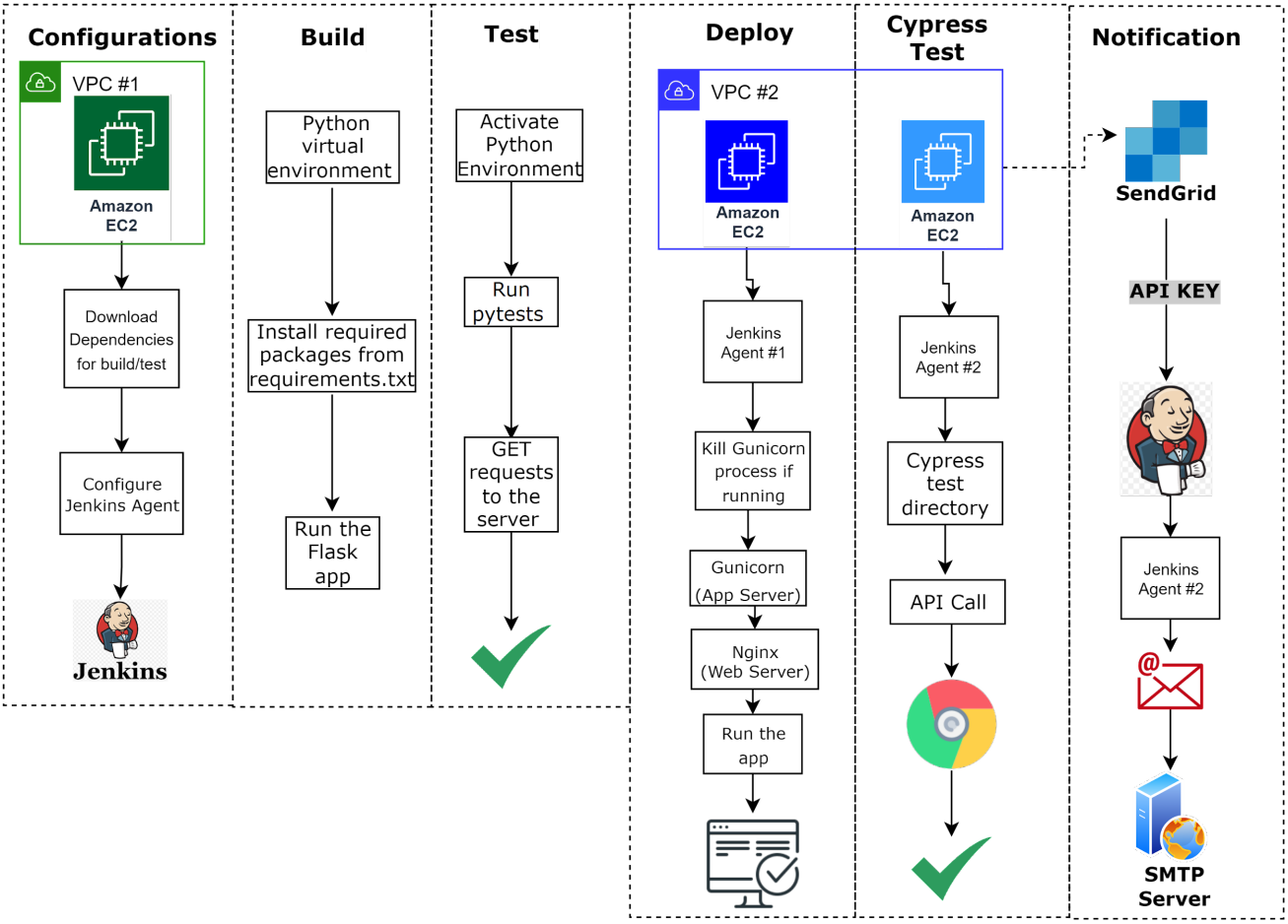
[Deployment 3 Documentation]
Author: Suborna Nath
Date: October 15, 2022

Description: In this deployment, we will be configuring a pipeline to deploy a Flask application to a custom VPC.

Diagram



Configurations: Setup EC2 and Jenkins Agent

Before getting started with the pipeline, we need to create three EC2s. One of the EC2 will be in the default VPC and the other two EC2 will be in the custom VPC. The custom VPC will have a public subnet and port access to 22 and 5000.

In all three EC2, we need to install all the dependencies before configuring the pipeline. The dependencies include default-jre, python3-pip, python3-venv, Nginx, and cypress test dependencies.

When setting up the pipeline, we will need to install the 'mailer' plugin and 'pipeline keep running' plugin in the Jenkins server.

Jenkins Pipeline Walkthrough

1. Declarative: Checkout SCM

- In this initial stage, Jenkins will access the GitHub repo and pull the latest version of the source code. It will add the project files to the Jenkins workspace.

2. Build

- Python environment is getting created and activated
- Latest version of pip is installed
- Installing all the required packages from requirements.txt
- Setting the application name "application.py" as an environment variable
- Running the Flask application

3. Test

- Activating the python environment
- Running the pytests written in ./test_app.py file and saving the results as results.xml
    - ➔ Testing the homepage to make sure we can access it
    - ➔ Testing an invalid URL to make sure it gives a 404 error
    - ➔ Testing a saved URL to make sure it redirects the request with a 302 status code

4. Deploy (Jenkins Agent #1)

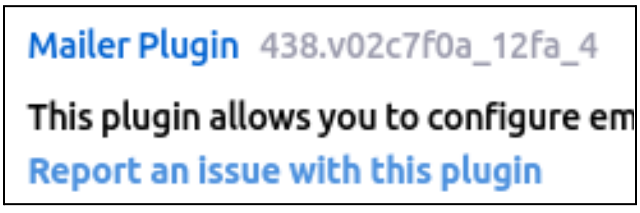
- Install all required packages from requirements.txt and Gunicorn
- Deploy the Flask app with Gunicorn as the application server and Nginx as the web server
- Nginx works as the webserver to receive and send requests
- Gunicorn handles the HTTP requests and interprets it for Python to handle the requests
- Once the app is successfully deployed, it can be accessed with the public IP of the EC2, at port 5000

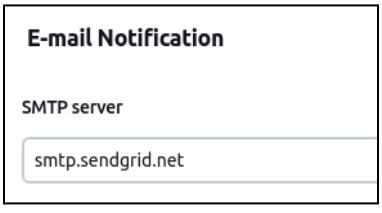
Troubleshoot: I needed to make sure "Pipeline keep running" plugin is installed properly. Also, I needed to install "pip" to run the commands for deployment. After making the changes to the Nginx "default" file, I needed to restart the server to make sure the changes were applied.

5. Cypress Test (Jenkins Agent #2)

- We changed the directory from the project folder to the cypress_test folder
- We download all the dependencies and cypress with npm
- The test is written in the ./cypress/e2e folder
- We read the test.cy.js file and run the test
- To run the test, we are making an API call with package "ChromeDriver" and "google-chrome-stable" to check if the webpage has the proper title.

6. Notification





- I used an email notification plugin “Mailer” with an SMTP server and SSL connection for sending the emails. I used SendGrid as the email server.
- This notification is set up to trigger at the end of each successful build and send an email to the specified address with the status report of the job.

What can be improved?

For the deploy stage, I can create a webhook to auto-deploy the application every time there is a new commit in the main branch.
For the Cypress test, I can implement more tests to check the webpage functionalities.
For notifications, I want to set up conditions so that the email triggers when any stage of the pipeline fails.