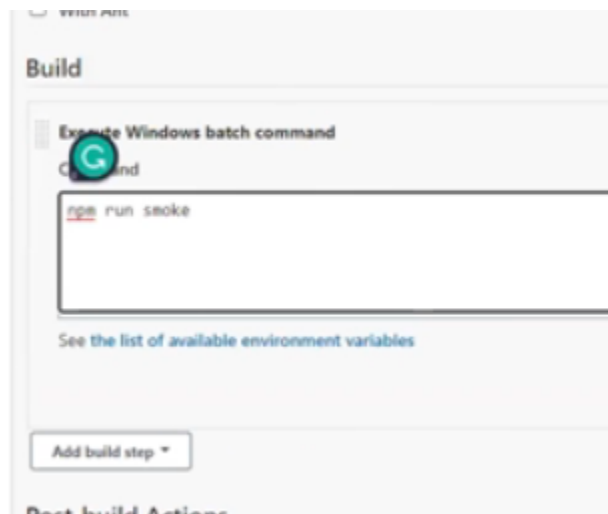


Trigger options—



The screenshot shows the 'Triggers' section of a Jenkins job configuration. It features two 'Schedule' fields. The first field contains the cron expression '30 19 * * *'. Below it, a warning message states: 'Spread load evenly by using 'H 19 * * *' rather than '30 19 * * *''. It also provides the last and next run times: 'Would last have run at Friday, October 14, 2022 7:30:43 PM IST; would next run at Saturday, October 15, 2022 7:30:43 PM IST.' Below the warning, there are two checkboxes: 'Github hook trigger for GITSCM polling' (unchecked) and 'Poll SCM' (checked). The second 'Schedule' field contains the cron expression 'H/2 * * *'. Below this field, it shows the last and next run times: 'Would last have run at Saturday, October 15, 2022 7:27:57 PM IST; would next run at Saturday, October 15, 2022 7:29:57 PM IST.'

Poll scm - it will keep polling the github at the time mentioned.



The screenshot shows the 'Build' section of a Jenkins job configuration. It features a 'Execute Windows batch command' step. The command field contains the text 'run smoke'. Below the command field, there is a link that says 'See the list of available environment variables'. At the bottom of the section, there is a button labeled 'Add build step'.

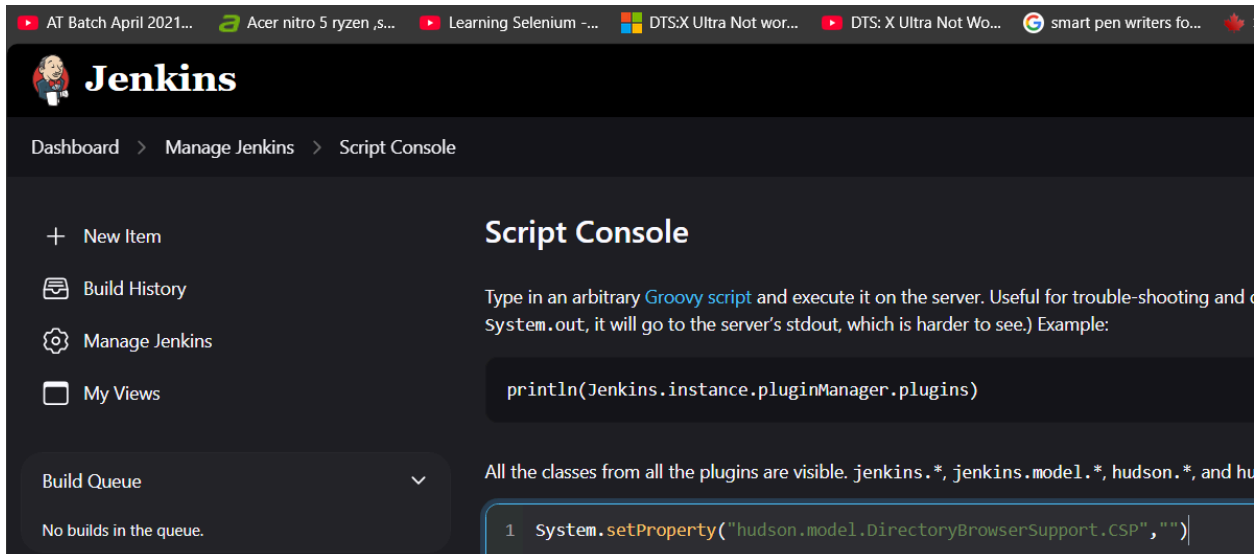
Html reports cannot be directly seen in jenkins.

Go to manage jenkins.

Script console.



Clear picture -



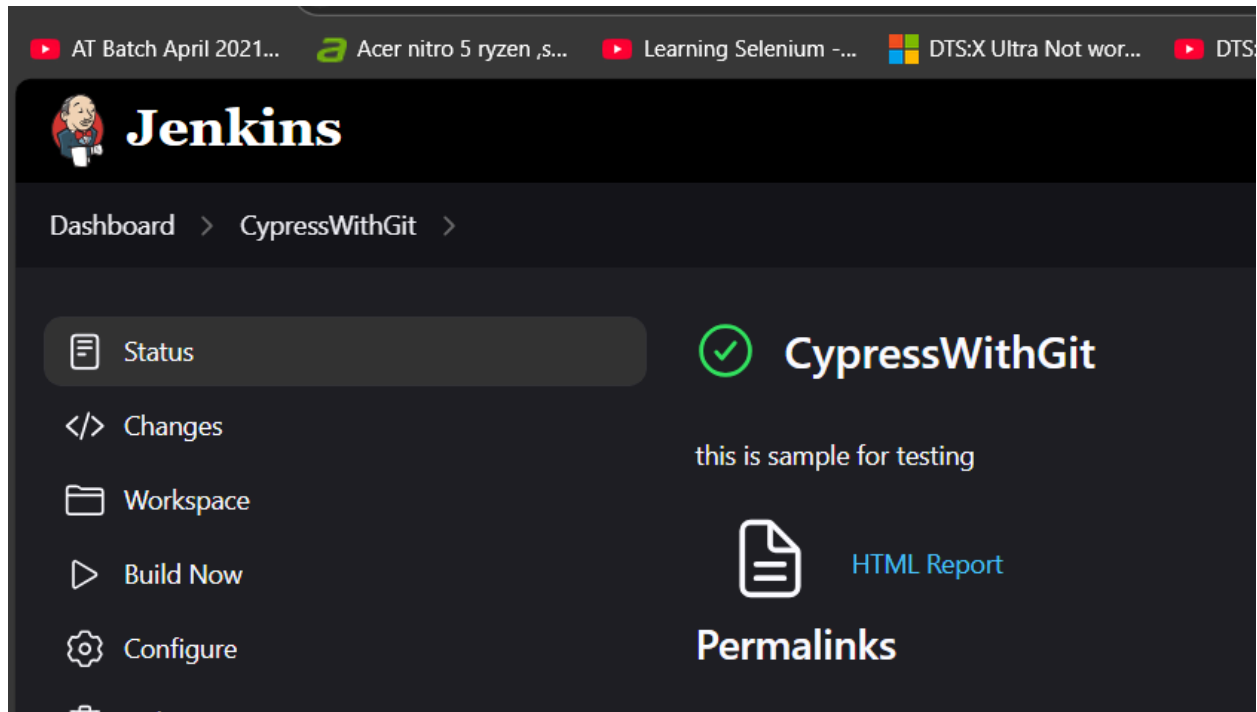
This script mukesh took from somewhere.

Click run.

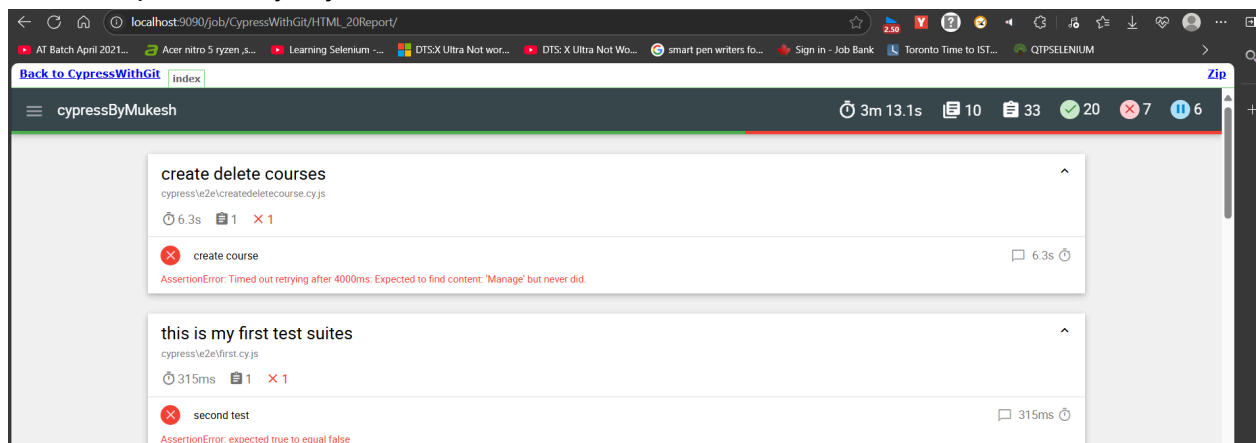
Till jenkins is up and running it will work.

Once jenkins closed then we need to rerun this.

Now click html -



We see report directly in jenkins-



Why we added groovy -
Known bug in jenkins for not showing html report.

Note-

If we want to run different jobs one option is create different executables.
Second way and best is parameterise.

[Plain text] [Preview](#)

☐ Discard old builds

☐ GitHub project

☐ This build requires lockable resources

☒ This project is parameterized

String Parameter

Name [?](#)

Default Value [?](#)

Description [?](#)

Please provide the name of test suite that you want to trigger

[Plain text] [Preview](#)

