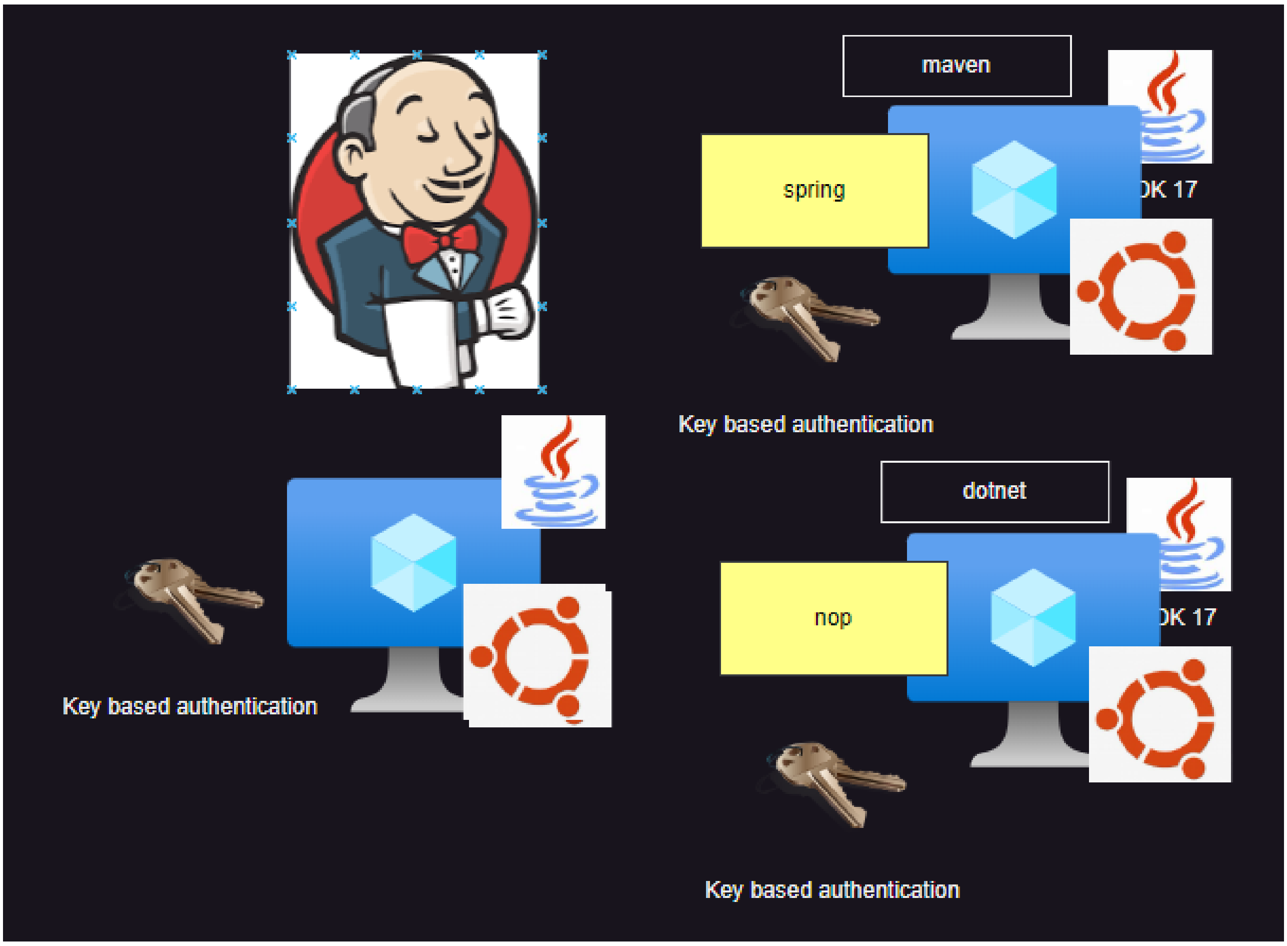
