**Deployment 1**

# Step1

Install Jenkins on an EC2

* Create an Ubuntu EC2
* Security group ports: 80, 8080, 22
* Set up Jenkins:
  + - * + $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/ > /etc/apt/sources.list.d/jenkins.list'
        + $sudo apt update && sudo apt install jenkins -y
        + $sudo systemctl start jenkins $sudo systemctl status Jenkins

# Step2

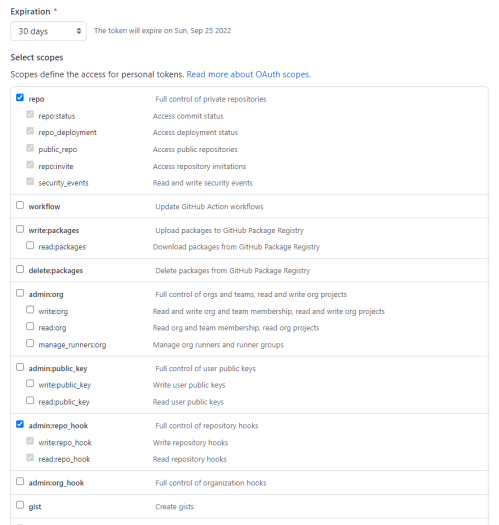
Install Virtual Environment on the EC2

* sudo apt python3-pip
  + sudo ap python3-10-venv

# Step3

Connect Github to Jenkins Server

* Fork the Deployment repo: https://github.com/kura-labs-org/kuralabs\_deployment\_1. git
* Next, create an access token from GitHub:
  + - * + Navigate to your GitHub settings, select developer settings
        + Select personal access token and create a new token
        + Select the settings you see below for access token permissions



# Step4

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
* Click on Add and then select Jenkins
* Under username enter your GitHub username
* Under password enter your token
* Enter your URL to the repository and you can validate by selecting validate
* Make sure it says Jenkinsfile
* Select Apply and then Save

# Step 5

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.
* 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
* After you have compressed/zipped your files, head over to AWS Elastic Beanstalk
* Select create a new environment
* Select Web Server environment
* Enter the following configurations:

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

# Step6

Trouble Shooting & Observation

* The argument for wget did not work for me. I went to Jenkins homepage to use the installation instructions.

# Diagram

Diagram

Description automatically generated