Chanesh Mahadeo
9/1/2022
KuraLabs

Deployment Documentation

Deployment

1. Create ec2 with ports 80 8080 and 22 open
2. Install jenkins on ec2
    a. $sudo apt update && sudo apt install default-jre
    b. $wget -q -O - https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo gpg--dearmor -o /usr/share/keyrings/jenkins.gpg
    c. $sudo sh -c 'echo deb [signed-by=/usr/share/keyrings/jenkins.gpg] http://pkg.jenkins.io/debian-stable binary/ > /etc/apt/sources.list.d/jenkins.list'
    d. $sudo apt update && sudo apt install jenkins -y
    e. $sudo systemctl start jenkins
    f. $sudo systemctl status jenkins
3. Configure jenkins https://www.jenkins.io/doc/tutorials/tutorial-for-installing-jenkins-on-AWS/
    a. http://[ec2ip]:8080
    b. sudo cat [locationofsecretkey]
    c. Install suggested plugins
    d. Create admin user
    e. Goto manage Jenkins on left hand side, go to manage plugins
    f. Install amazon ec2 without restart
    g. Go back to dashboard and click configure a cloud
    h. Add new drop down -> amazon ec2
4. Installing virtual environment so jenkins can run properly
    a. sudo apt install python3-pip
    b. sudo apt install python3.10-venv
5. Connect Github to Jenkins
    a. Fork desired repo
    b. Create an access token by going to github settings -> developer settings -> Personal access tokens -> generate new token. *SAVE THIS TOKEN FOR LATER*
    c. Select options repo and admin:repo_hook
    d. Log back into jenkins
    e. Select New Item
    f. Give build a name ->Select Multibranch Pipeline-> and press okay
    g. Give the project a name and description
    h. Now add a branch source by selecting add source and select github
    i. Under credentials select add and Jenkins

    j. Under username input your github username and under password enter your access token from earlier
    k. Select add then select the user you just created from the Credentials dropdown menu
    l. In the repository HTTPS URL enter the url of YOUR forked repo.
    m. Click on validate and if it works you did everything right so far. If not, go back and retrace your steps.
    n. Make sure under Build Configuration Mode says “by Jenkinsfile” and Script Path is “Jenkinsfile”. Then click apply then save.
    o. If a build doesn't happen automatically click scan or build repository, may need to refresh the page for updates.
6. Deploy to elastic beanstalk.
    a. Git clone the forked repo you created. You should now have the files locally.
    b. Cd into the repository
    c. Run $zip ../myapp.zip -r * .[^.]*
    d. Please note that the “myapp” portion can be changed to whatever app name you'd like.
    e. Log back into AWS and go to elastic beanstalk.
    f. Select create a new environment
    g. Select web server environment
    h. Fill in the details which match your app.
    i. For application code select local upload and upload the zip file that you created from your repository.
    j. Once it finishes creating, click on your link and go see your app running.
    k. Congrats you completed your deployment!!

ISSUES

I didn't run into too many issues with this deployment and it ran quite easy. One issue I did have is when following the instruction. When setting up the virtual environment the command “sudo apt install python3-10-venv” does not run. This is because the apt package name is python3.10-venv. The proper command to run is “sudo apt install python3.10-venv”.

Another issue would be when looking at the jenkins documentation. Most of the steps are already done so reading a large portion of the start of the file is a waste of time. Skipping to the configuration portion of the documentation is all you need to do.

IMPROVEMENTS

An improvement I would consider is obviously automating this process. The process of configuration and then building and then testing and then deployment is a bit tedious. If this process could somehow be functionally automated whilst still allowing modularity within each phase then it would significantly increase the speed at which an app could be deployed. Another thing I'd improve on is perhaps adding more tests to the Jenkinsfile. It's better to add multiple tests to fully stress and test the app to see what it can handle.

CI/CD Pipeline