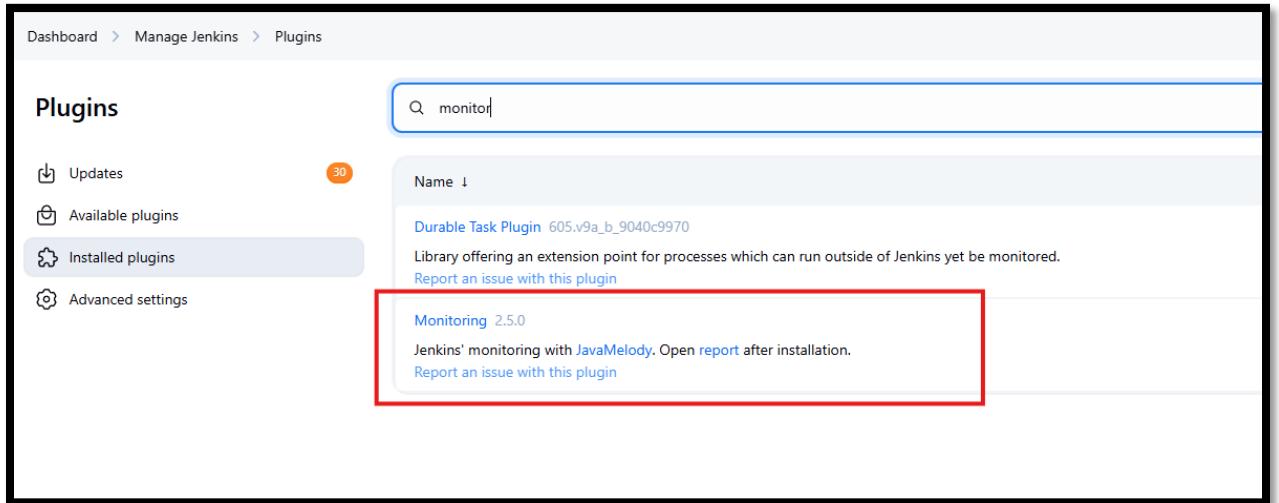
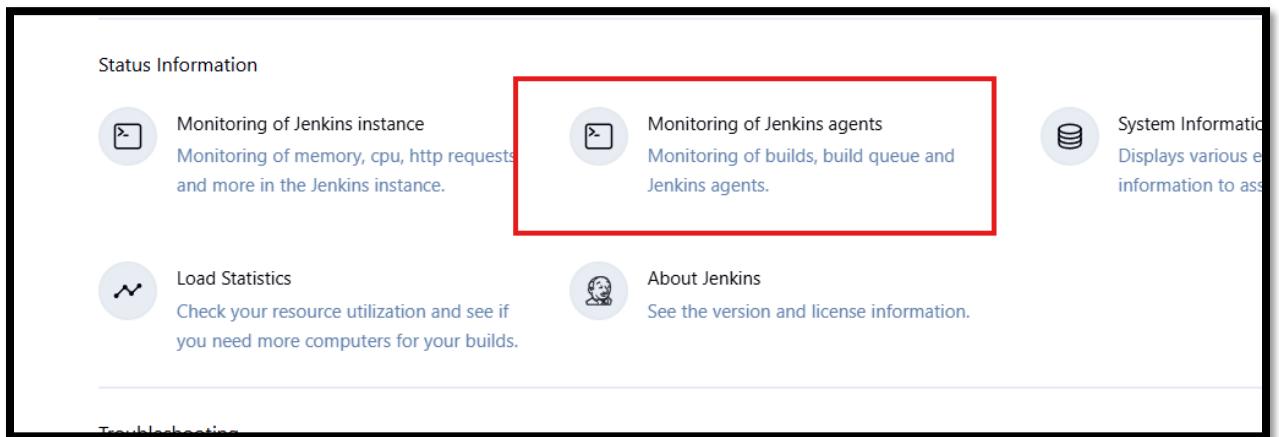
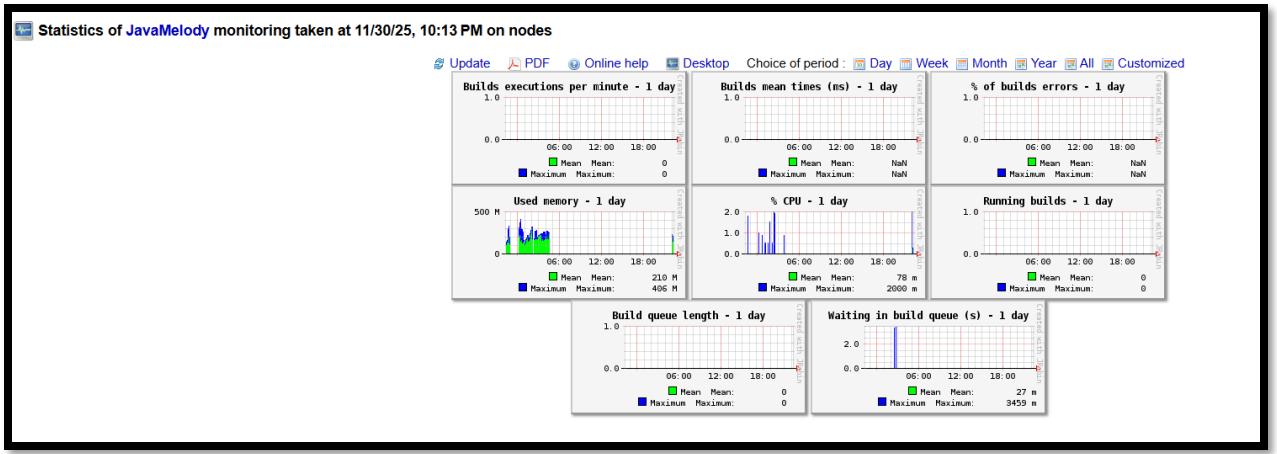


