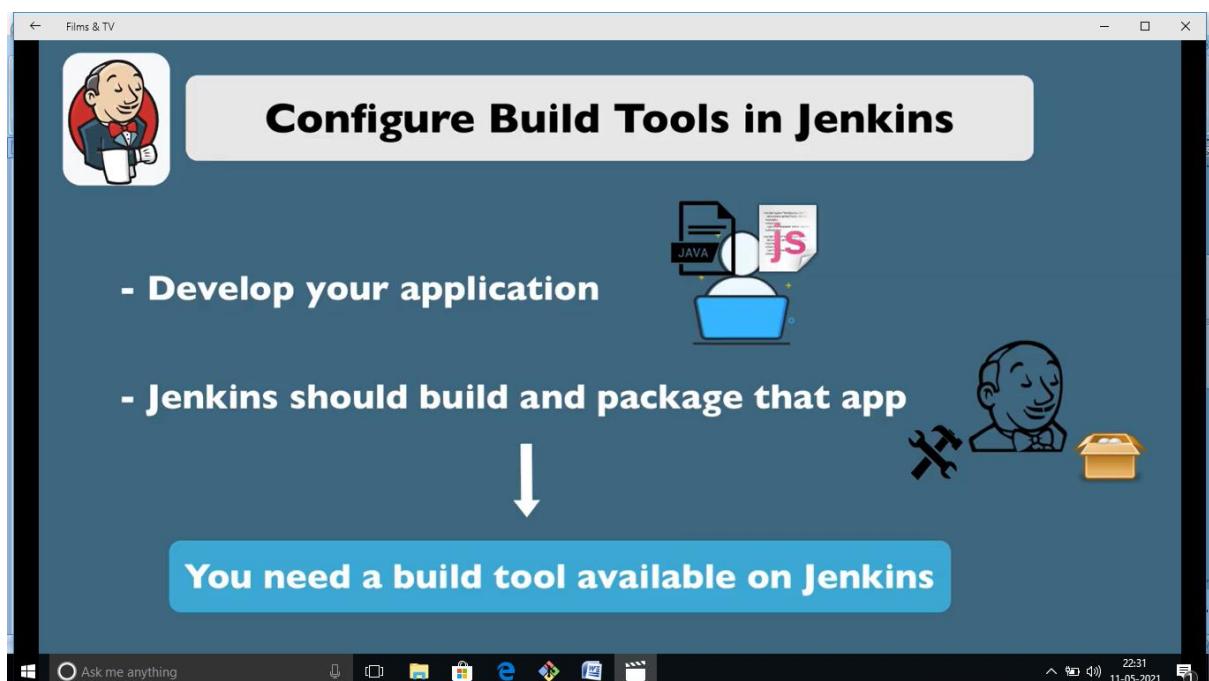
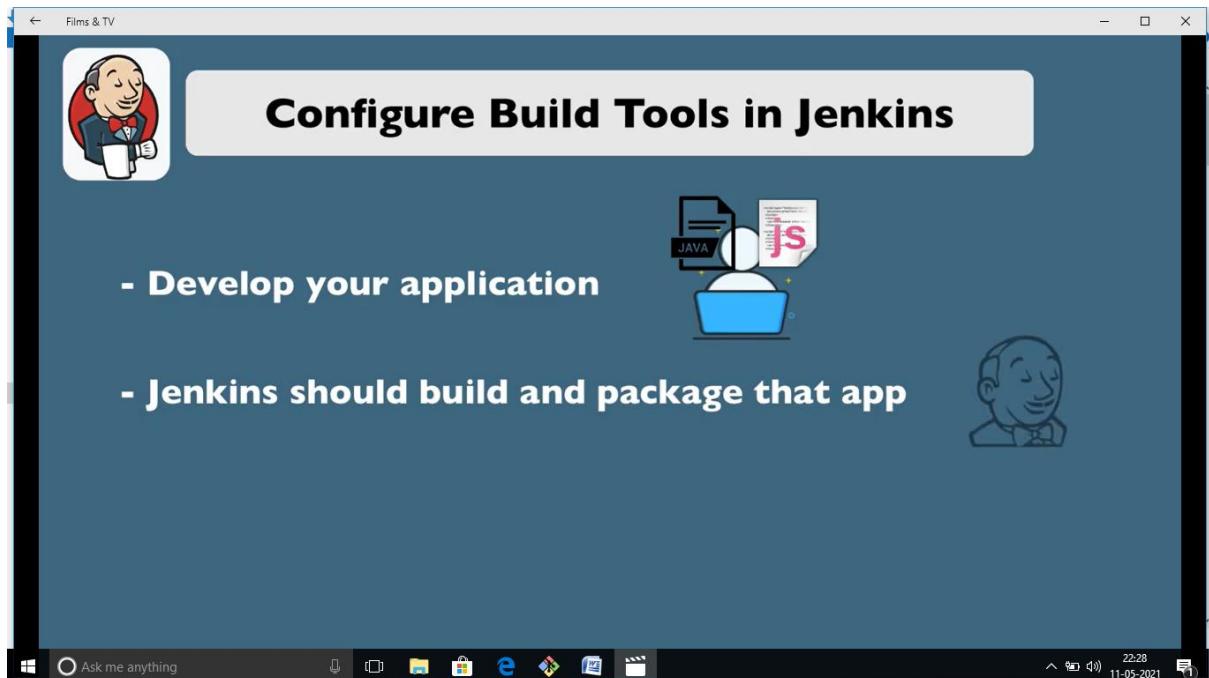
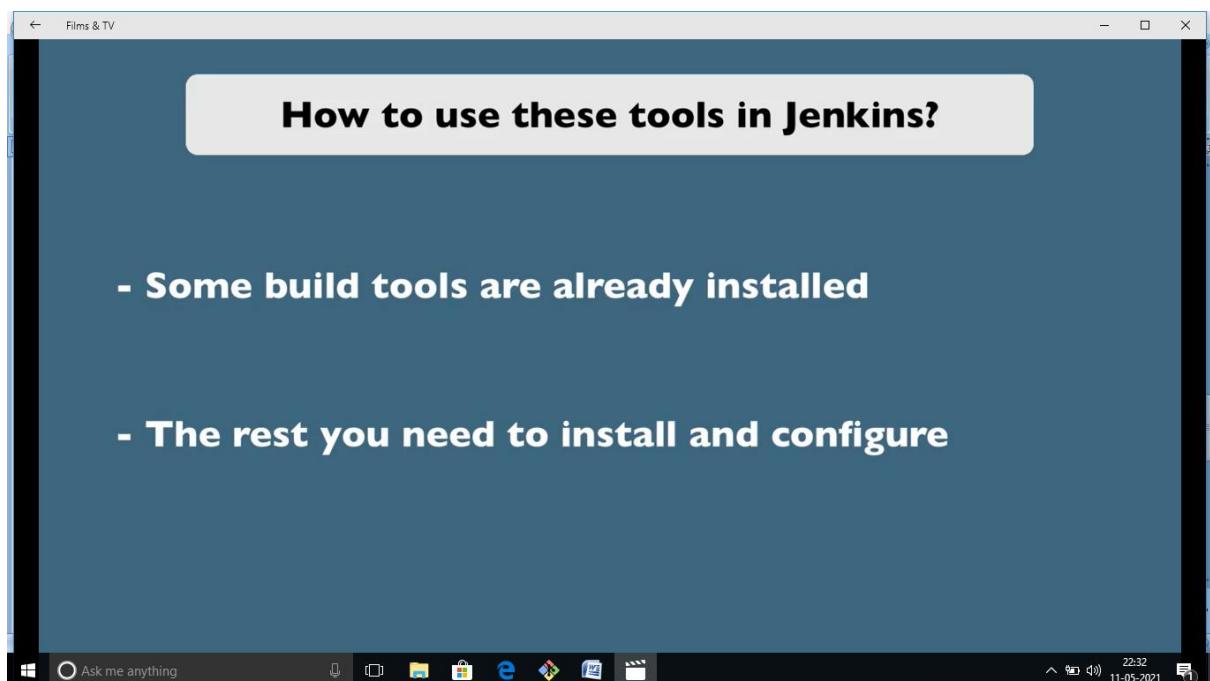
