

Tools Needed:

Access to OpenSSH Client

- 1) [WSL \(recommended for Windows users\)](#)
- 2) Command Line / Terminal (Mac)
- 3) Linux ... (If you're using this, I don't have to tell you.)

Accessing the Jenkins Server

- 1) Create an SSH Tunnel from your machine to the Jenkins Server
 - a. Servers
 - i. osc-mgmt.eng.auburn.edu
 1. username: comp2710
 2. password: comp2710
 - ii. 192.168.0.178 (only accessible from osc-mgmt – the cloud gateway)
 1. Username: jenkins
 2. Password: jenkins_p@ssw0rd (the 0 is the number 0 not capitol O)

Example:

```
ssh -L 8002:localhost:8080  
-J comp2710@osc-mgmt.eng.auburn.edu  
jenkins@192.168.0.178
```

Command Breakdown:

-L forward a local port to a remote machine

Blue (8002) is the local port

Orange (8080) is the port on the remote machine


Localhost is the remote machine

-J “jump” through multiple machines to get to the final destination. Format is comma separated list with final destination separated with space. Ex: **server1,server2,server3 final_server**

- 2) Open any web browser and go to the remote machine part of the address in the tunnel you made, followed by the local port that you are forwarding. For this example, this would be **localhost:8002**

localhost:8002/login?from=%2F

Auburn Email Auburn Canvas AU Access vocareum Zybooks Lemongrass Gmail Rodrigo Gmail Other favorites



Sign in to Jenkins

Username

Password

☐ Keep me signed in


Sign in

[or register](#)

3) If this is your first time accessing the server, register and create an account.

localhost:8002/login?from=%2F

Auburn Email Auburn Canvas AU Access vocareum Zybooks Lemongrass Gmail Rodrigo Gmail Other favorites



