The DynaSlave Plugin

Brian Moyles - Netflix
@bmoyles

NETFLIX

Hi!

NETFLIX

Hi everyone, thanks for sticking around. I’m Brian Moyles and I work on the Netflix Engineering Tools team.

“...and here is your room, Jenkins”

NETFLIX

Last year, we decided it would be a good idea to move our build infrastructure to the cloud to better deal with scalability problems.

At Netflix you are:

- QA
- Release Engineering
- DevOps
- On-call, often 24/7
- Physically take bits from DVD with tweezers, place into fiber optics

NETFLIX

Sunday, September 30, 12

It's important to note that at Netflix, teams generally handle all of their own QA, releng, devops, on-call duties, actually streaming the movies... not really.

The real cloud...

NETFLIX

We all know this is the real cloud...

What We Needed

- Fixed fleet of slaves for 24/7 instant-build coverage, scalable to meet load
- Resilient to the ephemeral nature of the cloud
- As hands-free as possible. Simple.

NETFLIX

Sunday, September 30, 12

Of course, the cloud is ephemeral, and if you don't think of it as such, the agents of chaos will help you out with that. We needed a fixed fleet of always-on slaves for instant building (no lag waiting for a node to be provisioned), we needed whatever we ended up with to deal with the ephemeral nature of the cloud (don't page me if a build node goes away, just replace it and leave me to my 20 minutes of sleep), and we needed it to be as close to hands-free as possible (with slaves coming/going, teams having specialized needs, we don't want to have to micromanage the slaves).

Lets bend the Jenkins slave setup a bit...

Slaves register themselves...

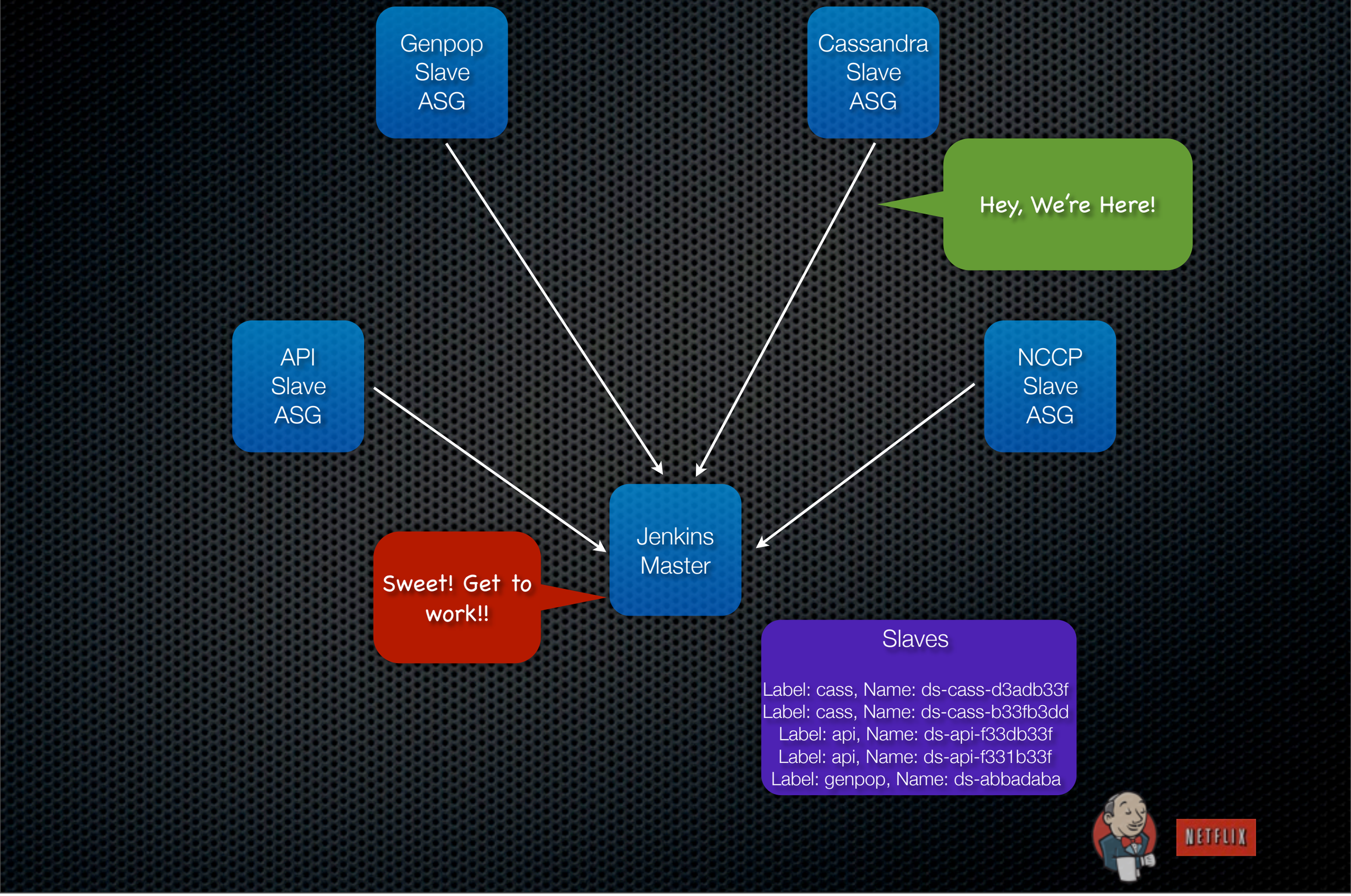
NETFLIX

We figured a good way to manage that would be to utilize some Amazon functionality we had spent the last few years wrangling. Slaves go in Amazon autoscaling groups, which are basically clusters of redundant nodes automatically managed by Amazon. Nodes disappear, they get replaced. Need more? Set a max, nodes automatically launch. Want slaves to disappear on a schedule? Okay. We could even feed back some metrics into Amazon to influence cluster size... But... now we have to give Jenkins some EC2 credentials, make it aware of EC2 itself, make it aware of our internal conventions... So why not make it simple and let the slaves register themselves?