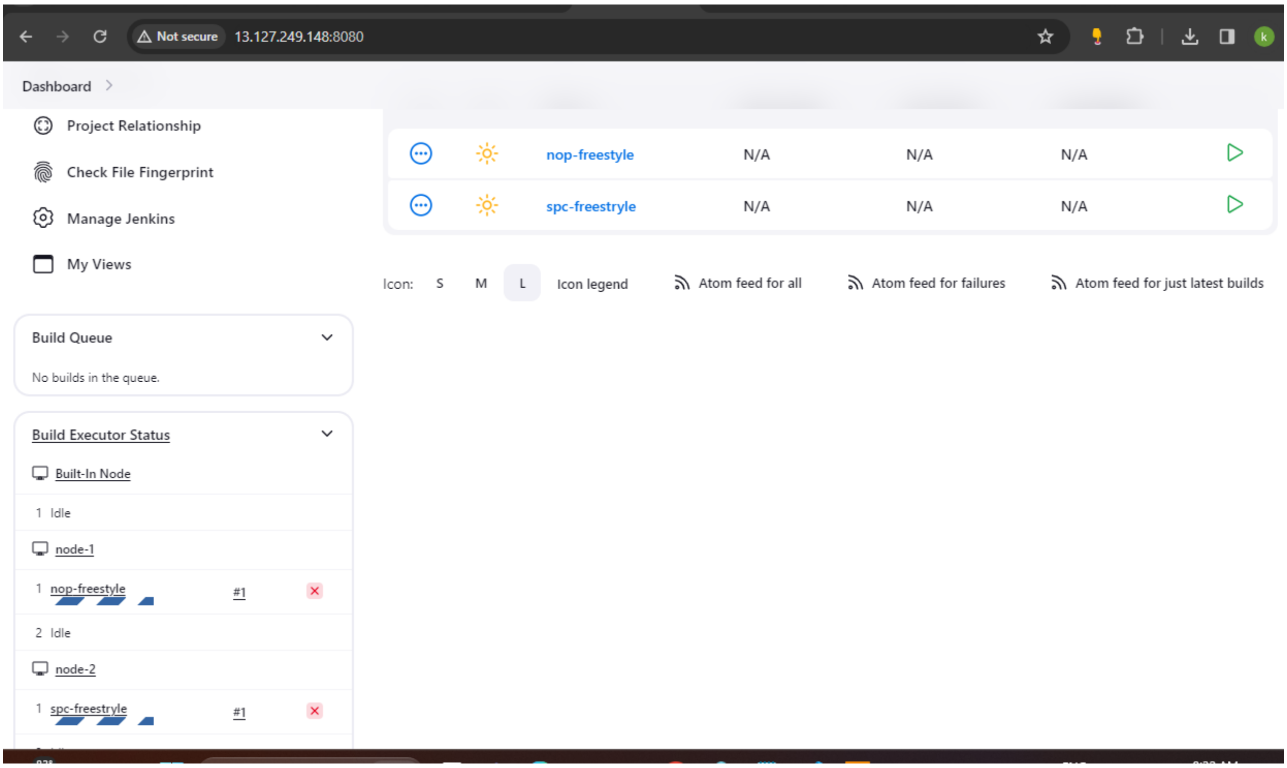
Distributed Build

- Build Executor:
  - On every node we can configure number of build executors, these determine how many projects can be built parllely on the node.
  - Generally most of orgs employ 1 executor per cpu
- We have configured nodes according the image below





nop-freestyle	N/A	N/A	N/A
spc-freestyle	N/A	N/A	N/A

Build Queue

No builds in the queue.

Build Executor Status

Built-In Node

1 Idle

node-1

1 nop-freestyle #1

2 Idle

node-2

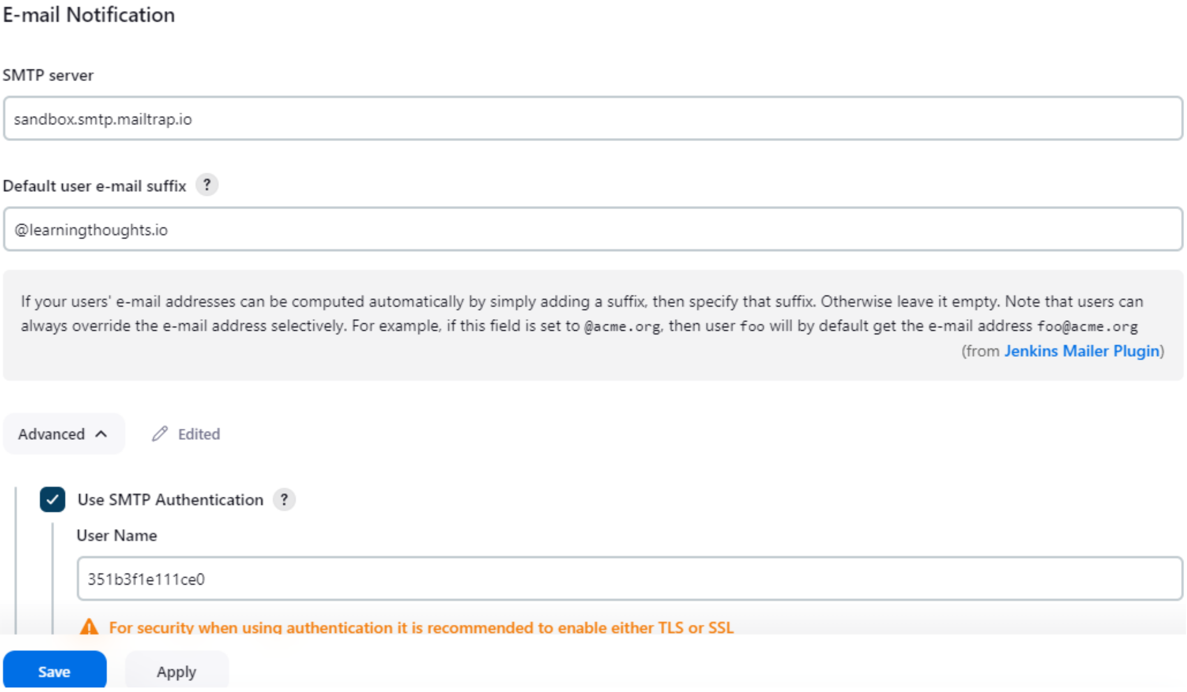
1 spc-freestryle #1

Day builds and Nightly Builds

- Day builds are generated on almost every change done by dev teams and Nightly builds are generated once in a day representing work done by dev team for the whole day.
- Most automated tests (selenium, postman etc) are execute on Nightly builds.
- Nightly builds execute
  - build, package and unit tests
  - static code analysis
  - artifacts are stored in artifact repository (jfrog, docker registry ...)
  - Run all automated tests
    - system tests
    - peformance tests
    - security tests
- Day builds
  - build, package and unit tests
  - static code analysis

Build Notifications

- Generally Build failure notifcations are sent on emails
- These days organizations prefer publishing build notifications on
  - Teams Channels
  - Slack Channels
- Lets configure our jobs to send emails after build failure
- Mail configuration Refer Here. Create a inbox in email testing (Fake smtp server)
- Navigate to Manage Jenkins => System



E-mail Notification

SMTP server

sandbox.smtp.mailtrap.io

Default user e-mail suffix ?