1. To implement monitoring, install the Monitoring plug-in by navigating from the Dashboard to “Manage Jenkins” -> “Plugins” -> “Available Plugins” -> and searching for the Monitoring plugin as shown below. Implementing this plugin is needed for maintaining an observable, high-quality pipeline for reasons including maintaining my application’s high uptime requirement, and the ability to provide fast, reliable feedback based on the KPIs being monitored.



2. Now that the plugin is installed, navigate to “Manage Jenkins” -> “Monitoring of Jenkins agents” to access the monitoring dashboard:





3. Next, implement automated feedback by integrating Slack notification anytime the CI/CD pipeline fails. This step first requires you to already have a Slack channel setup, create a Slack App, enables write permission, then invite the Slack Bot to the Slack Channel. Then, in Jenkins follow the same steps as mentioned above in Step #1 to download and install the “Slack Notification” plugin. Immediate feedback is critical to maintaining the pipeline and ensuring that any issues are addressed as soon as possible by automatically notifying all responsible parties.

[Slack Notification Plugin 795.v4b_9705b_e6d47](#)

Integrates Jenkins with Slack, allows publishing build statuses, messages and files to Slack channels.

[Report an issue with this plugin](#)

4. Navigate from the home Dashboard to “Manage Jenkins” -> “Credentials” to create a global credential using the OAuth bot token generated in Slack when creating a Slack App.

Update credentials

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

Secret
 Concealed

ID ?
slack-token

Description ?
bot token

Save

5. Lastly, update the Jenkinsfile to send the slack message saying that the build was either successful or failed:

```
PINECONE_API_KEY = credentials('pinecone-api-key')
SLACK_TOKEN = credentials('slack-token')

post {
    success {
        slackSend(
            teamDomain: 'jenkins-alertsglobal',
            channel: '#all-jenkins-alerts',
            color: 'good',
            message: "✓ SUCCESS: Job '${env.JOB_NAME}' build #${env.BUILD_NUMBER}",
            tokenCredentialId: 'slack-token',
            botUser: true,
            username: 'Jenkins',
            iconEmoji: ':jenkins:'
        )
    }

    failure {
        slackSend(
            teamDomain: 'jenkins-alertsglobal',
            channel: '#all-jenkins-alerts',
            color: 'danger',
            message: "✗ FAILED: Job '${env.JOB_NAME}' build #${env.BUILD_NUMBER}",
            tokenCredentialId: 'slack-token',
            botUser: true,
            username: 'Jenkins',
            iconEmoji: ':jenkins:'
        )
    }
}
```

6. To confirm the Slack automation is working correctly, after pushing to the main branch a Slack notification should appear (NOTE: the app/bot must be manually invited to the Slack channel for this to work properly):

