**Deployment #2**

Name: Biki Gurung

Date: 09/22/2022

**Description:**

**Pre-requisites:**

* AWS account
* CI tool of choice (Jenkins)
* GitHub repository you’d like to deploy

Tables of Contents:

[Install Jenkins on an EC2: 2](#_Toc112976926)

[Install Virtual Environment 4](#_Toc112976927)

[Connect GitHub to Jenkins Server 6](#_Toc112976928)

[Create an access token from GitHub 7](#_Toc112976929)

[How to setup Jenkins: 8](#_Toc112976930)

[Create a multibranch build 10](#_Toc112976931)

[How to Download application files from GitHub and Deploy to Elastic Beanstalk 15](#_Toc112976932)

[How to create an Elastic Beanstalk Env 16](#_Toc112976933)

[List of Issues and their solution 18](#_Toc112976934)

[Issue #1 – Couldn’t download zip file using SCP from the Server 18](#_Toc112976935)

[Issue #2: Elastic beanstalk kept on failing with HTTP 502 21](#_Toc112976936)

[What Improvements can be made? 23](#_Toc112976937)

[Idea #1) Rapid Deployment with Jenkins 23](#_Toc112976938)

[Upon sucessful Jenkins build and test. Use jenkins to compress the github repo and deploy to the Elasticbean automatically. 23](#_Toc112976939)

[Idea #2) Create a separate Staging and Production Env so you can test your codes before you sent the developers code to production. 23](#_Toc112976940)

[Idea #3) Scripts need to be checked against known attacks and malicious scripts databases to make sure that bad lines of code never get to the production environment. 23](#_Toc112976941)

[CI/CD Pipeline Diagram: 24](#_Toc112976942)