@learningthoughts.io

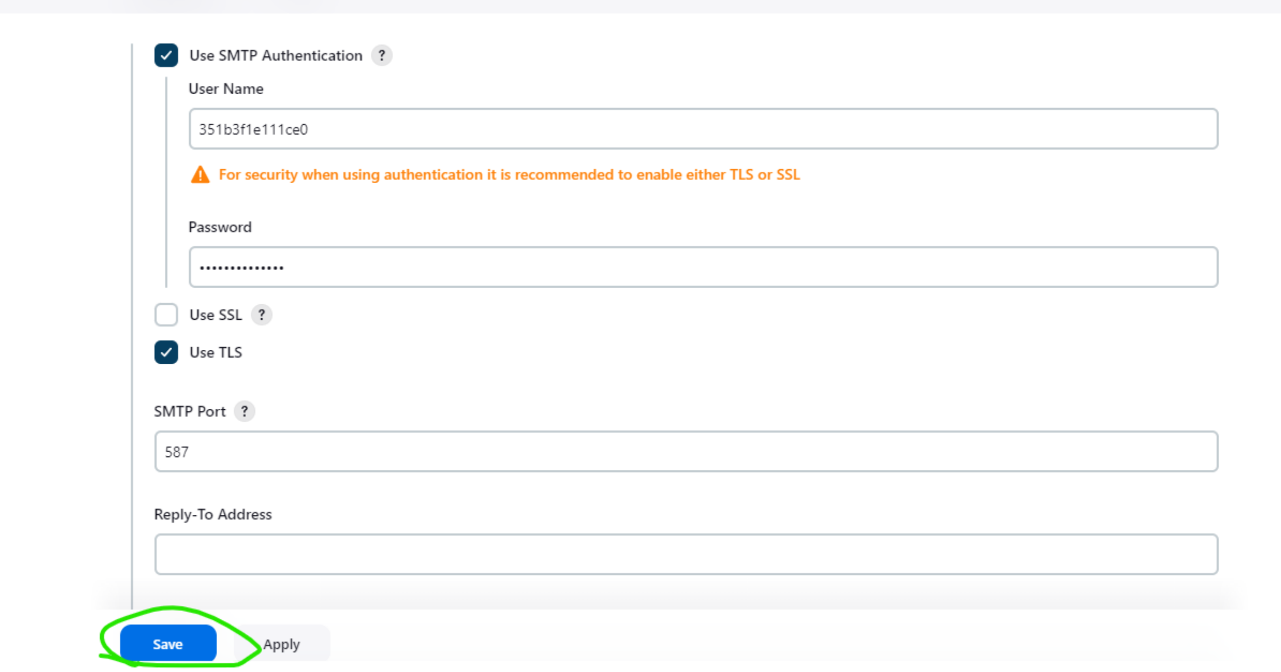
If your users' e-mail addresses can be computed automatically by simply adding a suffix, then specify that suffix. Otherwise leave it empty. Note that users can always override the e-mail address selectively. For example, if this field is set to `@acme.org`, then user `foo` will by default get the e-mail address `foo@acme.org` (from Jenkins Mailer Plugin)

☑ Use SMTP Authentication ?

User Name

351b3f1e111ce0

⚠ For security when using authentication it is recommended to enable either TLS or SSL



☑ Use SMTP Authentication ?

User Name

351b3f1e111ce0

⚠ For security when using authentication it is recommended to enable either TLS or SSL

Password

••••••••••••••

□ Use SSL ?