☐ Trim the string [?](#)

localhost:8080/view/cypressEveningBatch/job/cypressGithub/configure

CypressEveningBatch > CypressGithub >

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

Execute Windows batch command

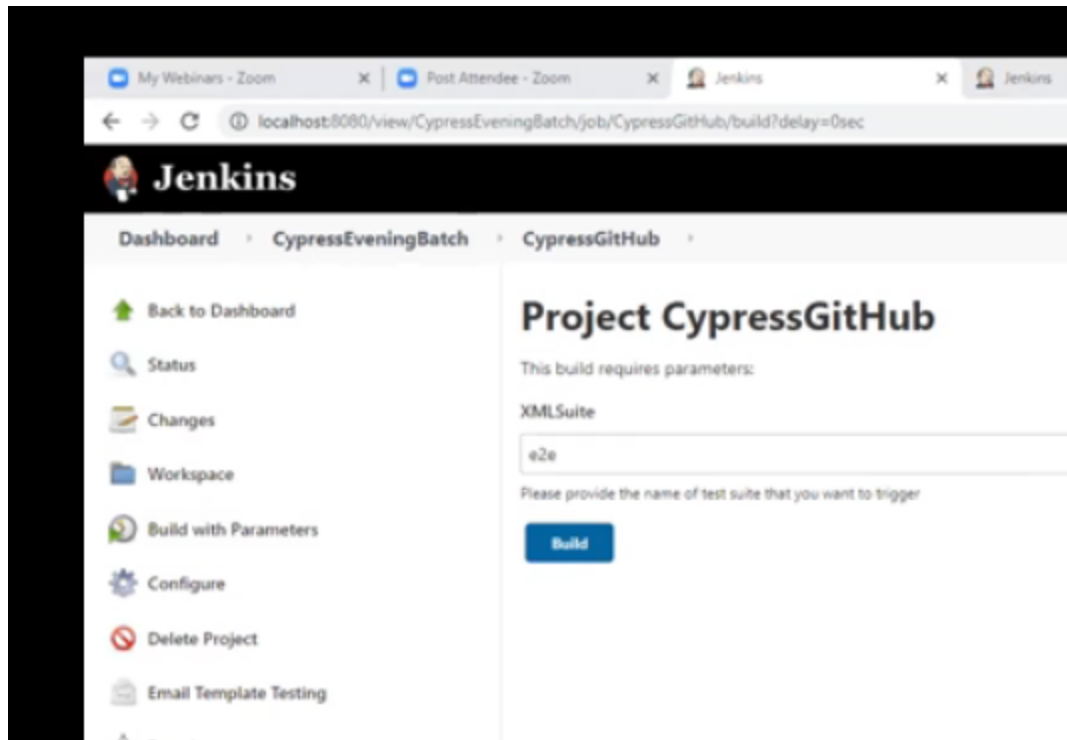
Runs a Windows batch script for building the project. The script will be run with the workspace as the current directory. The text you enter in the text box will be failure if at the end of the execution %ERRORLEVEL% is not 0.

If you already have a batch file in SCM, you can just type in the path of that batch file (again relative to the workspace directory), and simply execute that.

Command

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

See name of the build area is changed to build with parameters and the value how we pass-



Problem with string parameter, we give wrong values and the script fails.

Choice parameter-

[Plain text] [Preview](#)

☐ Discard old builds

☐ GitHub project

☐ This build requires lockable resources

☒ This project is parameterized

Choice Parameter

Name [?](#)

Choices [?](#)

e2e
smoke
regression
smoke-chrome

Description [?](#)

Please select suite which you want to execute

[Plain text] [Preview](#)

[Add Parameter](#)

☐ Throttle builds

[Disable this project](#)

☐ Color ANSI Console Output

☐ Inspect build log for published Gradle build scans

☐ With Ant

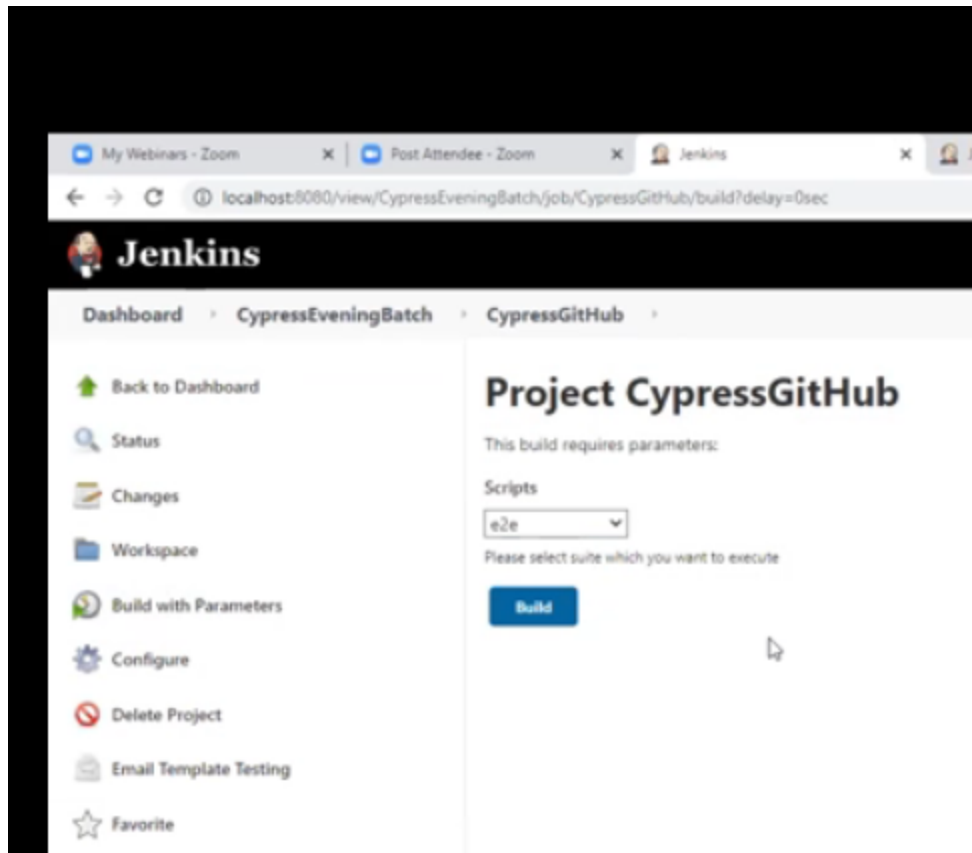
Build

Execute Windows batch command

Command

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

[Add build step](#)



In choice always keep the environment on which you want to run the cron jobs first, else it will pick the first one you mentioned.