|  |  |
| --- | --- |
| Install Jenkins on an EC2: | |
|  | 1. Create EC2 Ubuntu Instance <https://www.cloudbooklet.com/create-an-ec2-instance-on-aws-with-ubuntu-18-04/>      1. Download the .pem file found when creating the EC2 Instance cand store on your local P.      1. SSH into the EC2 using the .pem file and the Public IP Address of the Ec2 instance.   ssh -i "ubuntu\_ec2.pem" [ubuntu@ec2-52-90-159-255.compute-1.amazonaws.com](mailto:ubuntu@ec2-52-90-159-255.compute-1.amazonaws.com) | |
|  | Opening Ports 80,8080 and 22 on the ec2 instance    <https://aws.amazon.com/premiumsupport/knowledge-center/connect-http-https-ec2/> | |
|  | **Login to EC2 Instance:**  Once you’ve created the EC2, log into the EC2 and then  enter the commands below to install Jenkins:    $ ssh -i "ubuntu\_ec2.pem" ubuntu@ec2-52-90-159-255.compute-1.amazonaws.com | |
|  | $ sudo apt update && sudo apt install default-jre | |
|  | $wget -q -O - <https://pkg.jenkins.io/debian-stable/jenkins.io.key> |sudo gpg --dearmor -o/usr/share/keyrings/jenkins.gpg    $sudo sh -c 'echo deb [signed-by=/usr/share/keyrings/jenkins.gpg] [http://pkg.jenkins.io/debian-stable binary/](http://pkg.jenkins.io/debian-stable%20binary/) > /etc/apt/sources.list.d/jenkins.list'    $sudo apt update && sudo apt install jenkins -y    $sudo systemctl start jenkins      For assistance:   * [https://www.jenkins.io/doc/tutorials/tutorial-for-installing-jenkin s-on-AWS/](https://www.jenkins.io/doc/tutorials/tutorial-for-installing-jenkin%20s-on-AWS/)      * <https://www.digitalocean.com/community/tutorials/how-to-install-jenkins-on-ubuntu-20-04> | |
|  | To check the current status of Jenkins:    $sudo systemctl status jenkins | |
| Install Virtual Environment | | |
|  | Checking if python3 is intalled    $ python3 --version    Install updates:  $ sudo apt install    Install Python3-pip    Install Python pip $ sudo apt install python3-pip    Upgrade to latest version of pip  $ sudo pip3 install --upgrade pip            For Assistance:   * <https://phoenixnap.com/kb/how-to-install-pip-on-ubuntu> * <https://packaging.python.org/en/latest/guides/installing-using-pip-and-virtual-environments/> | |
|  | Install python3-10-venv    $sudoapt-get-y installpython3.10-venv    Or    $sudoapt-y installpython3.10-venv        Assistance:   * <https://installati.one/ubuntu/22.04/python3.10-venv/> | |
| Connect GitHub to Jenkins Server | | |
|  | First Fork the Deployment repo: <https://github.com/kura-labs-org/kuralabs_deployment_1>. git          mkdir deployment1  cd deployment1  git init    ssh-keygen -t ed25519 -C "bikigurung8@gmail.com"    ls ~/.ssh    cat id\_ed25519.pub | |
|  | Cat and copy the contents from the id\_ed25519.pub  Go to Github Settings > Add new key | |
|  | Fork the repo to your repository  And then git pull the repo from your github    $ git pull git@github.com:bikigrg11/kuralabs\_deployment\_1.git | |
| Create an access token from GitHub | | |
|  | Navigate to your GitHub settings, select developer settings    Select the settings you see below for access token permissions. | |
|  | Select personal access token and create a new token.    Select the settings you see below for access token permissions.    New token has been generated. | |
| How to setup Jenkins: | | |
|  | After you verified that the Jenkins has been started on your AWS EC2 Machine. | |
|  | Connect to http://<your\_server\_public\_DNS>:8080 from your browser. You will be able to access Jenkins through its management interface:  As prompted, enter the password found in /var/lib/jenkins/secrets/initialAdminPassword.  Use the following command to display this password:  $ sudo cat /var/lib/jenkins/secrets/initialAdminPassword | |
|  | Let the environment creation be finished | |
|  | Enter Username  Enter Password  Enter Confirm Password  Enter Full Name  Enter Email Password | |
|  | You can use this Jenkins URL to login and view your Jenkins application. | |
| Create a multibranch build | | |
|  | ● Log back into Jenkins and select “New item” | |
|  | ● Select multibranch pipeline  ● Enter a display name and brief description | |
|  | ● Add a Branch source by selecting Add source and select GitHub | |
|  | ● Select the Add button and select GitHub | |
|  | ● Click on Add and then select Jenkins | |
|  | Under username enter your GitHub username  ● Under password enter your token  ● (Optional) under ID and Description enter GitHub repo | |
|  | ● Enter your URL to the repository and you can validate by selecting validate. | |
|  | Make sure this says Jenkinsfile | |
|  | ● Select Apply and then Save | |
|  | Build Flask > Status > click On Main | |
|  | You should see a build happening. If you don’t, select Scan Repository.  The build has been completed which will run on your EC2 by pulling the Jenkinsfile from the github repo.  If you see green check box next to the Build No in this case #1 this means the build was completed successfully.  We can see that there was 2 Stages that were created.   * 1. Build   2. Test   More info: If you hover through the time (11s for the Build Stage) on how long each stage took to build you can see the logs and the results of the tests that took place in detail. | |
| How to Download application files from GitHub and Deploy to Elastic Beanstalk | | |
|  | Use git clone to copy deployment1 repo files of the flask application to your local computer.    $ git clone git@github.com:bikigrg11/kuralabs\_deployment\_1.git | |
|  | When you use the AWS Elastic Beanstalk console to deploy a new application or an application version, you'll need to upload a source bundle. Your source bundle must   * + Consist of a single ZIP file or WAR file (you can include multiple WAR files inside your ZIP file)   + Not exceed 512 MB   + Not include a parent folder or top-level directory (subdirectories are fine)     Documentation:  To compress the files from the repo, follow the how to from aws documentation: [https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/ applications-sourcebundle.html](https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/%20applications-sourcebundle.html) | |
|  | In order to zip the file first install zip  $ sudo apt install zip  $ zip kuralabs\_deployment\_1.zip kuralabs\_deployment\_1 | |
| How to create an Elastic Beanstalk Env | | |
|  | After you have compressed/zipped your files, head over to AWS Elastic Beanstalk     * Go to AWS Elastic Beanstalk      * Select the Web server Environment | |
|  | Next type in the following information  Application name: url-shortner  ○ Environment name: Urlshortner-env  ○ Platform: Python  ○ Platform branch: 3.8  ○ Platform version: 3.3.16  ○ Application code: Upload your code, local file      Resource:  <https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/GettingStarted.CreateApp.html> | |
|  | Click on locale file    Click on Choose file    Select the zip file downloaded from the server | |
| A picture containing graphical user interface  Description automatically generated | Click to Create Environment | |
|  | Wait for few minutes until the Environment is created and started. | |
|  |  | |
| List of Issues and their solution | | |
| Issue #1 – Couldn’t download zip file using SCP from the Server | | |
|  | When trying to download the zip file from the Ec2 server, I kept on getting permission denied error message. Although I was using the correct .pem file and using scp -I “.pem file” I was unable to download from the server.    so I downloaded and used FileZilla on my local machine to connect to the server    <https://dearsikandarkhan.medium.com/files-copying-between-aws-ec2-and-local-d07ed205eefa>  <https://filezilla-project.org/download.php?type=client> | |
|  | Go to Settings > General    Enter the Host Machine Information    Provide the path to the .pem key file    And connect | |
|  | If you are connected successfully then you can see the files from the remote server | |
|  | Select the folder on your local machine where you want the files to be downloaded | |
|  | On your Remote machine, select the files and right click on it  And click on Download    When the file is downloaded you can see the status. | |
| Issue #2: Elastic beanstalk kept on failing with HTTP 502 | | |
|  | After creating the Env for the Elastic Beanstalk the Health of my Elastic Beanstalk kept on getting Severe. | |
|  | I downloaded the log files and noticed that the artifact (Zip file) that was uploaded was not loading properly.  Because I was using MacBook to zip and upload to the elastic beanstalk env it was giving me this error message. | |
| Files selected in Mac OS X Finder | So, I followed what was written here:  Zipping files in Mac OS X Finder or Windows explorer:  When you create a ZIP file in Mac OS X Finder or Windows Explorer, make sure you zip the files and subfolders themselves, rather than zipping the parent folder. | |
| Compressing files in Mac OS X Finder  Icon  Description automatically generated | Once I followed this step and uploaded the new zip file to the elastic beanstalk the Health Turned to “OK” | |
| A picture containing timeline  Description automatically generated | If you click on the link you will be able to open the application that has started on the Elastic Beanstalk env. | |
|  | | |
| What Improvements can be made? | | |
| Idea #1) Rapid Deployment with JenkinsUpon sucessful Jenkins build and test. Use jenkins to compress the github repo and deploy to the Elasticbean automatically. | | |
|  | <https://www.serverkaka.com/2019/04/deploy-java-web-app-elastic-beanstalk-jenkins.html>  This will make the process seamless and without needs to zip download and then upload to the elastic beanstalk env. | |
| Idea #2) Create a separate Staging and Production Env so you can test your codes before you sent the developers code to production. | | |
| Idea #3) Scripts need to be checked against known attacks and malicious scripts databases to make sure that bad lines of code never get to the production environment. | | |

# CI/CD Pipeline Diagram:

Diagram

Description automatically generated