

CI_CD Project

Ayman Hesham

Email: ayman19982016@outlook.com

Github:

https://github.com/Ayman1231/Booster_CI_CD_Project

Project requirements:

Create CI/CD pipeline using jenkinsfile to deploy simple django web app as a microservice running on docker container locally.

Steps

1- Fork this repo to your account

2- write dockerfile inside the forked repo to create new image from base image ubuntu and install python3.6 and pip3 and copy the source code files of the app to this image and configure it to start the server when creating container (check the below section for steps to start the django server)

3- configure ubuntu slave to use it for the pipeline

4- create slck workspace and integrate it with jenkins

5- install any plugin from your choice to create statistics about builds

6- write jenkinsfile with the following four stages for both dev and master branch

- preparation: checkout the code
- build image: build image using the dockerfile
- push image: push the built image to docker registry(docker hub)
- deploy: deploy a container from the pushed image
- notification: send slack message with the build status

7- configure job in jenkins using multibranch pipeline type with the forked git repo url

Step:1) Fork this repo to your account

The screenshot shows a web browser displaying the GitHub repository page for 'Ayman1231/Booster_CI_CD_Project'. The repository is a fork of 'mahmoud254/Booster_CI_CD_Project'. The page includes a navigation bar with links to 'Pull requests', 'Issues', 'Marketplace', and 'Explore'. Below the navigation bar, the repository name is displayed along with statistics: 0 Watch, 0 Star, and 15 Fork. The 'Code' tab is selected, showing a list of files and their commit history. The files listed are 'simpleApp', 'Dockerfile', 'Jenkinsfile', 'README.md', 'manage.py', and 'requirements.txt'. The commit history for each file is shown, including the commit message and the time since the commit. The right sidebar contains sections for 'About', 'Releases', and 'Packages'.

Ayman1231 / Booster_CI_CD_Project
forked from mahmoud254/Booster_CI_CD_Project

Search or jump to... Pull requests Issues Marketplace Explore

Watch 0 Star 0 Fork 15

<> Code Pull requests Actions Projects Wiki Security Insights Settings

master 2 branches 0 tags Go to file Add file Code

This branch is 39 commits ahead, 1 commit behind mahmoud254:master. Pull request Compare

File	Commit Message	Time
Ayman1231 Update Jenkinsfile	8d093e9	15 hours ago 48 commits
simpleApp	first commit	5 months ago
Dockerfile	Update Dockerfile	17 hours ago
Jenkinsfile	Update Jenkinsfile	15 hours ago
README.md	Update README.md	8 days ago
manage.py	first commit	5 months ago
requirements.txt	Update requirements.txt	16 hours ago