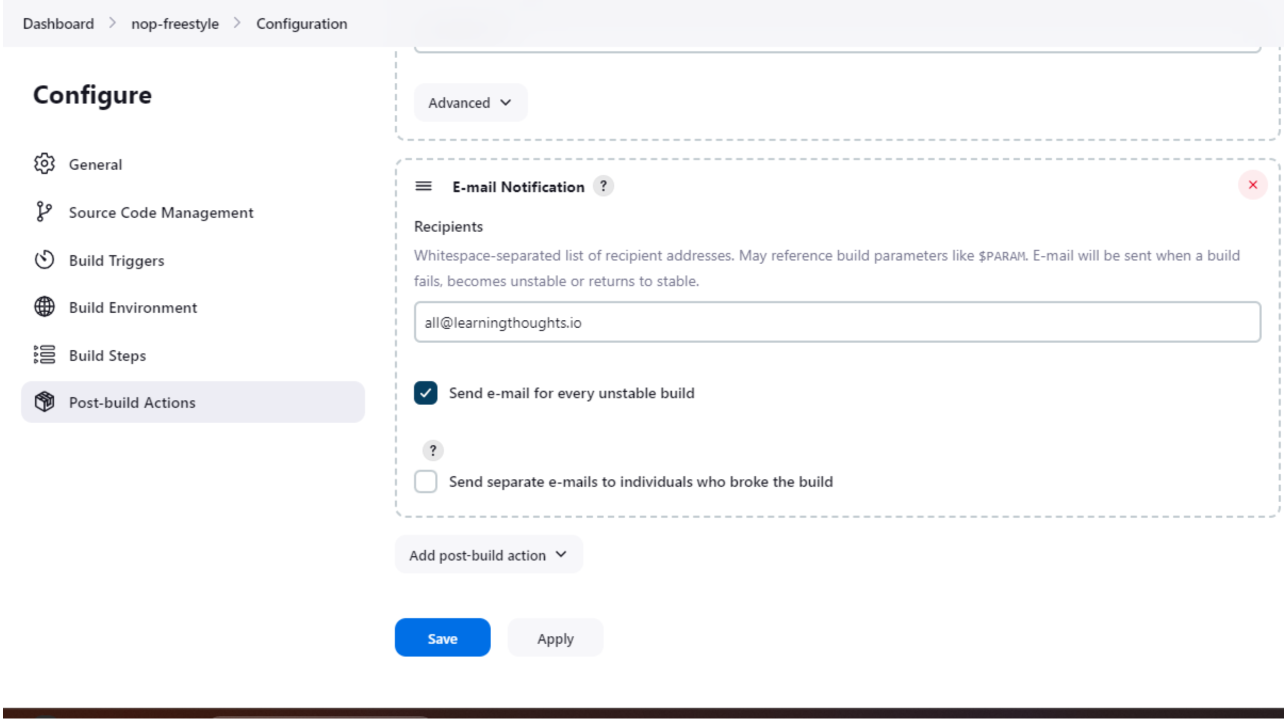
☑ Use TLS

SMTP Port ?

587

Reply-To Address

- Configure postbuild action to send email on failure



E-mail Notification ?

Recipients

Whitespace-separated list of recipient addresses. May reference build parameters like $PARAM. E-mail will be sent when a build fails, becomes unstable or returns to stable.

