Task 4: Jenkins. Automate, Manage and Control

Important points:

1. Read about Jenkins. What is Jenkins and what is it used for? Ways of using. What is a declarative and imperative approach?

Tasks:

1. Install Jenkins. It must be installed in a docker container.
2. Add maven/gradle plugins like: checkstyle, spotbugs, dependency-check.
3. Create several branches in your Git repository: one branch should contain good code, other branches should violate one of the plugins above.
4. Create a Jenkins pipeline that will build your project. It should fail if there are any violations of above.
5. Create a ***yaml linter*** for the spring config. Use ***github actions.*** If the linter fails you shouldn't be able to merge the PR(The option can be configured for free but won't be operational. That is OK).

EXTRA:

1. Configure integration between Jenkins and your Git repo. Jenkins project must be started automatically if you push or merge to master, you also must see Jenkins last build status(success/unsuccess) in your Git repo.

2. Configure several (2-3) build agents. Agents must be run in docker.

3. Create Pipeline which will execute **docker ps -a in** *docker agent*, running on *Jenkins master's Host*.

Hint: