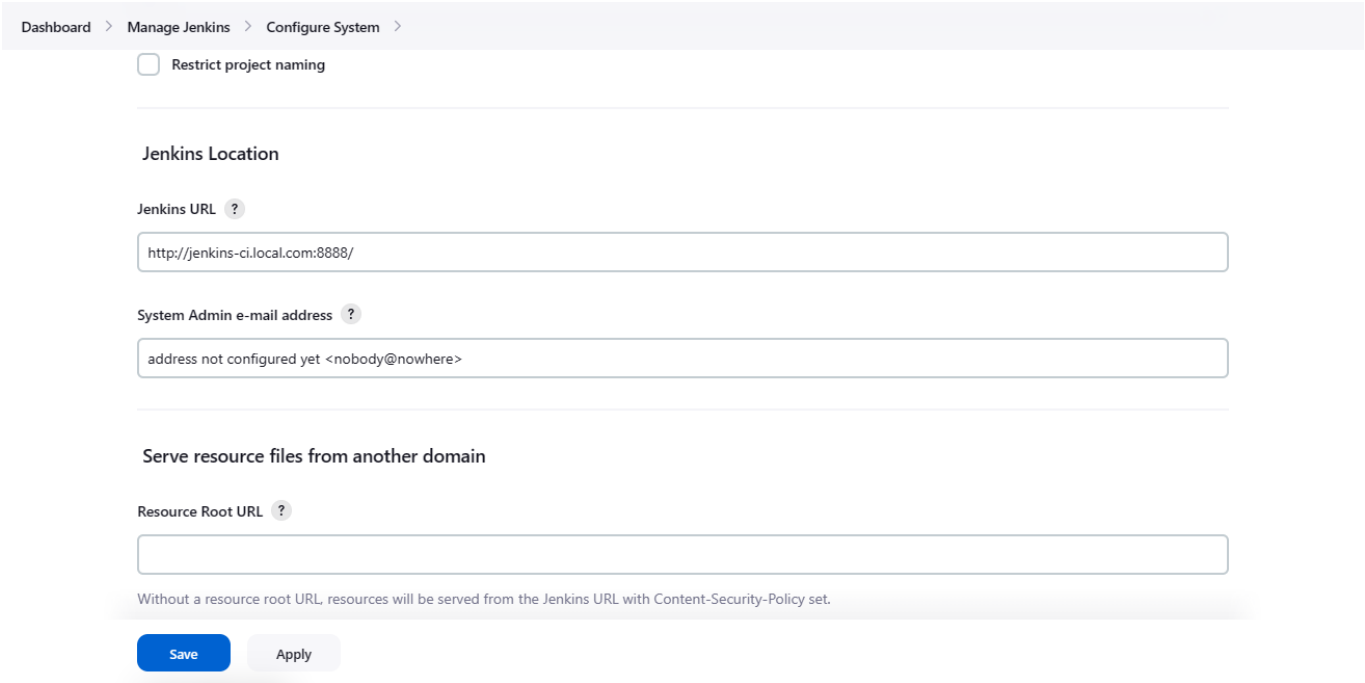


Go to the mentioned location and use the password to unlock jenkins. pressing continue should take you to a new page that lets you create a new user. Create a user and login into Jenkins.

Jenkins Custom Port & Domain

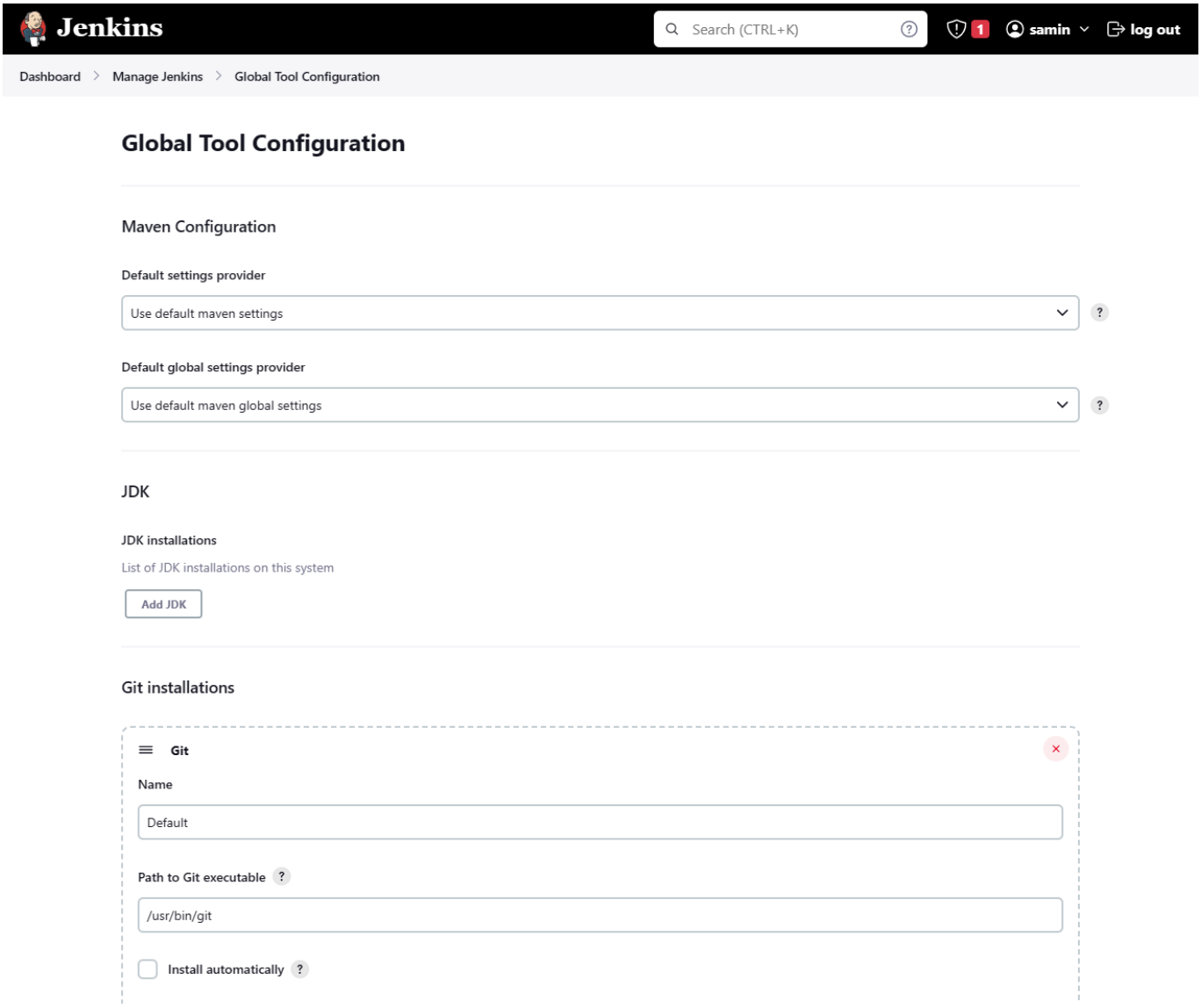
Go to **`Manage Jenkins --> Configure System`** and set a custom domain and port for making it easier to access jenkins.