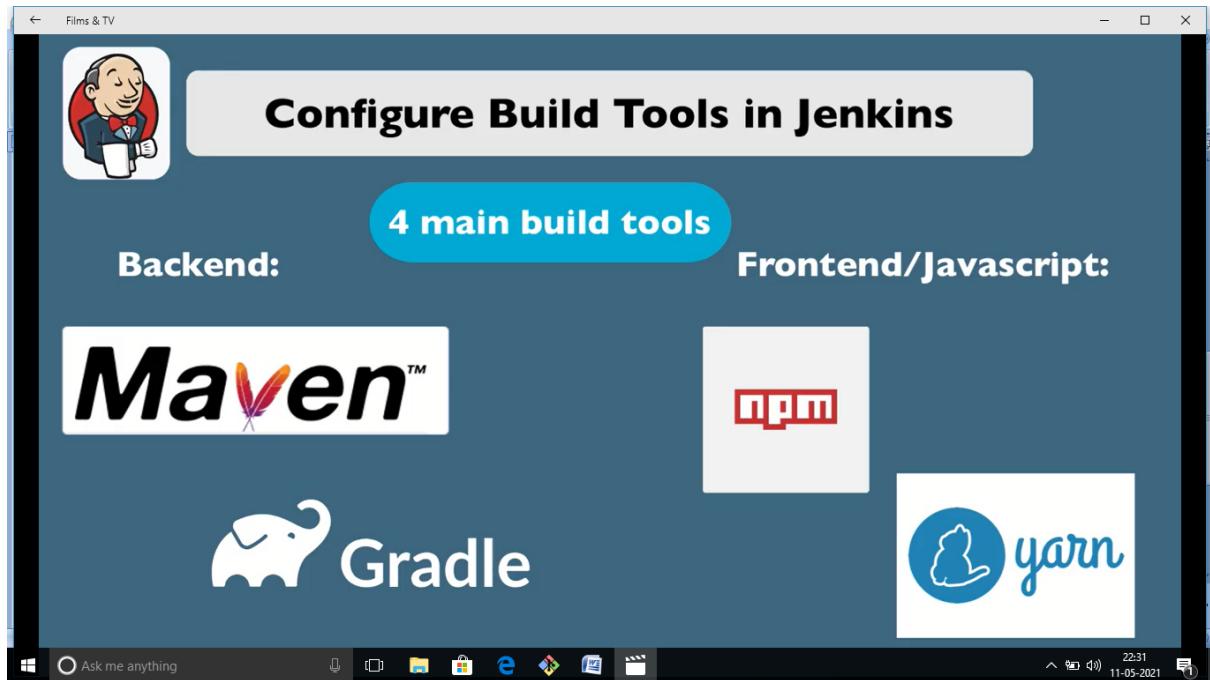
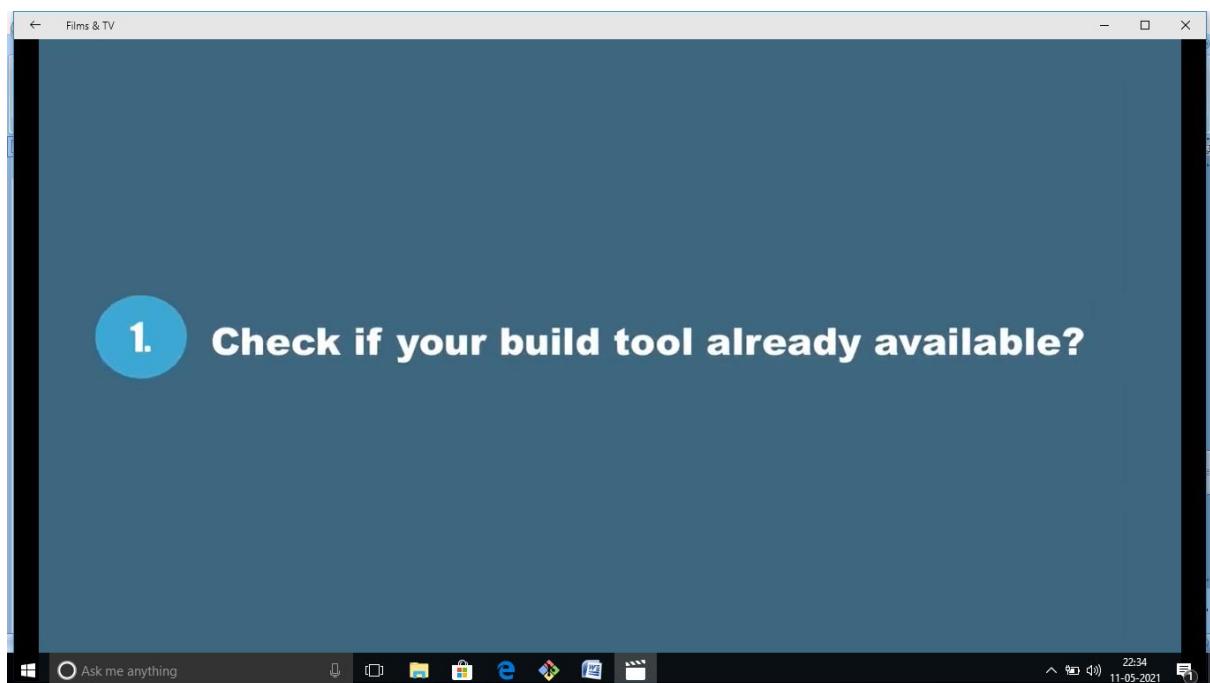
