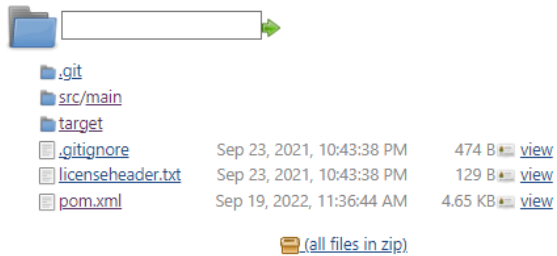


# Jenkins-Maven Project Job

## \*) Workspace

### Workspace of analytics-corner-dev on master



[.git](#)

[src/main](#)

[target](#)

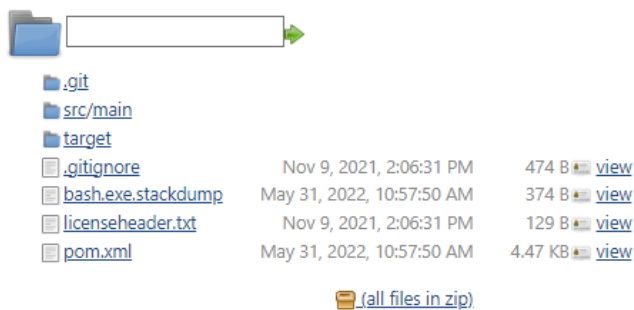
[.gitignore](#) Sep 23, 2021, 10:43:38 PM 474 B [view](#)

[licenseheader.txt](#) Sep 23, 2021, 10:43:38 PM 129 B [view](#)

[pom.xml](#) Sep 19, 2022, 11:36:44 AM 4.65 KB [view](#)

[\(all files in zip\)](#)

### Workspace of analytics-corner-test on master



[.git](#)

[src/main](#)

[target](#)

[.gitignore](#) Nov 9, 2021, 2:06:31 PM 474 B [view](#)

[bash.exe.stackdump](#) May 31, 2022, 10:57:50 AM 374 B [view](#)

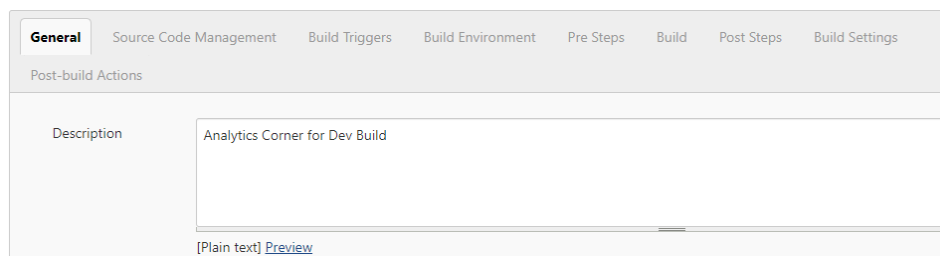
[licenseheader.txt](#) Nov 9, 2021, 2:06:31 PM 129 B [view](#)

[pom.xml](#) May 31, 2022, 10:57:50 AM 4.47 KB [view](#)

[\(all files in zip\)](#)

## \*) Description

Analytics Corner for Dev Build



**General** Source Code Management Build Triggers Build Environment Pre Steps Build Post Steps Build Settings

Post-build Actions

Description Analytics Corner for Dev Build

[Plain text] [Preview](#)

## \*) Discard old builds

Strategy

Log Rotation

Days to keep build: 2

Max # of builds to keep: 2

☒ Discard old builds

Strategy Log Rotation

Days to keep builds 2  
if not empty, build records are only kept up to this number of days

Max # of builds to keep 2  
if not empty, only up to this number of build records are kept

## **\*) Source Code Management**

Git-Repositories-Repository URL

Branches to build \*/develop

Repository browser: (Auto)

**Source Code Management**

☐ None  
☒ Git

Repositories

Repository URL /home/SDMS/backend/microservices/SDMS-analytics-corer-service

Credentials - none - Add

Advanced...  
Add Repository

Branches to build

Branch Specifier (blank for 'any') \*/develop

Add Branch

Repository browser (Auto)

Additional Behaviours Add

## **\*) Build Triggers**

Build whenever a SNAPSHOT dependency is Built

**Build Triggers**

☒ Build whenever a SNAPSHOT dependency is built

☐ Schedule build when some upstream has no successful builds

## **\*)Build**

Root POM

pom.xml

Goals and options

clean install package

**Build**

Root POM:

Goals and options:

[Advanced...](#)

### **\*) Post Steps-Run only if build succeeds-Execute Shell-Command**

**Post Steps**

☒ Run only if build succeeds ☐ Run only if build succeeds or is unstable ☐ Run regardless of build result

Should the post-build steps run only for successful builds, etc.

**Execute shell**

Command

```
scp -o StrictHostKeyChecking=no -r target/* -jpr jenkins@172.21.34.34:/home/jenkins/microback/ac_dev
scp -o StrictHostKeyChecking=no -r /home/trp/backend/microservices/analytics-corner-service/ipr.sh jenkins@172.21.34.34:/home/jenkins/microback/ac_dev
```

[See the list of available environment variables.](#)

Above image does not have the command to send the Dockerfile from jenkins hosted server to destination server

**Post Steps**

☒ Run only if build succeeds ☐ Run only if build succeeds or is unstable ☐ Run regardless of build result

Should the post-build steps run only for successful builds, etc.

**Execute shell**

Command

```
scp -o StrictHostKeyChecking=no -r /home/trp/backend/microservices/analytics-corner-service/Dockerfile jenkins@172.21.34.34:/home/jenkins/microback/ac_test
scp -o StrictHostKeyChecking=no -r target/* -jpr jenkins@172.21.34.34:/home/jenkins/microback/ac_test
scp -o StrictHostKeyChecking=no -r /home/trp/backend/microservices/analytics-corner-service/ipr.sh jenkins@172.21.34.34:/home/jenkins/microback/ac_test
```

### **Send the docker file to a particular directory from local jenkins server to VM**

```
scp -o StrictHostKeyChecking=no -r /home/trp/backend/microservices/analytics-corner-service/Dockerfile jenkins@172.21.34.34:/home/jenkins/microback/ac_dev
```

### **Send the artifact to a particular directory from jenkins workspace to VM**

```
scp -o StrictHostKeyChecking=no -r target/* -jpr jenkins@172.21.34.34:/home/jenkins/microback/ac_dev
```

### **Send ipr.sh file from local jenkins server to VM**

```
scp -o StrictHostKeyChecking=no -r /home/trp/backend/microservices/analytics-corner-service/ipr.sh jenkins@172.21.34.34:/home/jenkins/microback/ac_dev
```

### **Actual Command**

```
scp -o StrictHostKeyChecking=no -r /home/trp/backend/microservices/analytics-corner-service/Dockerfile jenkins@172.21.34.34:/home/jenkins/microback/ac_dev
```

```
scp -o StrictHostKeyChecking=no -r target/* -jpr jenkins@172.21.34.34:/home/jenkins/microback/ac_dev
```

```
scp -o StrictHostKeyChecking=no -r /home/trp/backend/microservices/analytics-corner-service/ipr.sh jenkins@172.21.34.34:/home/jenkins/microback/ac_dev
```

### **\*) Post-build Actions**

Send build artifacts over SSH-SSH Publisher-SSH Server

Name: TCSINOMICS04

## Transfer-Exec command

```
cd microback/ac_dev // Go to the directory
```

```
chmod u+x ipr.sh // Give execute permission to the user
```

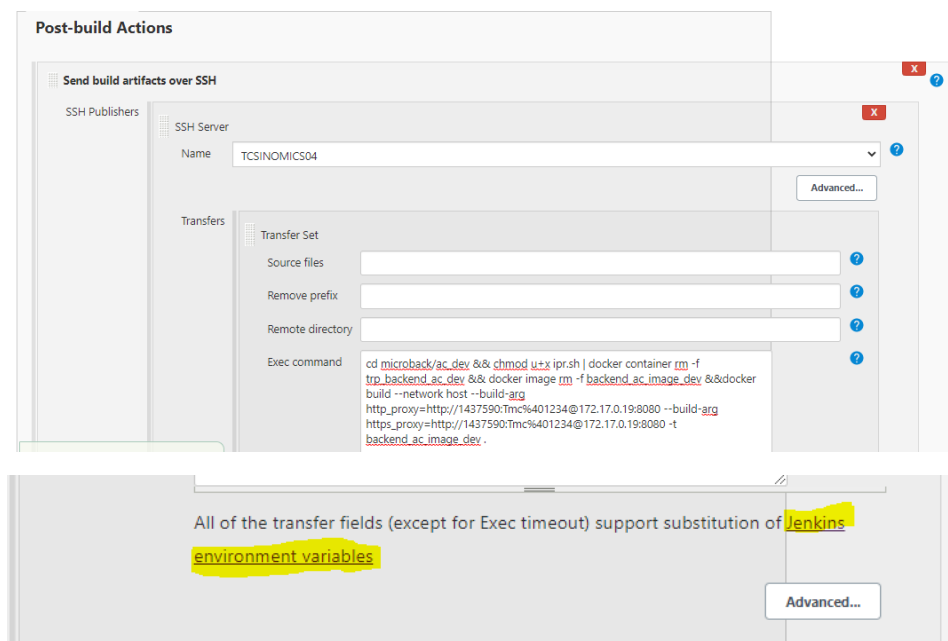
```
docker container rm -f trp_backend_ac_dev // Force remove earlier running container
```

```
docker image rm -f backend_ac_image_dev // Force remove earlier created image
```

```
docker build --network host --build-arg http_proxy=http://1437590:Tmc%401234@172.17.0.19:8080 --  
build-arg https_proxy=http://1437590:Tmc%401234@172.17.0.19:8080 -t backend_ac_image_dev .  
// Build a new image, and networking option as host
```