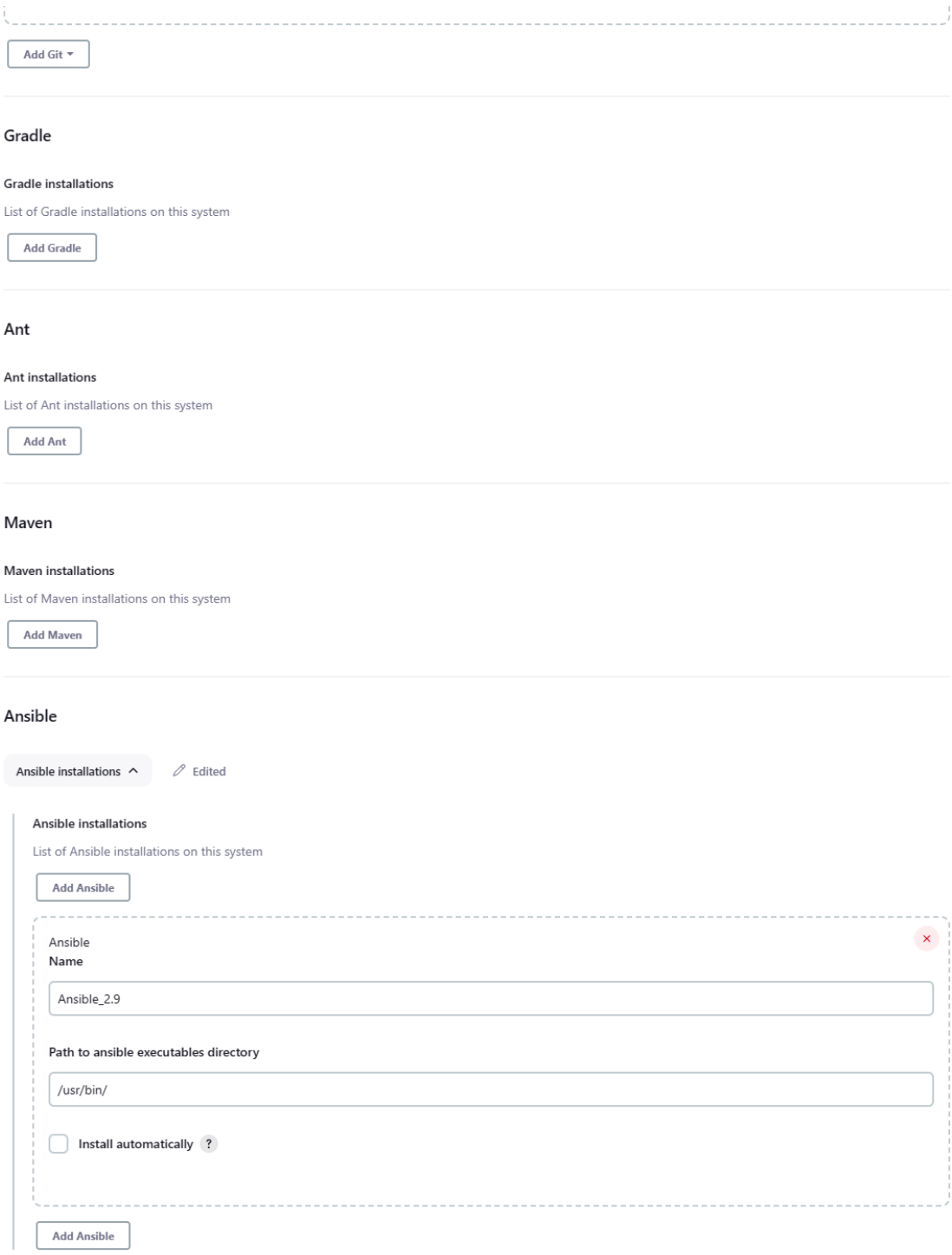
> 🔴IMPORTANT❗🔴 If changing port settings of Jenkins configuration file doesn't change the port even after restarting, you need to modify jenkins service file inside /usr/lib/systemd/system/jenkins.service.