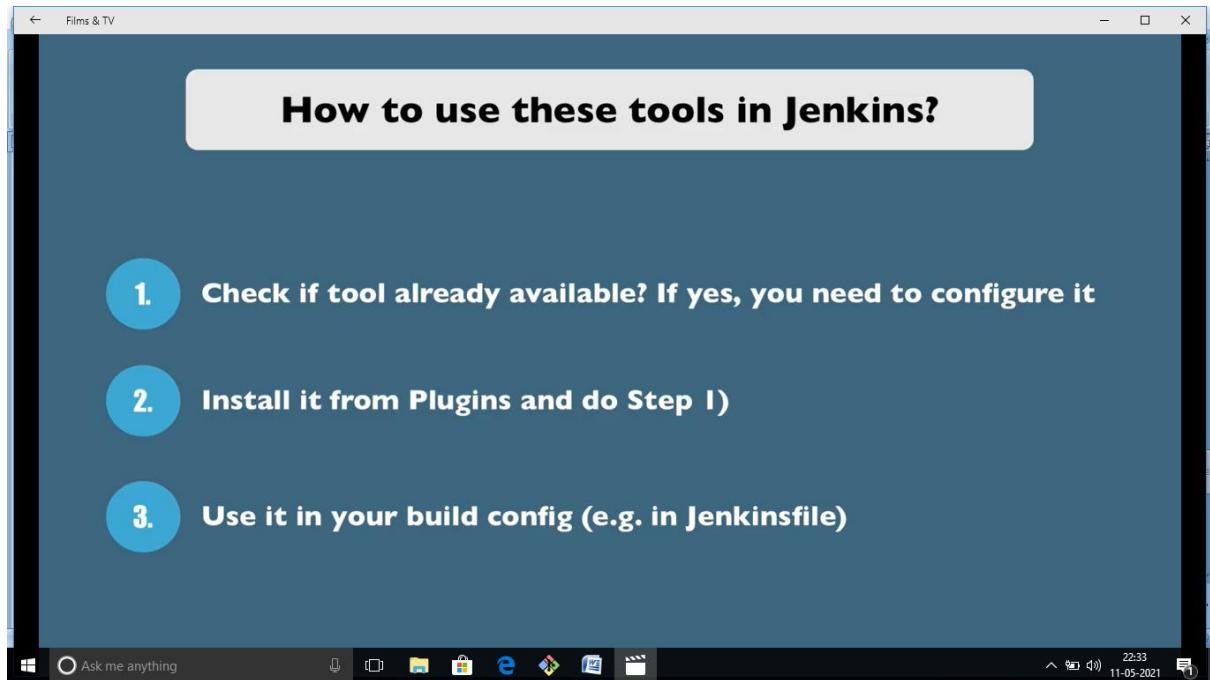
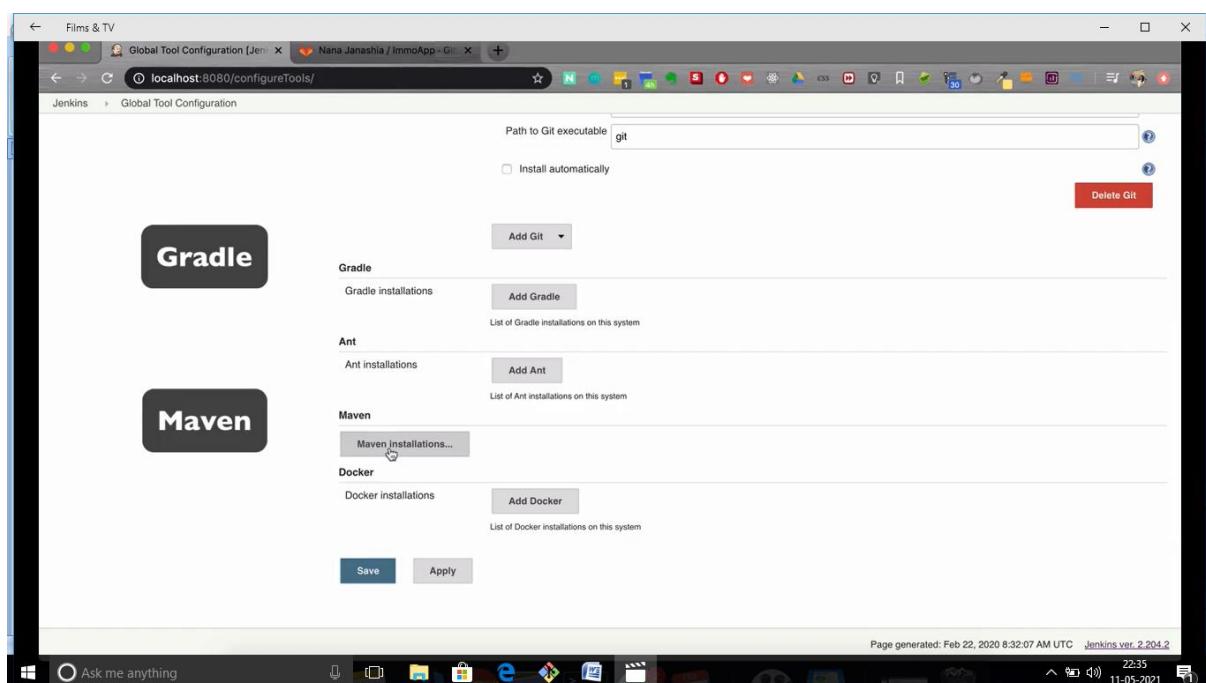
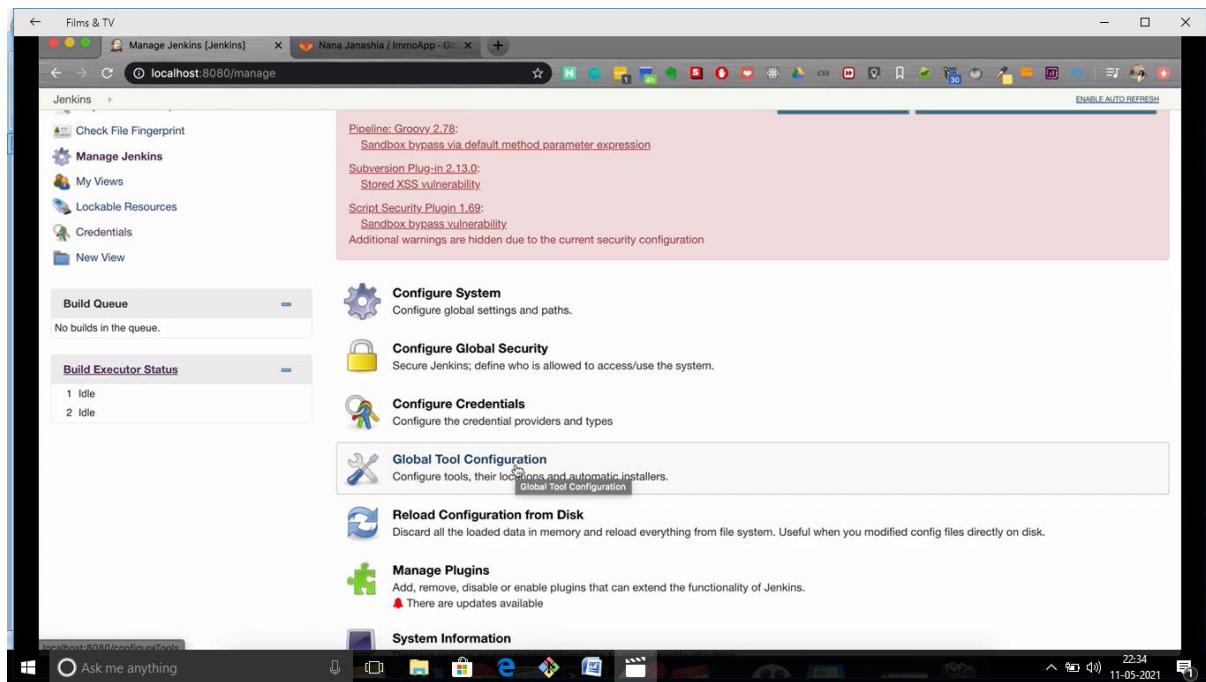


