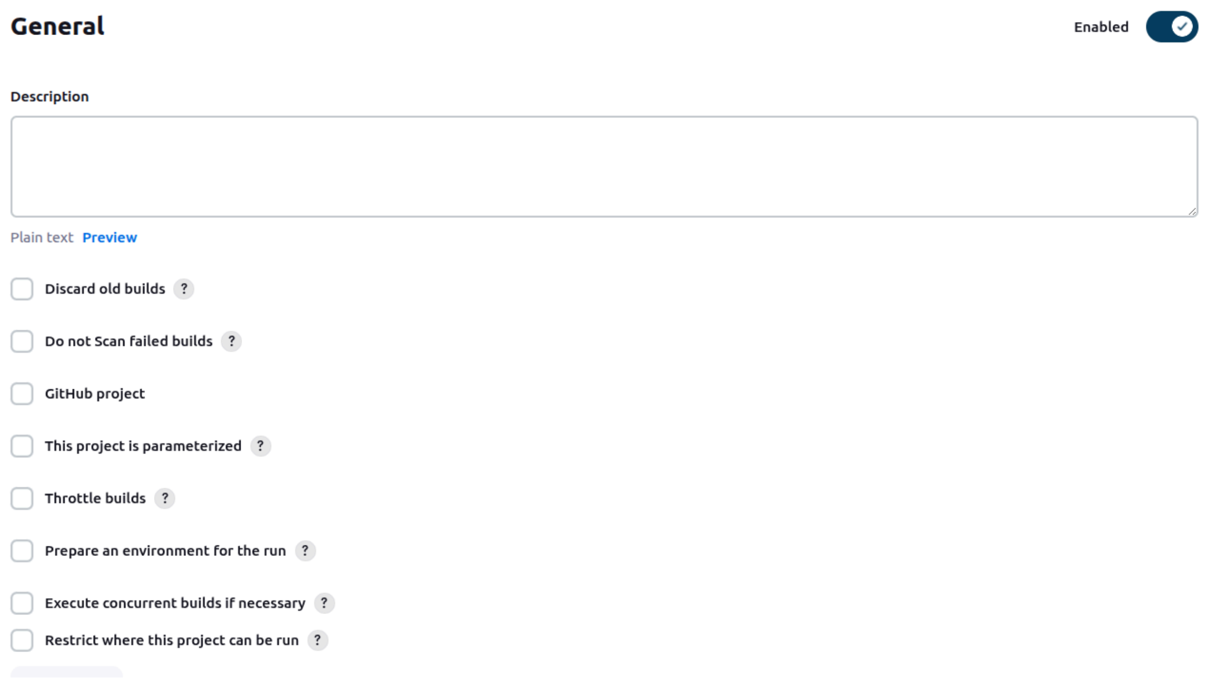
jenkins class-03

so in this class free style jobs what are left we will go through here
in jenkins we can install through the file i.e when the file excute it will install like shell scipt but .sh is compulsary

harish.sh
sudo wget -O /etc/yum.repos.d/jenkins.repo https://pkg.jenkins.io/redhatstable/jenkins.repo
sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key
amazon-linux-extras install java-openjdk11 -y
systemctl start jenkins
systemctl status jenkins

after that excutable permission
if excute the script it will download jenkins



General

Enabled

Description

Plain text Preview

□ Discard old builds ?

□ Do not Scan failed builds ?

□ GitHub project

□ This project is parameterized ?

□ Throttle builds ?

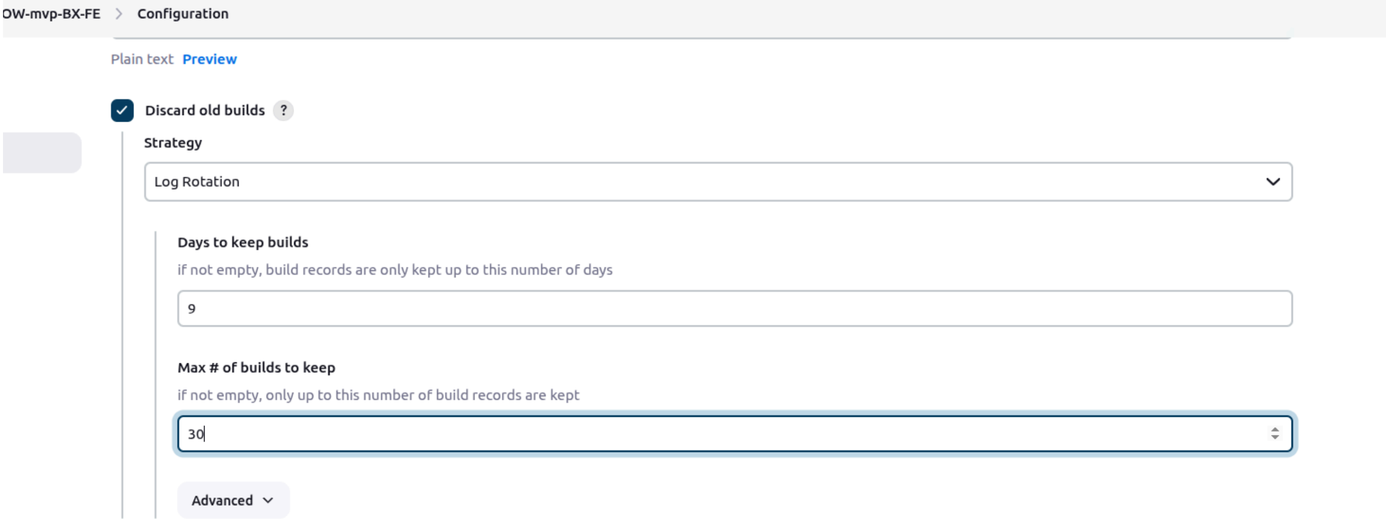
□ Prepare an environment for the run ?

□ Execute concurrent builds if necessary ?

□ Restrict where this project can be run ?

**we will discuss above points today with clearly
1. Description means the job description at what and why
2. Discard Builds: means build numbers are more then 1000 let say like per day 20 job build like week 120 build like that month keeep going we have more build for that
so for that what i want is last 7 days latest 40 jobs build i need console so thini kosam nenu configure chesukuntam and one more is let say 50 builds are done but
** **but iam storing here 40 only and remaing 10 builds will not deleted it will discard it will store in server but in jenkins dashboard it is hide so if we want we can take in jenkins , it store in jenkins default path /var/lib/jenkins so this sutuation**

using discard build



OW-mvp-BX-FE > Configuration

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

9

Max # of builds to keep

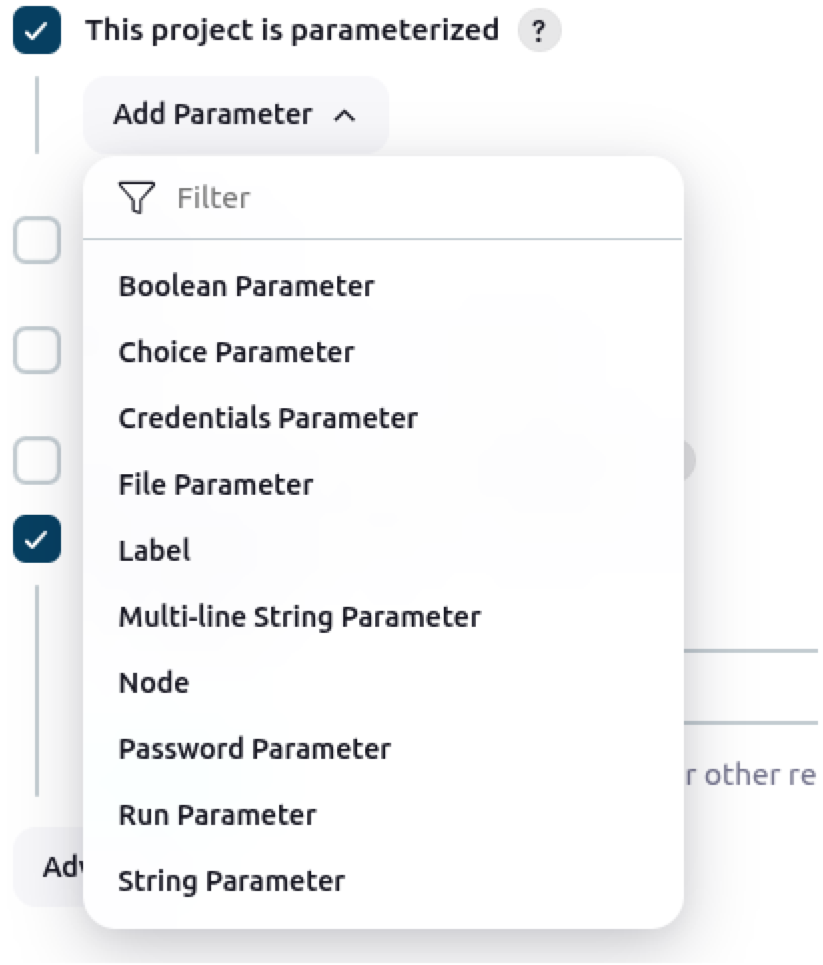
if not empty, only up to this number of build records are kept

30

GITHUB URL

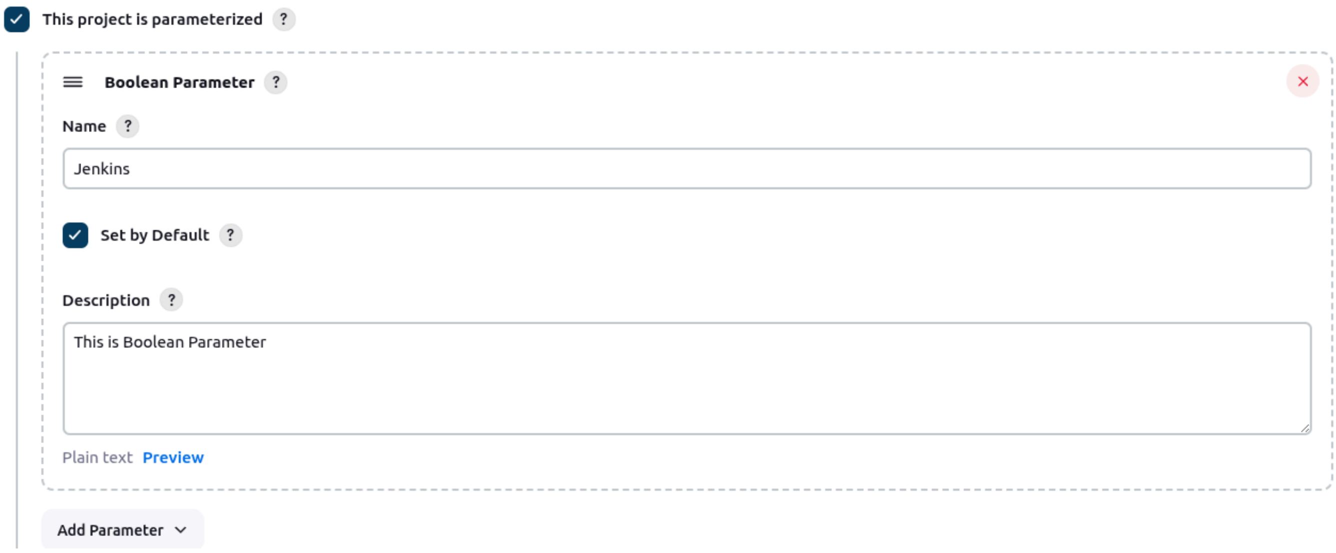
our related project url we will give here if we give or give not also it will work because source code we will give so this only for description purpose

PARAMETERIZED PROJECT



☑ This project is parameterized ?