Jenkins Global Tool Configuration

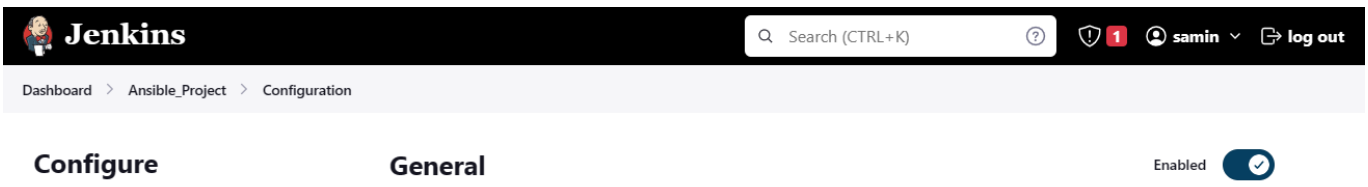
Go to **`Manage Jenkins > Global Tool Configuration`** and modify the settings according the the image below:





Pipeline Setup

Create a new pipeline job from the jenkins dashboard. Go to **`Configuration`** menu of the project. Now set up the pipeline and additional configuration



General
Advanced Project Options
Pipeline

Description

[Plain text] Preview

Discard old builds ?
Do not allow concurrent builds
Do not allow the pipeline to resume if the controller restarts
GitHub project
Project url ?
https://github.com/samin-irtiza-bjit/TetrisPHP/
Advanced
Pipeline speed/durability override ?
Preserve stashes from completed builds ?
This project is parameterised ?
Throttle builds ?

Build Triggers

Build after other projects are built ?
Build periodically ?
GitHub Branches
GitHub Pull Requests ?
GitHub hook trigger for GITScm polling ?
Poll SCM ?
Schedule ?
H/2 * * * *
Would last have run at Thursday, May 25, 2023 at 8:20:43 AM Eastern Daylight Time; would next run at Thursday, May 25, 2023 at 8:20:43 AM Eastern Daylight Time.
Ignore post-commit hooks ?
Quiet period ?
Trigger builds remotely (e.g., from scripts) ?

Advanced Project Options

Advanced

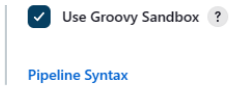
Pipeline

Definition

Pipeline script

Script ?

```
pipeline {
    agent any
    triggers{
        pollSCM('H/2 * * * *')
    }
    stages {
        stage('Clone Git') {
            steps {
                git branch: 'main', credentialsId: 'ef2a60b7-c8b3-45c2-b01c-58650f2d6542', url: 'https://github.co
            }
        }

        stage ('Run Playbook'){
            steps{
                ansiblePlaybook become: true, colorized: true, credentialsId: 'ef2a60b7-c8b3-45c2-b01c-58650f2d654
            }
        }
```


Use Groovy Sandbox
Pipeline Syntax

Pipeline Script

```
pipeline {
    agent any
    triggers{
        pollSCM('H/2 * * * *')
    }
    stages {
        stage('Clone Git') {
            steps {
                git branch: 'main', credentialsId: {your_credential_id}, url:
'https://github.com/samin-irtiza-bjit/TetrisPHP'
            }
        }

        stage ('Run Playbook'){
            steps{
                ansiblePlaybook become: true, colorized: true, credentialsId:
{your_credential_id}, installation: 'Ansible_2.9', inventory:
'/etc/ansible/hosts', playbook: '/opt/Ansible_Project/project.yml'
            }
        }
    }
}
```

Ansible Installation

Now we will install Ansible on the master VM.

1. Firstly we need to install the epel-release package so that we can install ansible from that package. For this we need to run the below command:

   `sudo yum install epel-release`

2. Now we will install ansible by below command:

   `yum install ansible`

3. Login as the root user and create a user named ansible on both client VMs and run the following command to set visudo config to be able to access without a password prompt.

   `echo 'ansible ALL=(ALL) NOPASSWD:ALL' > /etc/sudoers.d/ansible`

4. Add the following configurations to the master/controller VM's ansible configuration file placed at **/etc/ansible/ansible.cfg**

```
[defaults]
```

```
# Remote user name
remote_user = ansible

# SSH private key file
private_key_file = /root/.ssh/id_rsa

# Set the default module timeout to 60 seconds
timeout = 60

# Set the default number of forks to 10
forks = 10

# Enable fact gathering
gather_facts = smart

# Set the default ansible_python_interpreter to /usr/bin/python3
ansible_python_interpreter = /usr/bin/python3

# Enable pipelining
pipelining = True

# Limit the maximum number of parallel connections per host to 5
max_parallel_connections = 5

# Limit the maximum number of seconds to wait for a connection to a managed node
timeout_connection = 10
```

5. We will add ansible clients which is our Web Server VM, Database Server VM ip address in ansible configuration file.

   `sudo vim /etc/ansible/hosts`

```
[db]
192.168.56.110
[web]
192.168.56.111
```

6. We need to copy the ssh key of the master server in ansible client machines. So that the Ansible server to its nodes communication will be passwordless.

```
ssh-copy-id root@192.168.56.111 [Web Server IP]
ssh-copy-id root@192.168.56.110 [Database Server IP]
```