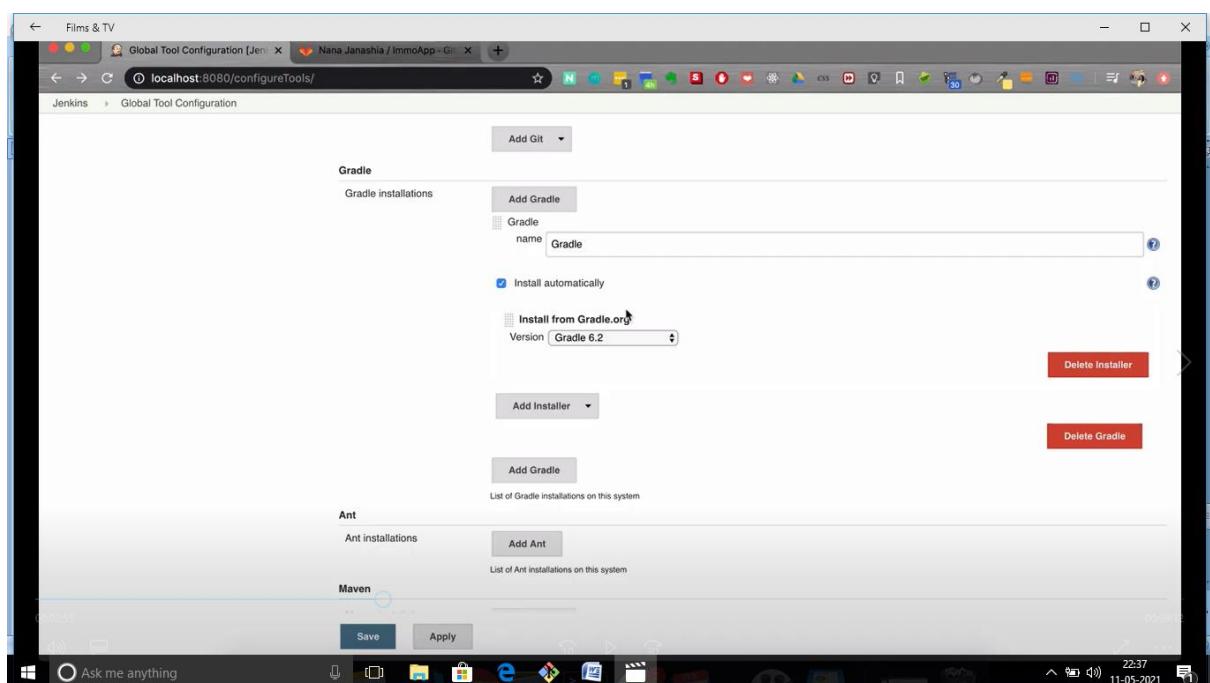
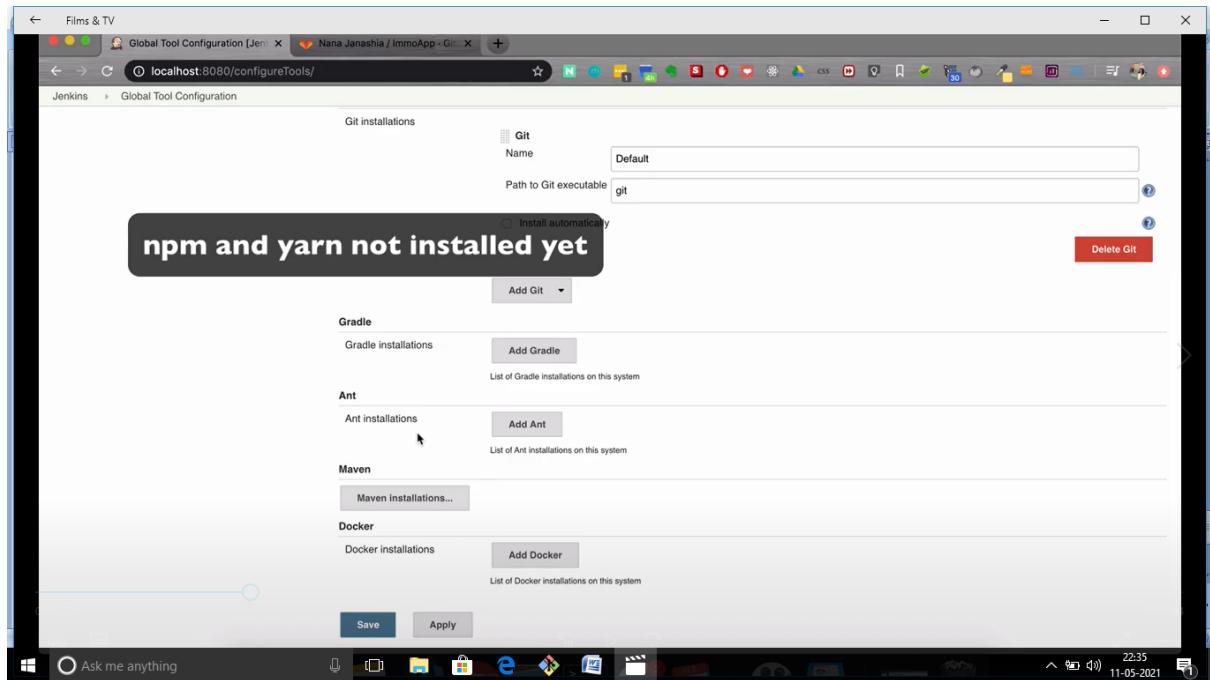
CONFIGURE BUILD TOOLS IN JENKINS