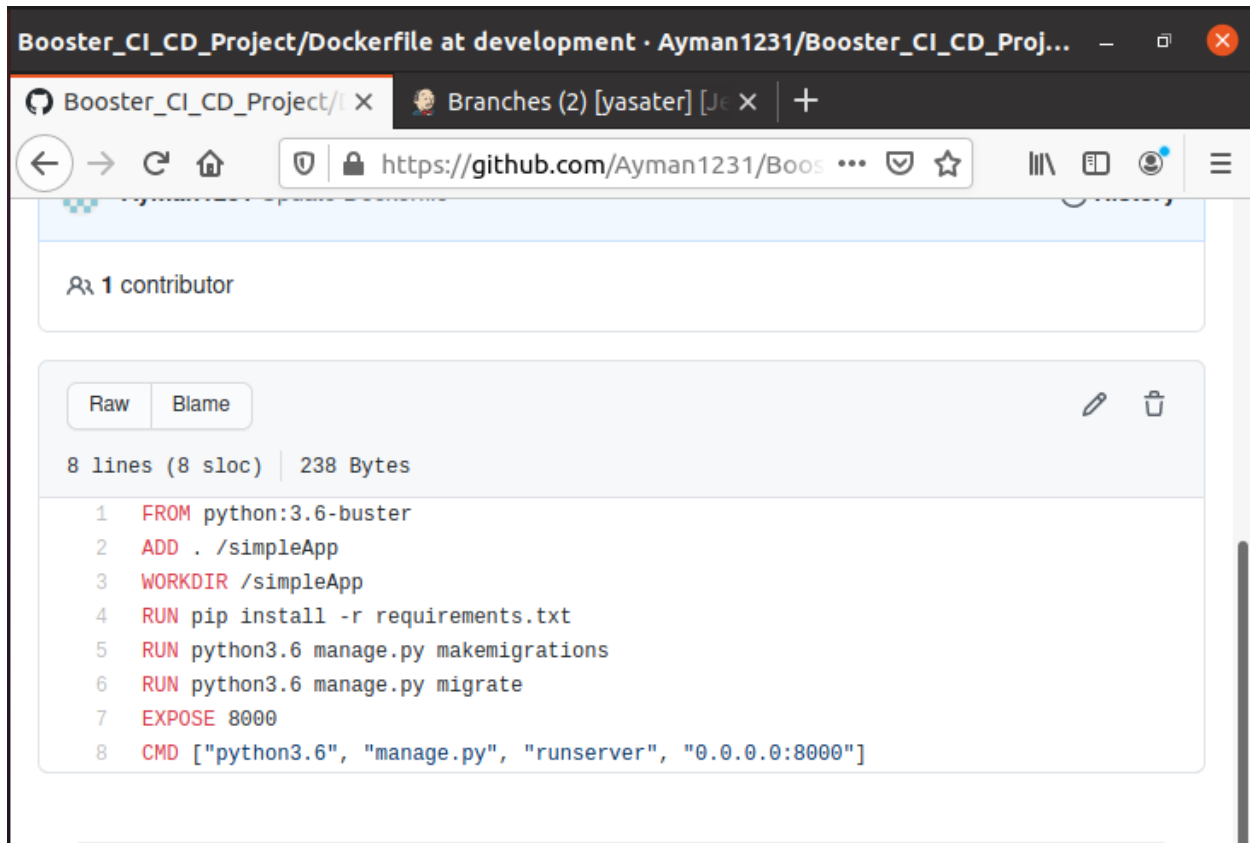
https://github.com/Ayman1231/Booster_CI_CD_Project/security

About
No description, website, or topics provided.
Readme

Releases
No releases published
[Create a new release](#)

Packages
No packages published
[Publish your first package](#)

Step:2) write dockerfile inside the forked repo to create new image from base image ubuntu and install python3.6 and pip3 and copy the source code files of the app to this image and configure it to start the server when creating container (check the below section for steps to start the django server)



The screenshot shows a web browser displaying a GitHub repository page for 'Booster_CI_CD_Project/Dockerfile'. The page title is 'Booster_CI_CD_Project/Dockerfile at development · Ayman1231/Booster_CI_CD_Proj...'. The browser's address bar shows the URL 'https://github.com/Ayman1231/Boos...'. The repository page indicates '1 contributor'. Below this, there are tabs for 'Raw' and 'Blame', with 'Raw' selected. The file size is '8 lines (8 sloc) | 238 Bytes'. The Dockerfile content is as follows:

```
1 FROM python:3.6-buster
2 ADD . /simpleApp
3 WORKDIR /simpleApp
4 RUN pip install -r requirements.txt
5 RUN python3.6 manage.py makemigrations
6 RUN python3.6 manage.py migrate
7 EXPOSE 8000
8 CMD ["python3.6", "manage.py", "runserver", "0.0.0.0:8000"]
```

Step:3) configure ubuntu slave to use it for the pipeline

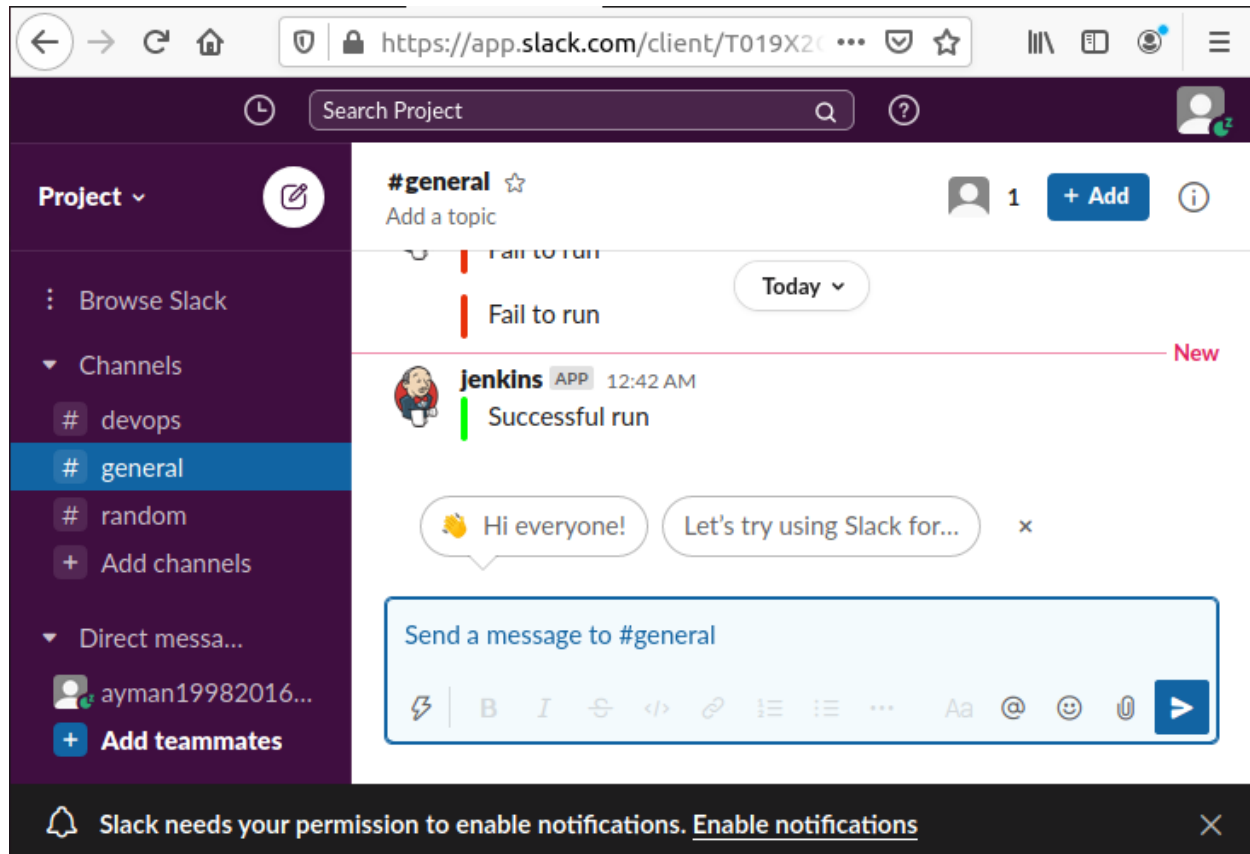
ssh-keygen (to create the private and public key)

cat id_rsa_pub (to put the value of the public key into the slave)

docker inspect bridge (to get the ip address so we can connect the master with the slave)

