1 1

Dashboard Python_appReleasePipeline_VOIS #13

- Status
- Changes
- Console Output
  - View as plain text
- Edit Build Information
- Delete build ‘#13’
- Git Build Data
- Restart from Stage
- Replay
- Pipeline Steps
- Workspaces
- Previous Build

Console Output

```
Started by user mac
Obtained Jenkinsfile from git https://github.com/MakariosNassef/Automated-WebApp-on-Amazon-EKS.git
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in
/var/lib/jenkins/workspace/Python_appReleasePipeline_VOIS
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Checkout SCM)
```

```
[Pipeline] checkout
Selected Git installation does not exist. Using Default
The recommended git tool is: NONE
using credential git-credential
 > git rev-parse --resolve-git-dir
/var/lib/jenkins/workspace/Python_appReleasePipeline_VOIS/.git # timeout=10
Fetching changes from the remote Git repository
 > git config remote.origin.url https://github.com/MakariosNassef/Automated-
WebApp-on-Amazon-EKS.git # timeout=10
Fetching upstream changes from https://github.com/MakariosNassef/Automated-
WebApp-on-Amazon-EKS.git
 > git --version # timeout=10
 > git --version # 'git version 2.34.1'
using GIT_ASKPASS to set credentials
 > git fetch --tags --force --progress --
https://github.com/MakariosNassef/Automated-WebApp-on-Amazon-EKS.git
+refs/heads/*:refs/remotes/origin/* # timeout=10
 > git rev-parse refs/remotes/origin/master^{commit} # timeout=10
Checking out Revision 6d4f0621e06d1197b860ba91944c112319b93904
(refs/remotes/origin/master)
 > git config core.sparsecheckout # timeout=10
 > git checkout -f 6d4f0621e06d1197b860ba91944c112319b93904 # timeout=10
Commit message: "edit ingress service name"
 > git rev-list --no-walk 9e4c8bdaa3568dc12424207b244441aae3488549 #
timeout=10
[Pipeline] }
[Pipeline] // stage
[Pipeline] withEnv
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Checkout external proj 🙈🙈🙈)
[Pipeline] git
Selected Git installation does not exist. Using Default
The recommended git tool is: NONE
using credential git-credential
 > git rev-parse --resolve-git-dir
/var/lib/jenkins/workspace/Python_appReleasePipeline_VOIS/.git # timeout=10
Fetching changes from the remote Git repository
 > git config remote.origin.url https://github.com/MakariosNassef/Automated-
WebApp-on-Amazon-EKS.git # timeout=10
Fetching upstream changes from https://github.com/MakariosNassef/Automated-
WebApp-on-Amazon-EKS.git
 > git --version # timeout=10
 > git --version # 'git version 2.34.1'
using GIT_ASKPASS to set credentials
 > git fetch --tags --force --progress --
https://github.com/MakariosNassef/Automated-WebApp-on-Amazon-EKS.git
```

```
+refs/heads/*:refs/remotes/origin/* # timeout=10
 > git rev-parse refs/remotes/origin/master^{commit} # timeout=10
Checking out Revision 6d4f0621e06d1197b860ba91944c112319b93904
(refs/remotes/origin/master)
 > git config core.sparsecheckout # timeout=10
 > git checkout -f 6d4f0621e06d1197b860ba91944c112319b93904 # timeout=10
 > git branch -a -v --no-abbrev # timeout=10
 > git branch -D master # timeout=10
 > git checkout -b master 6d4f0621e06d1197b860ba91944c112319b93904 #
timeout=10
Commit message: "edit ingress service name"
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Build Docker image Python app and push to ecr 🚚📌)
[Pipeline] script
[Pipeline] {
[Pipeline] sh
+ pwd
/var/lib/jenkins/workspace/Python_appReleasePipeline_VOIS
+ cd
/var/lib/jenkins/workspace/Python_appReleasePipeline_VOIS/flask_app/FlaskApp/
+ aws ecr get-login-password --region us-east-1
+ docker login --username AWS --password-stdin 993269114119.dkr.ecr.us-east-
1.amazonaws.com
WARNING! Your password will be stored unencrypted in
/var/lib/jenkins/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
+ docker build -t python_app:app_13 .