NOTE-

If we have two parameters, then the first one is picked up by default. Second parameter is not picked up. So either use choice or string or boolean or etc etc.

Not a good practice-

migration

☐ GitHub project

☒ This project is parameterized ?

≡ **String Parameter** ?

Name ?

enterValue

Default Value ?

smoke

Description ?

please provide test name to run

Plain text [Preview](#)

☐ Trim the string ?

> Configuration

Review

☐ Trim the string ?

≡ **Choice Parameter** ?

Name ?

environments

Choices ?

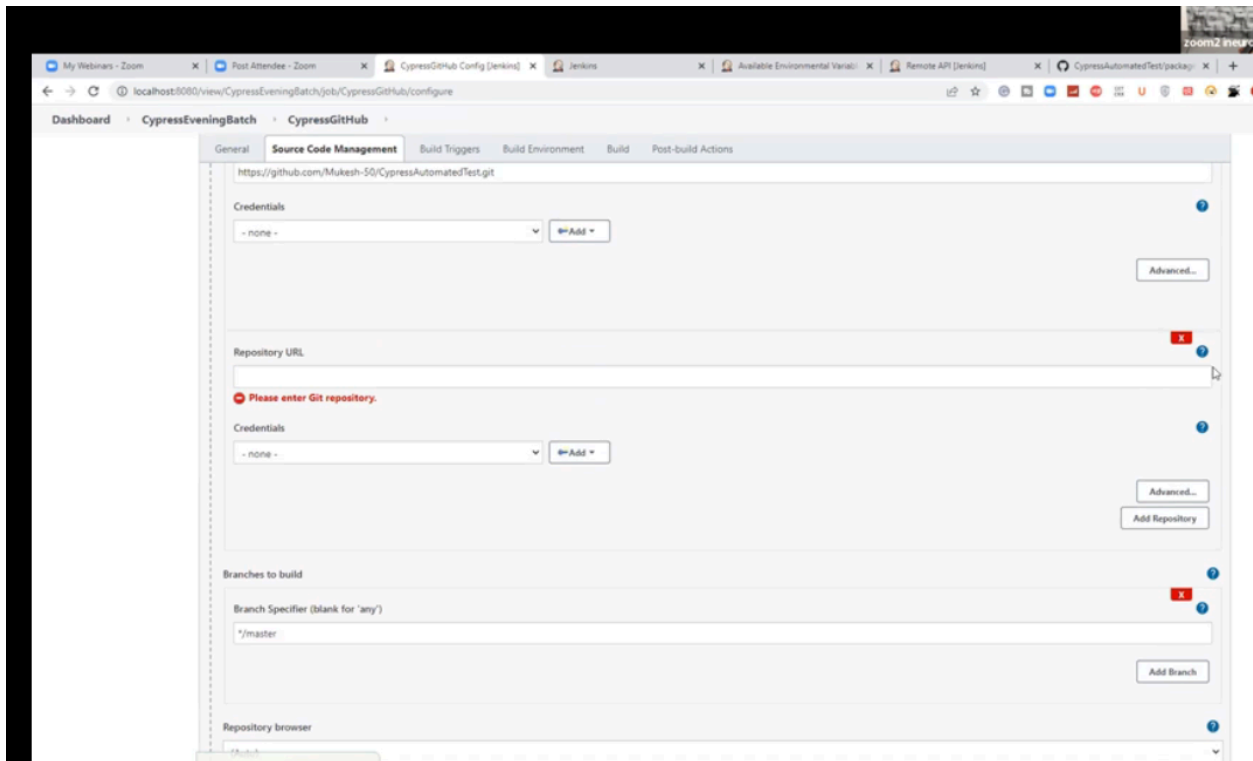
test
smoke
regression
e2e
smokeonchromebrowserheaded

Description ?

Plain text [Preview](#)

Add Parameter ✕

We can add multiple repo and specify the branches where the repo should run.



Pipeline-
If one job fails entire pipeline fails.