Sign in to Jenkins

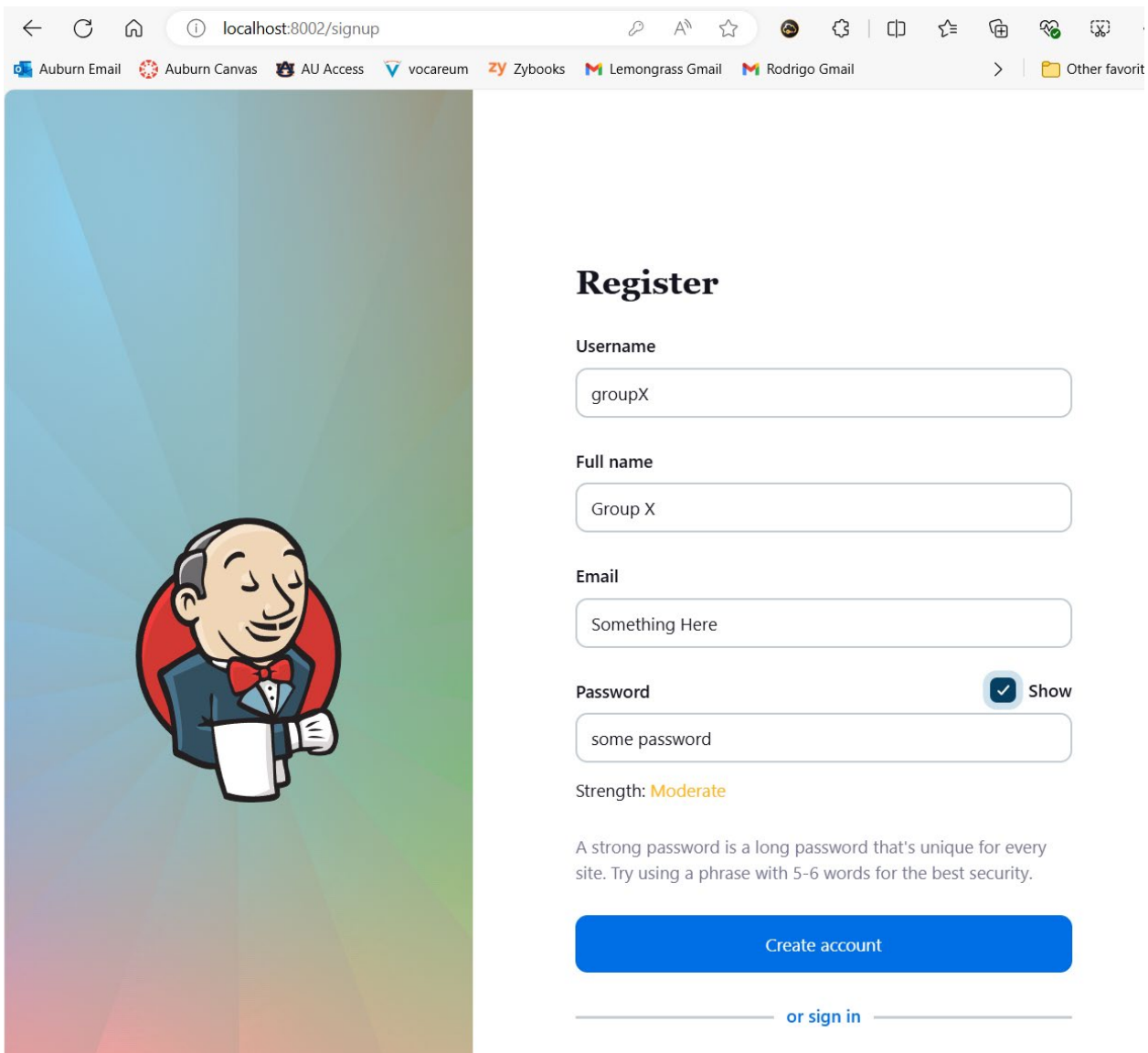
Username

Password

☐ Keep me signed in

Sign in

[or register](#)



← ↻ 🏠 ⓘ localhost:8002/signup 🔍 🌐 ⚙️ 📄 ⭐ 📁 📧 📧

Auburn Email Auburn Canvas AU Access vocareum zy Zybooks Lemongrass Gmail Rodrigo Gmail > Other favorit

Register

Username

Full name

Email

Password ☒ Show

Strength: **Moderate**

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

[or sign in](#)

- Be sure to use your auburn ID as your username.
Example: **ras0054**
- For full name, you can put your name

Using Jenkins

The Jenkins server is a shared server. Please use responsibly and DO NOT modify or delete or run other people's jobs. You will all be using the same Jenkins server, meaning that you will have access to everyone's jobs.

Please Limit the number of jobs created to 1 job per assignment per group. For example, for homework 1, each group will create a single job. That's 24 jobs. For the second assignment, each group will create another job. That would be a total of 48 jobs.

Naming Conventions:

In order to help the TA and I with grading, please name your jobs as follows:

GroupX-HWY Where the X is your group number and the Y is the assignment number. Use the same case as the example here. For example:

Group1-HW1 is good.

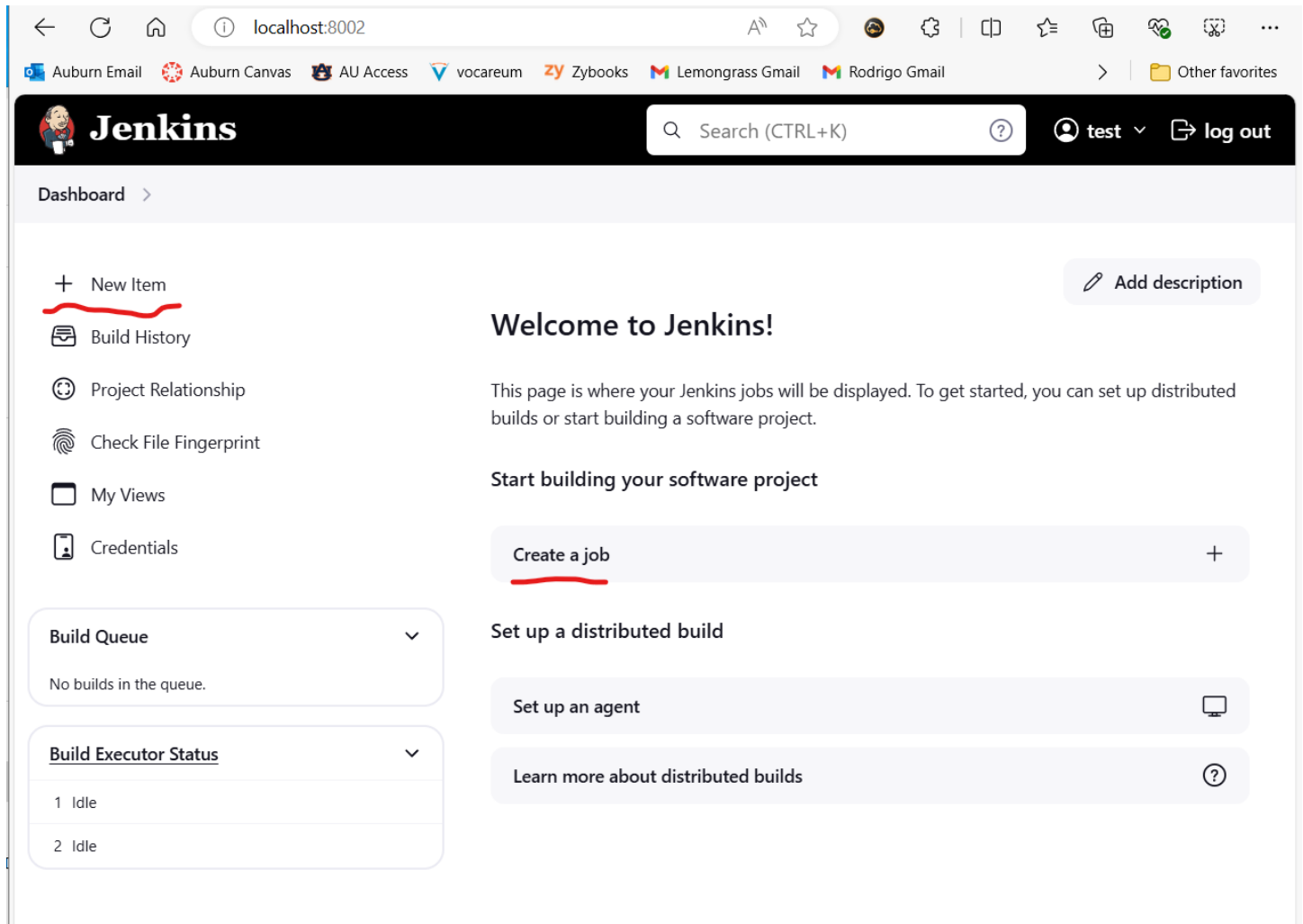
group1-hw1 is NOT good because the case is different.

This will make it easier for the TA to grade your assignments as he/she can easily create a filter for all assignments starting with your group.

Creating a Jenkins Job

Jenkins is a tool that has the ability to automate tasks for you. For example, we are going to be using it to fetch code from a github repo, pull it locally, compile it, and run it. It can do other things (like launch docker containers, create vms, etc...

One way that you specify what kind of tasks you want it to do for you is by creating a job. A job is one or more tasks that you want to automate.



- 1) If there are no jobs on the server, your dashboard will look something like this. Select Create a job. You can also select “New Item”.

localhost:8002/view/all/newJob

Auburn Email Auburn Canvas AU Access vocareum Zybooks Lemongrass Gmail Rodrigo Gmail Other f

Jenkins

Search (CTRL+K) test log







Dashboard > All > New Item

New Item

Enter an item name

group1-hw1

Select an item type

-  **Freestyle project**
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.
-  **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.
-  **Multibranch Pipeline**
Creates a set of Pipeline projects according to detected branches in one SCM repository.
-  **Organization Folder**
Creates a set of multibranch project subfolders by scanning for repositories.

OK

- 2) For now, select “Freestyle project”. Note that the “item name” is groupX-hwX. The item name is also the job name. Be sure to check “Discard old builds” to save disk space on the server.

Configure

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

General

Enabled

Description

Plain text [Preview](#)

☒ Discard old builds ?

Strategy

Log Rotation

Days to keep builds

if not empty, build records are only kept up to this number of days

Max # of builds to keep

if not empty, only up to this number of build records are kept

1

Advanced

- 3) Selecting ok will take you to a page where you can configure your job. For now, we want to configure several things.
- Our github repo (each assignment you'll be assigned a new github repo that you can use to upload your work for that specific assignment).

Dashboard > group1-hw1 > Configuration

Configure

General

Source Code Management

Build Triggers

Build Environment

Build Steps

Post-build Actions

None

Git ?

Repositories ?

Repository URL ?

Please enter Git repository.

Credentials ?

- none -

+ Add

Advanced

Add Repository

Branches to build ?

Branch Specifier (blank for 'any') ?

Add Branch

Repository browser ?

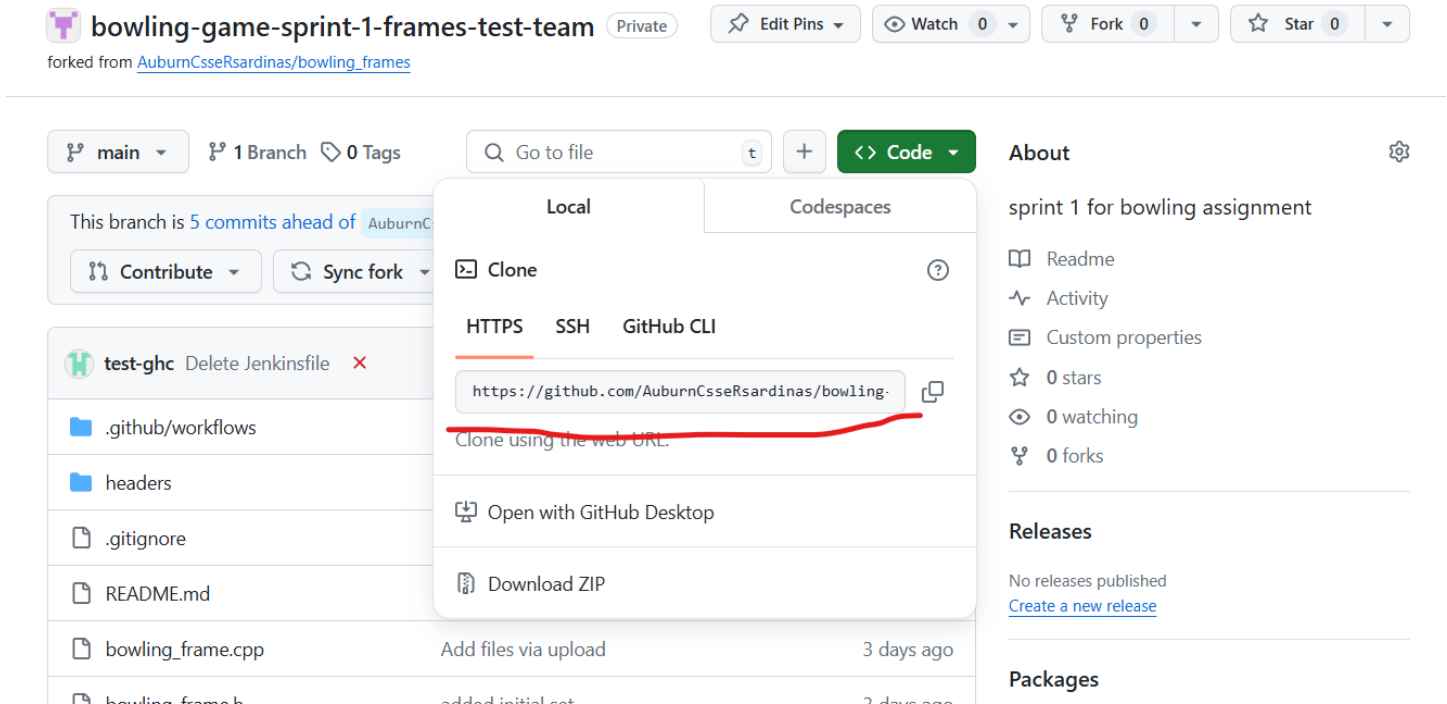
(Auto)

Additional Behaviours

Add

Save

Apply



- i. The repository url is the url you would use to clone the github repo.

Configure

General

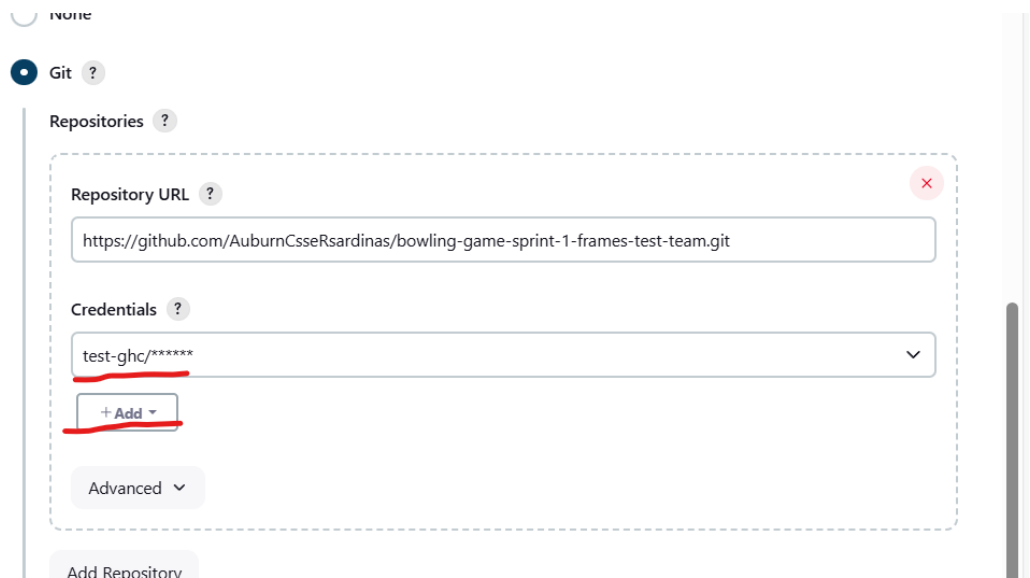
Source Code Management

Build Triggers

Build Environment

Build Steps

Post-build Actions



- b. Next you need to either select some credentials that will give you access to that repo (if you've already created them), or you need to create them. I have found that it is easiest to create an api key and use that to access all of your github repos. This will be covered in another section of this document.