#1 [internal] load build definition from Dockerfile
#1 transferring dockerfile:
#1 transferring dockerfile: 205B 0.0s done
#1 DONE 0.1s

#2 [internal] load .dockerignore
#2 transferring context: 2B 0.0s done
#2 DONE 0.0s

#3 [internal] load metadata for docker.io/library/python:3.6
#3 DONE 0.2s

#4 [1/5] FROM
docker.io/library/python:3.6@sha256:f8652afaf88c25f0d22354d547d892591067aa402
6a7fa9a6819df9f300af6fc
```

```
#4 DONE 0.0s

#5 [internal] load build context
#5 transferring context: 754B done
#5 DONE 0.0s

#6 [3/5] COPY requirements.txt /app
#6 CACHED

#7 [4/5] RUN pip install -r requirements.txt
#7 CACHED

#8 [2/5] WORKDIR /app
#8 CACHED

#9 [5/5] COPY . /app
#9 CACHED

#10 exporting to image
#10 exporting layers done
#10 writing image
sha256:77d4f8f9f2c247e3be3ff46459163e88fbe986c1d1fa7199a670613f5dcee498 done
#10 naming to docker.io/library/python_app:app_13 0.0s done
#10 DONE 0.0s
+ docker tag python_app:app_13 993269114119.dkr.ecr.us-east-
1.amazonaws.com/python_app:app_13
+ docker push 993269114119.dkr.ecr.us-east-1.amazonaws.com/python_app:app_13
The push refers to repository [993269114119.dkr.ecr.us-east-
1.amazonaws.com/python_app]
e4b9fea38520: Preparing
f62ea9d06abe: Preparing
abe66d440fba: Preparing
5352e94a0268: Preparing
aa4c808c19f6: Preparing
8ba9f690e8ba: Preparing
3e607d59ef9f: Preparing
1e18e7e1fcc2: Preparing
c3a0d593ed24: Preparing
26a504e63be4: Preparing
8bf42db0de72: Preparing
31892cc314cb: Preparing
11936051f93b: Preparing
8ba9f690e8ba: Waiting
3e607d59ef9f: Waiting
1e18e7e1fcc2: Waiting
c3a0d593ed24: Waiting
26a504e63be4: Waiting
```

```
8bf42db0de72: Waiting
31892cc314cb: Waiting
11936051f93b: Waiting
f62ea9d06abe: Layer already exists
5352e94a0268: Layer already exists
aa4c808c19f6: Layer already exists
abe66d440fba: Layer already exists
e4b9fea38520: Layer already exists
8ba9f690e8ba: Layer already exists
3e607d59ef9f: Layer already exists
1e18e7e1fcc2: Layer already exists
26a504e63be4: Layer already exists
c3a0d593ed24: Layer already exists
8bf42db0de72: Layer already exists
31892cc314cb: Layer already exists
11936051f93b: Layer already exists
app_13: digest:
sha256:090bb3bef83391938aaa3940e0e2a8e3ec950724753968db3cda095359518a13 size:
3053
+ echo Docker Cleaning up 🗑️
Docker Cleaning up 🗑️
+ docker rmi 993269114119.dkr.ecr.us-east-1.amazonaws.com/python_app:app_13
Untagged: 993269114119.dkr.ecr.us-east-1.amazonaws.com/python_app:app_13
Untagged: 993269114119.dkr.ecr.us-east-
1.amazonaws.com/python_app@sha256:090bb3bef83391938aaa3940e0e2a8e3ec950724753
968db3cda095359518a13
[Pipeline] }
[Pipeline] // script
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Build Docker image mysql and push to ecr 🚚📌)
[Pipeline] script
[Pipeline] {
[Pipeline] sh
+ pwd
/var/lib/jenkins/workspace/Python_appReleasePipeline_VOIS
+ cd /var/lib/jenkins/workspace/Python_appReleasePipeline_VOIS/flask_app/db/
+ docker login --username AWS --password-stdin 993269114119.dkr.ecr.us-east-
1.amazonaws.com
+ aws ecr get-login-password --region us-east-1
WARNING! Your password will be stored unencrypted in
/var/lib/jenkins/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
```

```
+ docker build -t python_db:db_13 .