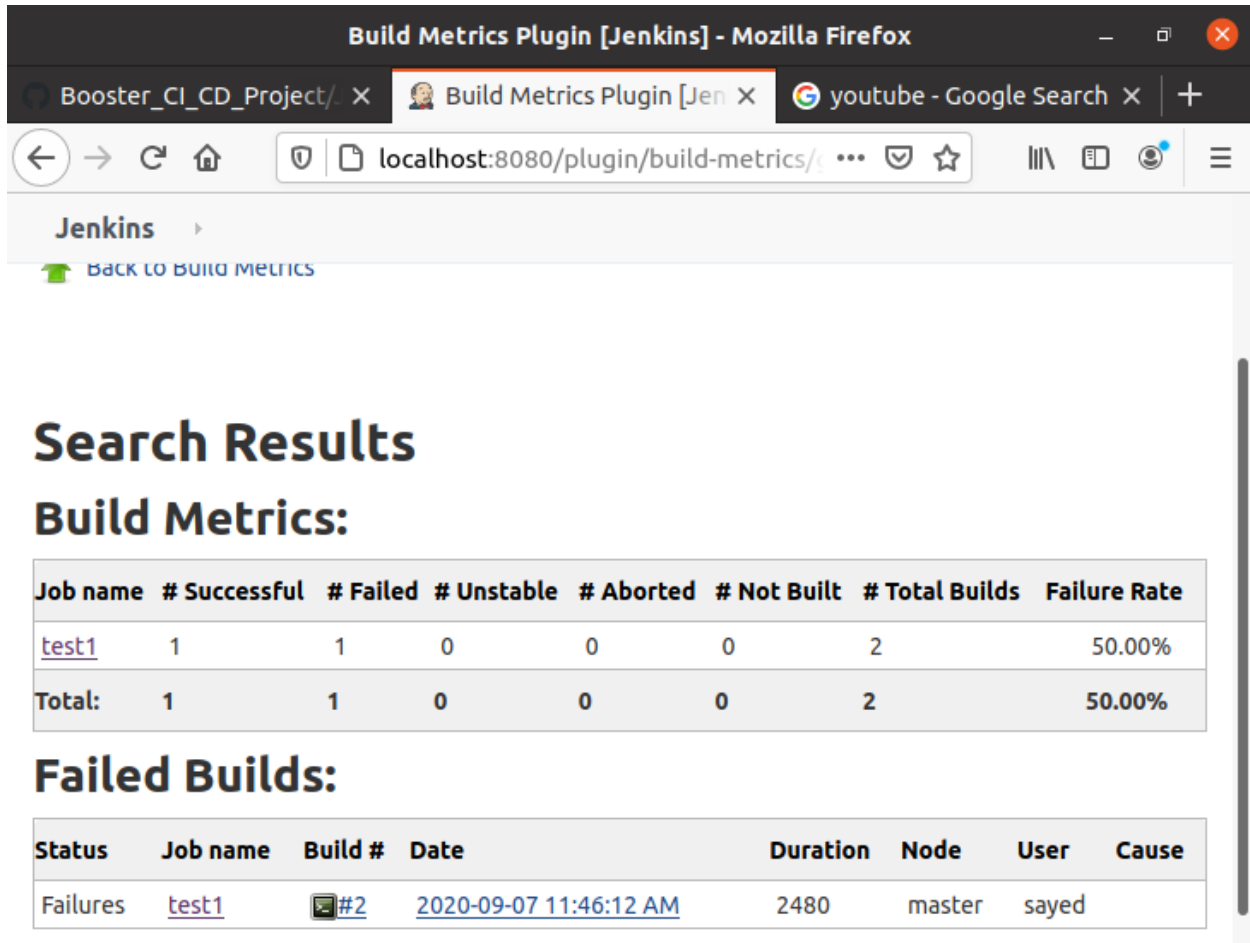
service ssh start

Step:4) create slck workspace and integrate it with jenkins



Step:5) install any plugin from your choice to create statistics about builds

I used the Build Metrics plugin that get the stats of the project (how many builds that are successful and how many of them are failed)



Build Metrics Plugin [Jenkins] - Mozilla Firefox

Booster_CI_CD_Project/ X Build Metrics Plugin [Jen X youtube - Google Search X +

localhost:8080/plugin/build-metrics/ ...

Jenkins


Back to Build Metrics

Search Results

Build Metrics:

Job name	# Successful	# Failed	# Unstable	# Aborted	# Not Built	# Total Builds	Failure Rate
test1	1	1	0	0	0	2	50.00%
Total:	1	1	0	0	0	2	50.00%

Failed Builds:

Status	Job name	Build #	Date	Duration	Node	User	Cause
Failures	test1	 #2	2020-09-07 11:46:12 AM	2480	master	sayed	

Step:6) write jenkinsfile with the following four stages for both dev and master branch