Repositories ?

Repository URL ?

https://github.com/AuburnCseRsardinas/bowling-game-sprint-1-frames-test-team.git

Credentials ?

test-ghc/*****

+ Add ^

Jenkins

Advanced ^

Jenkins Credentials Provider: Jenkins

Add Credentials

Domain

Global credentials (unrestricted)

Kind

Username with password

Scope ?

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

Username ?

test_gch

☐ Treat username as secret ?

Password ?

Api Key

ID ?

Description ?

Cancel

Add

- Username should be your github username (only one of the group members needs to do this).
- Password should be the API key you created in github.

Configure

General

Source Code Management

Build Triggers

Build Environment

Build Steps

Post-build Actions

None

Git ?

Repositories ?

Repository URL ?

`https://github.com/AuburnCsseRsardinas/bowling-game-sprint-1-frames-test-team.git`

Failed to connect to repository : Command "git ls-remote -h -- https://github.com/AuburnCsseRsardinas/bowling-game-sprint-1-frames-test-team.git HEAD" returned status code 128:

stdout:

stderr: remote: Support for password authentication was removed on August 13, 2021. remote: Please see <https://docs.github.com/get-started/getting-started-with-git/about-remote-repositories#cloning-with-https-urls> for information on currently recommended modes of authentication. **fatal:** Authentication failed for 'https://github.com/AuburnCsseRsardinas/bowling-game-sprint-1-frames-test-team.git/'

Credentials ?

- current -

+ Add

Cannot find any credentials with id test1

Advanced

Configure

General

Source Code Management

Build Triggers

Build Environment

Build Steps

Post-build Actions

None

Git ?

Repositories ?

Repository URL ?

`https://github.com/AuburnCsseRsardinas/bowling-game-sprint-1-frames-test-team.git`

Credentials ?

test-ghc/*****

+ Add

Advanced

e. If you see this screen, (all the red text) then most likely your credentials aren't working correctly.

f. Note how the warning goes away once the correct credentials are selected.

Configure

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

Add Branch

Repository browser ?

(Auto)

Additional Behaviours

Add

Build Triggers

- ☐ Trigger builds remotely (e.g., from scripts) ?
- ☐ Build after other projects are built ?
- ☐ Build periodically ?
- ☐ GitHub hook trigger for GITScm polling ?

☒ Poll SCM ?

Schedule ?

H/60 * * * *

Would last have run at Thursday, August 22, 2024, 1:11:30 AM Coordinated Universal Time; would next run at Thursday, August 22, 2024, 2:11:30 AM Coordinated Universal Time.

4) Next we want to configure what triggers a “build”. A build is an activation of our job.

- Since our Jenkins server doesn’t have a public URL we can’t configure it to work with Github hook Triggers. Ideally this is what you would want for a CI/CD type job. With this type of trigger, everytime some event happens in your github repo (like someone pushing new code to a specific branch), github would notify your Jenkins server and this would trigger some job to run.
- Since we can’t do this, we will have to settle for checking periodically. The schedule uses CRON notation (automated linux task scheduler) to determine how often to run. For now, just configure it to run hourly (you can use exactly what I did above). That means that each hour, go to check the github repo, and if there were any changes, trigger the build.
- Once you enter a schedule, you will see text below detailing exactly how often it is going to run by way of an example.
- You also want to select the option to abort the job if the log file gets too big. This usually happens (in this class) with infinite loops.

Build Environment

- ☐ Delete workspace before build starts
- ☐ Use secret text(s) or file(s) ?
- ☒ Abort the build if its log file size is too big

Use job specific maximum log size instead global value (0 MB)

- ☒ Use job specific size

Job specific size ?

Job specific max log size in MB

- ☒ Fail the build ?

If checked, the build will be marked as failed rather than marked as aborted. (In any case, the build does not continue when the maximum file size is reached).

(from [Jenkins build log file size checker plugin](#))

- ☐ Add timestamps to the Console Output
- ☐ Inspect build log for published build scans
- ☐ Terminate a build if it's stuck
- ☐ With Ant ?

Build Steps

Add build step ^

Filter

- Execute Windows batch command
- Execute shell
- Invoke Ant
- Invoke Gradle script
- Invoke top-level Maven targets
- Run with timeout
- Set build status to "pending" on GitHub commit

- 5) Next, we want to configure what actions the job is going to take. For now, we will simply execute shell commands. For all of your homework assignments, you should create makefiles (covered later) that will compile your code for you. We will simply execute those files, then run the executable file that was created.

Build Steps

≡ **Execute shell** ?

Command

See [the list of available environment variables](#)

```
make
```

Advanced ▾

Add build step ^

Filter

Execute Windows batch command

Execute shell

Invoke Ant







Invoke Gradle script

Invoke top-level Maven targets

Run with timeout

Set build status to "pending" on GitHub commit

Configure

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

Build Steps

≡ **Execute shell** ?



Command

See [the list of available environment variables](#)

```
make
```

Advanced ▾

≡ **Execute shell** ?



Command

See [the list of available environment variables](#)

```
./a.out
```

Advanced ▾

Add build step ▾

Post-build Actions

Save

Apply

- 6) Finally, select the option to clean up the workspace when the build is complete. Again, this is to save disk space on the Jenkins server.

Post-build Actions

Delete workspace when build is done

Advanced ▾

Add post-build action ▾

Save

Apply

Record fingerprints of files to track usage

Git Publisher

E-mail Notification

Editable Email Notification

Set GitHub commit status (universal)

Set build status on GitHub commit [deprecated]

Delete workspace when build is done

Add post-build action ^

Save

Apply

- 7) When you're done, you can click save to save the job.

Dashboard > group1-hw1 >

Status

</> Changes

Workspace

Build Now

Configure

Delete Project

Git Polling Log

Rename

group1-hw1

Permalinks

Add description

Build History

trend

No builds

Atom feed for all

Atom feed for failures

Once you save, it should take you to this screen where you can see information about the job, among other things. Note that you can manually trigger the build (job) and modify it (configure) among other things.

Jenkins

Search (CTRL+K) test log out

Dashboard > group1-hw1 >

Status

</> Changes

Workspace

Build Now

Configure

Delete Project

Git Polling Log

Rename

group1-hw1

Permalinks

Add description

Build History

trend

Filter...


#1

Aug 22, 2024, 2:10 AM

Atom feed for all

Atom feed for failures

Once the build has run at least once, you will see a build history (each time the build was triggered). If no changes are made, it should not trigger again. If you make changes to your github repo, then whenever it polls your github repo, it should trigger a new build.

**Jenkins**

Search (CTRL+K) ? test log out

Dashboard > group1-hw1 > #1

Status

</> Changes

Console Output

Edit Build Information

Delete build '#1'

Timings

Git Build Data

✓ #1 (Aug 22, 2024, 2:10:07 AM)

Add description

Keep this build forever

Started by user test

Started 2 min 3 sec ago
Took 0.96 sec

This run spent:

- 46 ms waiting;
- 0.96 sec build duration;
- 1 sec total from scheduled to completion.


Revision: 0cb9ca75b4bcca9e115f25bb3e46b9712aab5fc6

Repository: <https://github.com/AuburnCsseRsardinas/bowling-game-sprint-1-frames-test-team.git>

- origin/main

</> No changes.

Selecting a build number will take you to a page that will show you information about that instance of the job running.

**Jenkins**

Search (CTRL+K) ? test log out

Dashboard > group1-hw1 > #1 > Console Output

Status

</> Changes

Console Output

Edit Build Information

Delete build '#1'

Timings

Git Build Data

✓ Console Output

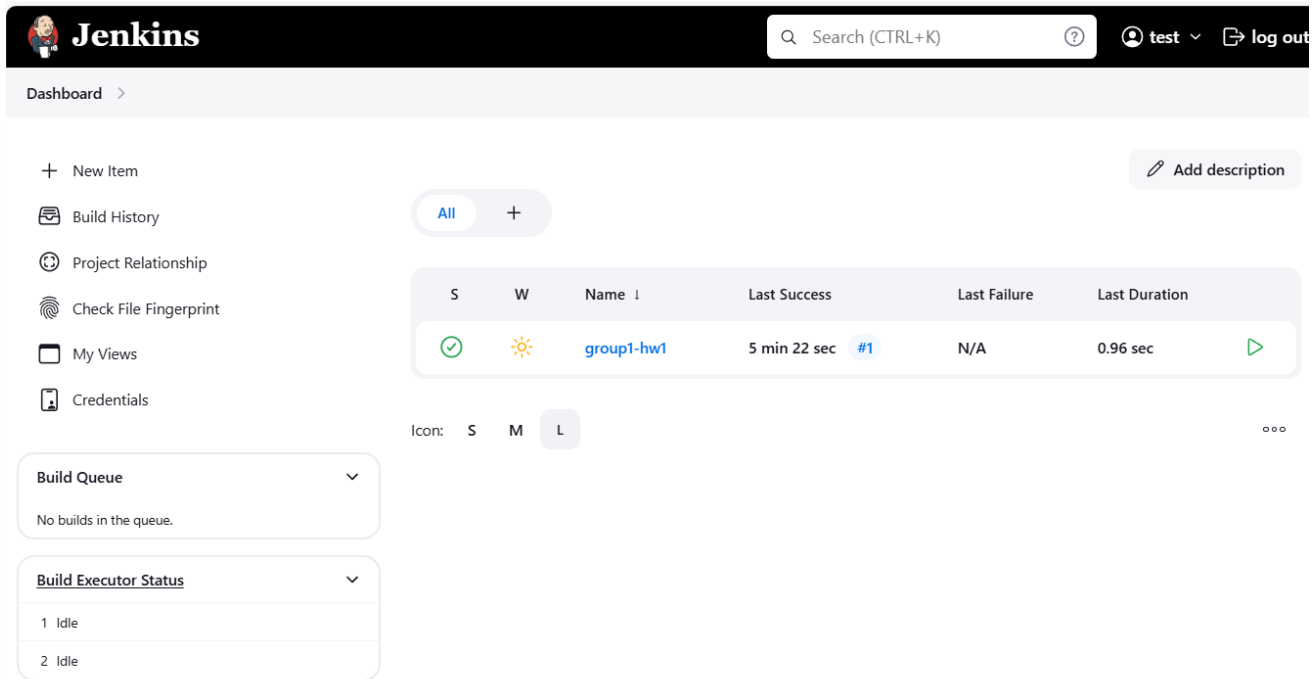
Download

Copy

View as plain text

```
Started by user test
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/group1-hw1
The recommended git tool is: NONE
using credential api-token
> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/group1-hw1/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/AuburnCsseRsardinas/bowling-game-sprint-1-frames-test-team.git # timeout=10
Fetching upstream changes from https://github.com/AuburnCsseRsardinas/bowling-game-sprint-1-frames-test-team.git
> git --version # timeout=10
> git --version # 'git version 2.34.1'
using GIT_ASKPASS to set credentials
> git fetch --tags --force --progress -- https://github.com/AuburnCsseRsardinas/bowling-game-sprint-1-frames-test-team.git +refs/heads/*:refs/remotes/origin/* # timeout=10
Seen branch in repository origin/main
Seen 1 remote branch
> git show-ref --tags -d # timeout=10
Checking out Revision 0cb9ca75b4bcca9e115f25bb3e46b9712aab5fc6 (origin/main)
> git config core.sparsecheckout # timeout=10
> git checkout -f 0cb9ca75b4bcca9e115f25bb3e46b9712aab5fc6 # timeout=10
Commit message: "Delete Jenkinsfile"
First time build. Skipping changelog.
[group1-hw1] $ /bin/sh -xe /tmp/jenkins14939086462858623608.sh
+ make
g++ -std=c++11 bowling_frame.o main.o
[group1-hw1] $ /bin/sh -xe /tmp/jenkins9471278698738169999.sh
+ ./a.out
0
Score 1: 8
Score 2: 2
Next Frame Score 1: 7
Finished: SUCCESS
```