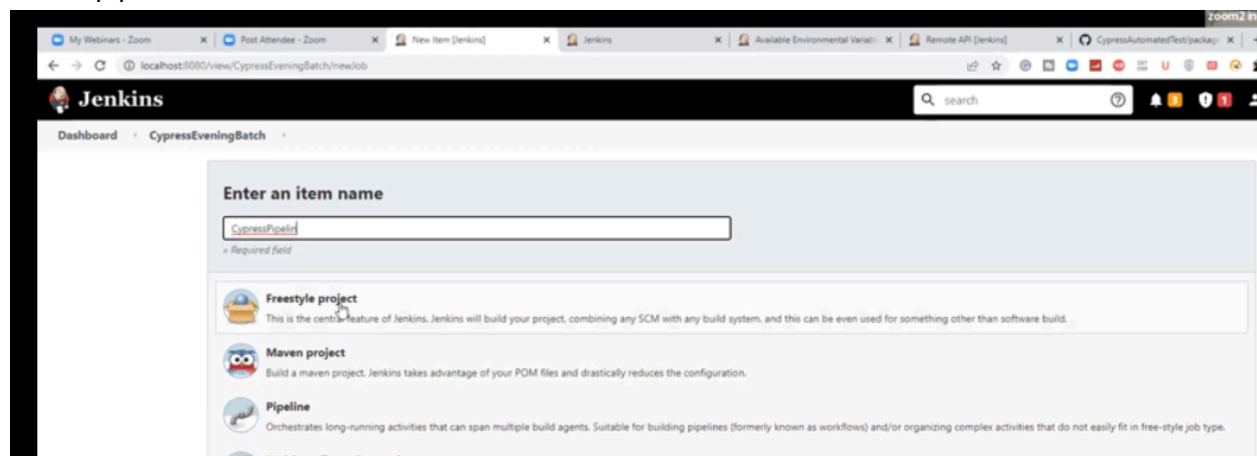


Agent means where the pipeline will run.

Agent any means any agent which is free will run the code.

Enter name.

Select pipeline.

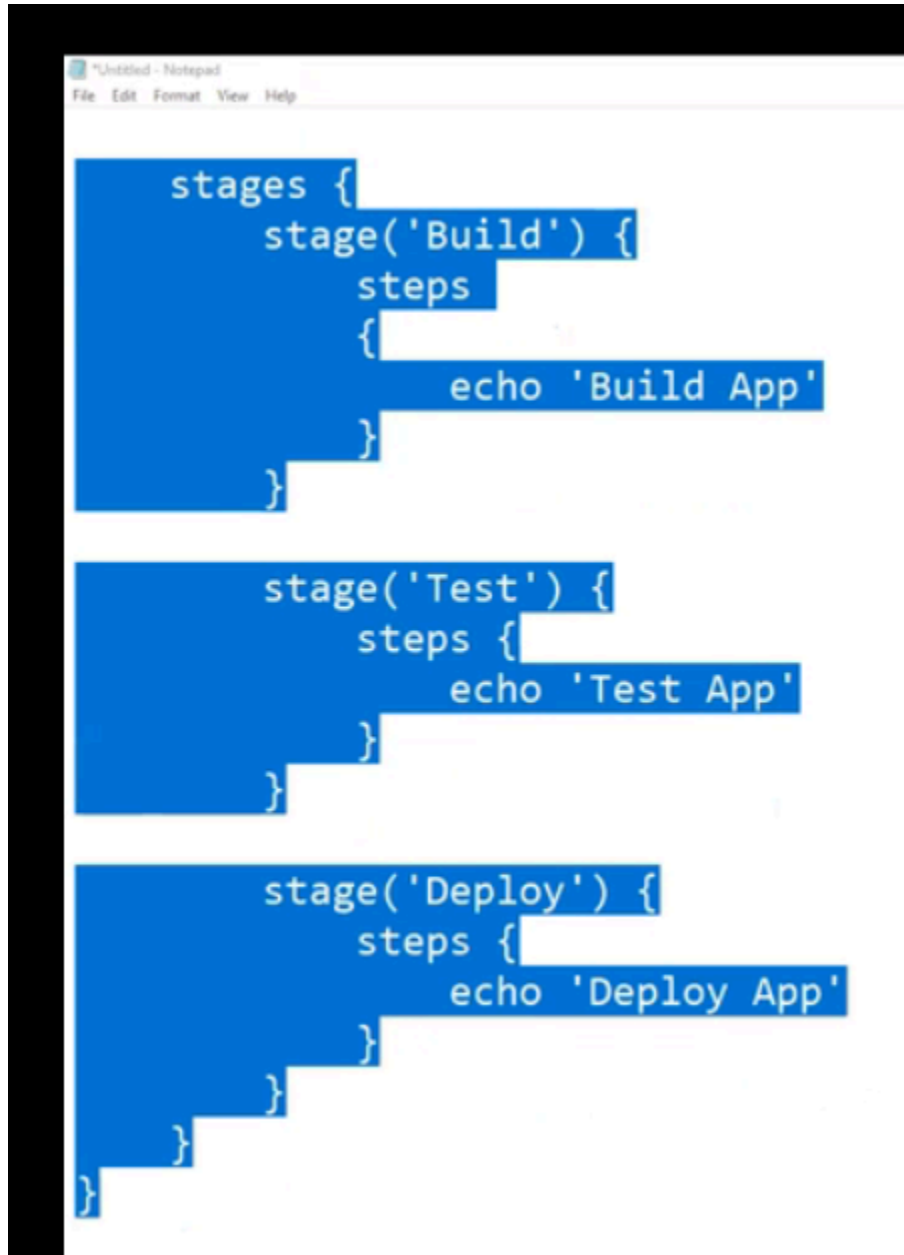


Note-

Scroll down and you will see pipeline generator.

There is pipeline syntax generator already present, you select what you want and the script will be generated. You copy paste in your code.