Boolean Parameter: is used for true or false condition we use 0/1 active/activenot male/feamle created/deleted when we have two option we use Boolean so how to configure will see select Boolean Parameter-->Name Jenkins-->Description(this is boolean paramter)-->save



☑ This project is parameterized ?

Boolean Parameter ?

Name ?

Jenkins

☑ Set by Default ?

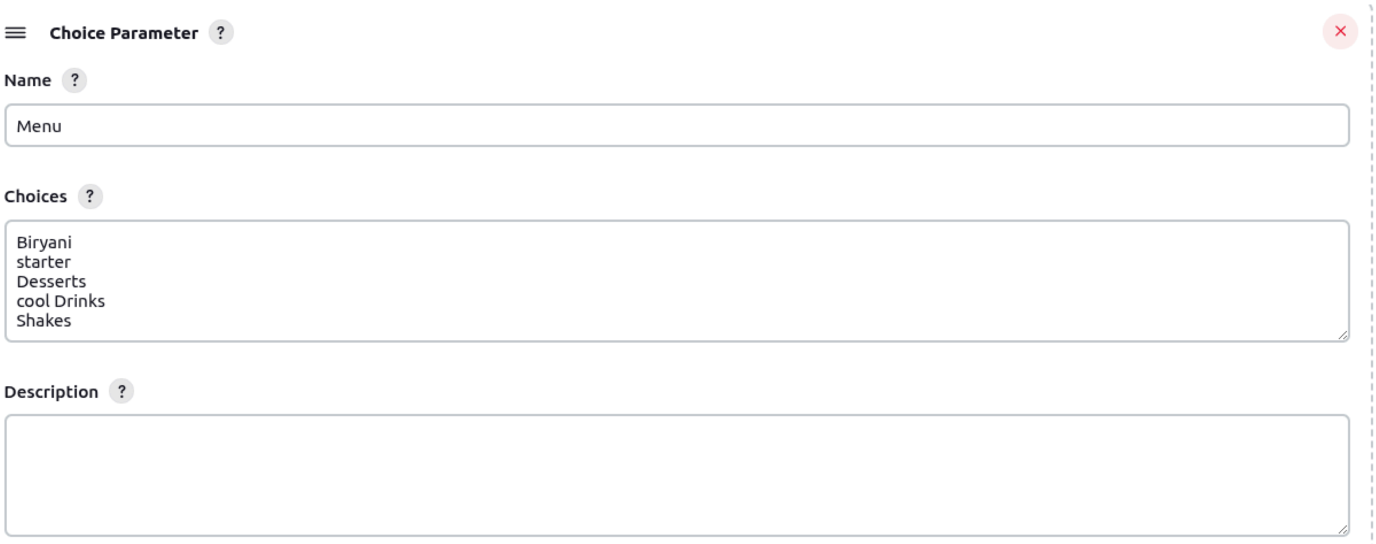
Description ?

This is Boolean Parameter

tick means: true
unticket means: false

choice parameter:

**when we have multiple option we use choice paramter
restarent: biryani,starter,dessert,shakes,harddrinks,



Choice Parameter ?

Name ?

Menu

Choices ?

Biryani
starter
Desserts
cool Drinks
Shakes

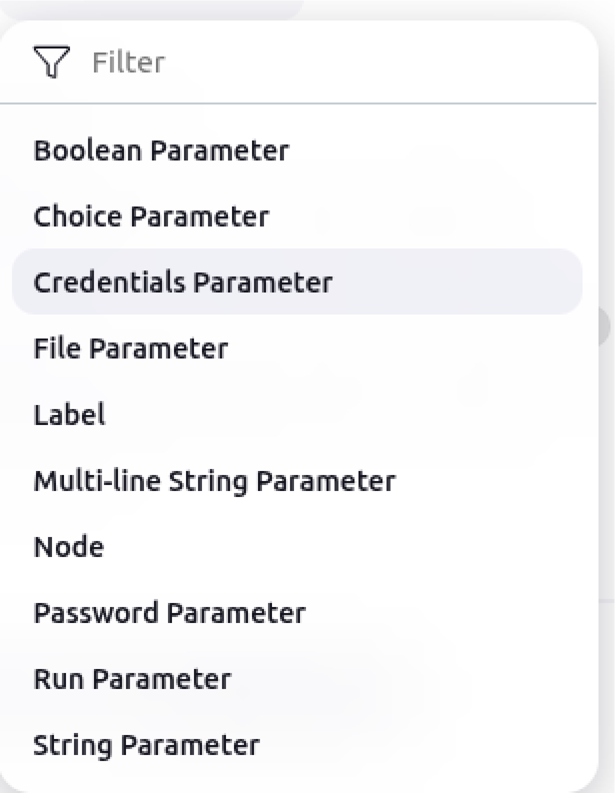
Description ?

**

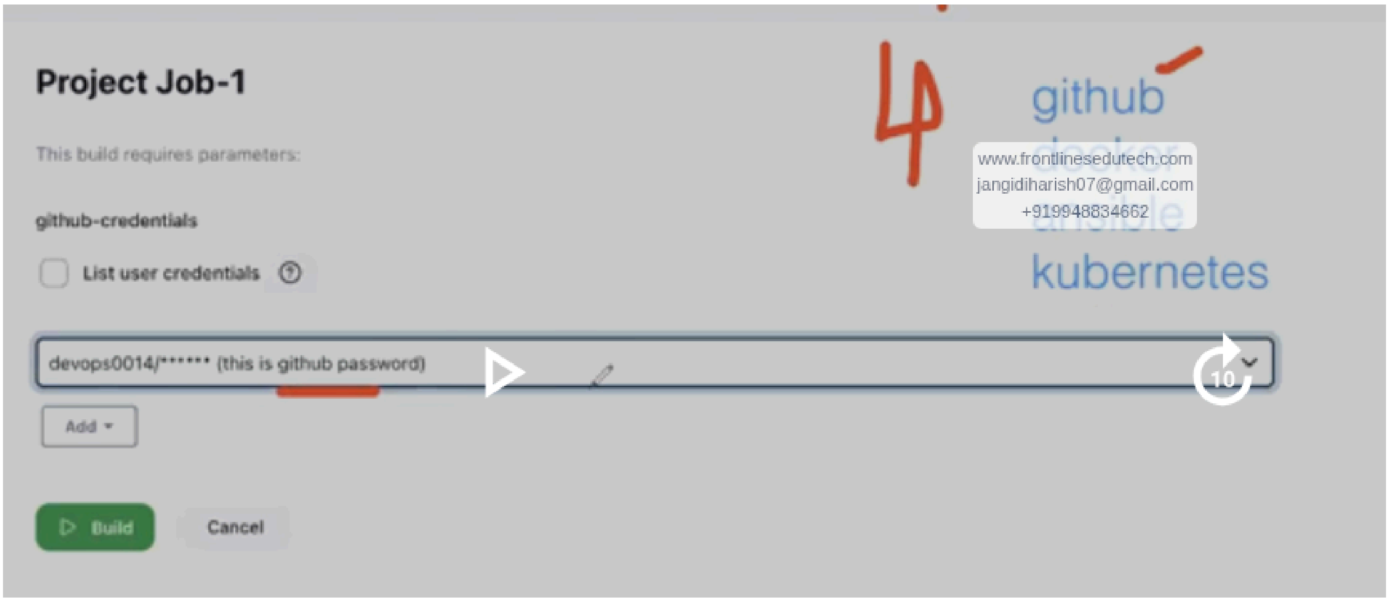
so why we have to use means generally be deafult we use master when select github project so after time manager asking dev branch so agin i need to configure so instead of that we can use choice paramter so it is manual and time waste so for the automatic i use chooice parameter so if manger ask i donot go configure mode just build with paramter

credentials Paramter

in earlier when i build the project suppose private repo we use github credentials so here we can add directly through paramter also



Name is Any thing letsay: github credentials and type is username with password in the default value we hae to add jenkins --> add jenkins -->where as username give has github/bitbucker username where as password enter token of the repo where we can get settings