Ansible Playbook Codes

Ansible Roles Playbook Folder Structure

```
.
├── project.yml
└── roles
    ├── db
    │   ├── files
    │   │   ├── db-load-script.sql
    │   │   └── my.cnf
    │   ├── handlers
    │   │   └── main.yml
    │   ├── tasks
    │   │   └── main.yml
    │   ├── templates
    │   └── vars
    │       └── main.yml
    └── web
        ├── handlers
        │   └── main.yml
        ├── tasks
        │   └── main.yml
        ├── templates
        └── vars
            └── main.yml
```

project.yml

```yaml
---
- name: DB Service
  hosts: db
  become: yes
  roles:
    - db
- name: Web Service
  hosts: web
  become: yes
  roles:
    - web
```

roles/db/tasks/main.yml

```yaml
- name: Installation Services
  yum:
```

```
    name:
    - libselinux-python
    - libsemanage-python
    - mariadb-server
    - MySQL-python
    - php-mysql
    state: installed
  tags: install
- name: Start firewalld
  service: name=firewalld state=started enabled=yes
  tags: start firewalld
- name: Insert firewalld rule for mysql
  firewalld: port={{ mysql_port }}/tcp permanent=true state=enabled immediate=yes
  tags: enable mysql port
- name: Restart firewalld
  service: name=firewalld state=reloaded enabled=yes
  tags: restarted firewalld
- name: Copy Mysql configuration file
  copy: src=files/my.cnf dest=/etc/my.cnf
  tags: mysql conf copy
- name: Start MariaDB Service
  service: name=mariadb state=started enabled=yes
  tags: start mariadb
- name: Create Application Database
  mysql_db: name={{ dbname }} state=present
  tags: create database
- name: Create Application DB User
  mysql_user: name={{ dbuser }} password={{ dbpassword }} priv=*.*:ALL
host='192.168.56.110' state=present
  tags: create user
- name: Move db-load-script to db host
  copy:
    src: files/db-load-script.sql
    dest: /tmp/db-load-script.sql
  tags: copy sql
- name: Load Inventory Data
  shell: mysql -f < /tmp/db-load-script.sql
  tags: run sql
```

roles/db/vars/main.yml

```
---
mysql_port: 3306
dbname: tetris
dbuser: tetris
dbpassword: tetrispass
```

roles/db/files/db_load_scripts.sql

```
GRANT ALL PRIVILEGES ON *.* TO 'tetris'@'192.168.56.111' IDENTIFIED BY
'tetrispass' WITH GRANT OPTION;
FLUSH PRIVILEGES;

USE tetris;

CREATE TABLE Users (
  Username VARCHAR(50) PRIMARY KEY,
  FirstName VARCHAR(50) NOT NULL,
  LastName VARCHAR(50) NOT NULL,
  Password VARCHAR(50) NOT NULL,
  Display VARCHAR(50) NOT NULL
);
CREATE TABLE Scores (
  Username VARCHAR(50) NOT NULL,
  Score INT NOT NULL,
  PRIMARY KEY (Username, Score)
);
```

roles/db/files/my.cnf

```
[mysqld]
bind-address=0.0.0.0
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock
symbolic-links=0
[mysqld_safe]
log-error=/var/log/mariadb/mariadb.log
pid-file=/var/run/mariadb/mariadb.pid
!includedir /etc/my.cnf.d
```

roles/web/tasks/main.yml

```
- name: Installation Services
  yum:
    name:
    - libselinux-python
    - libsemanage-python
    - httpd
    - git
    - php
    - php-mysql
    state: installed
  tags: install

- name: Start firewalld
  service: name=firewalld state=started enabled=yes
  tags: start firewalld
```

```
- name: Insert firewalld rule for httpd
  firewalld: port={{ httpd_port }}/tcp permanent=true state=enabled immediate=yes
  tags: enable httpd port

- name: insert firewalld rule for mysql
  firewalld: port={{ mysql_port }}/tcp permanent=true state=enabled immediate=yes
  tags: enable mysql port

- name: Set index.php as the default page
  replace:
    path: /etc/httpd/conf/httpd.conf
    regexp: 'DirectoryIndex index.html'
    replace: '#DirectoryIndex index.html \nDirectoryIndex index.php'
  tags: rename html file

- name: http service state
  service: name=httpd state=started enabled=yes
  tags: httpd start

- name: copy repo code
  git:
    repo: '{{ repository }}'
    dest: '/var/www/html/'
    force: yes
```

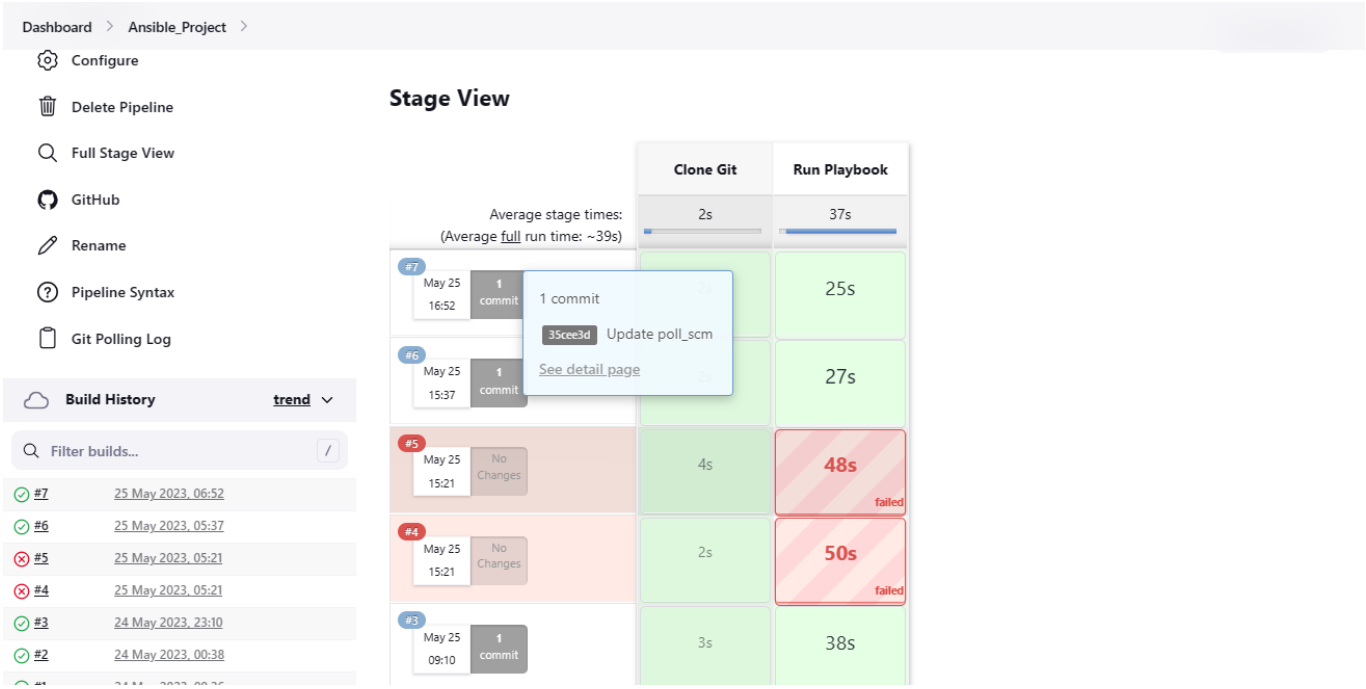
roles/web/vars/main.yml

```
---
httpd_port: 80
mysql_port: 3306
repository: https://github.com/samin-irtiza-bjit/TetrisPHP.git
```

