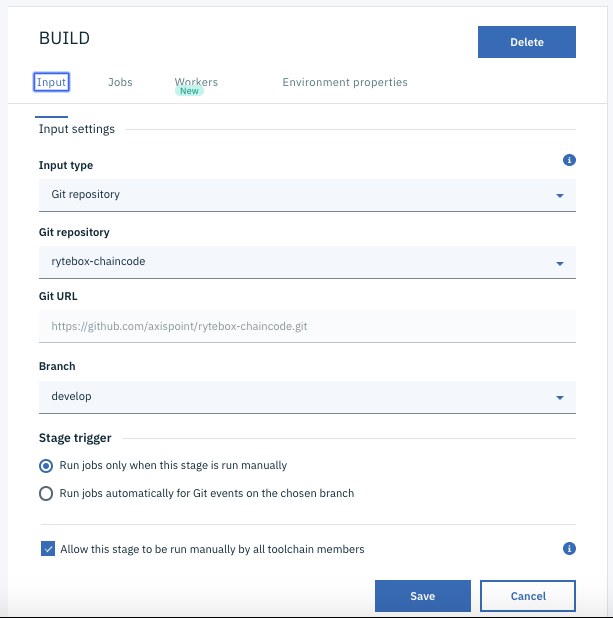
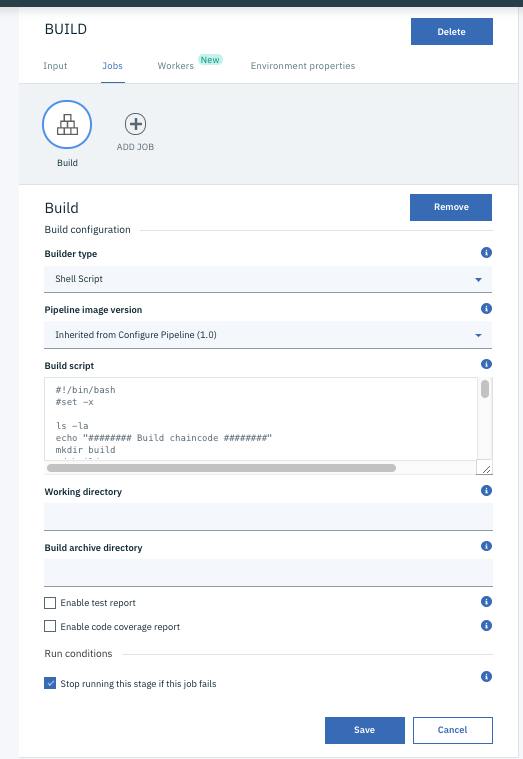
BUILD





#!/bin/bash

#set -x

ls -la

echo "######## Build chaincode ########"

mkdir build

cd build

# Download and install go binaries in current directory

# This script also sets the GOPATH and GOROOT variables used later in this script

echo "######## Begin install and configure Go ########"

sudo apt-get update

sudo apt-get install -y libtool

export GO\_VERSION="1.10.2"

echo "######## Extracting and decompressing Go version ${GO\_VERSION} ########"

pwd

curl -O https://storage.googleapis.com/golang/go${GO\_VERSION}.linux-amd64.tar.gz

tar -xvf go${GO\_VERSION}.linux-amd64.tar.gz

# set location for go executables

export GOROOT=$(pwd)/go

export PATH=${GOROOT}/bin:$PATH

export GOPATH=$(pwd)

echo "GOPATH: ${GOPATH}"

echo "GOROOT: ${GOROOT}"