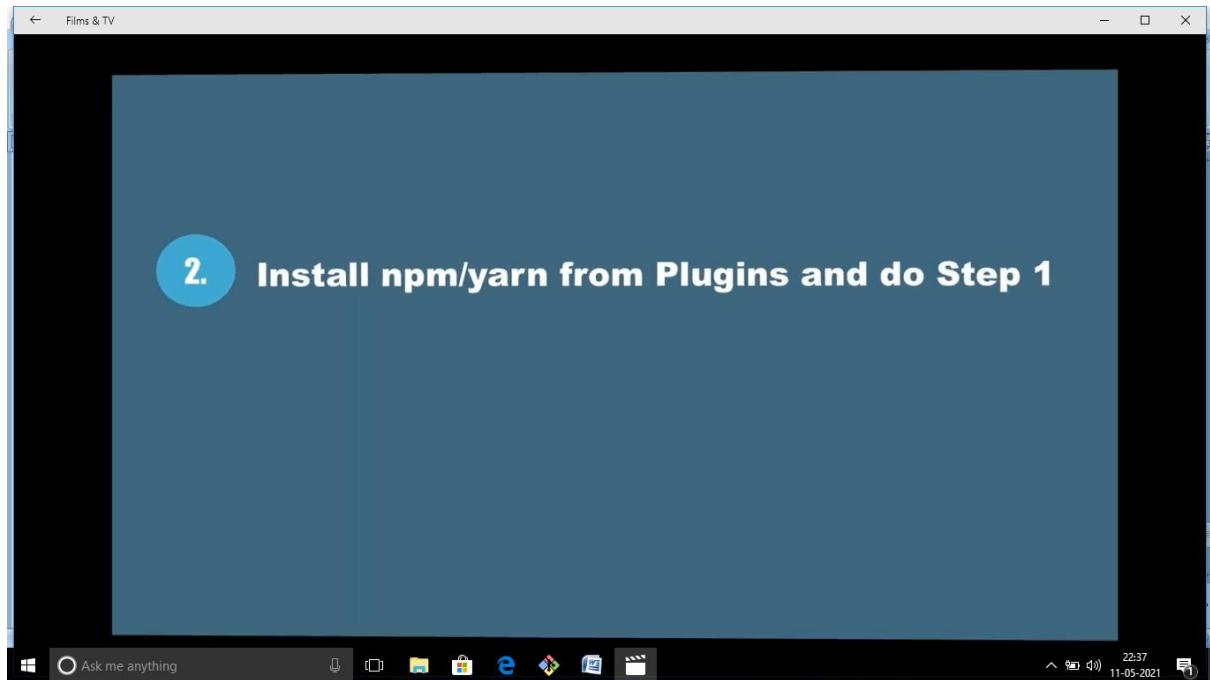




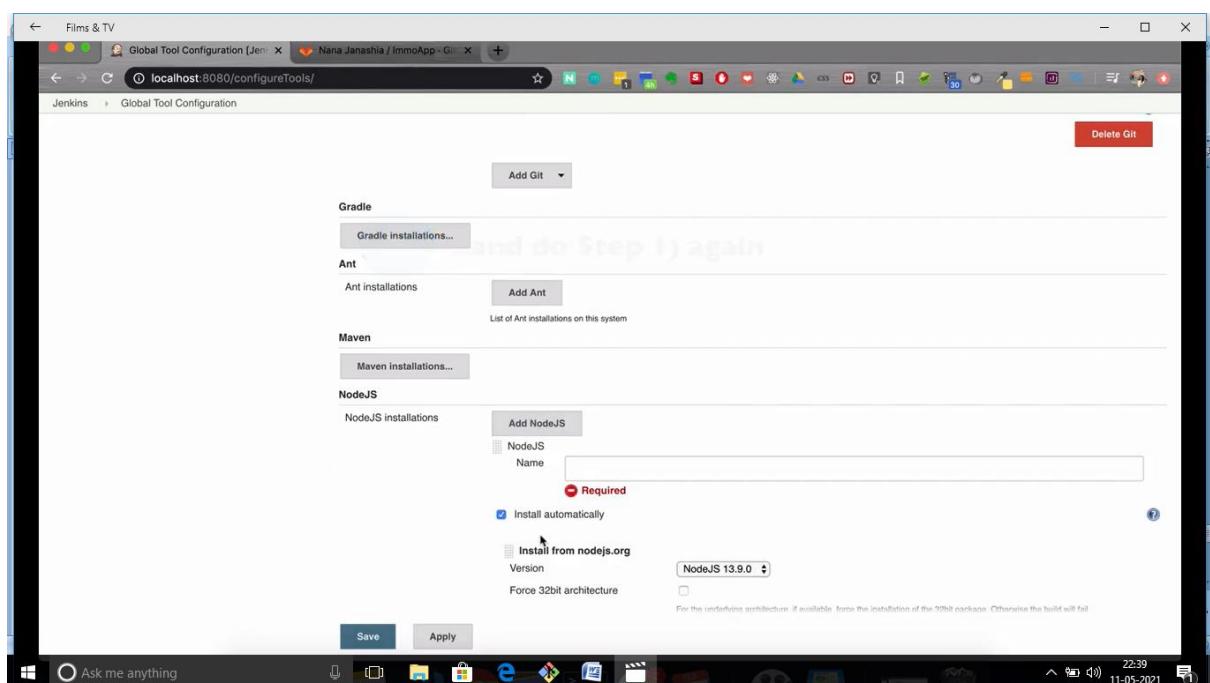
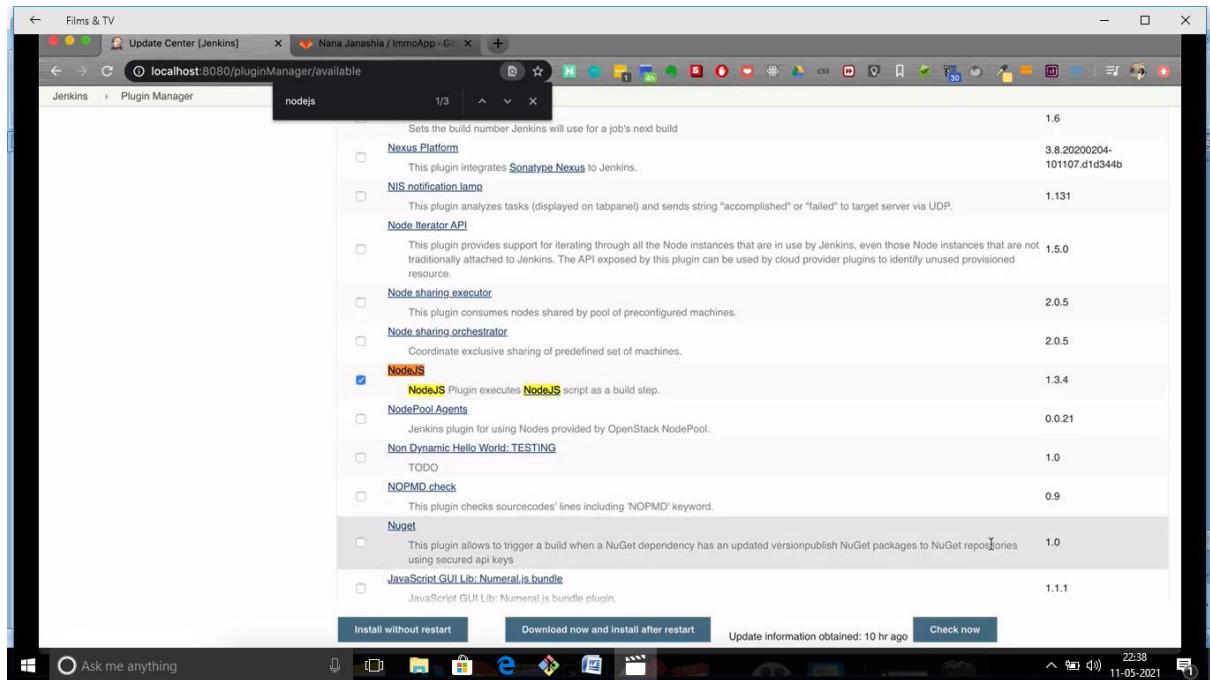


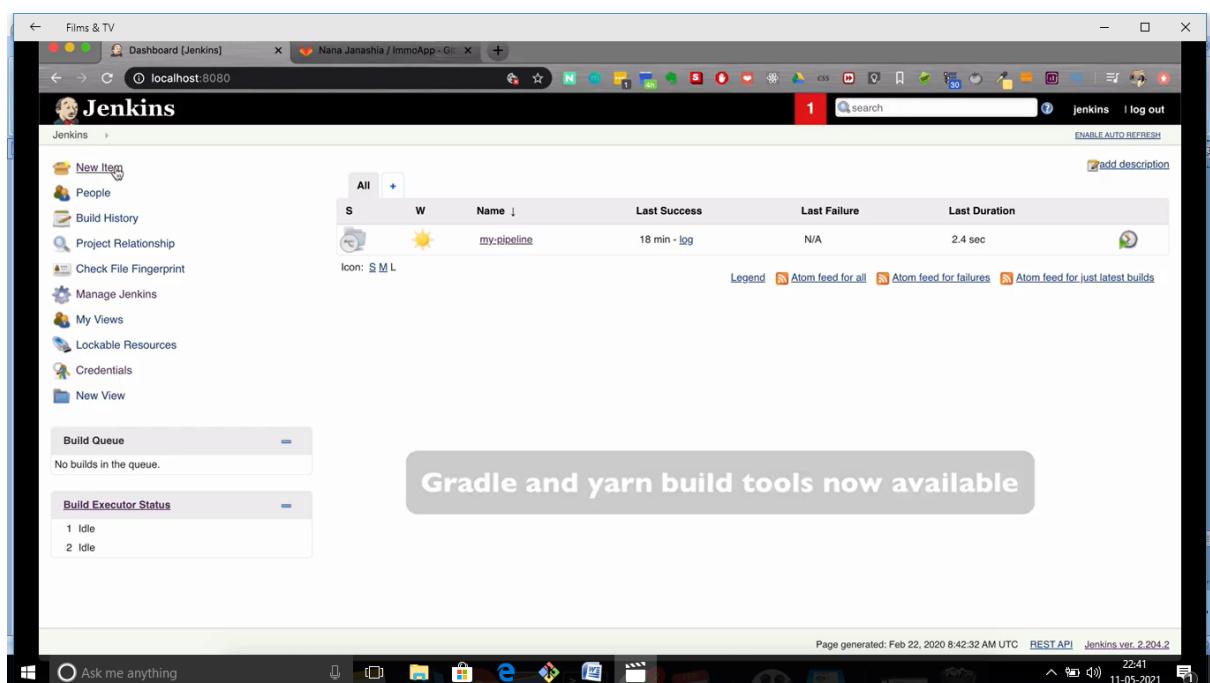
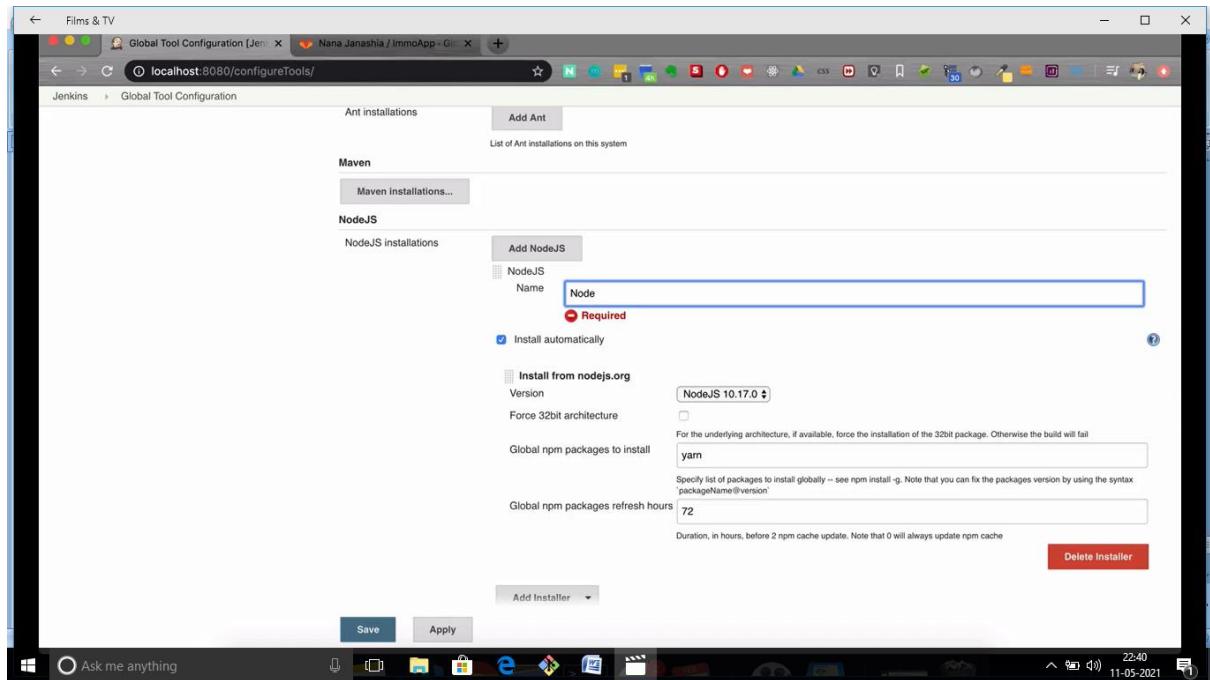