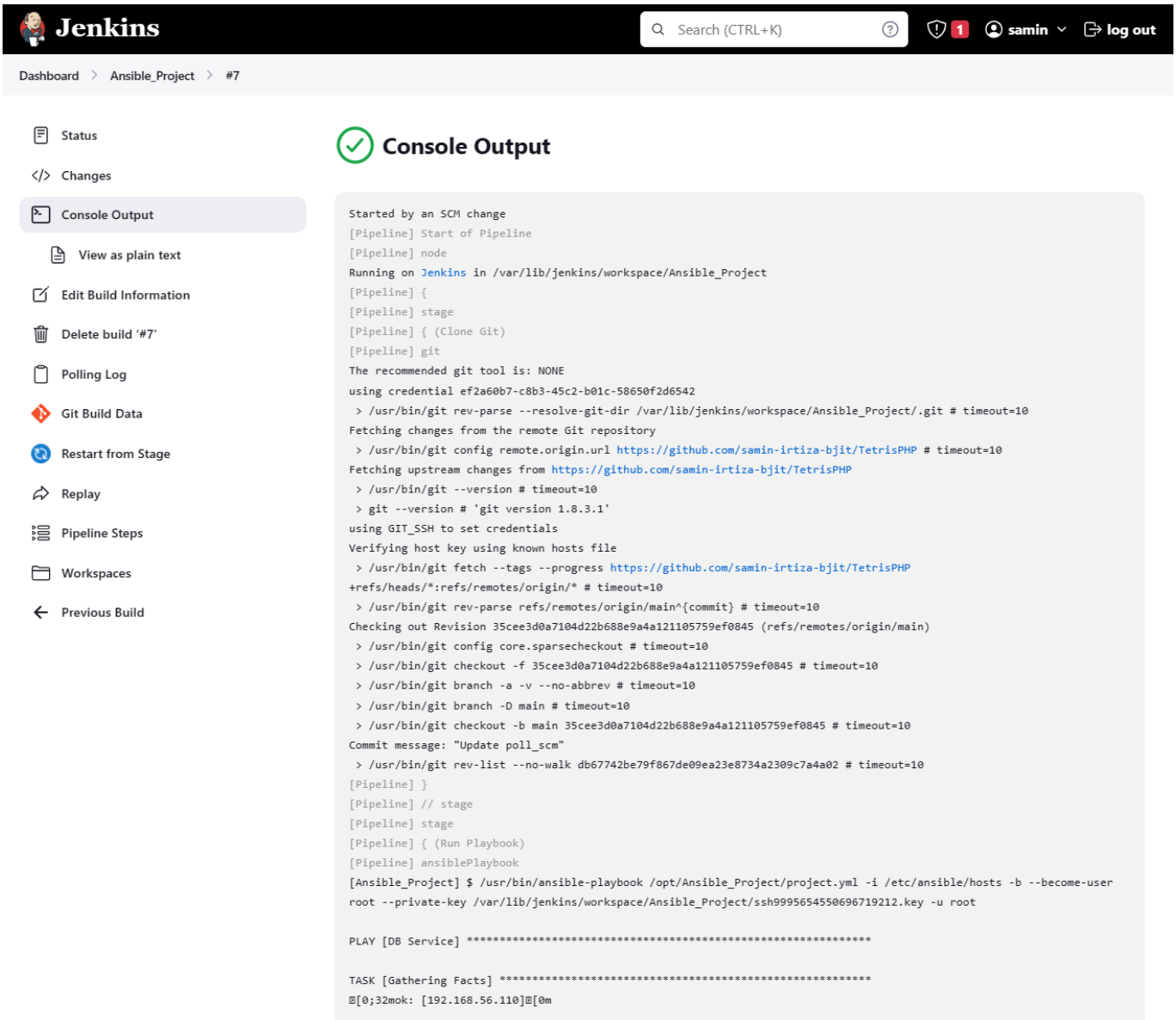
Working Pipeline

If everything is set up correctly, jenkins should start an auto build and deploy the website upon pushing a commit into the GitHub repository:

Auto Polling from GitHub



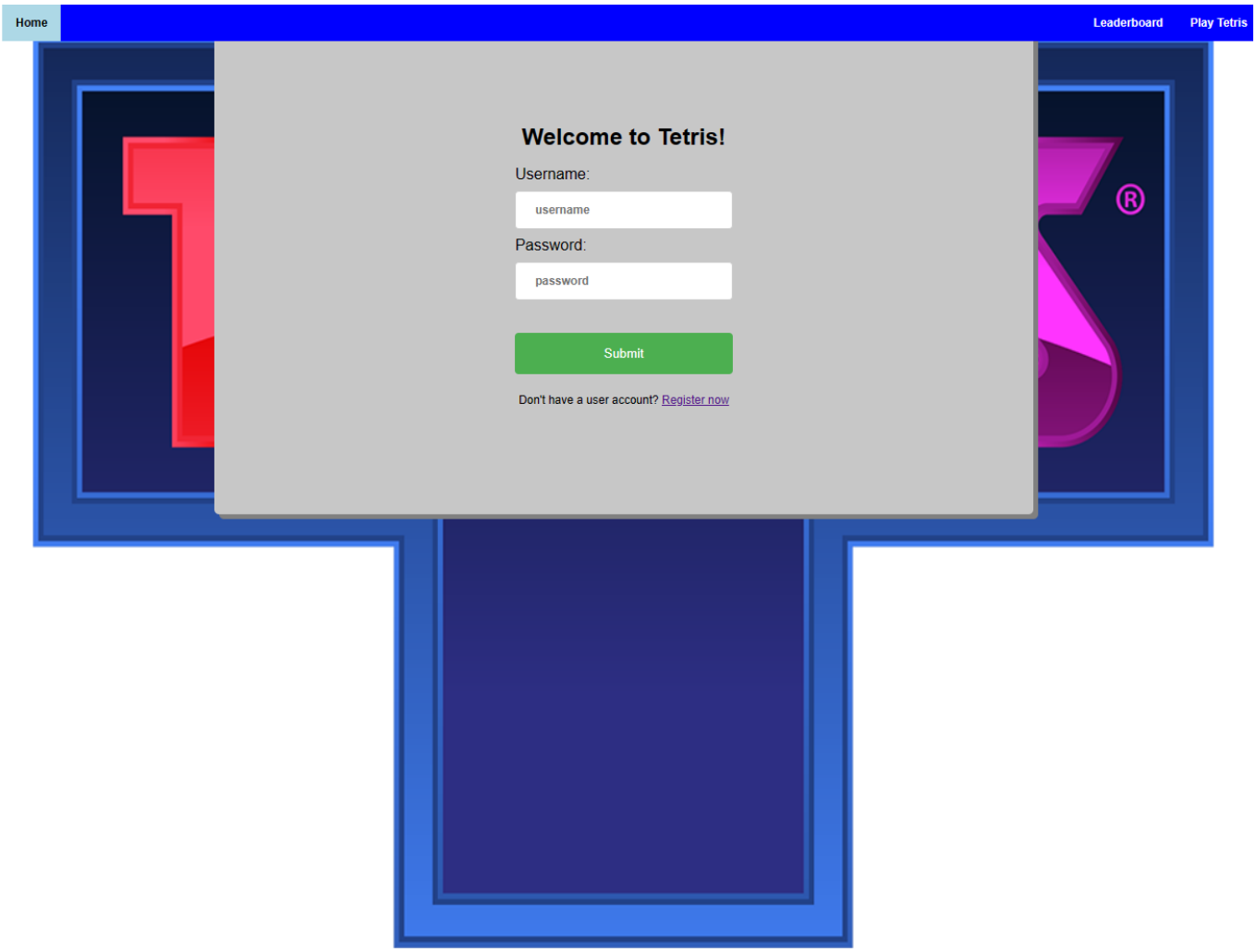
Playbook output from the pipeline

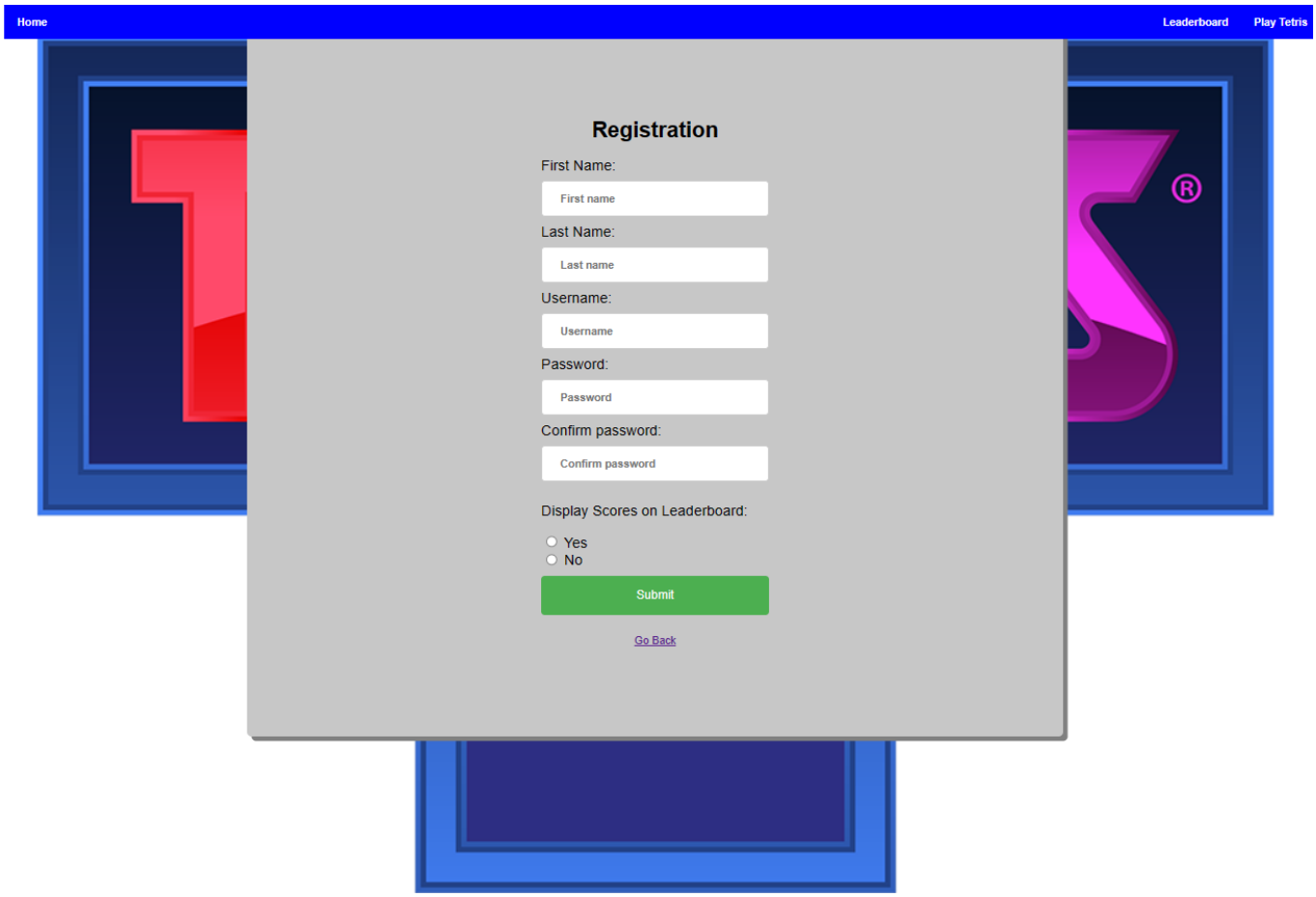


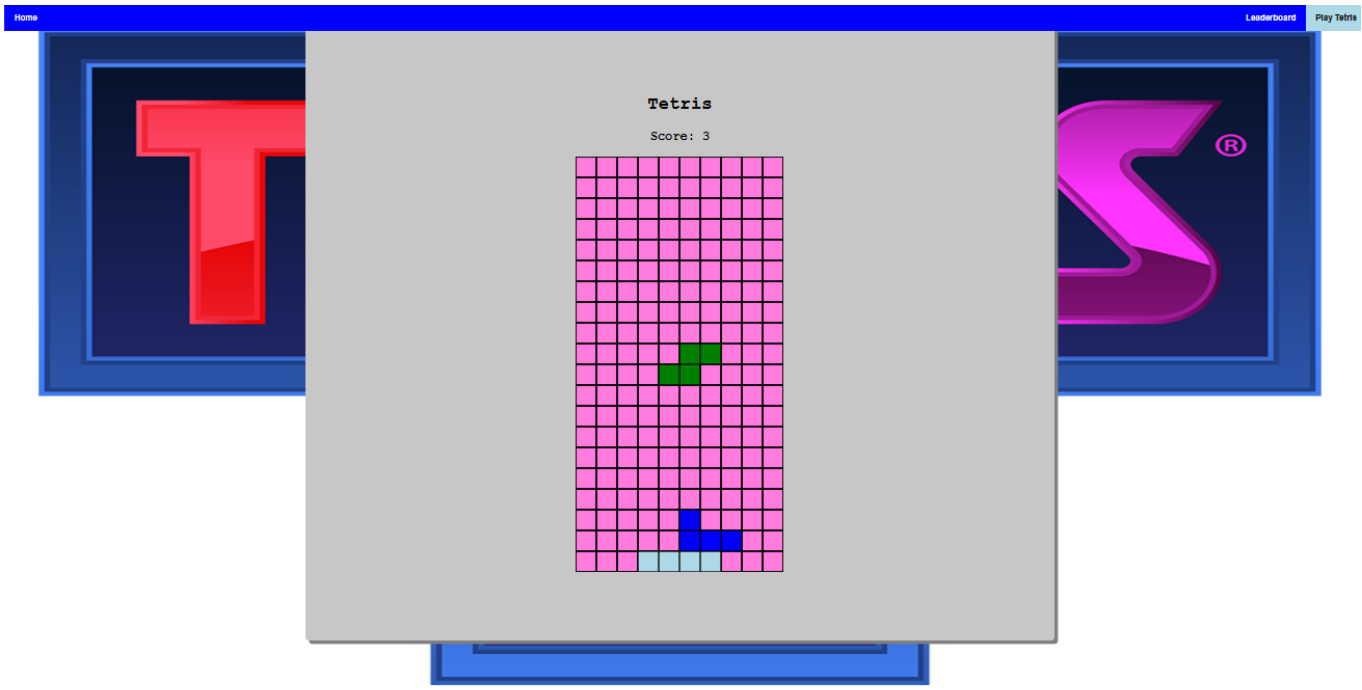
```
TASK [db : Installation Services] **********************************************
[0;32mok: [192.168.56.110][0m

TASK [db : Start firewalld] ****************************************************
[0;32mok: [192.168.56.110][0m

TASK [db : Insert firewalld rule for mysql] ************************************
[0;32mok: [192.168.56.110][0m

TASK [db : Restart firewalld] **************************************************
[0;33mchanged: [192.168.56.110][0m

TASK [db : Copy Mysql configuration file] **************************************
[0;32mok: [192.168.56.110][0m

TASK [db : Start MariaDB Service] **********************************************
[0;32mok: [192.168.56.110][0m

TASK [db : Create Application Database] ****************************************
[0;32mok: [192.168.56.110][0m

TASK [db : Create Application DB User] *****************************************
[1;35m[WARNING]: Module did not set no_log for update_password[0m