#1 [internal] load .dockerignore
#1 transferring context: 2B 0.0s done
#1 DONE 0.0s

#2 [internal] load build definition from Dockerfile
#2 transferring dockerfile: 240B 0.0s done
#2 DONE 0.1s

#3 [internal] load metadata for docker.io/library/mysql:5.6.51
#3 DONE 0.2s

#4 [1/2] FROM
docker.io/library/mysql:5.6.51@sha256:20575ecebe6216036d25dab5903808211f1e9ba
63dc7825ac20cb975e34cfcae
#4 DONE 0.0s

#5 [internal] load build context
#5 transferring context: 36B done
#5 DONE 0.0s

#6 [2/2] ADD BucketList.sql /docker-entrypoint-initdb.d
#6 CACHED

#7 exporting to image
#7 exporting layers done
#7 writing image
sha256:d67803fe61ac58845c4e1c350c81cb9893114eb6cadd77a85eae807083934f92 done
#7 naming to docker.io/library/python_db:db_13 0.0s done
#7 DONE 0.0s
+ docker tag python_db:db_13 993269114119.dkr.ecr.us-east-
1.amazonaws.com/python_db:db_13
+ docker push 993269114119.dkr.ecr.us-east-1.amazonaws.com/python_db:db_13
The push refers to repository [993269114119.dkr.ecr.us-east-
1.amazonaws.com/python_db]
70b1fe0059fa: Preparing
7137327a7221: Preparing
49a1ca1cd2b8: Preparing
7c5a5c1986b1: Preparing
eba393347f89: Preparing
2612088e90f6: Preparing
e3dce1c82d4e: Preparing
7ea96a4e341b: Preparing
4085e588967d: Preparing
d414fdead0b9: Preparing
2e1029557391: Preparing
2b83e5699838: Preparing
```

```
2612088e90f6: Waiting
e3dce1c82d4e: Waiting
7ea96a4e341b: Waiting
4085e588967d: Waiting
d414fdead0b9: Waiting
2e1029557391: Waiting
2b83e5699838: Waiting
eba393347f89: Layer already exists
49a1ca1cd2b8: Layer already exists
7c5a5c1986b1: Layer already exists
7137327a7221: Layer already exists
70b1fe0059fa: Layer already exists
2612088e90f6: Layer already exists
e3dce1c82d4e: Layer already exists
7ea96a4e341b: Layer already exists
4085e588967d: Layer already exists
d414fdead0b9: Layer already exists
2b83e5699838: Layer already exists
2e1029557391: Layer already exists
db_13: digest:
sha256:f0391b283a206a8af8743079a0c635190cea979eba829b41781784ca9e5f7247 size:
2828
+ echo Docker Cleaning up 🗑️
Docker Cleaning up 🗑️
+ docker rmi 993269114119.dkr.ecr.us-east-1.amazonaws.com/python_db:db_13
Untagged: 993269114119.dkr.ecr.us-east-1.amazonaws.com/python_db:db_13
Untagged: 993269114119.dkr.ecr.us-east-
1.amazonaws.com/python_db@sha256:f0391b283a206a8af8743079a0c635190cea979eba82
9b41781784ca9e5f7247
[Pipeline] }
[Pipeline] // script
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Apply Kubernetes files 🚀 🎉 )
[Pipeline] script
[Pipeline] {
[Pipeline] sh
+ pwd
+ sed -i s|image:.*|image: 993269114119.dkr.ecr.us-east-
1.amazonaws.com/python_app:app_13|g
/var/lib/jenkins/workspace/Python_appReleasePipeline_VOIS/kubernetes_manifest
_file/deployment_flaskapp.yml
+ pwd
+ sed -i s|image:.*|image: 993269114119.dkr.ecr.us-east-
1.amazonaws.com/python_db:db_13|g