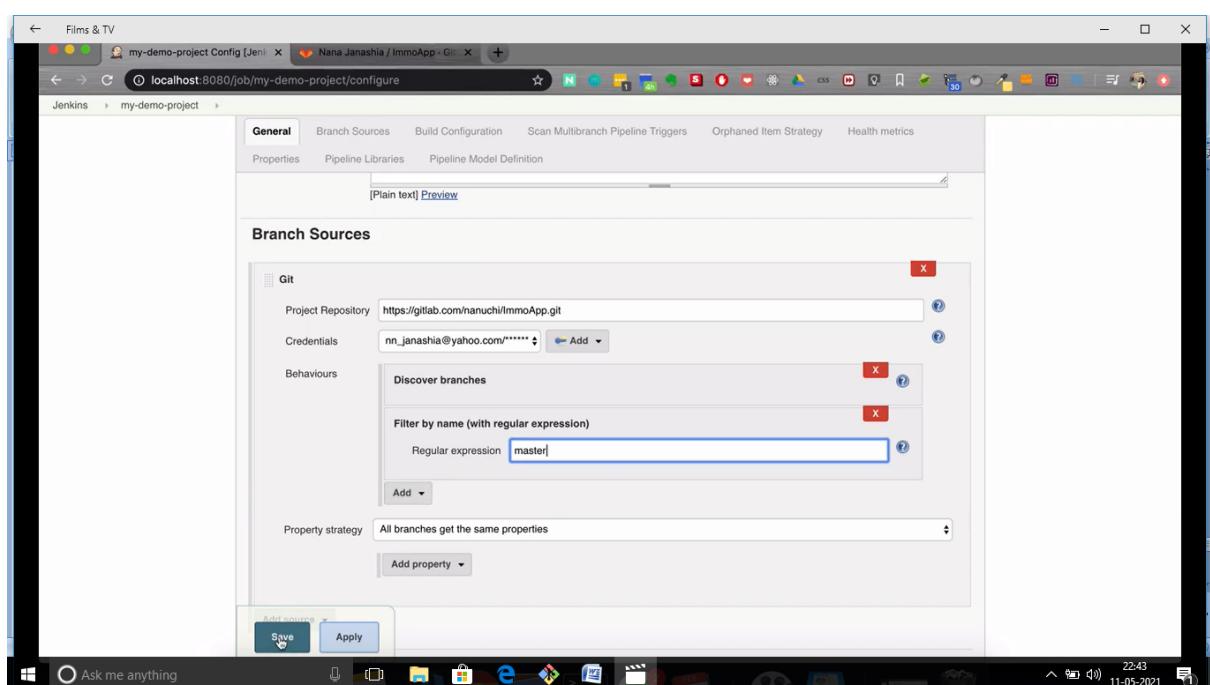
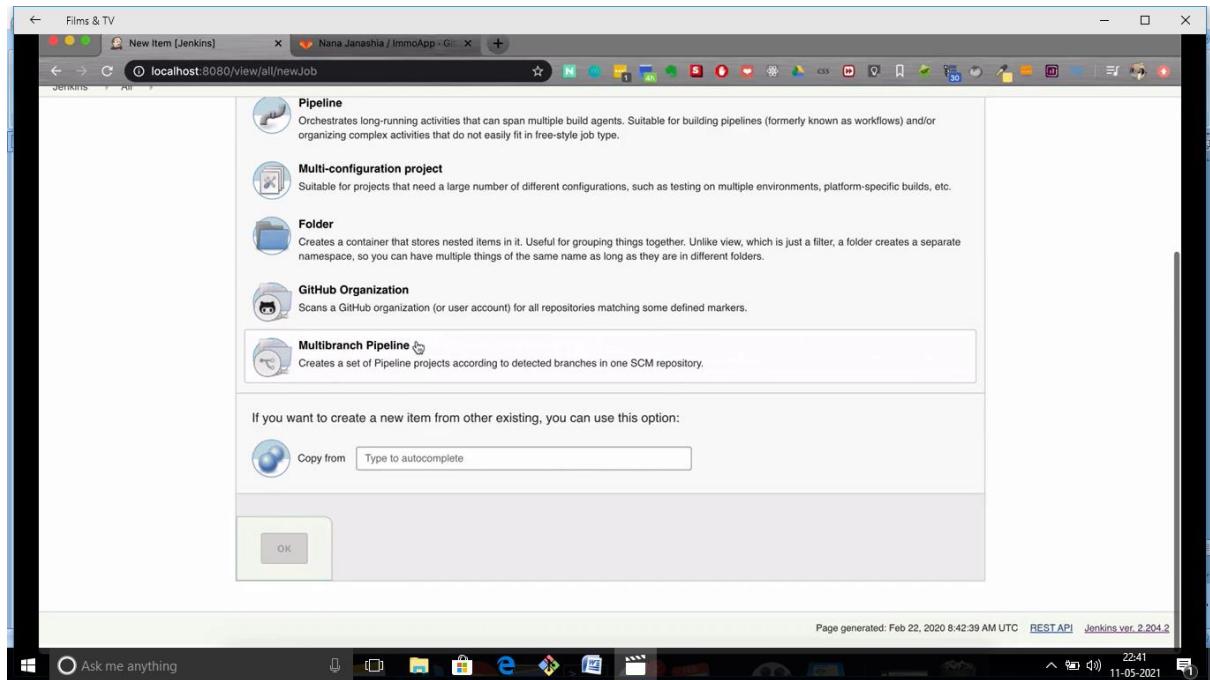




A screenshot of a web browser displaying the Jenkins Plugin Manager. The title bar shows 'localhost:8080/pluginManager/'. The main content area is titled 'Updates' and lists several Jenkins plugins with their names, descriptions, and current versions. A 'Download now and install after restart' button is at the bottom left, and a 'Check now' button is at the bottom right. The browser's address bar shows 'localhost:8080/pluginManager/' and the user 'Nana Janashia / ImmoApp - Gl...'. The status bar at the bottom indicates 'Waiting for localhost' and the date/time '11-05-2021 22:38'.







A screenshot of a Windows desktop environment. In the center is a browser window displaying the Jenkins Multibranch Pipeline Log for a project named 'my-demo-project'. The log output shows the process of indexing branches, including cloning from a GitLab repository and checking for Jenkinsfiles. The Jenkins interface includes a sidebar with various management options like 'Scan Multibranch Pipeline Now' and 'Build History'. The taskbar at the bottom shows several pinned icons, and the system tray indicates the date and time as 11-05-2021 22:43.

```
Started
[Sat Feb 22 08:43:32 UTC 2020] Starting branch indexing...
> git --version # timeout=10
using GIT_ASKPASS to set credentials
> git ls-remote --symref -- https://gitlab.com/nanuchi/ImmoApp.git # timeout=10
> git rev-parse --is-inside-work-tree # timeout=10
Setting origin to https://gitlab.com/nanuchi/ImmoApp.git
> git config remote.origin.url https://gitlab.com/nanuchi/ImmoApp.git # timeout=10
Fetching & pruning origin...
Fetching remote references...
> git config --get remote.origin.url # timeout=10
using GIT_ASKPASS to set credentials
> git ls-remote -h -- https://gitlab.com/nanuchi/ImmoApp.git # timeout=10
Fetching upstream changes from origin
> git config --get remote.origin.url # timeout=10
using GIT_ASKPASS to set credentials
> git fetch --tags --progress --prune --origin +refs/heads/*:refs/remotes/origin/* # timeout=10
Checking branches...
Checking branch master
'Jenkinsfile' not found
Does not meet criteria
Processed 20 branches
[Sat Feb 22 08:43:40 UTC 2020] Finished branch indexing. Indexing took 8.5 sec
Finished: SUCCESS
```