#Download Hyperledger Fabric src in current directory

#also set the FABRIC\_SRC\_DIR variable used later in this script

ls -la

echo "######## Begin download Fabric ########"

export HLF\_VERSION="1.2.0"

echo "######## Extracting and decompressing Fabric version ${HLF\_VERSION} ########"

curl -O -L https://github.com/hyperledger/fabric/archive/v${HLF\_VERSION}.tar.gz

tar -xvf v${HLF\_VERSION}.tar.gz

export FABRIC\_SRC\_DIR=$(pwd)/fabric-${HLF\_VERSION}

echo "FABRIC\_SRC\_DIR: ${FABRIC\_SRC\_DIR}"

echo "######## Placing source in directory expected by go build ########"

mkdir ${GOPATH}/src

# Let's put fabric source into gopath so that go can resolve dependencies with Fabric libraries

mkdir -p ${GOPATH}/src/github.com/hyperledger

mv ${FABRIC\_SRC\_DIR} ${GOPATH}/src/github.com/hyperledger/fabric

# Copy chaincode into gopath

echo "Copy chaincode into gopath"

cp -pR ../src/axispoint-cc ${GOPATH}/src

ls -la ${GOPATH}/src/axispoint-cc/

echo "######## Building chaincode ########"

go build -v -x axispoint-cc

pwd

set -x

sudo apt-get update

sudo apt-get install -y libtool

sudo apt-get install -y jq

sudo apt-get install zip gzip tar

CC\_VERSION=$(date +%Y%m%d)-${BUILD\_NUMBER}

echo $CC\_VERSION

PROJECT="axispoint-cc"

URL="https://blockchain-starter.ng.bluemix.net"

CHAINCODE\_ARGS=$CC\_VERSION

pwd

ls -la

ls -la src/

cd src

ls -la $PROJECT/

rm -rf $PROJECT/$PROJECT

ls -la $PROJECT/

MYZIP="zip -r $PROJECT.zip $PROJECT"

$MYZIP

ls -la

curl -w '%{http\_code}' -X POST -H 'accept: application/json' -u "org1:$ORG1\_SECRET" -H 'Content-Type: multipart/form-data' -F "files=@$PROJECT.zip;type=application/zip" -F "chaincode\_id=$PROJECT" -F "chaincode\_version=$CC\_VERSION" -F 'chaincode\_type=golang' "$URL/api/v1/networks/$NETWORK\_ID/chaincode/install"

echo ""

echo "Install succeeded, look at the result above"

echo ""

curl -w '%{http\_code}' -X POST -H 'accept: application/json' -u "org2:$ORG2\_SECRET" -H 'Content-Type: multipart/form-data' -F "files=@$PROJECT.zip;type=application/zip" -F "chaincode\_id=$PROJECT" -F "chaincode\_version=$CC\_VERSION" -F 'chaincode\_type=golang' "$URL/api/v1/networks/$NETWORK\_ID/chaincode/install"

echo ""

echo "Install succeeded, look at the result above"

echo ""

BODY="{ \"chaincode\_id\": \"$PROJECT\", \"chaincode\_version\": \"$CC\_VERSION\", \"chaincode\_type\": \"golang\", \"chaincode\_arguments\": [ \"$CC\_VERSION\" ], \"endorsement\_policy\": { \"identities\": [ { \"role\": { \"name\": \"member\", \"mspId\": \"org1\" } }, { \"role\": { \"name\": \"member\", \"mspId\": \"org2\" } } ], \"policy\": { \"1-of\": [ { \"signed-by\": 0 }, { \"signed-by\": 1 } ] } }}"

BODY\_FILE=body.json

echo "$BODY" >"$BODY\_FILE"

cat $BODY\_FILE

curl -w '%{http\_code}' -X POST -H 'accept: application/json' -u "org1:$ORG1\_SECRET" -H 'Content-Type: application/json' -d '@body.json' "$URL/api/v1/networks/$NETWORK\_ID/channels/$CHANNEL\_NAME/chaincode/instantiate"

echo ""

echo "Instatiate succeeded, look at the result above"

echo ""

<https://cloud.ibm.com/devops/pipelines/4567764b-5603-4241-941e-663acf70cfed?env_id=ibm:yp:us-south>