[0;32mok: [192.168.56.110][0m

TASK [Move db-load-script to db host] ******************************************
[0;32mok: [192.168.56.110][0m

TASK [db : Load Inventory Data] ************************************************
[0;33mchanged: [192.168.56.110][0m

PLAY [Web Service] *************************************************************

TASK [Gathering Facts] *********************************************************
[0;32mok: [192.168.56.111][0m

TASK [web : Installation Services] *********************************************
[0;32mok: [192.168.56.111][0m

TASK [web : Start firewalld] ***************************************************
[0;32mok: [192.168.56.111][0m

TASK [web : Insert firewalld rule for httpd] ***********************************
[0;32mok: [192.168.56.111][0m

TASK [web : insert firewalld rule for mysql] ***********************************
[0;32mok: [192.168.56.111][0m

TASK [web : Set index.php as the default page] *********************************
[0;33mchanged: [192.168.56.111][0m

TASK [web : http service state] ************************************************
[0;32mok: [192.168.56.111][0m

TASK [web : copy repo code] ****************************************************
[0;33mchanged: [192.168.56.111][0m

PLAY RECAP *********************************************************************
[0;33m192.168.56.110[0m             : [0;32mok=11  [0m [0;33mchanged=2   [0m unreachable=0    failed=0
skipped=0    rescued=0    ignored=0
[0;33m192.168.56.111[0m             : [0;32mok=8   [0m [0;33mchanged=2   [0m unreachable=0    failed=0
skipped=0    rescued=0    ignored=0

[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

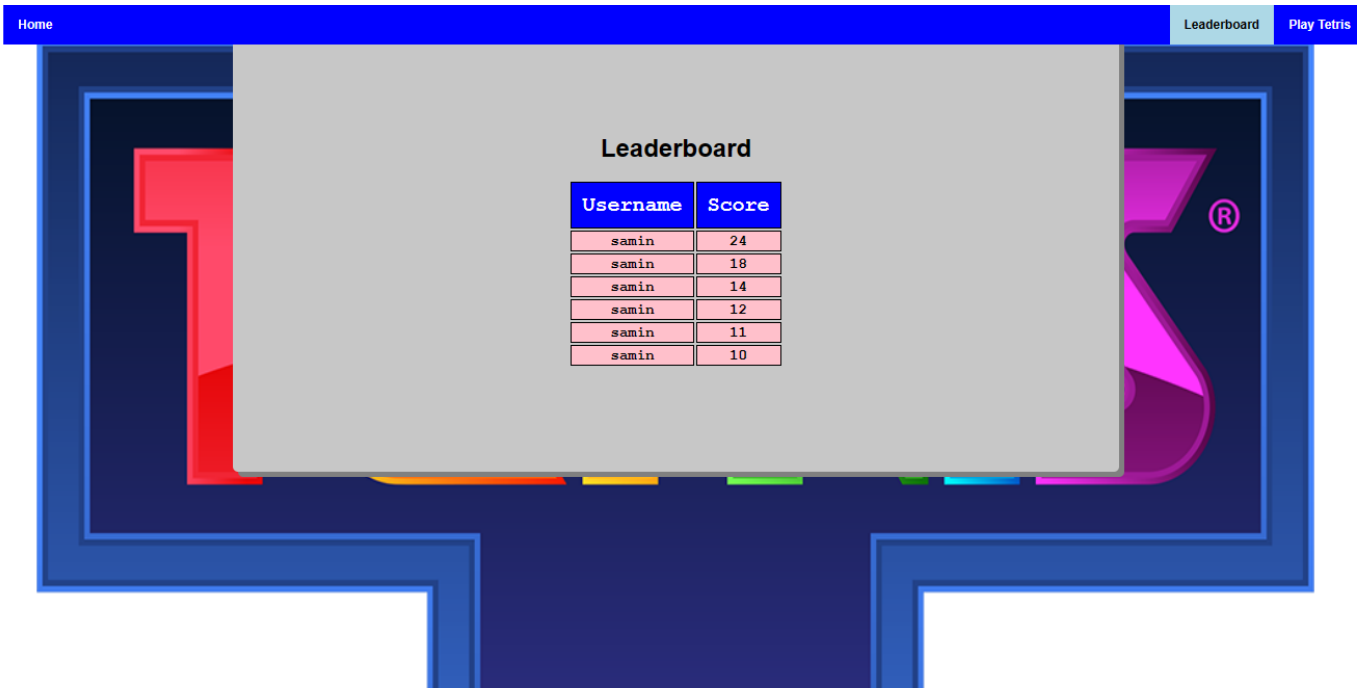
REST API Jenkins 2.387.3

Deployed Website


Home
Leaderboard
Play Tetris
Welcome to Tetris!
Username:
username
Password:
password
Submit
Don't have a user account? Register now


Home
Leaderboard
Play Tetris
Registration
First Name:
First name
Last Name:
Last name
Username:
Username
Password:
Password
Confirm password:
Confirm password
Display Scores on Leaderboard:
Yes
No
Submit
Go Back


Home
Leaderboard
Play Tetris
Tetris
Score: 3


Home
Leaderboard
Play Tetris
Leaderboard
Username
Score
samin 24
samin 18
samin 14
samin 12
samin 11
samin 10