HOW NETFLIX AUTOSCALES CI

Rahul Somasunderam

WHAT DOES CI LOOK 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

CLUSTERS AND ASGS

- AWS has Auto Scaling Groups

CLUSTERS AND ASGS

- AWS has Auto Scaling Groups
- Spinnaker calls them Server Groups

CLUSTERS AND ASGS

- AWS has Auto Scaling Groups
- Spinnaker calls them Server Groups
- `<Application>-<Stack>-<Detail>-v<Version>`

CLUSTERS AND ASGS

- AWS has Auto Scaling Groups
- Spinnaker calls them Server Groups
- `<Application>-<Stack>-<Detail>-v<Version>`
- `jenkins-unstable-agent-highlander-v123`

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

SYSTEM METRICS

CPU/Memory/Disk IO/Network throughput

- Natively supported by cloud providers and most metrics solutions

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

Queue Depth seems adequately proportional.

QUEUE DEPTH

Queue Depth seems adequately proportional.

However, it is a trailing metric.

AGENT UTILIZATION

AGENT UTILIZATION

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

AGENT UTILIZATION

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

Sum these up by ASG.

AGENT UTILIZATION

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

Sum these up by ASG.

Compute utilization as $\frac{busy + offline}{busy + offline + e}$

MEASURING AGENT UTILIZATION

AN AGENT'S ASG

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

CAPTURING METRICS

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

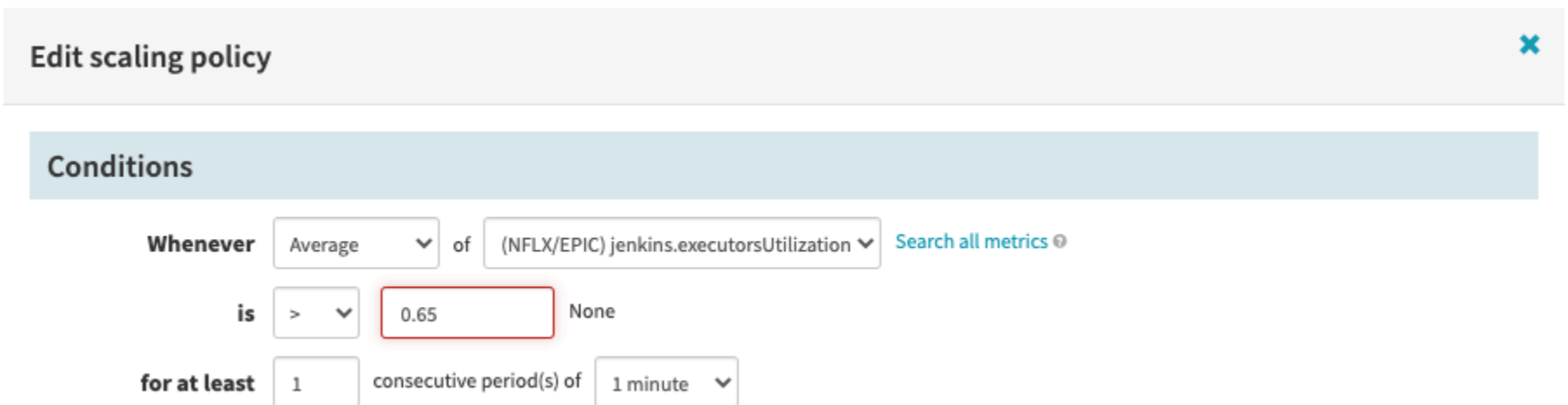
AUTOSCALING

HOW TO AUTOSCALE

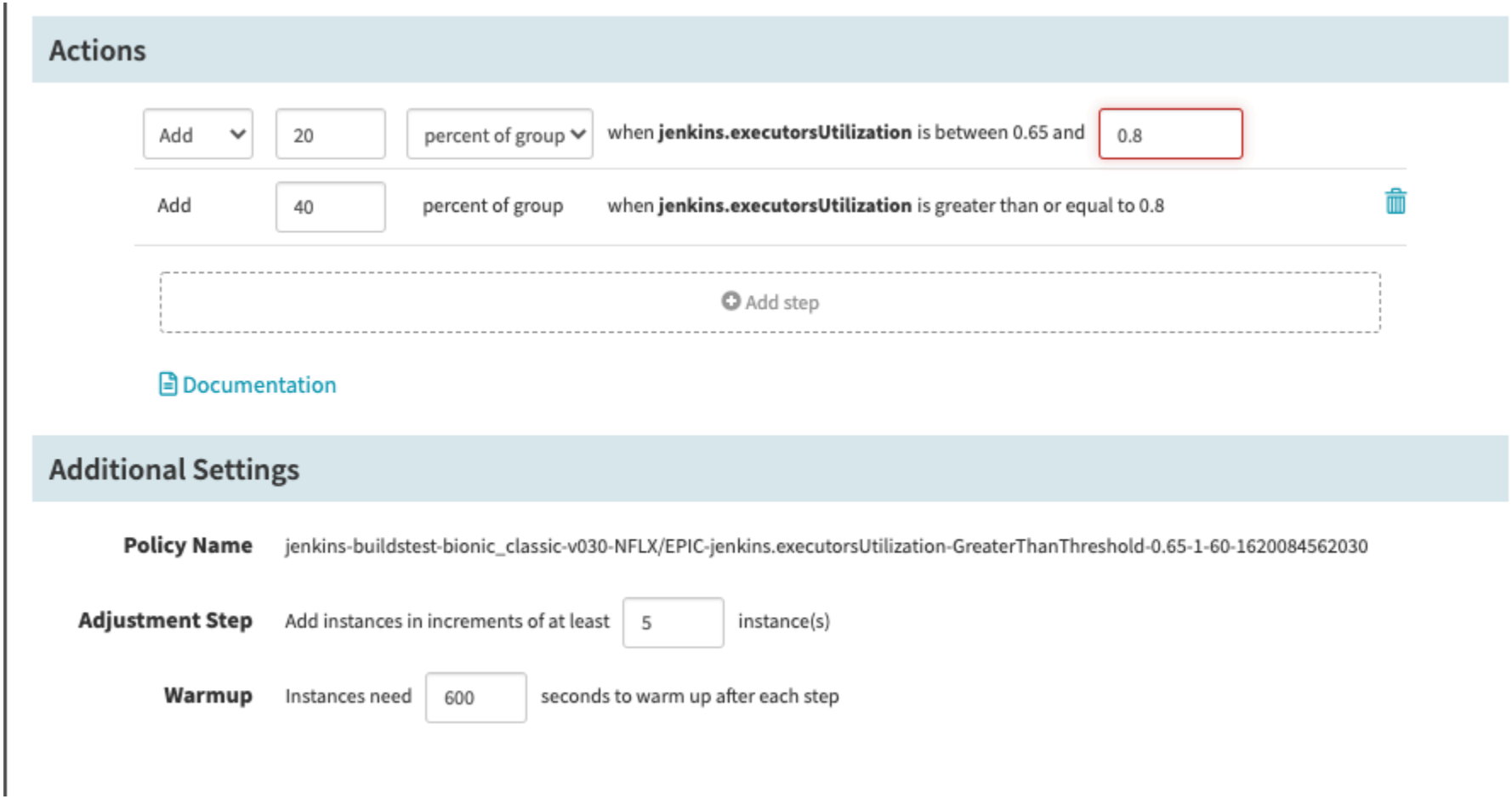
AWS offers 2 ways to scale

- Target Tracking
- Step Scaling

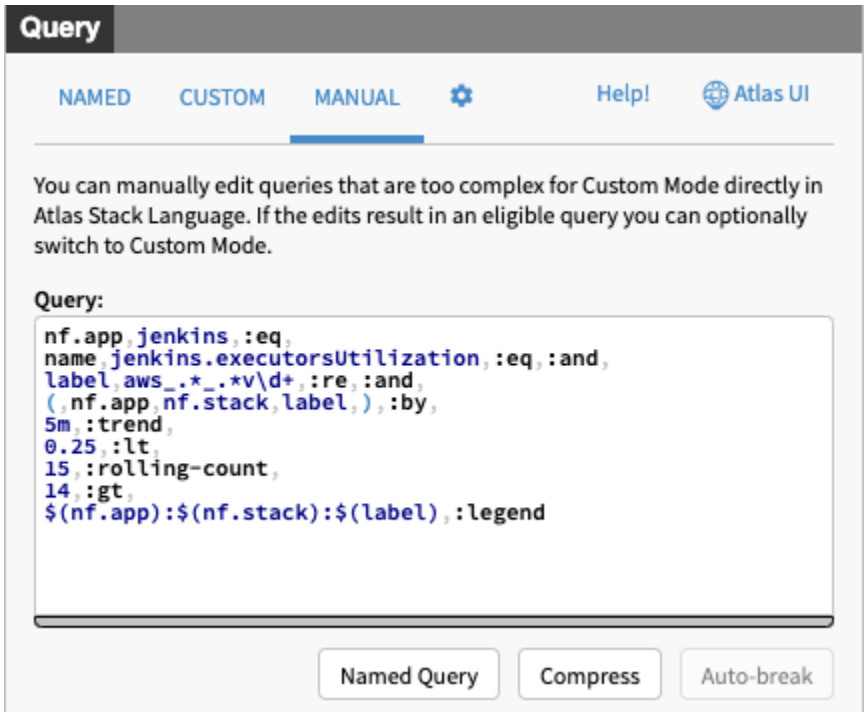
WHEN TO SCALE UP



HOW TO SCALE UP



WHEN TO SCALE DOWN



HOW TO SCALE DOWN

Controller	ASG	Exception	Idl	Tot	Rto	IC	TC	ZC	Count
jenkins/mce	test/us-east-1/jenkins-mce-bionic_classic-1-v020		19	20	6	6	6	6	6
	OK i-091aa9055f8dac251								
	OK i-08aeaf14573f2653d								
	OK i-04414343adb901c59								
	OK i-06a513fe9d989f10a								
	OK i-0f6e7eec07f0c3421								
	OK i-007fe724966b114bc								
	Terminate and shrink 6								

RECAP

WHAT WE LEARNT

WHAT WE LEARNT

- This improved support experience

WHAT WE LEARNT

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

THANK YOU!

jobs.netflix.com