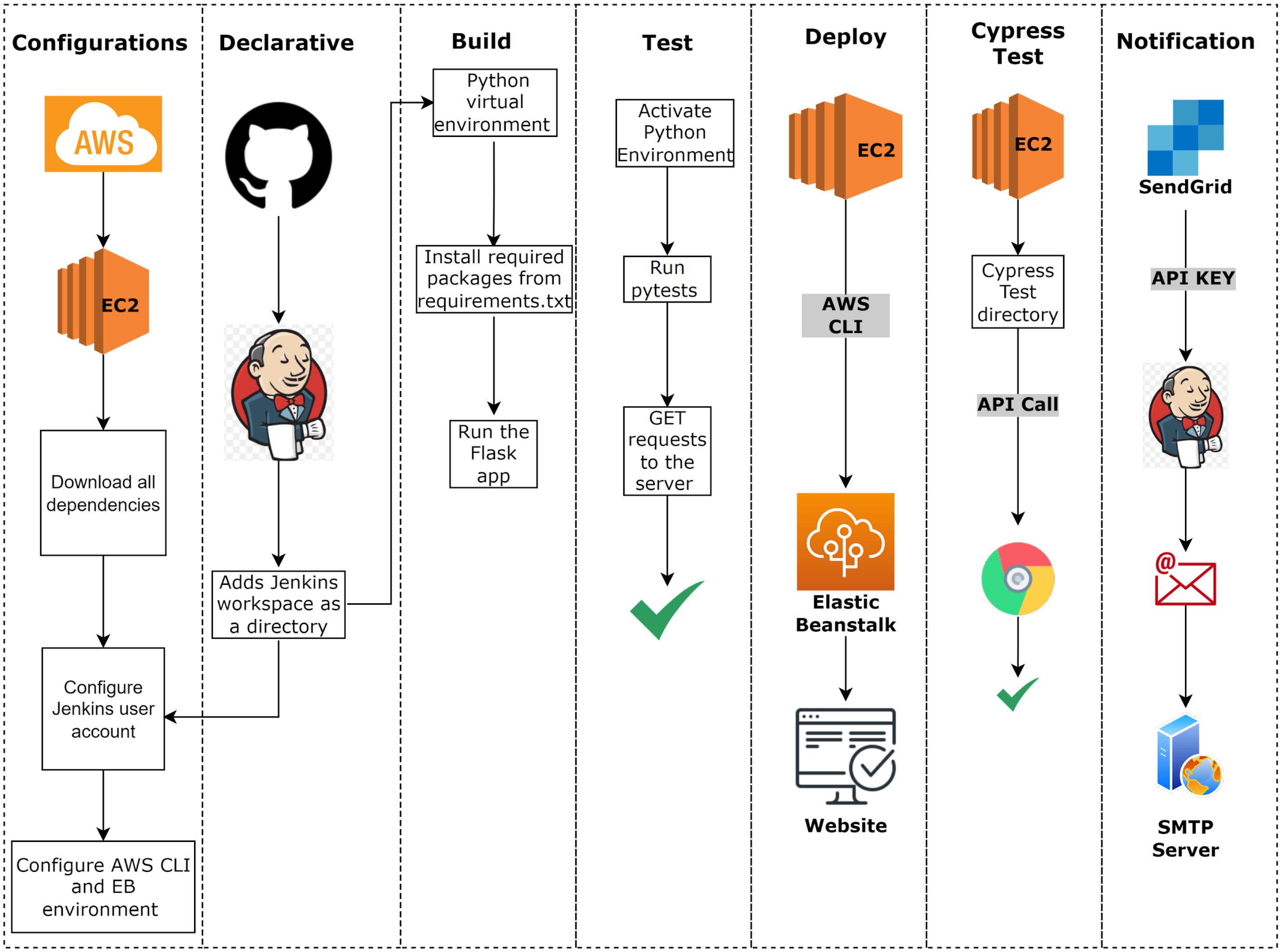
[Deployment 2 Documentation]

Author: Suborna Nath
Date: September 27, 2022

Description: In this deployment, we are configuring a Jenkins pipeline to deploy a flask application from a GitHub repository using AWS CLI, EC2, and Elastic Beanstalk CLI.

Deployment 2 Diagram


Configurations
AWS
EC2
Download all dependencies
Configure Jenkins user account
Configure AWS CLI and EB environment
Declarative
Adds Jenkins workspace as a directory
Build
Python virtual environment
Install required packages from requirements.txt
Run the Flask app
Test
Activate Python Environment
Run pytests
GET requests to the server
Deploy
EC2
AWS CLI
Elastic Beanstalk
Website
Cypress Test
EC2
Cypress Test directory
API Call
Notification
SendGrid
API KEY
SMTP Server