Project Job-1

This build requires parameters:

github-credentials

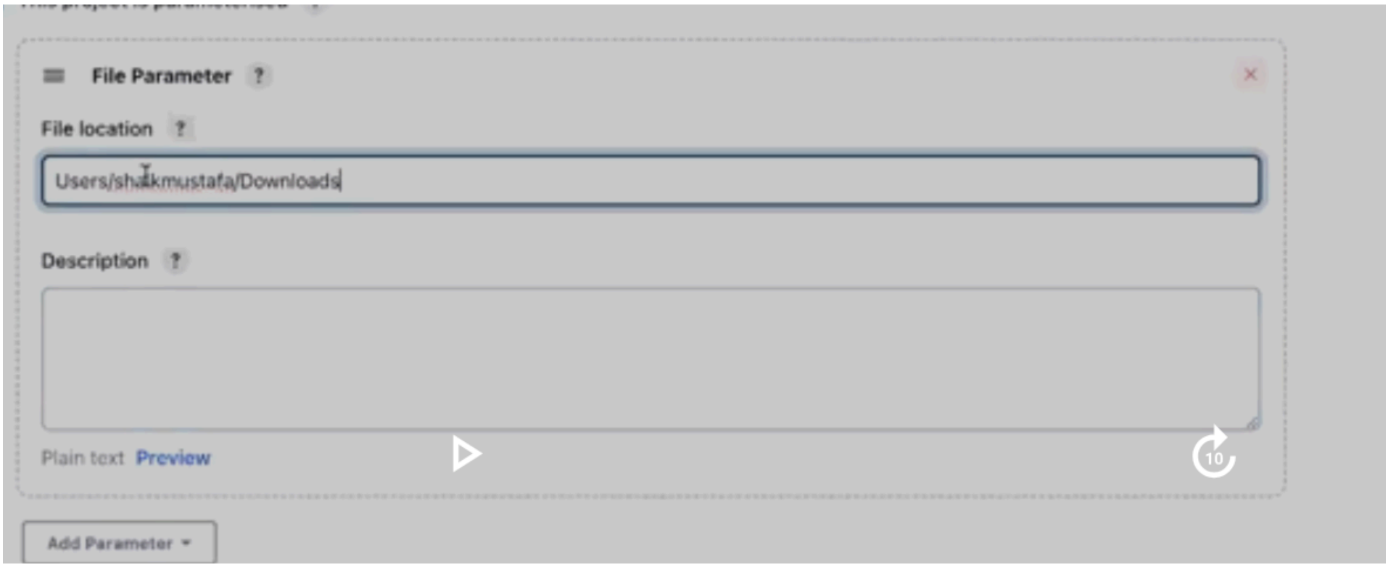
□ List user credentials

devops0014/****** (this is github password)

whichever we want we can use if we are using now docker we can select docker credentilas and as well github as well bitbucket like that we can select whichever we want while first time configure setup done

FILE PARAMETER

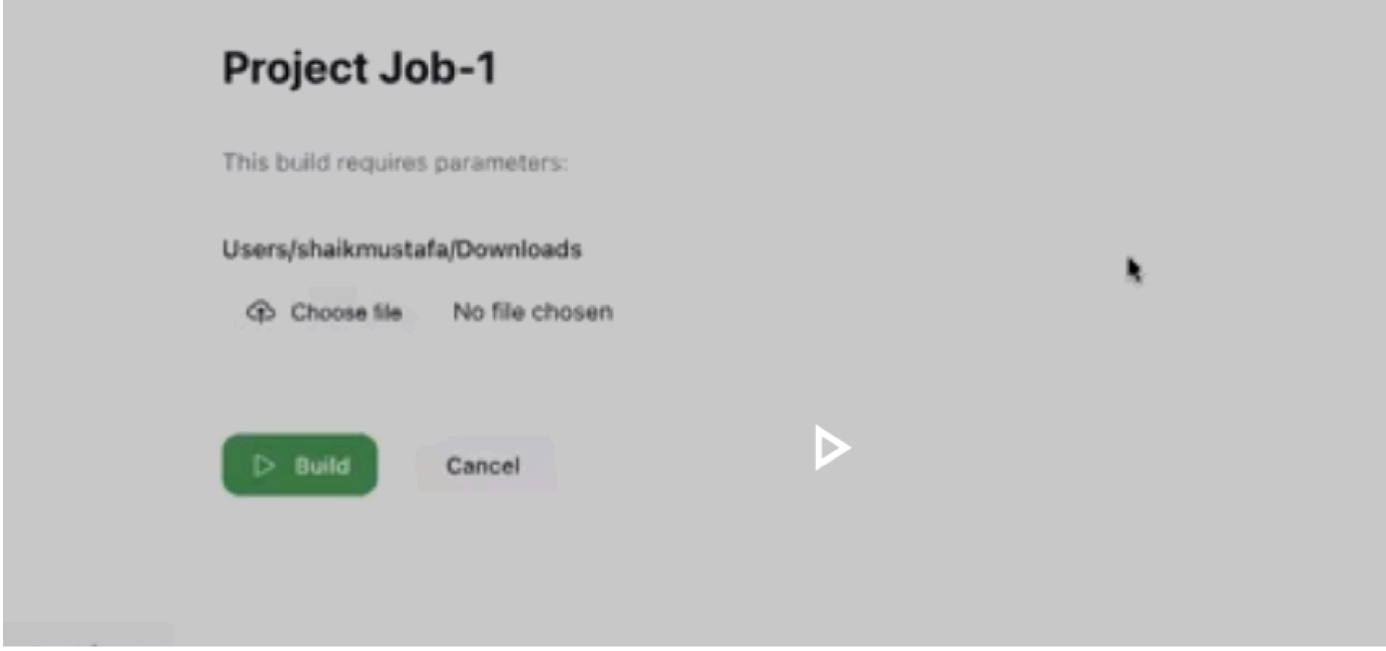
suppose in your local have a file then how to bring that file local to ci server at the time we can use file parameter



File location: Users/shaikmustafa/Downloads

Description:

just give the file path while build it will show the upload file then it will go this location and we did not select the file



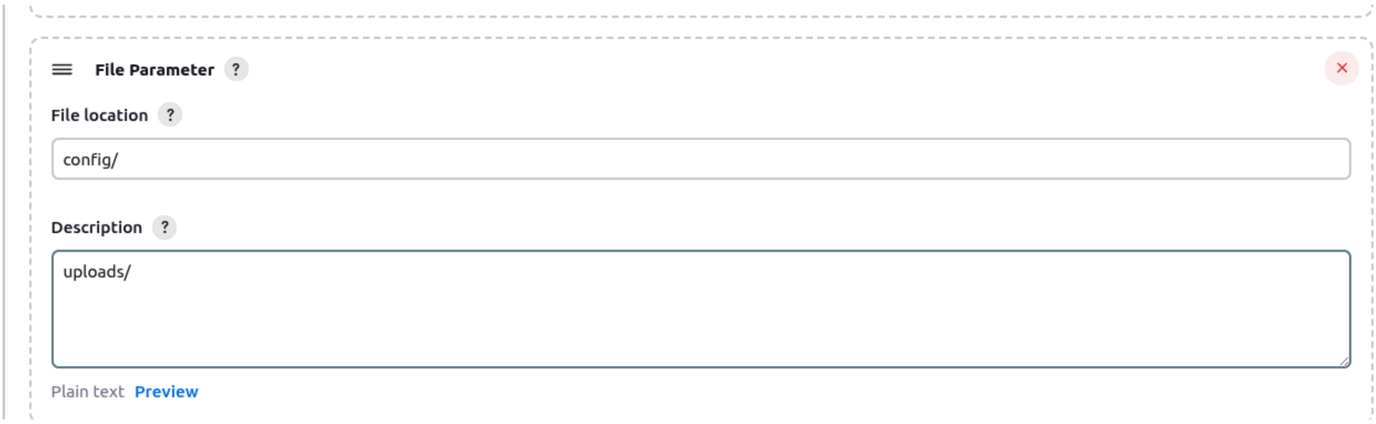
Project Job-1

This build requires parameters:

Users/shaikmustafa/Downloads

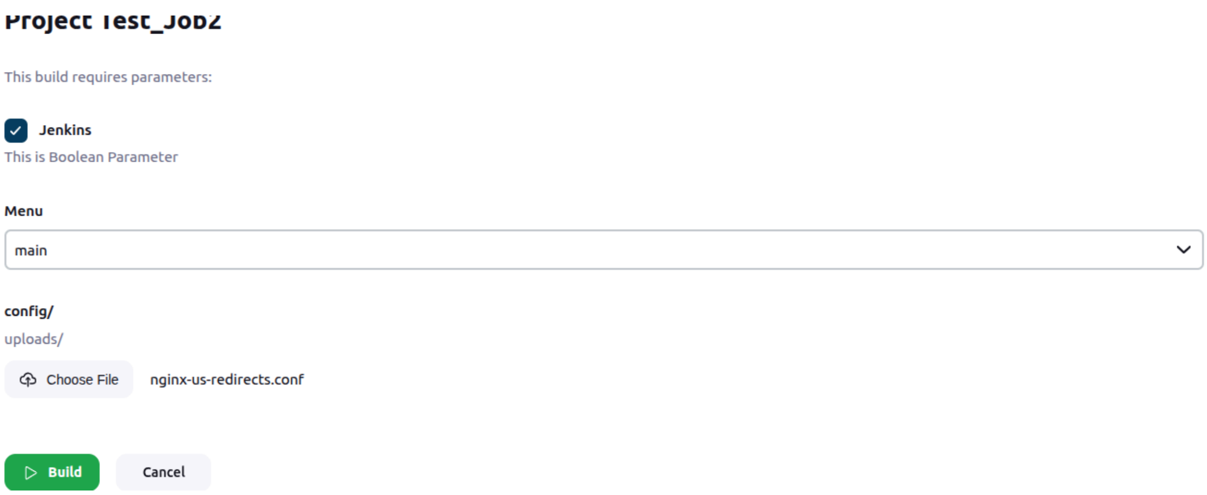
No file chosen

you have to choose the file path



File location: config/

Description: uploads/



Project Test_Job2

This build requires parameters:

☑ Jenkins

This is Boolean Parameter

Menu

main

config/

uploads/

nginx-us-redirects.conf

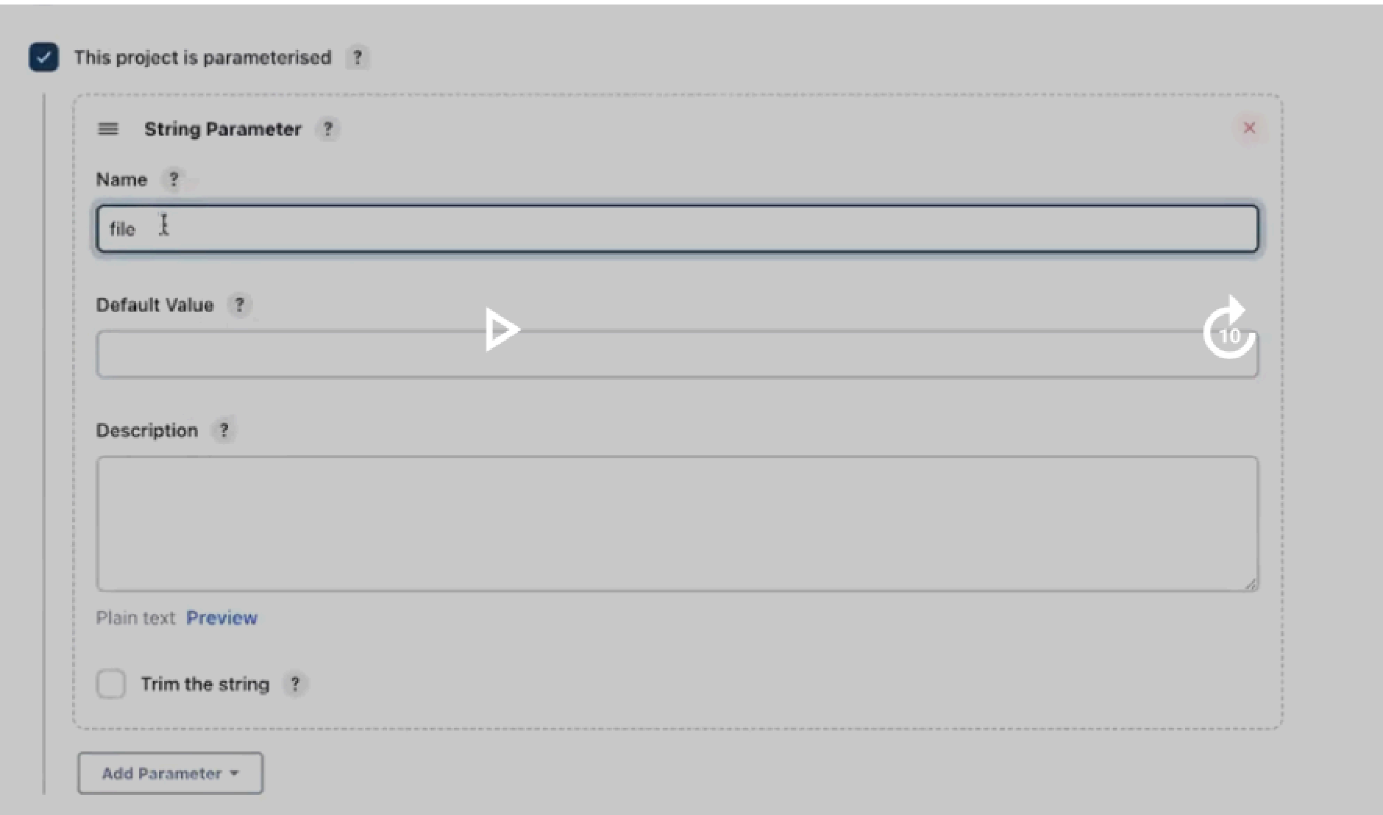
Now next we have

String Parameter

while build if we want to pass anyinformation we can use for this

If we want perform any action from the jenkins we use excute sheel
touch harish
it will create workspace inside job
ex2: touch satya --> save build-->satya file created
evrytime iam going to cofigure so for that if we use string parameter if we enable

example select string parameter
name : file
default value nothing



☑ This project is parameterised ?

String Parameter ?

Name ?

file

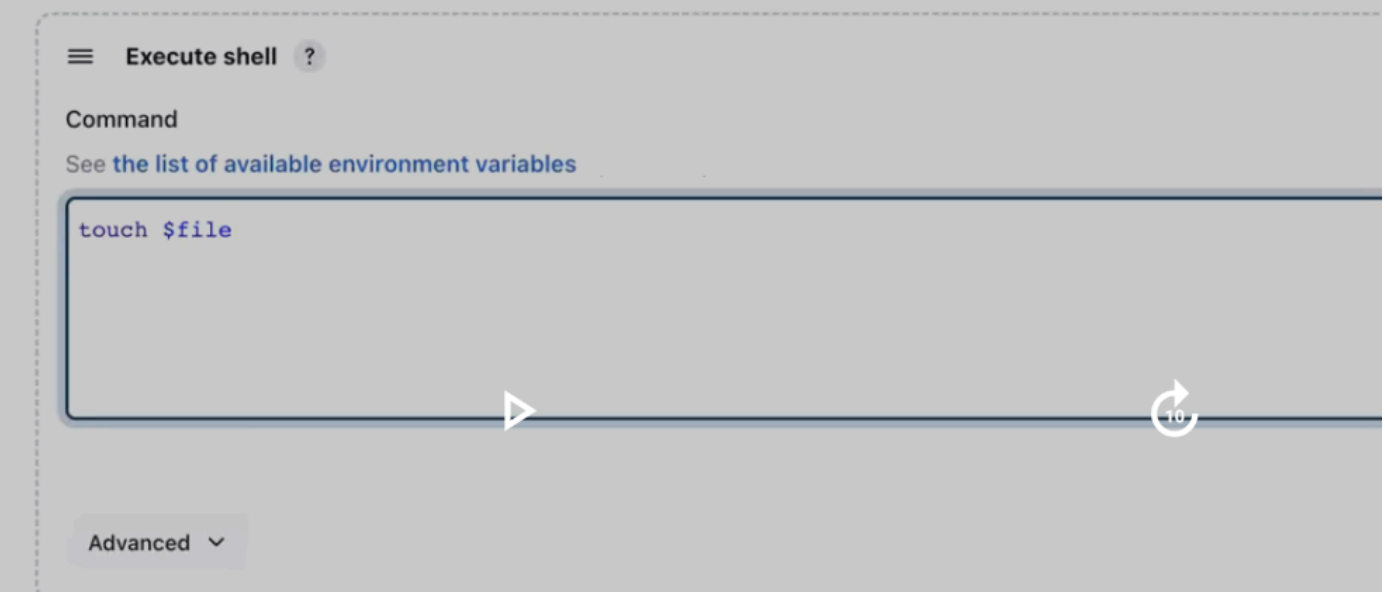
Default Value ?

Description ?

Plain text Preview

□ Trim the string ?

go to excute sheel inside command we have to use
touch $file
varible we taken we declare as $ and note file name and excute shell command same we have to give

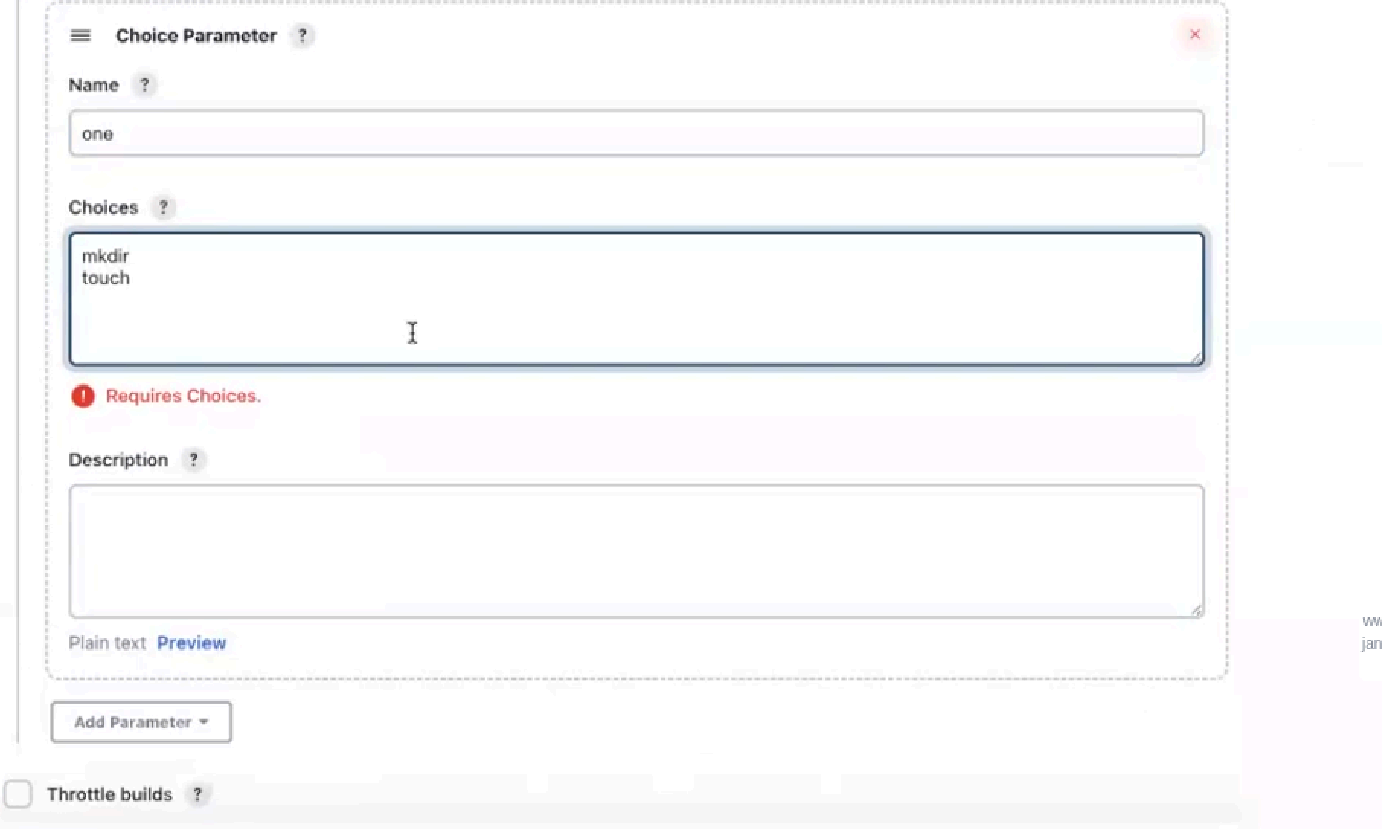


Command
See the list of available environment variables

```
touch $file
```

save and build with parameter
automatically rahul file will create so here everytime we donot need to create a file
ex2: sometimes file and folders needs to create

i select choice parameter as well as previous config
name : one
choices: mkdir ,touch

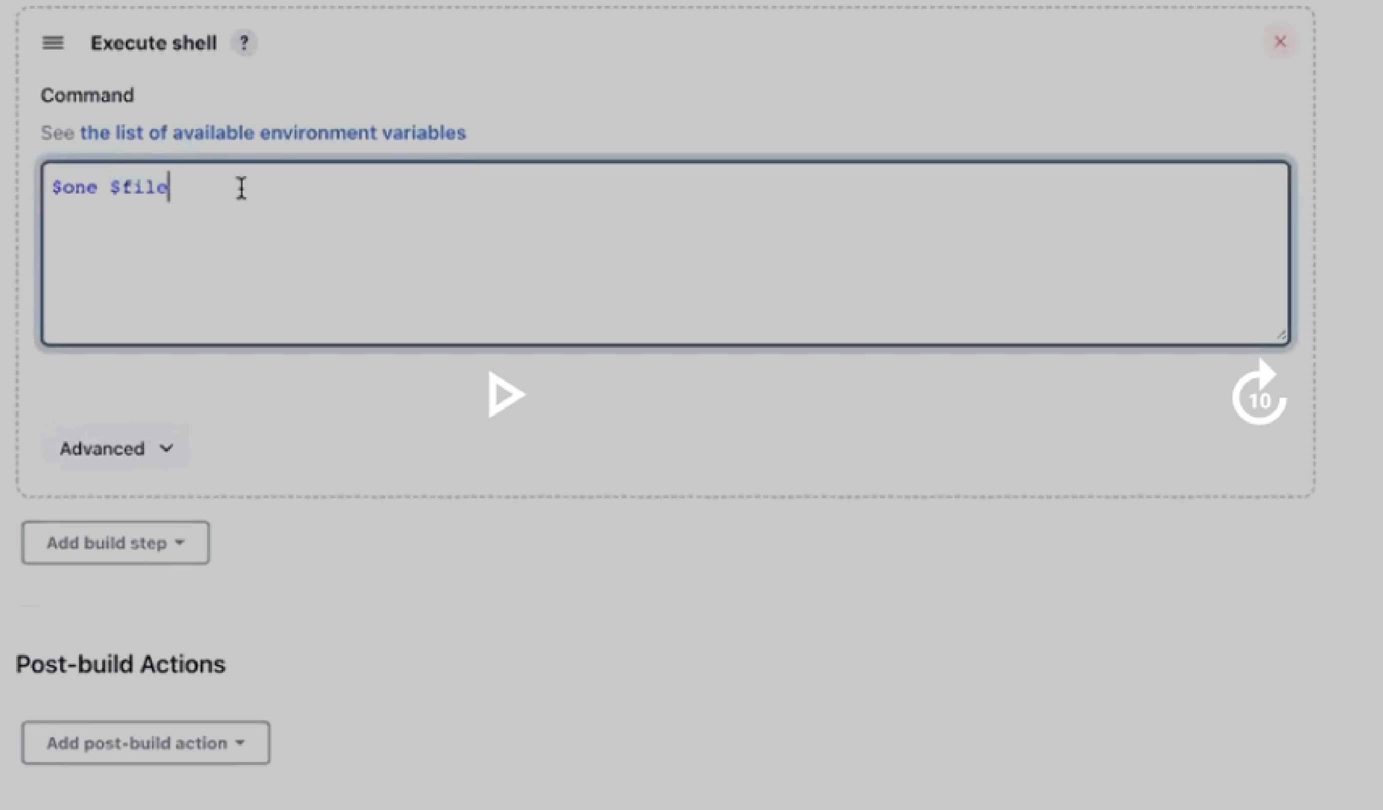


Name ?
one

Choices ?
mkdir
touch

Requires Choices.

Description ?



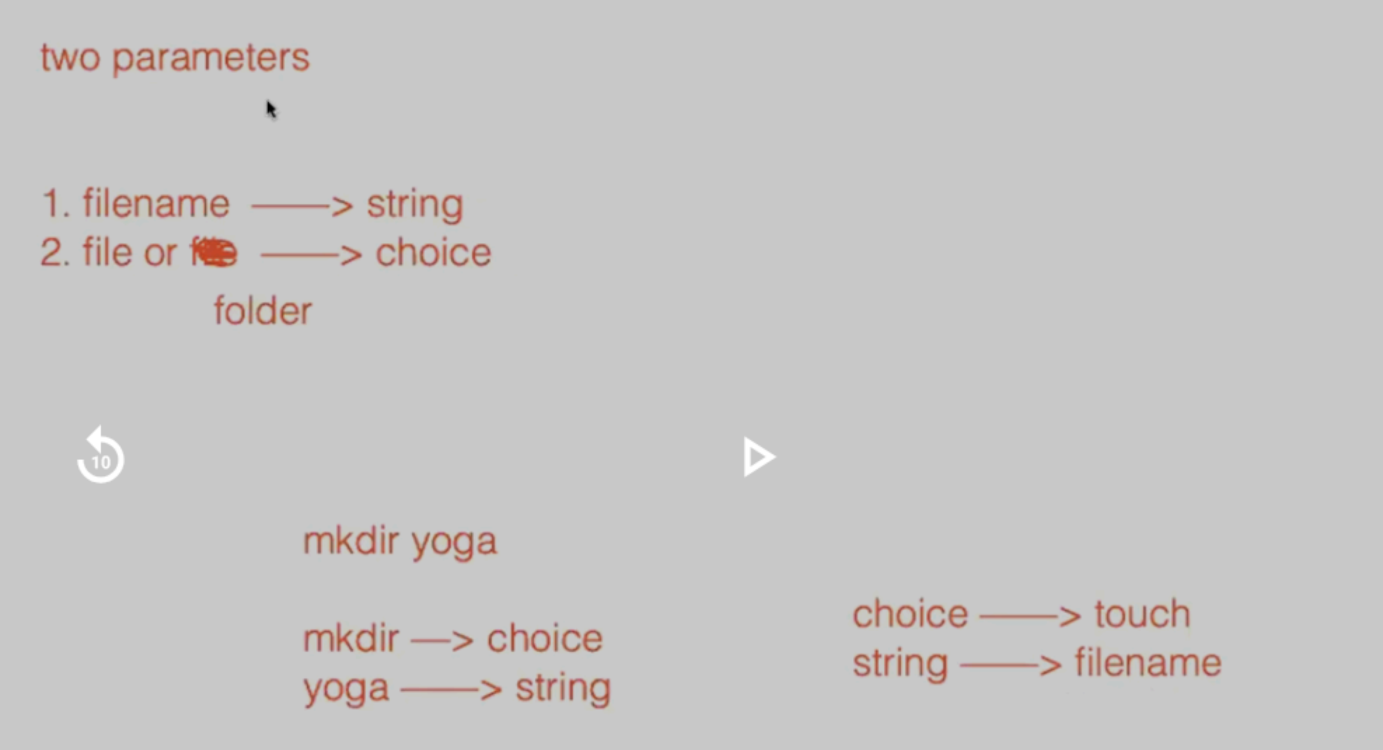
Execute shell

Command

See the list of available environment variables

```
$one $file
```

save and build



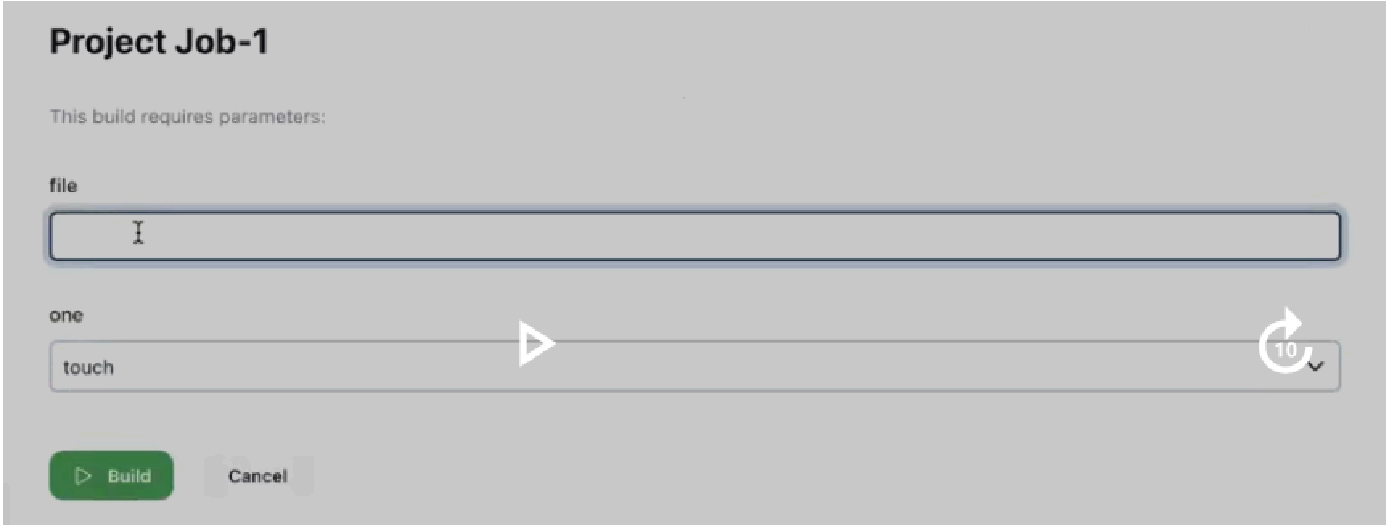
two parameters

1. filename ——> string
2. file or ~~file~~ ——> choice
folder

mkdir yoga

mkdir —> choice
yoga ——> string

choice ——> touch
string ——> filename



Project Job-1

This build requires parameters:

file

one

touch

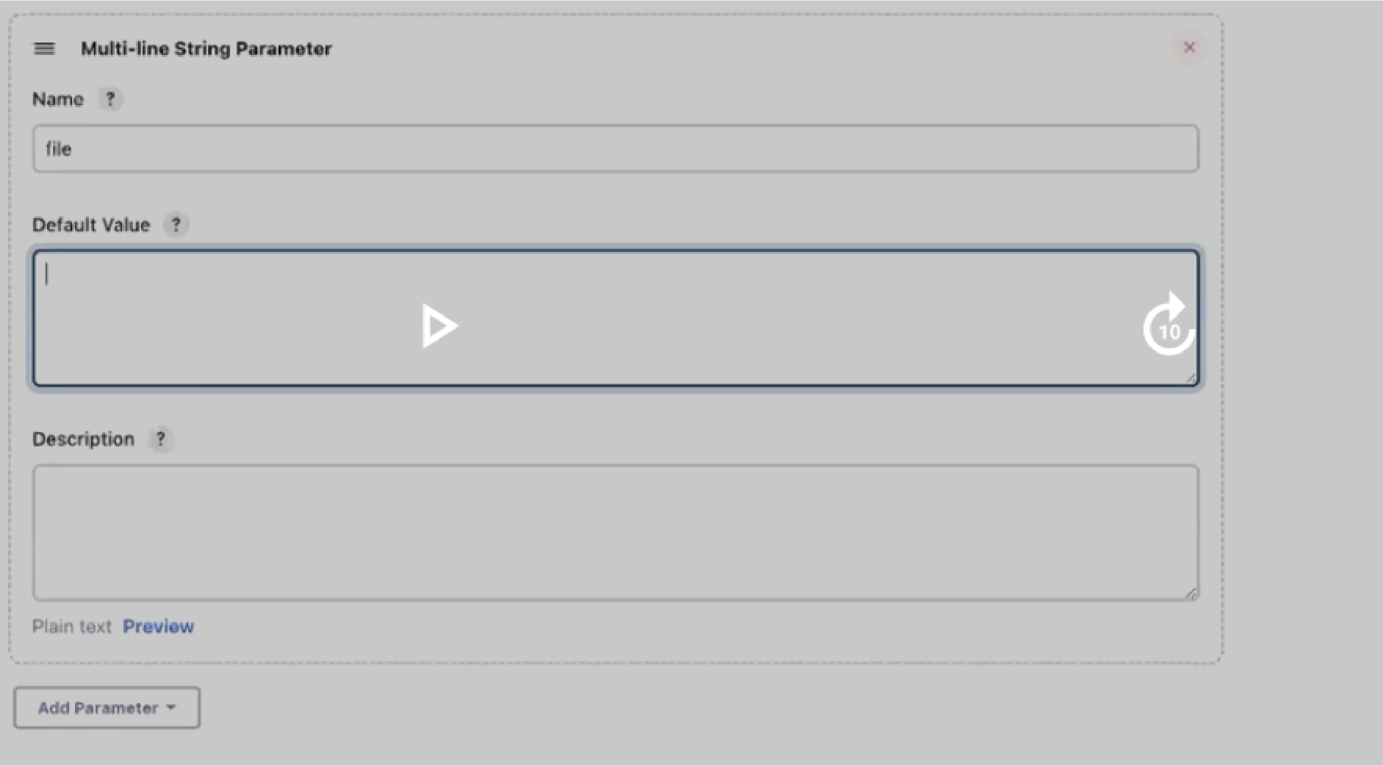
now while build it ask choice i.e name we gaven one and we can select which ever we choice i.e touch/file

now i select one name: i select touch and file name giving anil

Multiline String Parameter

Name: file

Default value nothing



Multi-line String Parameter

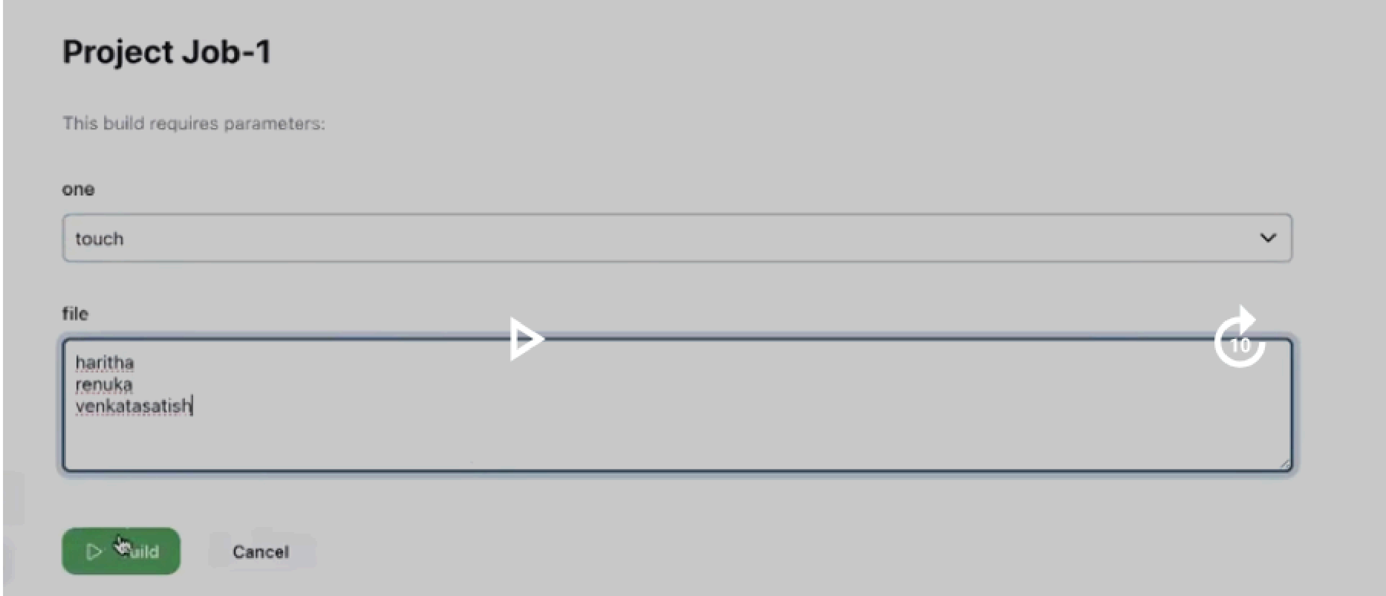
Name ?

file

Default Value ?

Description ?

Bow build with paramter



Project Job-1

This build requires parameters:

one

touch

file

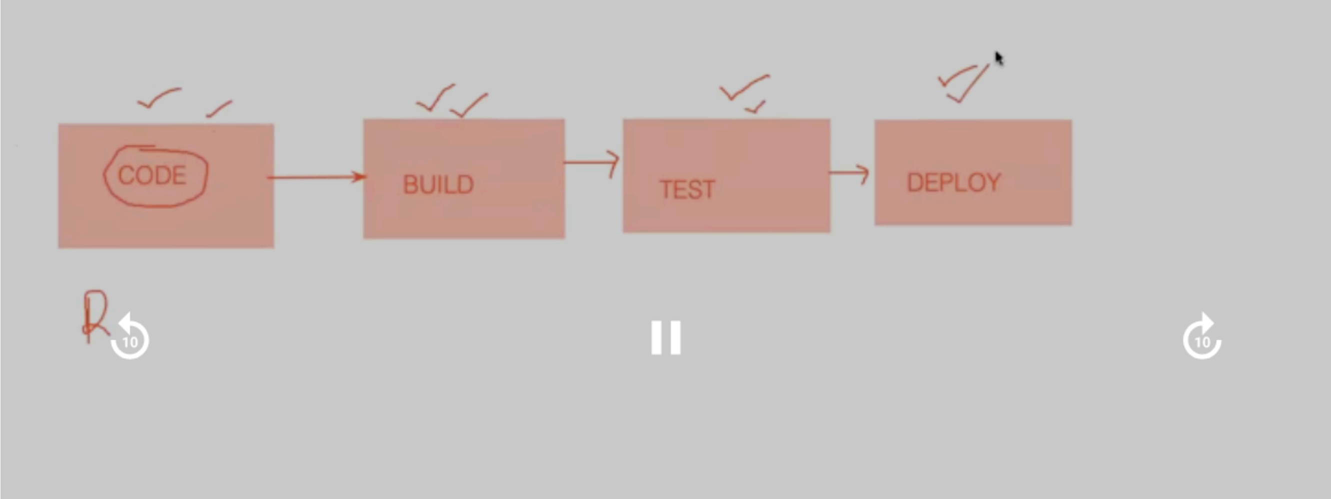
haritha
renuka
venkatasatish

now if we build it will create multiplt files
single file we can select string paramter

Now we have to come section Build Triggers:

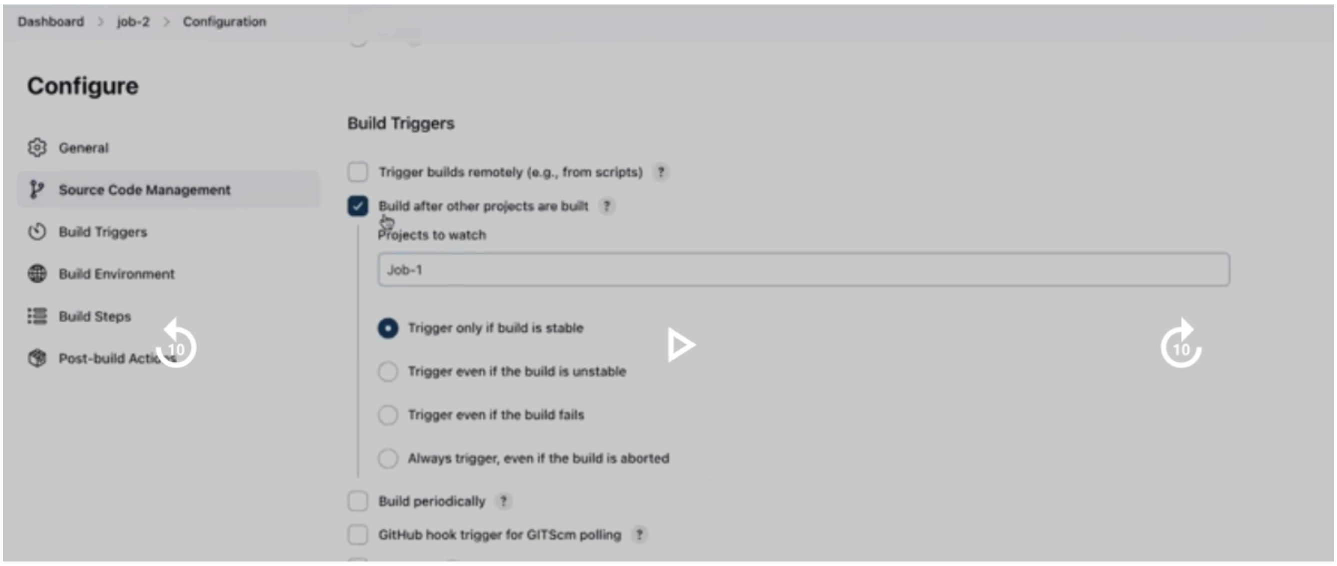
Build Triggers
1.Trigger buildremotly(this will explain later)

2. Build after other project build means :pipeline stages-> getting the code from devlopers(ersion)
   Code+build+Test+Deploy
   If build success then only it will test

   Take 4 jobs
   job1->getting code
   job2->build
   job3->test
   job4->deploy
3. means if code comes then only build happen if build happen then only test next deploy
   select other projects build
   go to job2 here go to build triggers(select build after other projects built) here select job1 which job

completed next to trgiigers job2 now save job1 linked to job2 simmilialru like job2 and job4



Dashboard > job-2 > Configuration

Configure

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

Build Triggers

☐ Trigger builds remotely (e.g., from scripts) ?