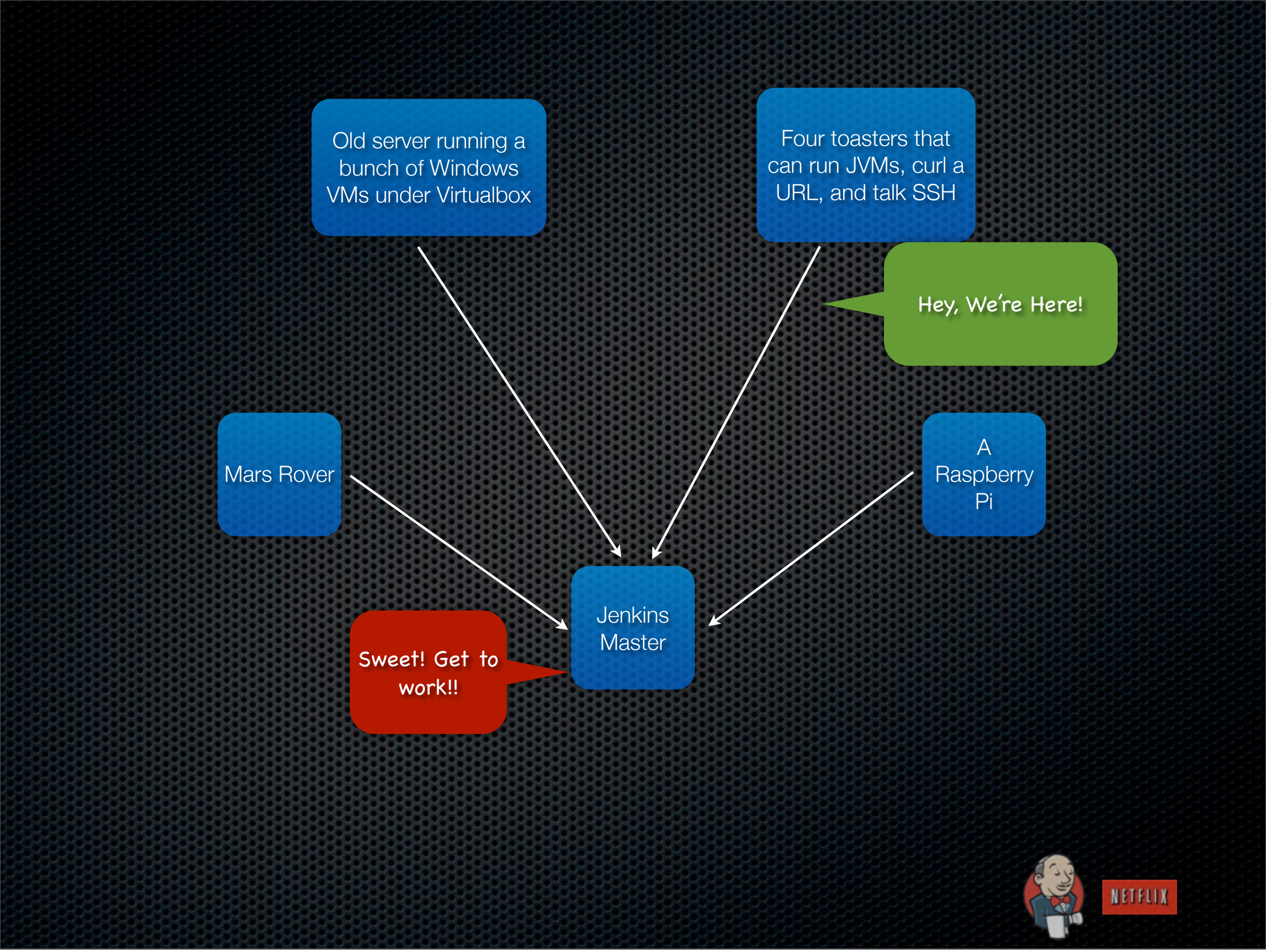
The DynaSlave Plugin

NETFLIX

We started with the swarm plugin as a base and ended up with the DynaSlave plugin. Today, it merely exposes a URL that slaves can poll to tell Jenkins they're around and need some work. Jenkins creates its internal representations of the slaves, uses a script to push out some bits that may have changed since their images were baked, and starts up slave.jar. We can triple the size of our general fleet in 5-10 minutes depending on how much we want or need to push to the slave once it's alive.



No presentation is complete without some sort of diagram.
Pretend you see autoscaling groups configured to use a standard image we assemble for all slaves to use.
Autoscaling groups are given meaningful names, and we can use those group names as labels in Jenkins. Jobs then tie to those labels.
Node dies? Amazon relaunches. We want to scale up for an event? Tell Amazon to grow. Or maybe have Jenkins tell Amazon we've got a backlog of jobs and could use some more horsepower... When we're quiet, we can tell Amazon to kill off a few nodes (via our console, Asgard, available on our Github site :))



Sunday, September 30, 12

In case it wasn't obvious, one interesting by-product of this is the plugin actually doesn't have to know about Amazon (at least today). Or any cloud, for that matter... If you can hit a url, run a JVM, you likely can use the Dynaslave to let those machines register themselves. You don't have to update a plugin or do much at all in the way of tweaking Jenkins itself if you have a new cloud provider, new OS, and so on.

The DynaSlave...Freed

github.com/netflix-skunkworks/dynaslave-plugin

NETFLIX

It's simple, and it's been useful to us, so we'd like others to give it a spin and perhaps help us make it better.
We're already looking at things like using the built-in cloud abstractions to handle grouping and automatically scaling better (things we handle now with naming conventions and system groovy scripts), but there are likely other great ideas we haven't thought of.
You can take a peek today at the github url above, and we'll be working with the Jenkins dev community to get it under the Jenkinsci org soon.

Further Reading

http://jobs.netflix.com/

http://netflix.github.com/

http://techblog.netflix.com/

http://www.slideshare.net/netflix

https://github.com/netflix-skunkworks/dynaslave-plugin/

@netflixoss - Netflix OSS

@bmoyles - Brian Moyles

@garethbowles - Gareth Bowles

NETFLIX

Thanks again for listening, hopefully this sounded interesting. These links will lead to happiness and prosperity and contain all you seek :)

Thank You To Our Sponsors

CloudBees

CLOUDANT

Sonatype

sauce LABS

salesforce

SendGrid
Email Delivery. Simplified.

LIFERAY

New Relic.

ZEROTURNAROUND

YAHOO!

hp

WANDISCO

NETFLIX