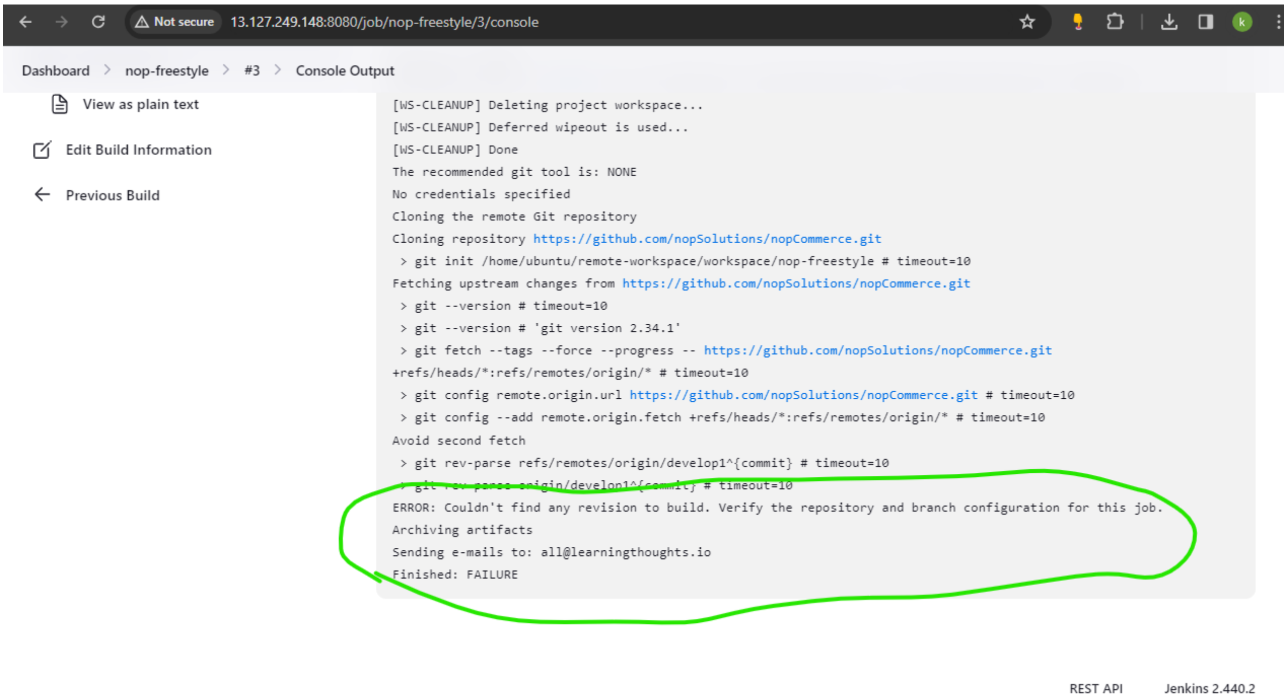
all@learningthoughts.io

☑ Send e-mail for every unstable build

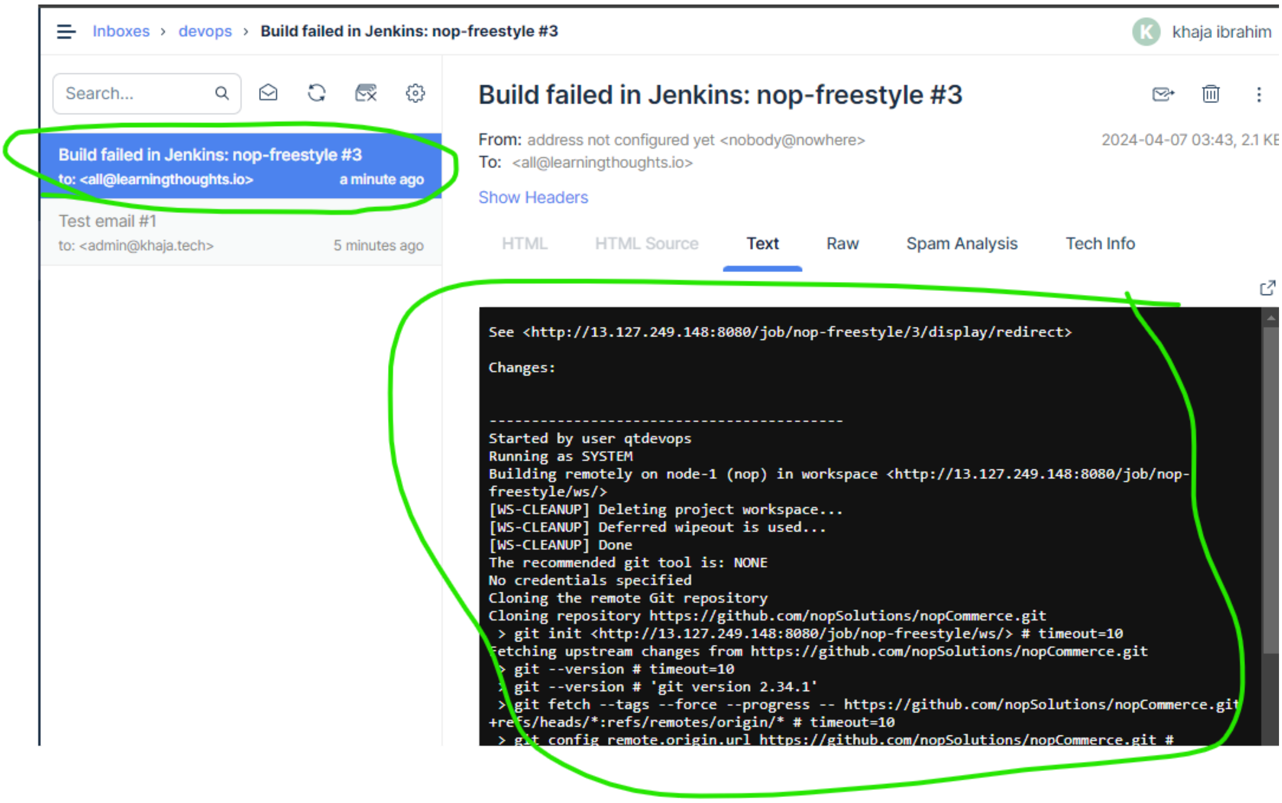
?