The simplest pipeline-

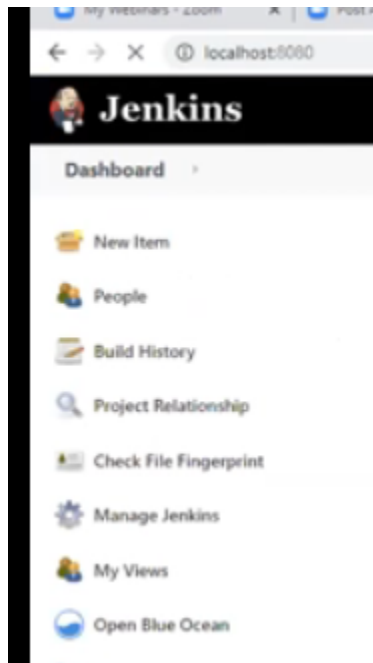
A screenshot of a Notepad window titled "Untitled - Notepad" with a menu bar (File, Edit, Format, View, Help). The window contains a YAML pipeline script. The script defines three stages: 'Build', 'Test', and 'Deploy'. Each stage has a 'steps' block containing an 'echo' command. The 'Build' stage echoes 'Build App', the 'Test' stage echoes 'Test App', and the 'Deploy' stage echoes 'Deploy App'. The script is formatted with blue highlighting for the stage and step definitions, and white space for the echo commands.

```
stages {  
  stage('Build') {  
    steps  
    {  
      echo 'Build App'  
    }  
  }  
  
  stage('Test') {  
    steps {  
      echo 'Test App'  
    }  
  }  
  
  stage('Deploy') {  
    steps {  
      echo 'Deploy App'  
    }  
  }  
}
```

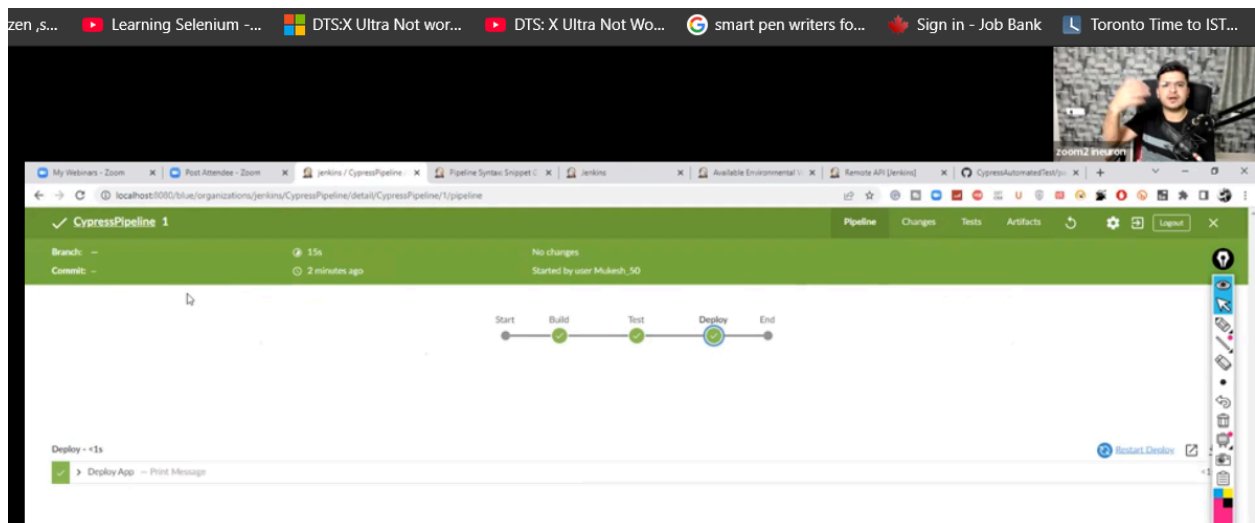
Click save. Click apply.

Build now.

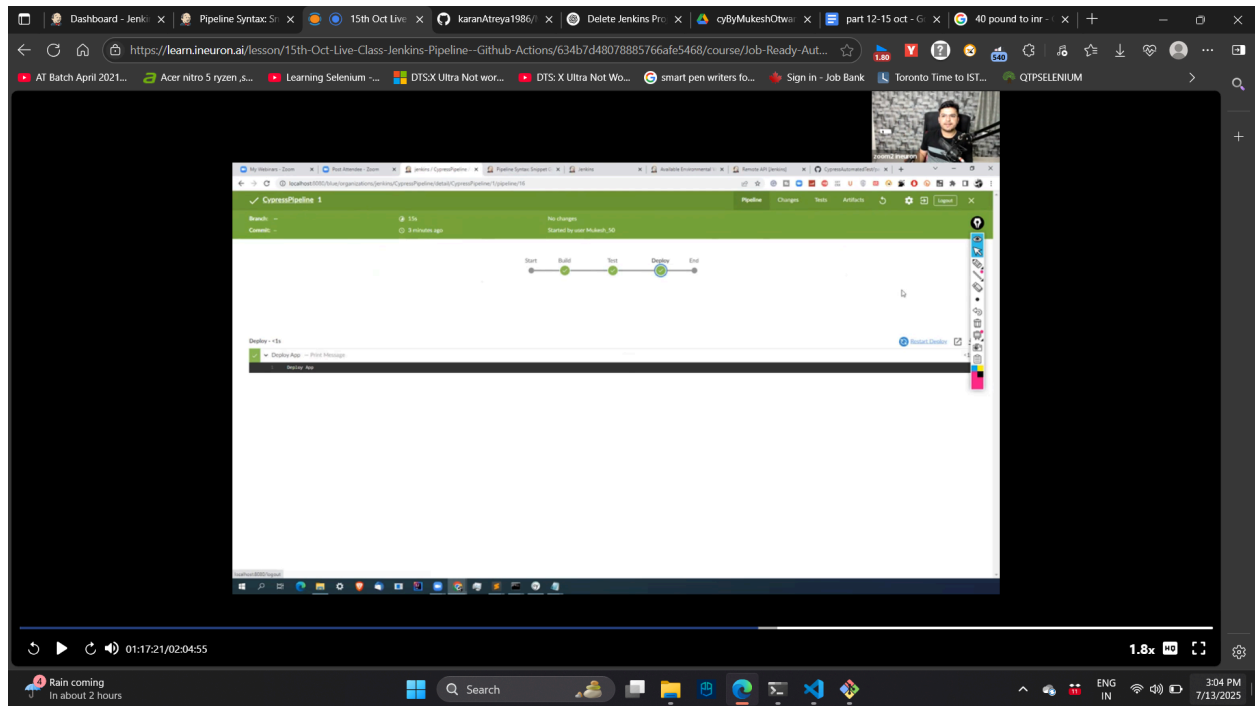
Blue ocean-



Click open blue ocean. It will give nice graphical view of the pipeline. Blue ocean is plugin we have to install.



Ocd-



Trying to build sample jobs-

Here we are adding our jenkins jobs itself to the pipeline. You pass the job name.



Another pipeline to run multiple jobs –

```
1 pipeline {
2     agent any
3
4     stages {
5         stage('Smoke') {
6             steps
7             {
8                 build 'NodeJS Sample Job'
9             }
10        }
11
12        stage('Regression') {
13            steps {
14                build 'NodeSecondJobWithEmail'
15            }
16        }
17
18        stage('E2E') {
19            steps {
20                build job: 'CypressGitHub'
21            }
22        }
23    }
```

codesnap.dev

How to use pipeline script generator-
We want to build another jenkins pipeline job.

Steps

Sample Step

build: Build a job

build ?

Project to Build ?

CypressWithChoiceParameter

Project to Build ?

CypressWithChoiceParameter

☒ Wait for completion ?

☐ Wait until the build starts ?

☒ Propagate errors ?

Quiet period ?

Parameters ?


environments

smokeonchromebrowserheaded

Generate Pipeline Script

```
build job: 'CypressWithChoiceParameter', parameters: [string(name: 'environments', value: 'smokeonchromebrowserheaded')]
```


Sample jobs created with pipeline help-



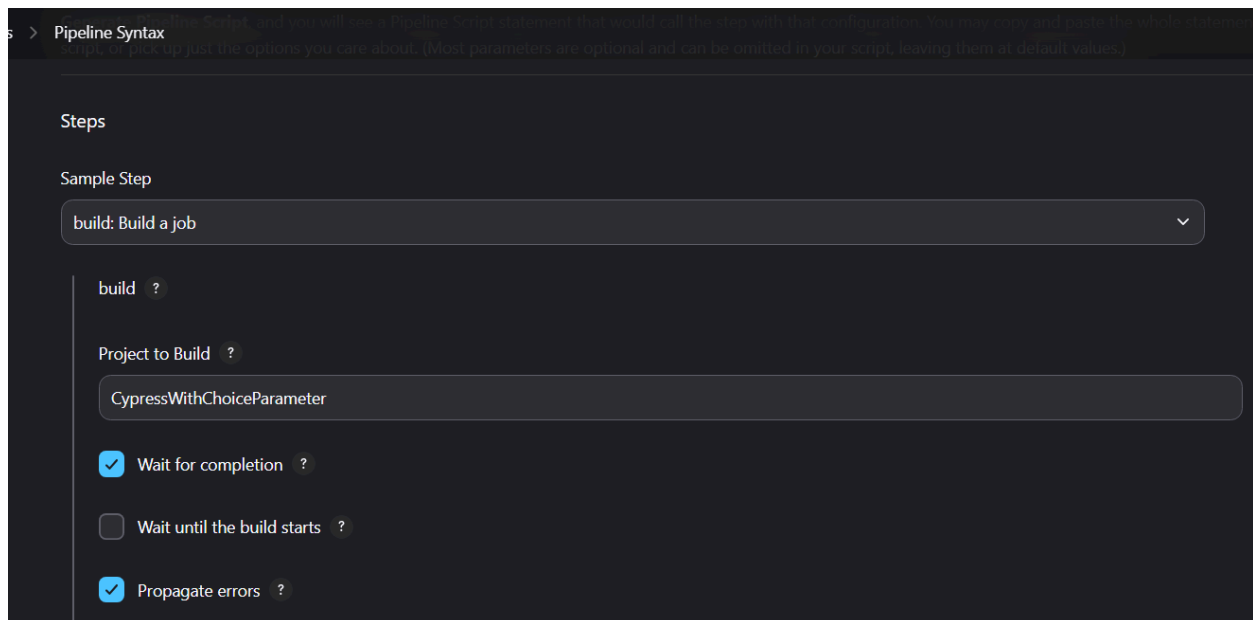
```
1 pipeline {
2   agent any
3
4   stages {
5     stage('smoke') {
6       steps {
7         build job: 'CypressWithChoiceParameter', parameters: [string(name: 'environments', value: 'smokeonchromebrowserheaded')]
8       }
9     }
10
11    stage('regression') {
12      steps {
13        build 'CypressTesting'
14      }
15    }
16
17    stage('e2e') {
18      steps {
19        build 'CypressWithSSHCheck'
20      }
21    }
22  }
23 }
24
```