☑ Build after other projects are built ?

Projects to watch

Job-1

● Trigger only if build is stable

○ Trigger even if the build is unstable

○ Trigger even if the build fails

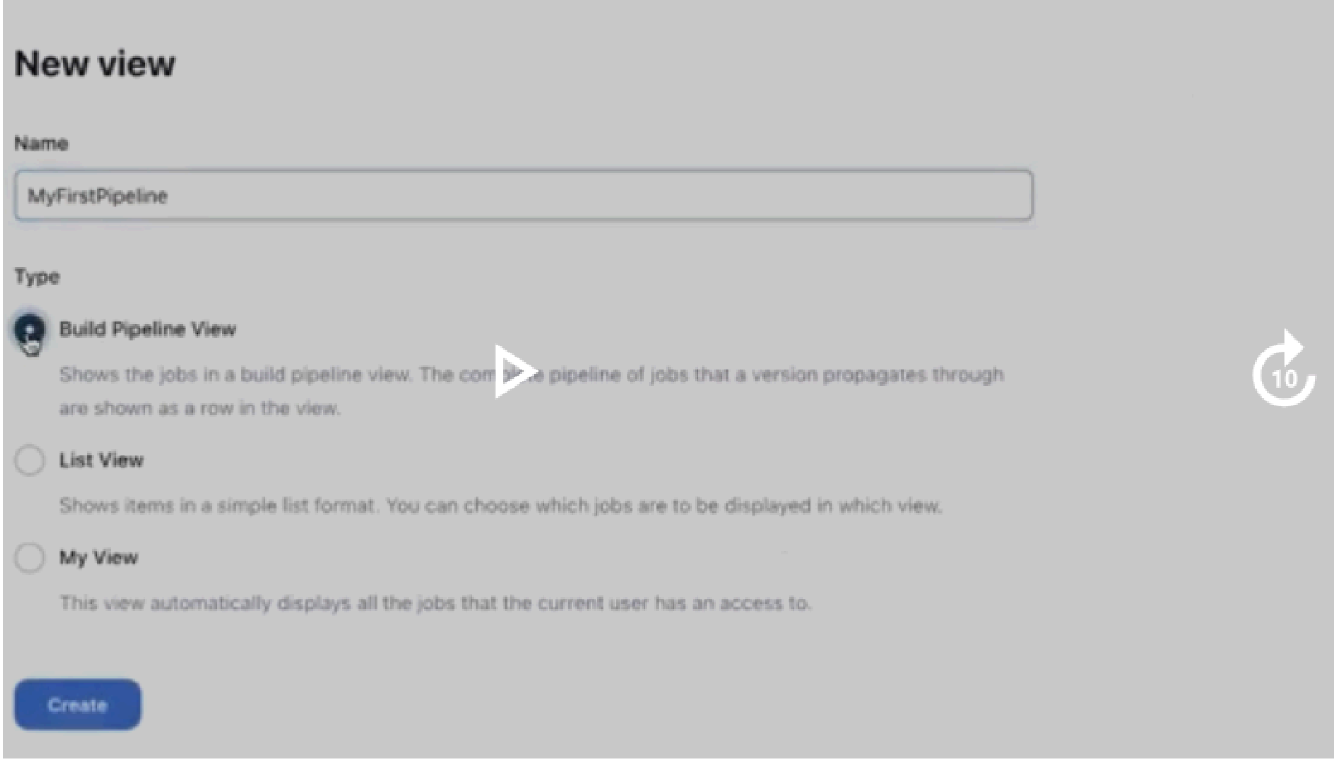
○ Always trigger, even if the build is aborted

☐ Build periodically ?

☐ GitHub hook trigger for GITScm polling ?

so this is Build after other projects build
if we start build job1 next after next it will complete all jobs we can see success status

SO if we want to see step by step process we have to see we can also see at one place one dashboard view we can see
if we want step by step process select + symbol that to be we want pipeline view we want pipeline view by defualt it will not come so we have to install plugin i.e build pipeline view



New view

Name

MyFirstPipeline

Type

● Build Pipeline View

Shows the jobs in a build pipeline view. The complete pipeline of jobs that a version propagates through are shown as a row in the view.

○ List View

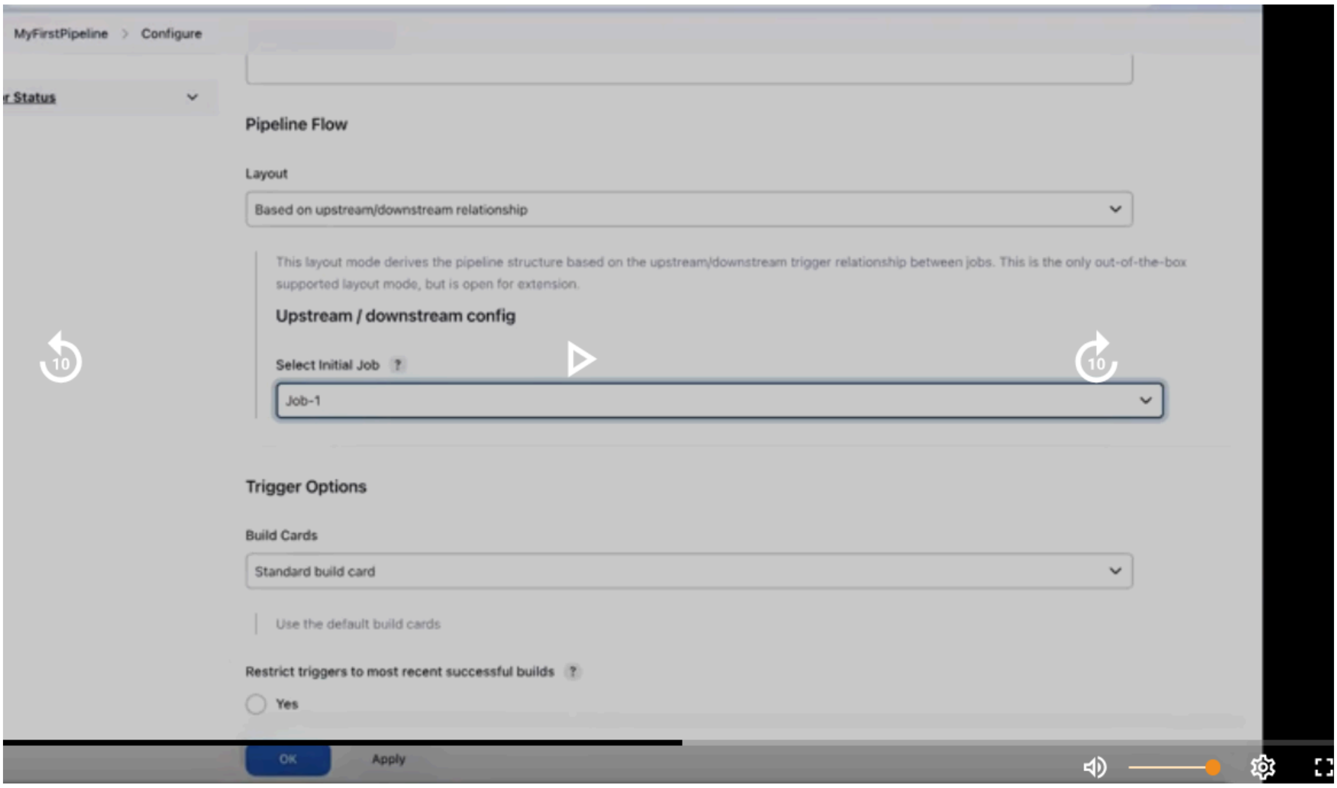
Shows items in a simple list format. You can choose which jobs are to be displayed in which view.

○ My View

This view automatically displays all the jobs that the current user has an access to.

Create

create
till now we have trigger job1 now here
we have to select job1 + click on okay so her


MyFirstPipeline > Configure

Pipeline Flow

Layout: Based on upstream/downstream relationship

This layout mode derives the pipeline structure based on the upstream/downstream trigger relationship between jobs. This is the only out-of-the-box supported layout mode, but is open for extension.

Upstream / downstream config

Select Initial Job: Job-1

Trigger Options

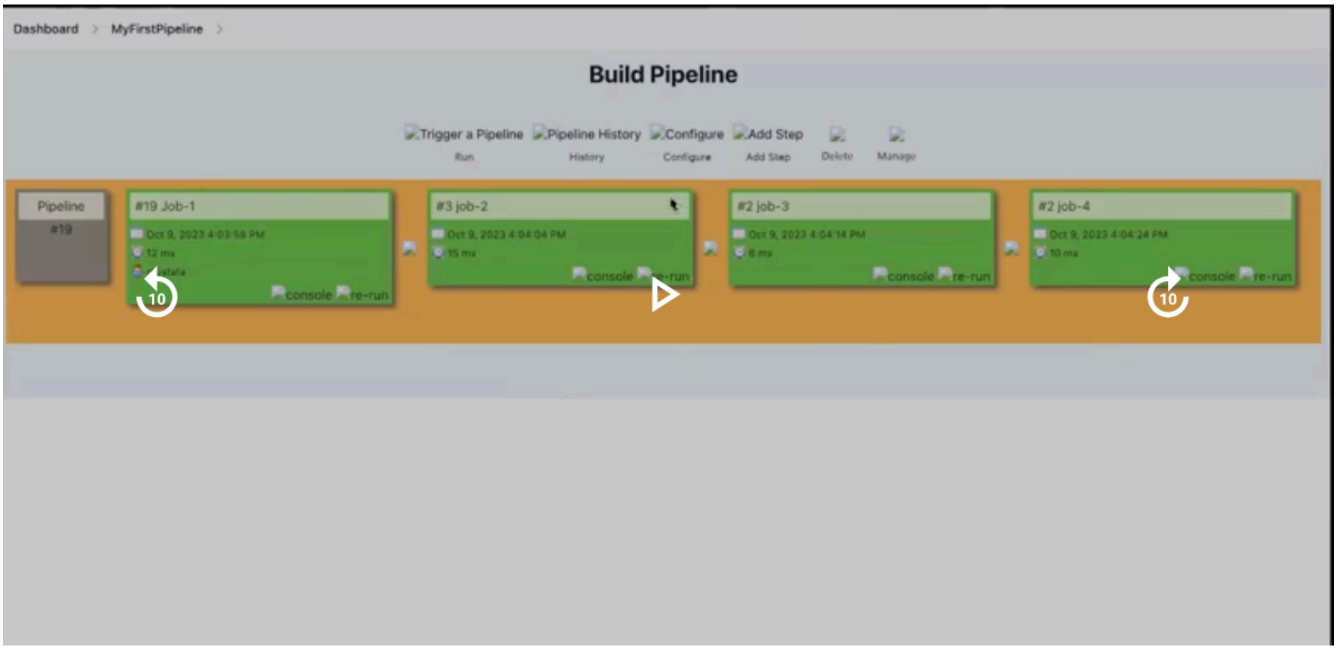
Build Cards: Standard build card

Use the default build cards

Restrict triggers to most recent successful builds

○ Yes

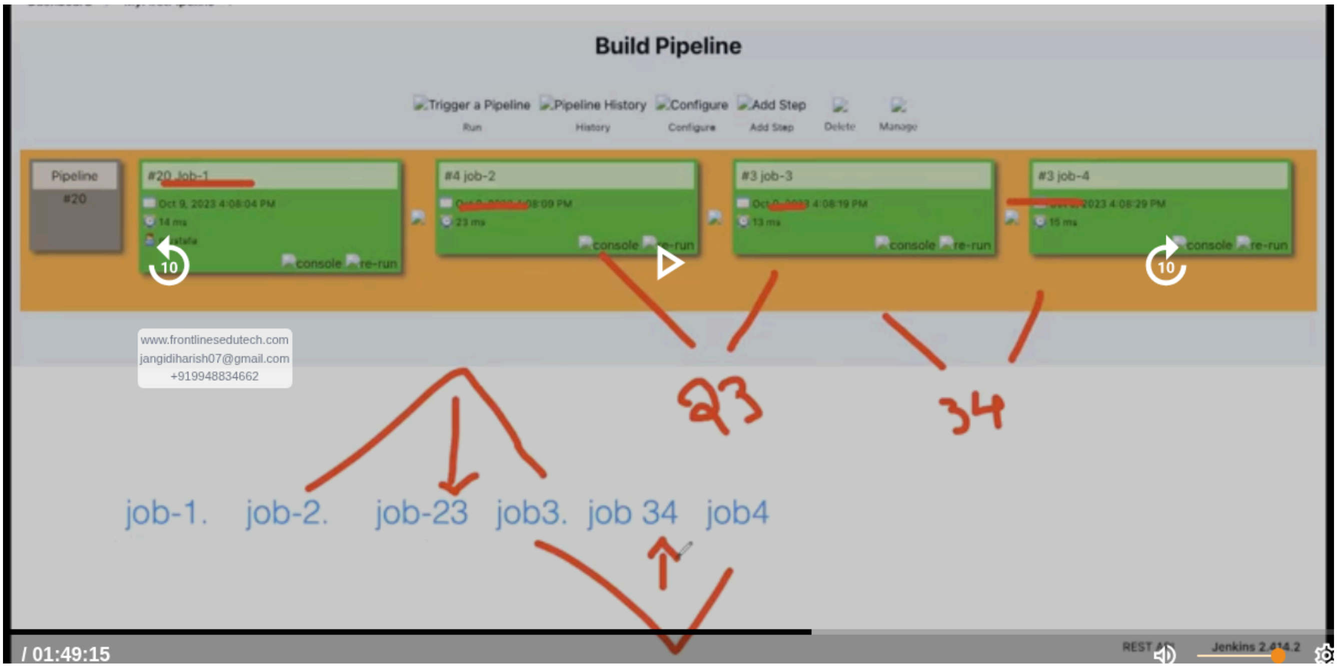
so in final we can see like this we have to give only job1 thats it it will show all jobs in build pipeline what we have integrate internall


Dashboard > MyFirstPipeline >

Build Pipeline

Pipeline #19	#19 Job-1	#3 job-2	#2 job-3	#2 job-4
	Oct 9, 2023 4:03:58 PM 12 ms	Oct 9, 2023 4:04:04 PM 15 ms	Oct 9, 2023 4:04:14 PM 6 ms	Oct 9, 2023 4:04:24 PM 10 ms

if we deploy here job1 it will be in queue process and we can also see history also



this is how work to do b/t job2and 3 create 23 as well as b/w job3and4 create job34

- **job-2 + job-3 → job-23**
- **job-3 + job-4 → job-34**

✅ FREESTYLE JOB SETUP (STEP-BY-STEP)

🔹 Jobs to create

You will have **5 freestyle jobs**:

1. `job-2`
2. `job-3`
3. `job-4`
4. `job-23`
5. `job-34`

🔹 STEP 1: Configure `job-23`

`job-23` should run **ONLY when job-2 AND job-3 succeed**

🔧 job-23 configuration

1. Open **job-23**
2. Click **Configure**
3. Go to **Build Triggers**
4. ✅ Check **Build after other projects are built**

5. In **Projects to watch**:

   `job-2, job-3`

6. In **Trigger only if build is stable** → ✅ checked

7. Save ✅

📌 This means:

- Jenkins waits for **both job-2 and job-3**
- If both are **SUCCESS**, job-23 runs

🔹 STEP 2: Configure `job-34`

`job-34` should run **ONLY when job-3 AND job-4 succeed**

🔧 job-34 configuration

1.
   1. Open **job-34**
   2. Click **Configure**
   3. Go to **Build Triggers**
   4. ✅ Check **Build after other projects are built**
   5. In **Projects to watch**:

      `job-3, job-4`

   6. **Trigger only if build is stable** → ✅ checked
   7. Save

🔹 STEP 3: Trigger base jobs (optional)

If you want **job-2, job-3, job-4** to start together:

Option A: Manually trigger them

- Click **Build Now** on each

Option B: Create a parent job (recommended)

Create a freestyle job called `parent-job`

parent-job → Configure

- Build Step → **Execute shell

  **

  curl -X POST http://JENKINS_URL/job/job-2/build

  curl -X POST http://JENKINS_URL/job/job-3/build

```
curl -X POST http://JENKINS_URL/job/job-4/build
```

🔹 FINAL FLOW (WHAT WILL HAPPEN)

```
job-2 ───────┐
         ├──▶ job-23
job-3 ───────┤
         ├──▶ job-34
job-4 ───────┘
```

**✔ job-23 waits for job-2 & job-3
✔ job-34 waits for job-3 & job-4
✔ job-3 is reused in both paths
✔ No pipeline required

Option B (`parent-job`) only starts jobs
**

job-2
job-3
job-4

**That’s it ✔
**

🔹 Who triggers job-23 and job-34 then?

🔧 job-23 configuration

- Trigger: **Build after other projects are built**
- Projects to watch:
  `job-2, job-3`

📌 Jenkins logic:

> “When BOTH job-2 and job-3 finish successfully → run job-23”

🔧 job-34 configuration

- Trigger: **Build after other projects are built**
- Projects to watch:
  `job-3, job-4`

📌 Jenkins logic:

> “When BOTH job-3 and job-4 finish successfully → run job-34”

🔹 FULL FLOW (END-TO-END)

```
parent-job
  |
  ├── job-2 ──┐
  │  ├── job-23
  ├── job-3 ──┼──────────┐
  │  │  └── job-34
  └── job-4 ──┘
```

✔ parent-job only triggers job-2,3,4
✔ job-23 & job-34 are **event-driven**, not manually started
✔ No duplication, no pipeline

🆚 Option A vs Option B (very short)

Option	Who starts job-2/3/4
Option A	**You (click Build Now)**
Option B	**parent-job (automatic)**

✅ When to use Option A

- Testing / learning
- One-time run
- No need for automation

✅ When to use Option B

- Daily deployments
- CI/CD
- One-click execution

🔹 Role of job-1

job-1 = parent / starter job

It **starts job-2, job-3, job-4**
It does **NOT** directly trigger job-23 or job-34.

✅ Final Job List (Freestyle)

You will have **6 jobs**:

1. **job-1** ✅ (starter)

2. job-2
3. job-3
4. job-4
5. job-23
6. job-34

🔧 Configure job-1 (IMPORTANT)

job-1 → Configure

1️⃣ Build Triggers

- (Optional) SCM / Schedule / Manual
- Or just click **Build Now**

2️⃣ Build Step

Use **Build other projects**
(or install *Parameterized Trigger Plugin* if needed)

Projects to build:

> job-2, job-3, job-4

✔ Check **Trigger even if the build is unstable** (optional)
✔ Save

📌 This makes job-2,3,4 start **together (parallel)**

**🔁 COMPLETE EXECUTION FLOW
**

```
job-1
  |
  ├── job-2 ──┐
  │  ├── job-23
  ├── job-3 ──┼──────────┐
  │  │  └── job-34
  └── job-4 ──┘
```

**🔍 Who triggers what?
**

Job	Triggered by
job-1	Manual / SCM / Schedule
job-2	job-1
job-3	job-1
job-4	job-1

Job	Triggered by
job-23	job-2 + job-3
job-34	job-3 + job-4

Final quick recap (freestyle, simple):

- **job-1** → you click **Build Now** (or schedule/SCM)
- **job-1** triggers:
  - `job-2`
  - `job-3`
  - `job-4` (in parallel)
- Jenkins automatically triggers:
  - **job-23** → after **job-2 + job-3** succeed
  - **job-34** → after **job-3 + job-4** succeed

You **never** manually run `job-23` or `job-34`.

```
job-1 | ├─ job-2 ─┐ ├─ job-3 ─┼─► job-23 └─ job-4 ─┘ └─► job-34
```

If later you want:

- artifacts sharing
- parameters passing
- failure handling
- or converting this to pipeline