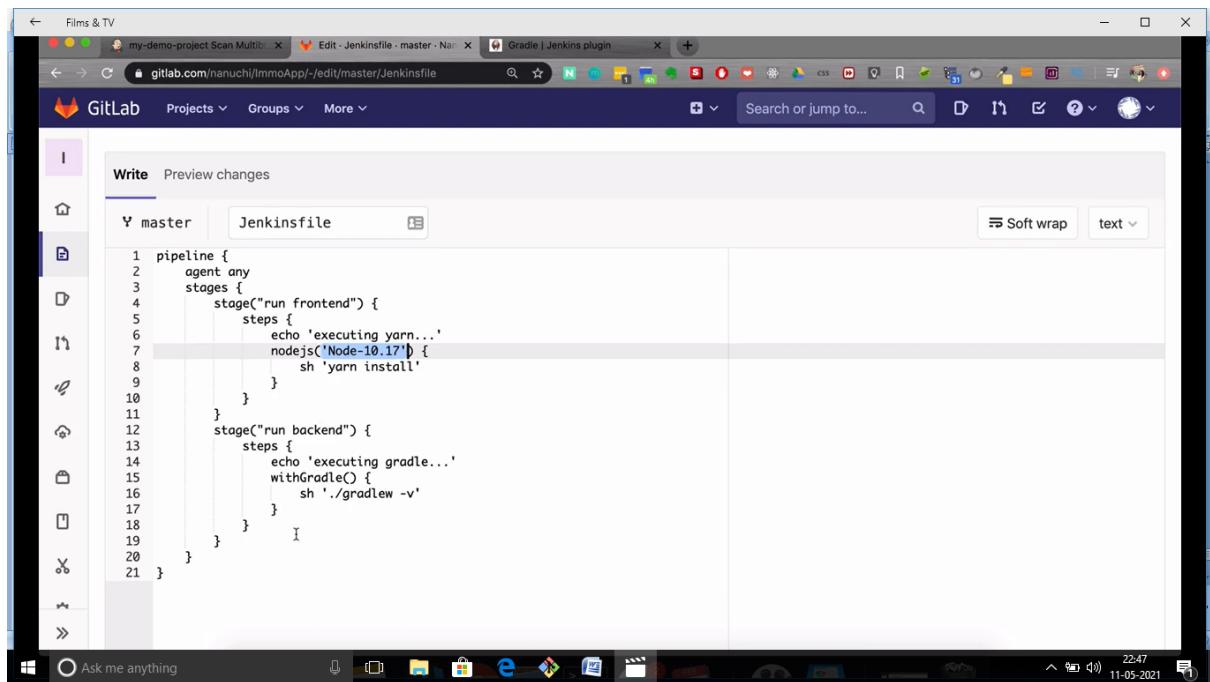
3. Use build tools in Jenkinsfile

A screenshot of a GitLab repository page for the project 'ImmoApp'. The sidebar on the left shows various project management sections like Project overview, Repository, Files, Commits, Branches, Tags, Contributors, Graph, Compare, Charts, Issues, Merge Requests, CI / CD, Operations, Packages, and Wiki. The main area displays a list of files and commits. A context menu is open over a file named 'gradle'. The menu options include 'New file' (which is highlighted), 'Upload file', 'New directory', and 'This repository' which includes 'New branch' and 'New tag'. The commit history table shows several commits from Nana Janashia, with the most recent being a merge branch 'develop' from 8 months ago. Other commits include 'Initial application generated by JHipster-4.13.1' and 'Refactor LoginVM and git ignore out/ folder Add bcrypt dependency...'. The commit table also lists 'Last update' times for each commit.

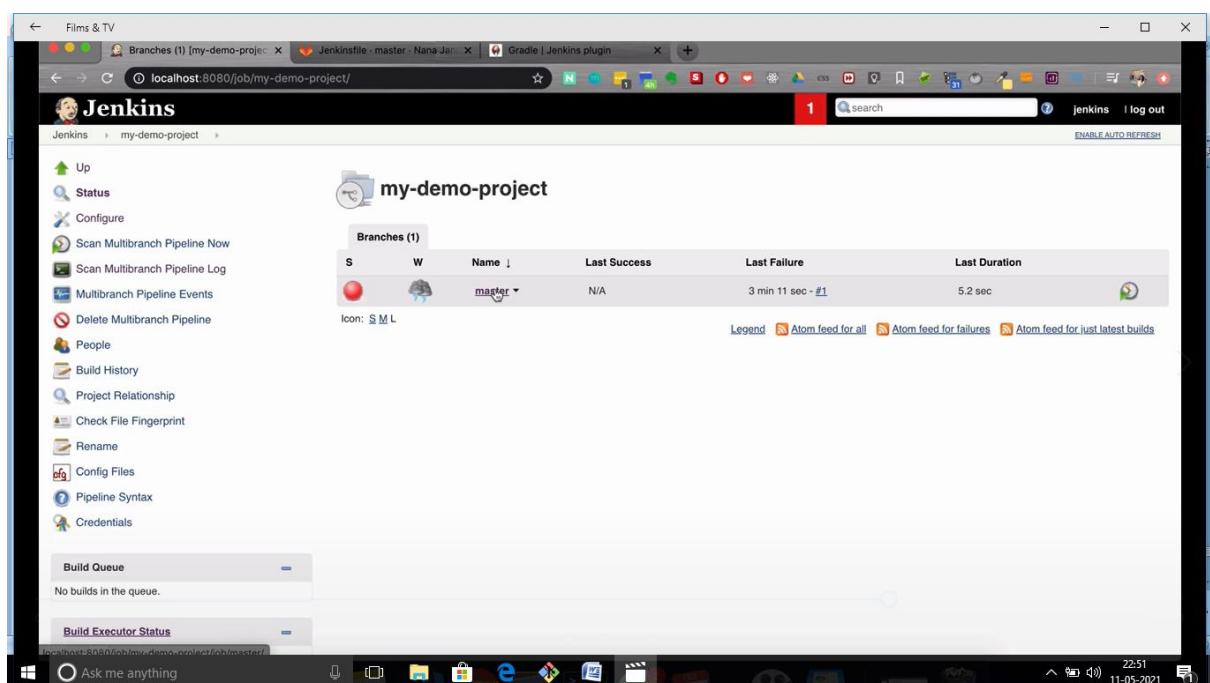
A screenshot of a Jenkinsfile editor within a GitLab interface. The title bar shows 'my-demo-project Scan Multi' and 'New File - master - Nana Janashia'. The editor displays a Jenkins pipeline configuration. The code in the Jenkinsfile is:

```
1 pipeline {
2     agent any
3     stages {
4         stage("run frontend") {
5             steps {
6                 echo 'executing yarn...'
7             }
8         }
9         stage("run backend") {
10            steps {
11                echo 'executing gradle...'
12            }
13        }
14    }
15 }
```

The editor has a toolbar at the top with 'Soft wrap' and 'text' buttons. On the left, there's a sidebar with icons for file operations like New file, Delete, Copy, Paste, and More. The bottom of the screen shows the Windows taskbar with various pinned icons.



```
1 pipeline {
2     agent any
3     stages {
4         stage("run frontend") {
5             steps {
6                 echo 'executing yarn...'
7                 nodejs('Node-10.17') {
8                     sh 'yarn install'
9                 }
10            }
11        }
12        stage("run backend") {
13            steps {
14                echo 'executing gradle...'
15                withGradle() {
16                    sh './gradlew -v'
17                }
18            }
19        }
20    }
21 }
```



Jenkins

my-demo-project

Branches (1)

S	W	Name ↓	Last Success	Last Failure	Last Duration
0	1	master	N/A	3 min 11 sec - #1	5.2 sec

Icon: S M L

Legend: Atom feed for all Atom feed for failures Atom feed for just latest builds

INTHE CONSOLE OUTPUT

```
[Pipeline] 
[Pipeline] // nodejs
[Pipeline] 
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (run backend)
[Pipeline] echo
[Pipeline] executing gradle...
[Pipeline] withGradle
[Pipeline] {
[Pipeline] sh
+ ./gradlew -v
-----
Gradle 4.2.1

Build time: 2017-10-02 15:36:21 UTC
Revision: a88ebd6be7840c2e59ae4782eb0f27fbe3405ddf

Groovy: 2.4.12
Ant: Apache Ant(TM) version 1.9.6 compiled on June 29 2015
JVM: 1.8.0_242 (Oracle Corporation 25.242-b08)
OS: Linux 4.9.13-moby amd64

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

