HOWNETFLIX AUTOSCALES CI

Rahul Somasunderam

WHAT DOES CILOOK LIKE AT NETFLIX

JENKINS @ NETFLIX

- 35 Jenkins controllers
- ~45k job definitions
- ~600k builds per week
- 650-1500 agents
- 1-100 executors per agent

THE SPINNAKER VIEW

- 1 Application
- 35 stacks (Controller Clusters)
- 180 Agent Clusters
- 1+ ASG per cluster
- All workloads on AWS

CLUSTERS AND ASGS

- AWS has Auto Scaling Groups
- Spinnaker calls them Server Groups
- <Application>-<Stack>-<Detail> v<Version>
- jenkins-unstable-agent-highlanderv123

HOW TO PLAN FOR CI INFRASTRUCTURE

INFINITE RESOURCES

- Provision capacity based on known maximum load
- Multiply by a safety factor for good measure
- Monitor and change the capacity as load increases



INFINITE PATIENCE

- Plan capacity based on median load
- Builds will sit in queue for long times

INSTANT RESOURCES

- You will get resources as soon as you request for them
- Works well with Containerizable builds
- Not all builds can be containerized
- Does not scale well with large numbers of short-lived builds

AUTOSCALING

- Set up minimum and maximum capacity
- Scale based on some metric

WHAT METRIC TO USE

SYSTEM METRICS

CPU/Memory/Disk IO/Network throughput

 Natively supported by cloud providers and most metrics solutions

Scaling Policies are supported by cloud providers



SYSTEM METRICS

Not very useful for CI

QUEUE DEPTH

Queue Depth seems adequately proportional.

However, it is a trailing metric.

AGENT UTILIZATION

For each agent, find [idle, busy, offline] executors.

Sum these up by ASG.

Compute utilization as

$$\frac{busy + offline}{busy + offline + idle}$$

MEASURING AGENT UTILIZATION

AN AGENT'S ASG

When launching agents, use labels to specify the placement of the agent.



Mark this node temporarily offline

Labels

Agent is connected.

asg:jenkins-unstable-bionic-v189 aws:test:us-east-1:jenkins-unstable-bionic-v189 bionic buildgroup:bionic carson.version:0.767.0 carson:true cloud:aws cluster:jenkins-unstable-bionic detail:bionic ec2.availZone:us-east-1e ec2.instanceType:m5d.xlarge ec2.region:us-east-1 env:test executors:4 iamRole:jenkinsInstanceProfile java.jvm:zulu8

java.runtime:1.8.0_292-b10 nf.account:test nf.app:jenkins nflx.agent.build:569 os.arch:amd64 os.codename:bionic os.distribution:ubuntu os.name:linux os.release:18.04

stack:unstable us-east-1

CAPTURING METRICS

We wrote a custom plugin that plays well with Atlas. You could write one for whatever your metrics capturing service is.