## Actual Command

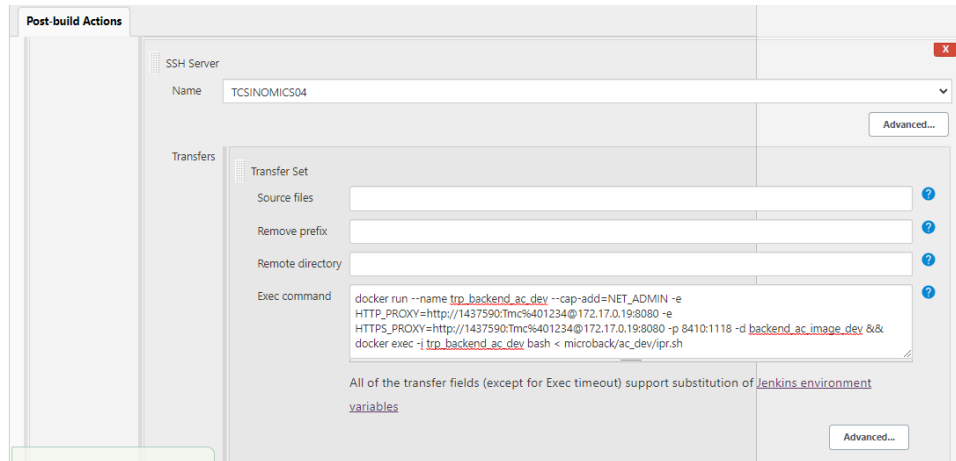
```
cd microback/ac_dev && chmod u+x ipr.sh | docker container rm -f trp_backend_ac_dev && docker  
image rm -f backend_ac_image_dev && docker build --network host --build-arg  
http_proxy=http://1437590:Tmc%401234@172.17.0.19:8080 --build-arg  
https_proxy=http://1437590:Tmc%401234@172.17.0.19:8080 -t backend_ac_image_dev .
```



Name: TCSNOMICS04

## Transfers-Exec command

```
docker run --name trp_backend_ac_dev --cap-add=NET_ADMIN -e  
HTTP_PROXY=http://1437590:Tmc%401234@172.17.0.19:8080 -e  
HTTPS_PROXY=http://1437590:Tmc%401234@172.17.0.19:8080 -p 8410:1118 -d  
backend_ac_image_dev // Run the newly created image to be a container  
docker exec -i trp_backend_ac_dev bash < microback/ac_dev/ipr.sh
```



## Actual Command

```
docker run --name trp_backend_ac_dev --cap-add=NET_ADMIN -e
HTTP_PROXY=http://1437590:Tmc%401234@172.17.0.19:8080 -e
HTTPS_PROXY=http://1437590:Tmc%401234@172.17.0.19:8080 -p 8410:1118 -d
backend_ac_image_dev && docker exec -i trp_backend_ac_dev bash < microback/ac_dev/ipr.sh
```

\*\*\*\*\*

Dockerfile and ipr for dev

FROM java:8

WORKDIR /

COPY ./analytics\*.jar ac.jar

EXPOSE 1118

#CMD java -Xmx1024m -jar es.jar

CMD java -Xmx256m -Dhttps.proxyUser=1437590 -Dhttps.proxyPassword="Tmc@1234" -  
Dhttp.proxyUser=1437590 -Dhttp.proxyPassword="Tmc@1234" -Dhttps.proxyHost=172.17.0.19 -  
Dhttps.proxyPort=8080 -Dhttp.proxyHost=172.17.0.19 -Dhttp.proxyPort=8080 -  
Dhttp.nonProxyHosts="localhost|127.0.0.1|10.\*.\*.\*|172.21.34.\*" -jar ac.jar

```
[root@TCSINOMICS01 analytics-corner-service]# ls
Dockerfile  ipr.sh  readme
[root@TCSINOMICS01 analytics-corner-service]# cat Dockerfile
FROM java:8
WORKDIR /
COPY ./analytics*.jar ac.jar
EXPOSE 1118
#CMD java -Xmx1024m -jar es.jar
CMD java -Xmx256m -Dhttps.proxyUser=1437590 -Dhttps.proxyPassword="Tmc@1234" -Dhttp.proxyUser=1437590 -Dhttp.proxyPassword="Tmc@1234" -D
https.proxyHost=172.17.0.19 -Dhttps.proxyPort=8080 -Dhttp.proxyHost=172.17.0.19 -Dhttp.proxyPort=8080 -Dhttp.nonProxyHosts="localhost|12
7.0.0.1|10.*.*.*|172.21.34.*" -jar ac.jar
[root@TCSINOMICS01 analytics-corner-service]# vi Dockerfile
[root@TCSINOMICS01 analytics-corner-service]#
```

cat ipr.sh--> ip route add 172.17.0.19/32 via 172.17.0.1 dev eth0

```
[root@TCSINOMICS01 analytics-corner-service]# cat ipr.sh
ip route add 172.17.0.19/32 via 172.17.0.1 dev eth0
[root@TCSINOMICS01 analytics-corner-service]#
```