```
1 pipeline {
2   agent any
3   stages {
4     stage("run frontend") {
5       steps {
6         echo 'executing yarn...'
7         nodejs('Node-10.17') {
8           sh 'yarn install'
9         }
10      }
11    }
12    stage("run backend") {
13      steps {
14        echo 'executing gradle...'
15        withGradle() {
16          sh './gradlew -v'
17        }
18      }
19    }
20  }
21 }
```

Alternative to using build tools

The screenshot shows a Jenkinsfile editor within a GitLab interface. The code defines a pipeline with stages for running frontend and backend code. A tooltip indicates that only three tools are supported: Maven, Gradle, and JDK.

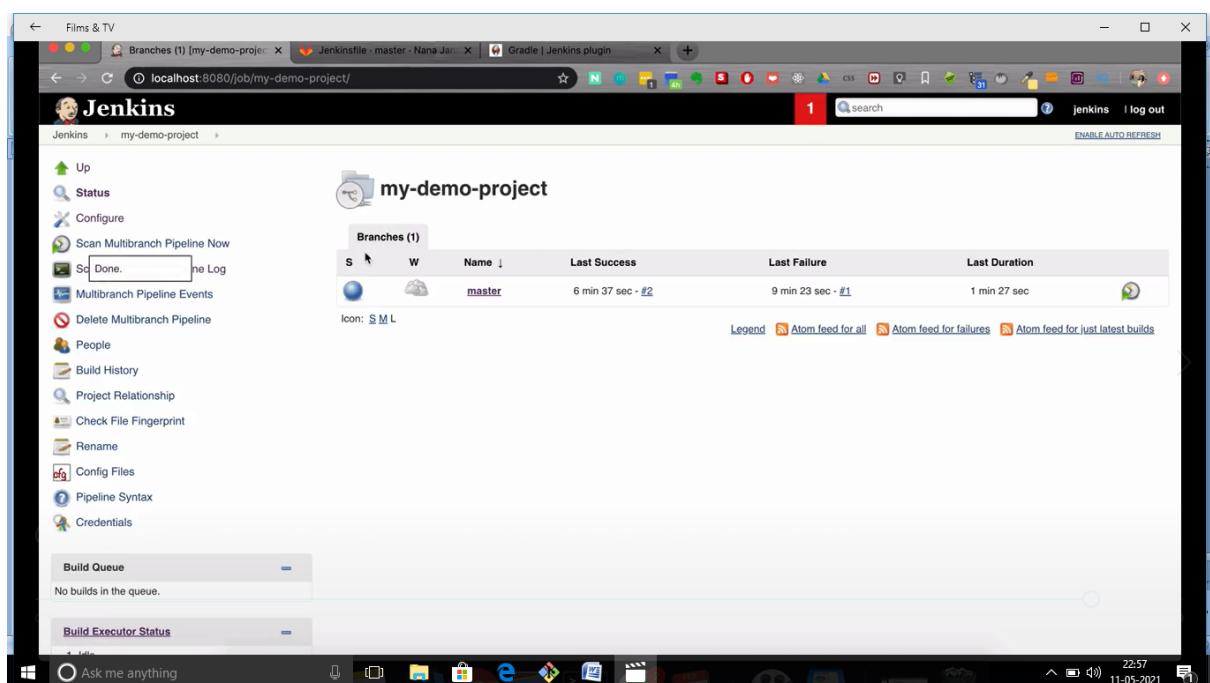
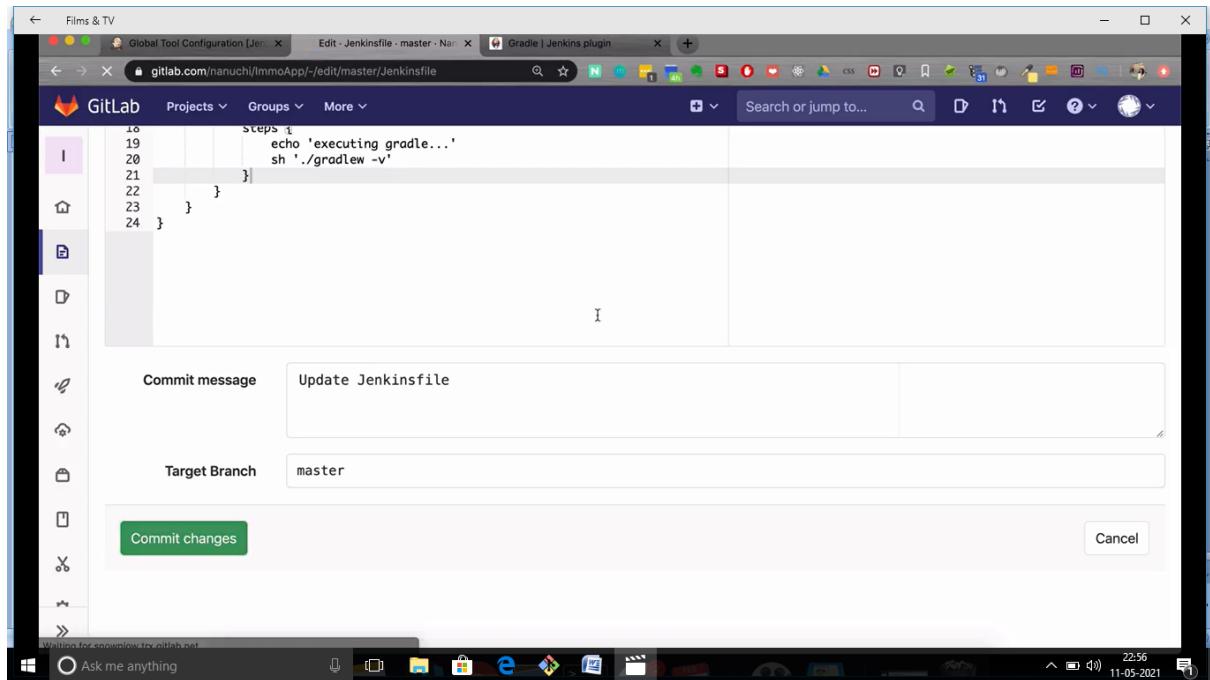
```
1 pipeline {
2     agent any
3
4     tools {
5         |           |
6     }
7
8     stages {
9         stage("run frontend") {
10            steps {
11                echo 'executing yarn...'
12                nodejs('Node-10.17') {
13                    sh 'yarn install'
14                }
15            }
16        }
17        stage("run backend") {
18            steps {
19                echo 'executing gradle...'
20                withGradle() {
21                    sh './gradlew -v'
22                }
23            }
24        }
25    }
26 }
```

Only 3 tools supported:

maven, gradle, jdk

The screenshot shows a Jenkinsfile editor within a GitLab interface, similar to the first one but with a specific tool configuration. The code defines a pipeline with stages for running frontend and backend code, specifying the use of Gradle 6.2.

```
1 pipeline {
2     agent any
3
4     tools {
5         gradle 'Gradle-6.2'
6     }
7
8     stages {
9         stage("run frontend") {
10            steps {
11                echo 'executing yarn...'
12                nodejs('Node-10.17') {
13                    sh 'yarn install'
14                }
15            }
16        }
17        stage("run backend") {
18            steps {
19                echo 'executing gradle...'
20                withGradle() {
21                    sh './gradlew -v'
22                }
23            }
24        }
25    }
26 }
```



The screenshot shows the Jenkins Pipeline master interface. The top navigation bar includes tabs for 'master (my-demo-project)', 'Jenkinsfile - master - Nana Jain', and 'Gradle | Jenkins plugin'. The main content area is titled 'Pipeline master' with the sub-section 'Stage View'. It displays a timeline of stages: 'Declarative: Checkout SCM' (2s), 'Declarative: Tool Install' (52s), 'run frontend' (1s), and 'Success run backend' (1s). A tooltip indicates an average stage time of 3s. Below the timeline is a 'Build History' section showing three builds: #3 (Feb 22, 2020 9:01 AM), #2 (Feb 22, 2020 8:54 AM), and #1 (Feb 22, 2020 8:51 AM). The bottom right corner of the window shows the Windows taskbar with the date and time as 11-05-2021 22:57.