#!/bin/bash

# script to deploy simcluster storm job to CI

set -u -e

cd "$(git rev-parse --show-toplevel)"

# shellcheck source=/dev/null

. "$(git rev-parse --show-toplevel)/devprod/source-sh-setup"

function usage {

cat <<EOF

$0 --env [devel | prod] --dc [atla | pdxa]

Optional:

--dc atla | pdxa

--env devel | prod

EOF

if [ -n "$1" ] && [ "$1" != "noargs" ]; then

echo ""

echo "Invalid app args encountered! Expecting: $1"

fi

}

if [ $# -lt 1 ]; then

usage noargs

exit 1

fi

CLUSTER=

ENV=

USER=cassowary

while [[ $# -gt 1 ]]; do

key="$1"

case $key in

--dc)

CLUSTER="$2"

shift

;;

--env)

ENV="$2"

shift

;;

\*)

# options ignored

;;

esac

shift

done

echo "Bundling..."

JAR\_NAME="tweet-simclusters-storm-job.tar"

JOB\_NAME="summingbird\_simclusters\_v2\_tweet\_alt\_job\_${ENV}"

BASE\_DIR="src/scala/com/twitter/simclusters\_v2/summingbird"

./bazel bundle --bundle-jvm-archive=tar ${BASE\_DIR}:tweet-simclusters-storm-job || exit 1

# initialize the aurora path for a heron job: <dc>/<role>/<env> where <env> can only be devel or prod

AURORA\_PATH=${AURORA\_PATH:="$CLUSTER/$USER/$ENV"}

AURORA\_JOB\_KEY="${AURORA\_PATH}/${JOB\_NAME}"

heron kill "$AURORA\_PATH" "$JOB\_NAME" || true

echo "Waiting 5 seconds so heron is sure its dead"

sleep 5

echo "AURORA\_JOB\_KEY: $AURORA\_JOB\_KEY"

echo "Starting your topology... for ${ENV} ${JOB\_NAME}"

#set -v

heron submit "${AURORA\_PATH}" "dist/${JAR\_NAME}" com.twitter.simclusters\_v2.summingbird.storm.TweetJobRunner --env "$ENV" --dc "$CLUSTER" --alt "alt" --usingLogFavScore