#!/bin/env groovy

/\*

Author: Alex Taylor, Allan Burdajewicz

Since: July 2019

Description: This script stop all or a series of running pipeline jobs

Parameters: None

Scope: Cloudbees Jenkins Platform

\*/

/\* This script can be used to kill off all running jobs which have been running for a certain amount of time. This

script will guaranteed take 30 seconds to run because we want to ensure that each build command has the time needed

to run. It is best to set Jenkins in a Quietdown (Shutdown) mode before running the script \*/

import groovy.time.TimeCategory

import hudson.model.Queue

import org.jenkinsci.plugins.workflow.job.WorkflowRun

import org.jenkinsci.plugins.workflow.support.steps.ExecutorStepExecution

/\*\*

\* Try to infer the WorkflowRun of the executable passed in.

\* @param executable The executable

\* @return The WorkflowRun, or null if this is not a Pipeline run

\*/

WorkflowRun getPipelineRunFromExecutable(Queue.Executable executable) {

if (executable instanceof WorkflowRun) {

return ((WorkflowRun) executable)

}

if (executable.parent instanceof ExecutorStepExecution.PlaceholderTask) {

def executorPlaceholderTask = ((ExecutorStepExecution.PlaceholderTask) executable.parent)

return ((WorkflowRun) executorPlaceholderTask.runForDisplay())

}

return null

}

def doForAllPipelineInProgress = { Closure closure ->

use(TimeCategory) {

def delay = 1.days

def processedPipeline = []

jenkins.model.Jenkins.instanceOrNull.getComputers().each { computer ->

computer.allExecutors.findAll { exec -> exec.isBusy() && exec.currentExecutable && exec.elapsedTime > delay.toMilliseconds() }.each { exec ->

def currentPipelineRun = getPipelineRunFromExecutable(exec.currentExecutable)

if (currentPipelineRun) {

def pipelineRunId = currentPipelineRun.getExternalizableId()

if(!processedPipeline.contains(pipelineRunId)) {

processedPipeline.add(pipelineRunId)

closure(exec, currentPipelineRun)

} else {

" (Already processed ${currentPipelineRun})"

}

}

}

}

}

}

boolean somethingHappened = false

doForAllPipelineInProgress { exec, run ->

println " \* Stopping ${run.fullDisplayName} that spent ${exec.elapsedTime}ms building on ${exec.owner.displayName} #${exec.number}..."

run.doStop()

somethingHappened = true

}

if(somethingHappened) {

somethingHappened = false

sleep(30000)

doForAllPipelineInProgress { exec, run ->

println " \* Forcibly Terminating ${run.fullDisplayName} that spent ${exec.elapsedTime}ms building on ${exec.owner.displayName} #${exec.number}..."

run.doTerm()

somethingHappened = true

}

}

if(somethingHappened) {

somethingHappened = false

sleep(30000)

doForAllPipelineInProgress { exec, run ->

println " \* Forcibly Killing ${run.fullDisplayName} that spent ${exec.elapsedTime}ms building on ${exec.owner.displayName} #${exec.number}..."

run.doKill()

somethingHappened = true

}

}

return