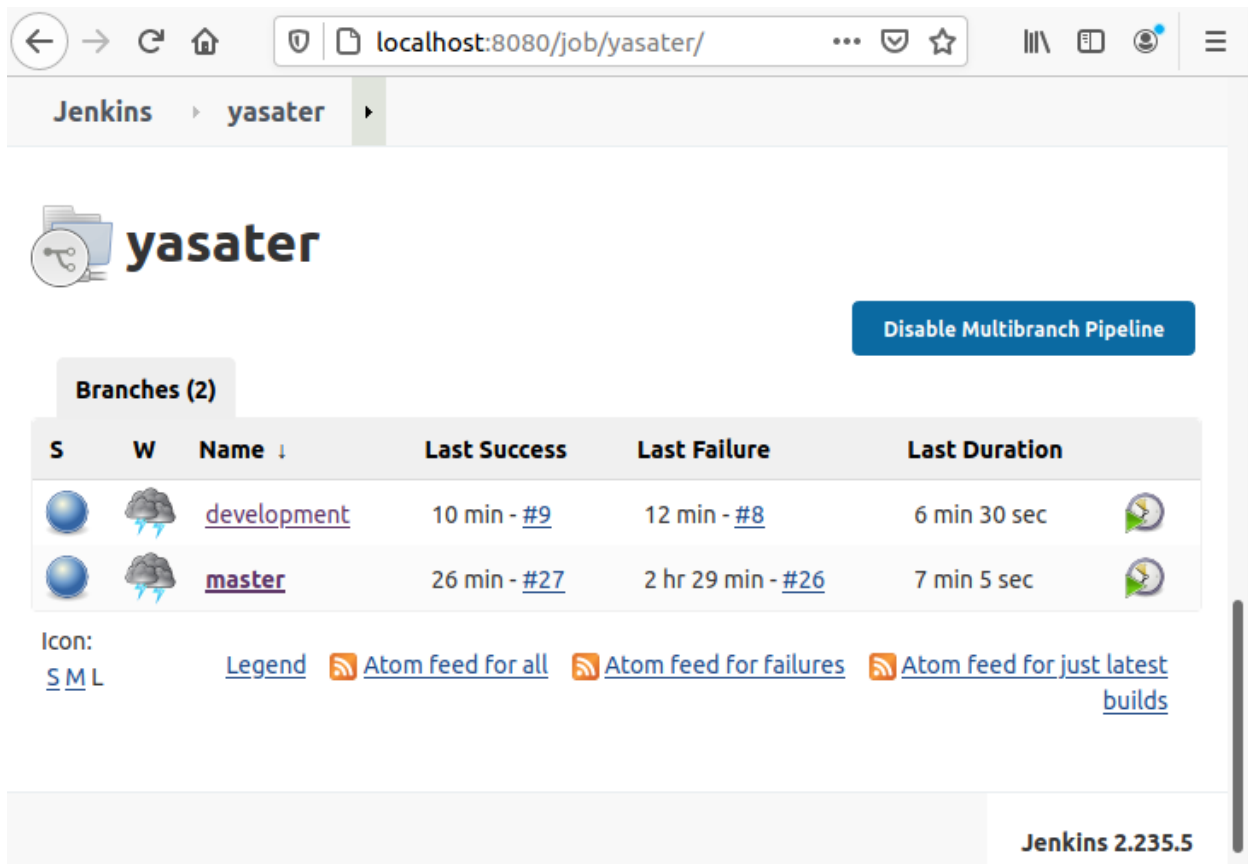
- preparation: checkout the code
- build image: build image using the dockerfile
- push image: push the built image to docker registry(docker hub)
- deploy: deploy a container from the pushed image
- notification: send slack message with the build status









```
1 pipeline {
2   agent any
3
4   stages {
5     stage('Build') {
6       steps {
7         sh 'docker build . -t aymanhesham/myproject:v1.0'
8       }
9     }
10
11     stage('Push') {
12       steps {
13         withCredentials([usernamePassword(credentialsId:"docker",usernameVariab:
14         sh 'docker login --username $USERNAME --password $PASSWORD'
15         sh 'docker push aymanhesham/myproject:v1.0 '
16       }
17     }
18
19     stage('Deploy') {
20       steps {
21         sh 'docker run -d -p 8000:8000 ayman/project'
22       }
23     }
24   }
25 }
```

Booster_CI_CD_Project/Jenkinsfile at development · Ayman1231/Booster_CI_CD_Proj...
Booster_CI_CD_Project/ X Branches (2) [yasater] [Jenkins icon] X +
https://github.com/Ayman1231/Boos ...
18 }
19 }
20 stage('Deploy') {
21 steps {
22 sh 'docker run -d -p 8000:8000 ayman/project'
23 }
24 }
25 }
26 post {
27 success {
28 slackSend (color: '#00FF00' , message: "Successful run")
29 }
30 failure {
31 slackSend (color: '#E83009' , message: "Fail to run")
32 }
33 aborted {
34 slackSend (color: '#E8E200' , message: "Aborted run")
35 }
36 }
37 }
38 }

Step:7) configure job in jenkins using multibranch pipeline type with the forked git repo url