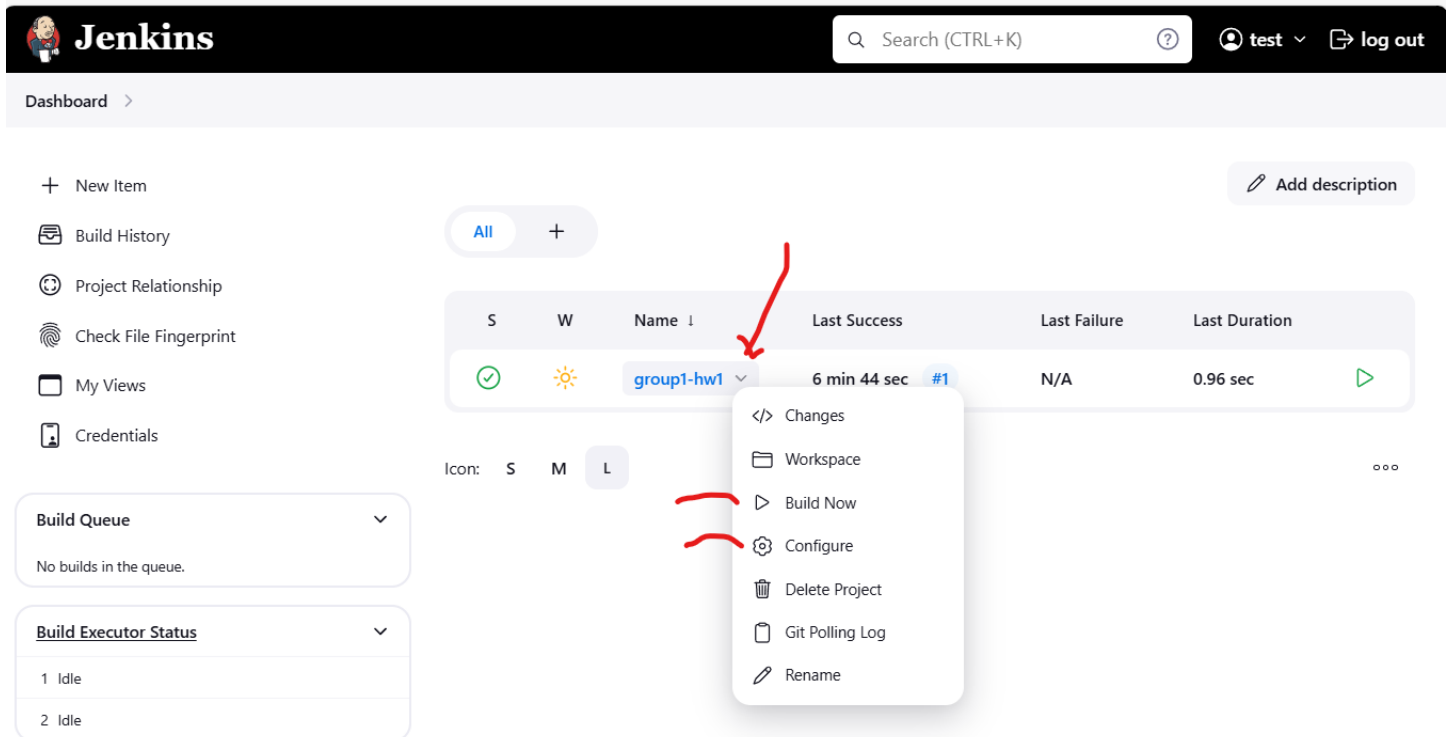
The tab we really care about will be the console output tab. Your submitted assignments should always produce output. When your executable is run, we can see that output here. Your assignments will be graded based on the output here.



A screenshot of the Jenkins Dashboard. The top navigation bar includes the Jenkins logo, a search bar with the text "Search (CTRL+K)", and user information "test" with a "log out" button. The main content area shows a sidebar with links: "New Item", "Build History", "Project Relationship", "Check File Fingerprint", "My Views", and "Credentials". The main panel displays a table of builds. The table has columns: "S" (Status), "W" (Workspace), "Name", "Last Success", "Last Failure", and "Last Duration". A single build is listed with status "Success" (green checkmark), workspace "group1-hw1", last success "5 min 22 sec", last failure "N/A", and last duration "0.96 sec". Below the table, there are two dropdown menus: "Build Queue" (showing "No builds in the queue.") and "Build Executor Status" (showing "1 Idle" and "2 Idle").

S	W	Name	Last Success	Last Failure	Last Duration
✓	☀	group1-hw1	5 min 22 sec #1	N/A	0.96 sec

If you go back to your dashboard, your default view will show you all jobs in the Jenkins server. This is one reason we want to have some kind of naming convention for jobs, to ensure that you ONLY work with or modify your jobs or even the job for the specific assignment. From here, you can select a job, rerun it, make modifications, etc...



A screenshot of the Jenkins Dashboard, similar to the one above, but with a context menu open for the "group1-hw1" job. A red arrow points to the job name in the table. The context menu is a white box with a shadow, containing the following options: "Changes", "Workspace", "Build Now", "Configure", "Delete Project", "Git Polling Log", and "Rename". Red brackets are drawn around the "Build Now" and "Configure" options. The rest of the dashboard interface is the same as in the previous screenshot.

S	W	Name	Last Success	Last Failure	Last Duration
✓	☀	group1-hw1	6 min 44 sec #1	N/A	0.96 sec

- Changes
- Workspace
- Build Now
- Configure
- Delete Project
- Git Polling Log
- Rename

Please be careful NOT to delete anyone else's jobs.