/var/lib/jenkins/workspace/Python_appReleasePipeline_VOIS/kubernetes_manifest
```

```
_file/statefulSet_flaskdb.yml
+ aws eks update-kubeconfig --region us-east-1 --name eks
Updated context arn:aws:eks:us-east-1:993269114119:cluster/eks in
/var/lib/jenkins/.kube/config
+ kubectl apply -f
/var/lib/jenkins/workspace/Python_appReleasePipeline_VOIS/kubernetes_manifest
_file
secret/app-secret unchanged
configmap/app-configmap unchanged
secret/db-secret unchanged
deployment.apps/flask-deployment configured
service/flask-ingress-service unchanged
persistentvolumeclaim/flask-pv-claim unchanged
service/flask-service-lb unchanged
namespace/ingress-nginx unchanged
serviceaccount/ingress-nginx unchanged
serviceaccount/ingress-nginx-admission unchanged
role.rbac.authorization.k8s.io/ingress-nginx unchanged
role.rbac.authorization.k8s.io/ingress-nginx-admission unchanged
clusterrole.rbac.authorization.k8s.io/ingress-nginx unchanged
clusterrole.rbac.authorization.k8s.io/ingress-nginx-admission unchanged
rolebinding.rbac.authorization.k8s.io/ingress-nginx unchanged
rolebinding.rbac.authorization.k8s.io/ingress-nginx-admission unchanged
clusterrolebinding.rbac.authorization.k8s.io/ingress-nginx unchanged
clusterrolebinding.rbac.authorization.k8s.io/ingress-nginx-admission
unchanged
configmap/ingress-nginx-controller unchanged
service/ingress-nginx-controller unchanged
service/ingress-nginx-controller-admission unchanged
deployment.apps/ingress-nginx-controller configured
job.batch/ingress-nginx-admission-create unchanged
job.batch/ingress-nginx-admission-patch unchanged
ingressclass.networking.k8s.io/nginx unchanged
validatingwebhookconfiguration.admissionregistration.k8s.io/ingress-nginx-
admission configured
statefulset.apps/mysql-set configured
service/mysql unchanged
persistentvolumeclaim/flaskdb-pv-claim unchanged
ingress.networking.k8s.io/ingress-service configured
service/flask-service unchanged
[Pipeline] }
[Pipeline] // script
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (INGRESS and LB URL 🚀 🎉 )
[Pipeline] script
```

```
[Pipeline] {
[Pipeline] sh
+ echo LB SVC URL 🎉 🎉 🎉
LB SVC URL 🎉 🎉 🎉
+ kubectl get svc flask-service-lb -o jsonpath=
{.status.loadBalancer.ingress[0].hostname}
+ echo ad30dd51cad3c4db2afa8a4074b8c274-1370091485.us-east-
1.elb.amazonaws.com:5002
ad30dd51cad3c4db2afa8a4074b8c274-1370091485.us-east-1.elb.amazonaws.com:5002
+ echo INGRESS URL 🎉 🎉 🎉
INGRESS URL 🎉 🎉 🎉
+ kubectl get ingress ingress-service -o jsonpath=
{.status.loadBalancer.ingress[0].hostname}
+ echo adc734f566e1a40c59e2582bb2a25885-638397776.us-east-1.elb.amazonaws.com
adc734f566e1a40c59e2582bb2a25885-638397776.us-east-1.elb.amazonaws.com
[Pipeline] }
[Pipeline] // script
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

Jenkins 2.387.3