Setup: SSH into an AWS EC2 & configure dependencies

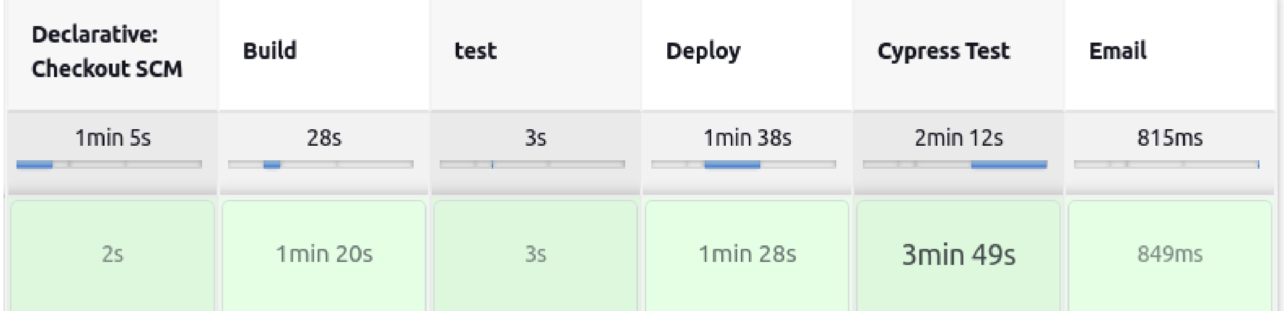
The first step of the deployment is to create an EC2 and SSH into the EC2. In the EC2, we download all the dependencies needed to run the Jenkins server and the Flask application.

We added the Jenkins user to a sudoers group so that we can install and configure the dependencies for the pipeline as a Jenkins user. The packages include AWS CLI, Elastic Beanstalk (EB) CLI, Cypress test dependencies, and Google Chrome (without GUI).

After configuring all the dependencies, we run the pipeline to add the workspace and project folder to the terminal. We create an EB environment in the project folder which is needed for deploying the app with the pipeline.

Before we run the Cypress test with the pipeline, we need to download the Cypress and node_modules in the cypress_test folder. These dependencies are required for the cypress test to work.

Jenkins Pipeline Walkthrough



Declarative: Checkout SCM	Build	test	Deploy	Cypress Test	Email
1min 5s	28s	3s	1min 38s	2min 12s	815ms
2s	1min 20s	3s	1min 28s	3min 49s	849ms

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 environment variable
    - Running the Flask application

3. ***Test***
    - Activating the python environment
    - Running the pytests written in “test_app.py” 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***
    - We have pre-configured AWS CLI with Jenkins and created an EB environment. This gives Jenkins pipeline access to deploy the app on AWS.
    - A copy of the application gets uploaded to EB
    - A EC2 instance gets created by EB
    - The application is deployed to the EC2 instance

    Troubleshoot: I ran into an issue where my app wasn’t deploying. The error in the console log mentioned that I didn’t have an environment setup and I need to run “eb init”. To fix the issue, I deleted the environment in AWS EB and reconfigured it from my terminal. That resolved the issue.

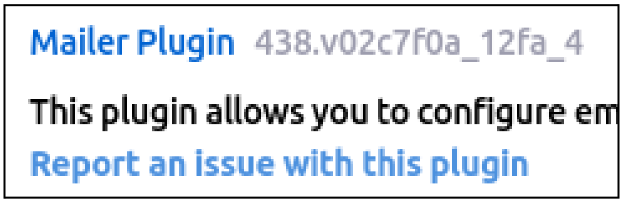
Another issue I had is the name of my environment was different in my pipeline than what was set in the terminal. I needed to make sure I am using the correct environment name in the pipeline.

5. ***Cypress Test***

- The dependencies for the test were pre-installed
- We changed the directory from the project folder to the cypress_test folder
- The test is written in the ./cypress/e2e folder
- We read the test.cy.js file and run the test
- With this test, we are checking if the website has the proper title

Troubleshoot: When I tried adding the installation of node modules and cypress in the pipeline, my EC2 will get stuck and I needed to start the EC2. So, I downloaded the packages before running the pipeline. I also was running into errors with the cypress versions. Downloading cypress 10.0.3 resolved the errors.
I also needed a browser to run the test. Since EC2 doesn't support any GUI, I downloaded Google Chrome and a ChromeDriver that allows access to chrome with an API protocol.

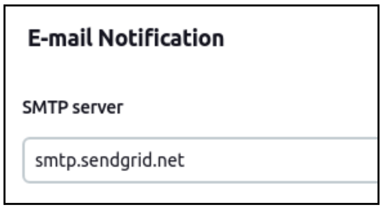
6. ***Notification***



Mailer Plugin 438.v02c7f0a_12fa_4

This plugin allows you to configure em

Report an issue with this plugin



E-mail Notification

SMTP server

smtp.sendgrid.net

- We 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.

Troubleshoot: At first, I tried setting up my Gmail for the notifications; it required my login credentials for the account. I was getting an error with sending the emails because my Jenkins server isn't set up with HTTPS. I resolved this issue with SendGrid which has an email server and it let me create separate credentials for Jenkins. I was able to see the emails in the server I set up with them.

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 a few more tests that check other functionalities.
For notifications, I can set up the email triggers to go off whenever any stage of the pipeline fails.