codesnap.dev

If you want to run the parameters like the one we selected from drop down—

Go to generator.

Give the job name which requires parameter.



Pipeline Syntax

Sample Step

build: Build a job

build ?

Project to Build ?

CypressWithChoiceParameter

☒ Wait for completion ?

☐ Wait until the build starts ?

☒ Propagate errors ?

You can select the environment where you want to run. Click generate script and done.

Quiet period ?

Parameters ?

environments

smoke

Generate Pipeline Script

```
build job: 'CypressWithChoiceParameter', parameters: [string(name: 'environments', value: 'smoke')]
```

📄

Sample yaml file-

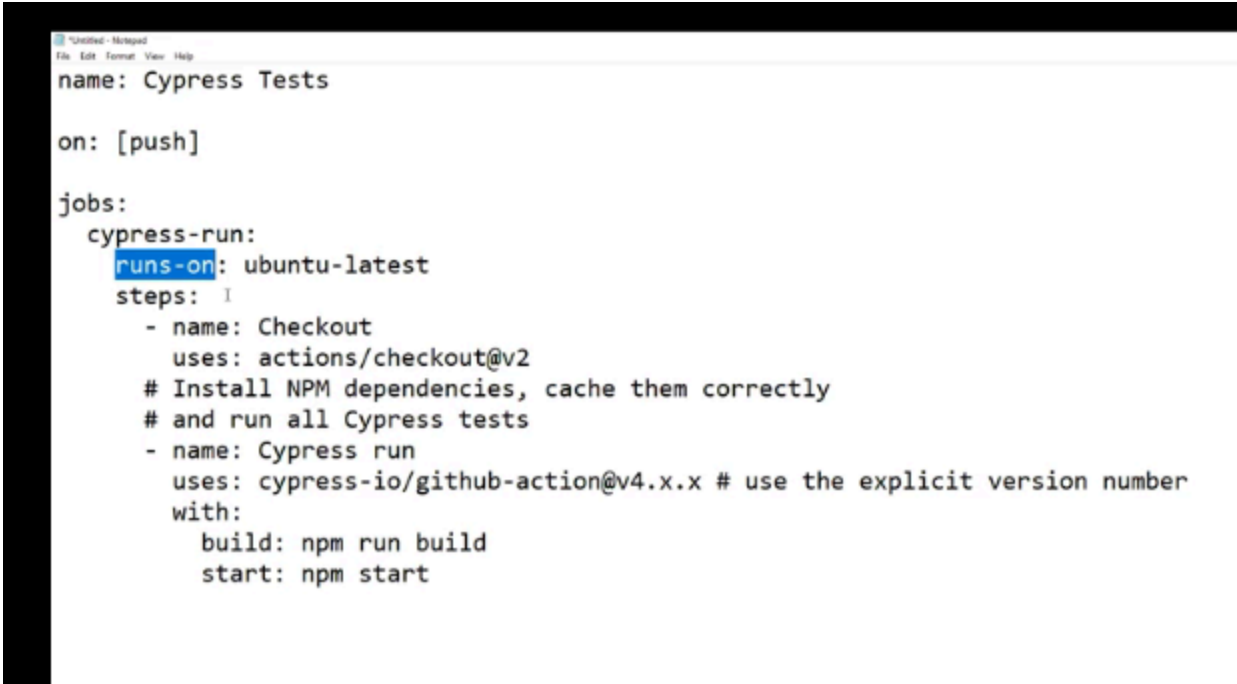
```
ci.yml

1 on: push
2 jobs:
3   test:
4     strategy:
5       matrix:
6         platform: [ubuntu-latest, macos-latest, windows-latest]
7     runs-on: ${{ matrix.platform }}
8     steps:
9       - uses: actions/checkout@v3
10      - uses: actions/setup-node@v3
11        with:
12          node-version: 16
13      - run: npm install-ci-test
14      - uses:
```

Note-

Cypress does not have parallel runs like pw.

Pay and use or take github actions help.



```
name: Cypress Tests

on: [push]

jobs:
  cypress-run:
    runs-on: ubuntu-latest
    steps:
      - name: Checkout
        uses: actions/checkout@v2
        # Install NPM dependencies, cache them correctly
        # and run all Cypress tests
      - name: Cypress run
        uses: cypress-io/github-action@v4.x.x # use the explicit version number
        with:
          build: npm run build
          start: npm start
```

Name - we give.

On - event.

Jobs - different jobs.

Runs on - where to run the job.

Steps- what individual task to run.

Name can be anything.

Checkout - takes code from github and puts in ubuntu.

With mainly for developers, to build and start the application.