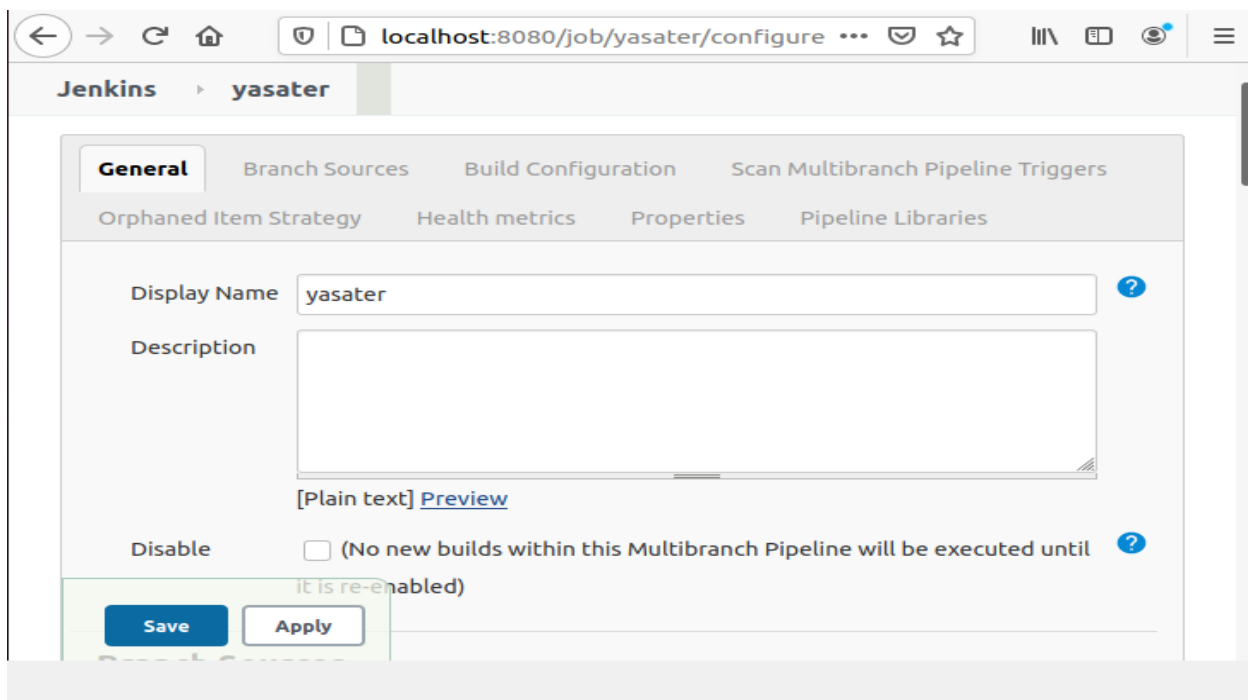


The screenshot shows the Jenkins job page for 'yasater'. The browser address bar indicates the URL is `localhost:8080/job/yasater/`. The page header shows 'Jenkins' and 'yasater'. Below the header, there is a 'Disable Multibranch Pipeline' button. The main section is titled 'Branches (2)' and contains a table with build history for two branches: 'development' and 'master'.

S	W	Name ↓	Last Success	Last Failure	Last Duration
		development	10 min - #9	12 min - #8	6 min 30 sec 
		master	26 min - #27	2 hr 29 min - #26	7 min 5 sec 

Below the table, there is a section for 'Icon: S M L' and a 'Legend' section with links for 'Atom feed for all', 'Atom feed for failures', and 'Atom feed for just latest builds'.

Jenkins 2.235.5



The screenshot shows the Jenkins job configuration page for 'yasater'. The browser address bar indicates the URL is `localhost:8080/job/yasater/configure`. The page header shows 'Jenkins' and 'yasater'. The configuration page has several tabs: 'General', 'Branch Sources', 'Build Configuration', 'Scan Multibranch Pipeline Triggers', 'Orphaned Item Strategy', 'Health metrics', 'Properties', and 'Pipeline Libraries'. The 'General' tab is selected, showing fields for 'Display Name' (set to 'yasater') and 'Description'. There is a 'Disable' checkbox which is currently unchecked, with a note that '(No new builds within this Multibranch Pipeline will be executed until it is re-enabled)'. At the bottom, there are 'Save' and 'Apply' buttons.

Branch Sources

 **Git** X

Project Repository ?

Credentials ?

Behaviors

Discover branches X ?

Build Configuration

Mode

Script Path ?