□ Send separate e-mails to individuals who broke the build

- Example failure


Not secure
13.127.249.148:8080/job/nop-freestyle/3/console
Dashboard > nop-freestyle > #3 > Console Output
View as plain text
Edit Build Information
Previous Build
REST API
Jenkins 2.440.2

```
[WS-CLEANUP] Deleting project workspace...
[WS-CLEANUP] Deferred wipeout is used...
[WS-CLEANUP] Done
The recommended git tool is: NONE
No credentials specified
Cloning the remote Git repository
Cloning repository https://github.com/nopSolutions/nopCommerce.git
 > git init /home/ubuntu/remote-workspace/workspace/nop-freestyle # timeout=10
Fetching upstream changes from https://github.com/nopSolutions/nopCommerce.git
 > git --version # timeout=10
 > git --version # 'git version 2.34.1'
 > git fetch --tags --force --progress -- https://github.com/nopSolutions/nopCommerce.git
+refs/heads/*:refs/remotes/origin/* # timeout=10
 > git config remote.origin.url https://github.com/nopSolutions/nopCommerce.git # timeout=10
 > git config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
Avoid second fetch
 > git rev-parse refs/remotes/origin/develop1^{commit} # timeout=10
 > git rev-parse origin/develop1^{commit} # timeout=10
ERROR: Couldn't find any revision to build. Verify the repository and branch configuration for this job.
Archiving artifacts
Sending e-mails to: all@learningthoughts.io
Finished: FAILURE
```



Build failed in Jenkins: nop-freestyle #3

From: address not configured yet <nobody@nowhere>
To: <all@learningthoughts.io>
2024-04-07 03:43, 2.1 KB

```
See <http://13.127.249.148:8080/job/nop-freestyle/3/display/redirect>

Changes:


------------------------------------------
Started by user qtdevops
Running as SYSTEM
Building remotely on node-1 (nop) in workspace <http://13.127.249.148:8080/job/nop-
freestyle/ws/>
[WS-CLEANUP] Deleting project workspace...
[WS-CLEANUP] Deferred wipeout is used...
[WS-CLEANUP] Done
The recommended git tool is: NONE
No credentials specified
Cloning the remote Git repository
Cloning repository https://github.com/nopSolutions/nopCommerce.git
 > git init <http://13.127.249.148:8080/job/nop-freestyle/ws/> # timeout=10
Fetching upstream changes from https://github.com/nopSolutions/nopCommerce.git
 > git --version # timeout=10
 > git --version # 'git version 2.34.1'
 > git fetch --tags --force --progress -- https://github.com/nopSolutions/nopCommerce.git
+refs/heads/*:refs/remotes/origin/* # timeout=10
 > git config remote.origin.url https://github.com/nopSolutions/nopCommerce.git #
```

Pipeline as Code

- Express CI/CD Pipelines as Code and store them in version control systems
- Any changes in CI/CD Pipelines will be done in these code files and history will be mainted by version control.
- Pipeline as Code uses different ways according to CI/CD tool
  - Azure DevOps: YAML
  - Github Actions: YAML
  - Git Lab: YAML
  - Jenkins

        - Jenkins DSL(Domain Specific language)

Pipeline as Code in Jenkins

- Jenkins started with Scripted Pipelines which were heavily dependent on groovy and then they introduced declarative pipelines which are relative simpler and learning curve is less (depending on jenkins experience)
- Scripted Pipelines are used when build steps are custom and complex and for regular CI/CD pipelines using Declarative Pipelines is easier and adoptable.