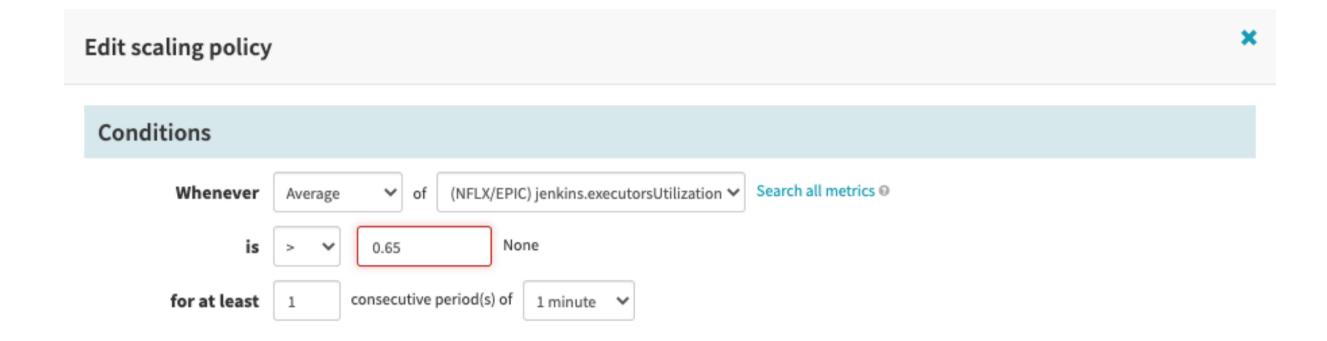
AUTOSCALING

HOW TO AUTOSCALE

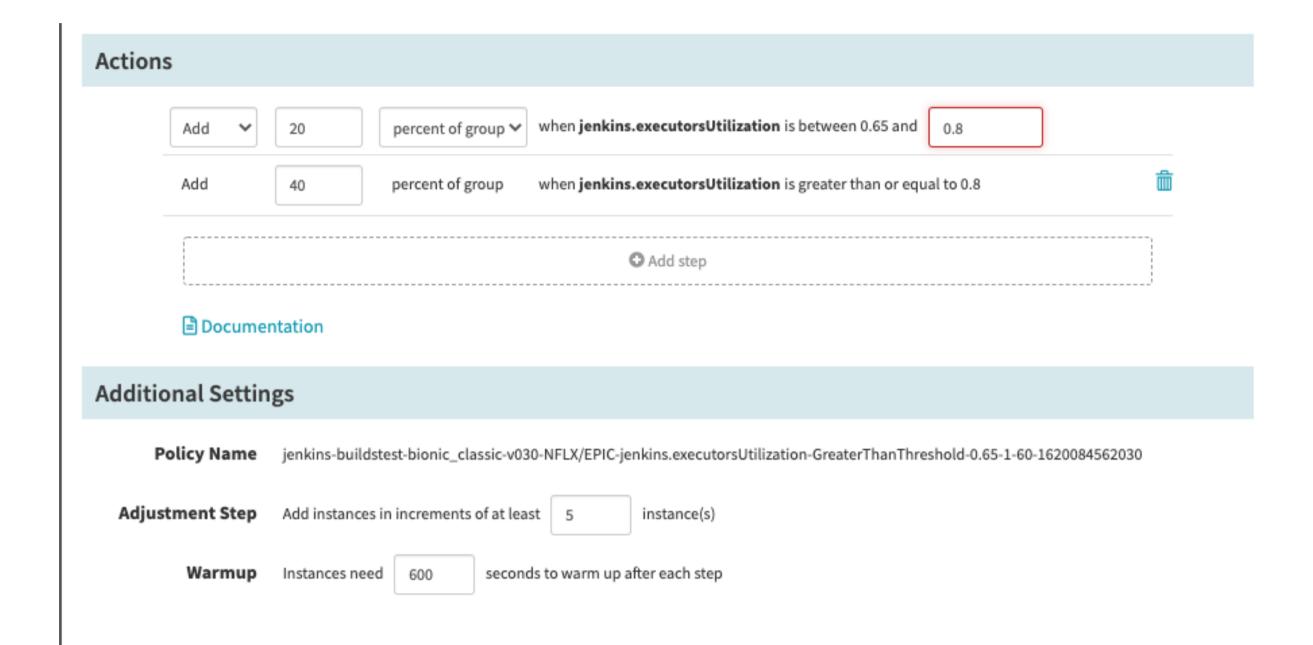
AWS offers 2 ways to scale

- Target Tracking
- Step Scaling

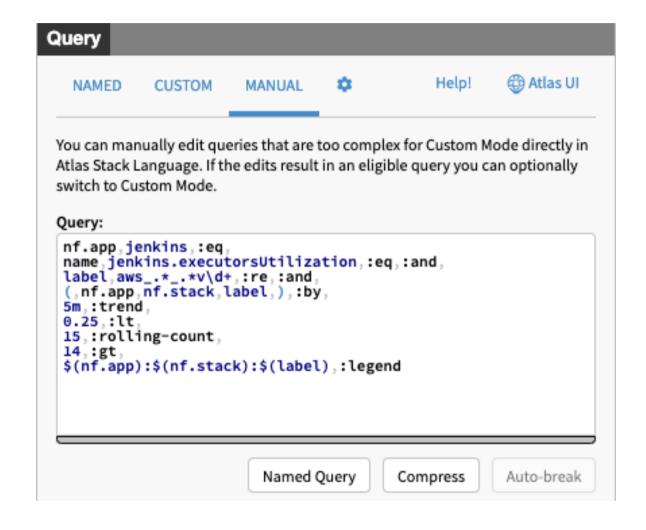
WHEN TO SCALE UP



HOW TO SCALE UP



WHEN TO SCALE DOWN



HOW TO SCALE DOWN

Controller ASG			E	cception				Rto	I	IC	TC	l zo	Cou	nt
jenkin		st/us-east-1/jenkins-mce-bionic_classic-1-v020			+ 	19	20		-+- 	6				6
	- 1	OK i-091aa9055f8dac251												
I	- 1	OK i-08aeaf14573f2653d												
I	- 1	OK i-04414343adb901c59												
I	- 1	OK i-06a513fe9d989f10a												
I	- 1	OK i-0f6e7eec07f0c3421												
I	- 1	OK i-007fe724966b114bc												
	1	Terminate and shrink 6	1		1			1	1				I	

RECAP

WHAT WE LEARNT

- This improved support experience
- This improved the experience for spiky workloads

THANK YOU!

jobs.netflix.com