
Using Docker for fun + profit

A story about (more generally) deploying and managing
containers in prod



Me

   @andrewhowdencom

Anything useful will be posted to Twitter.



Software engineer @ Sitewards

--

“Certified Kubernetes Administrator”
· Playing with containers since 2015

Please Interrupt.



Please Contribute!



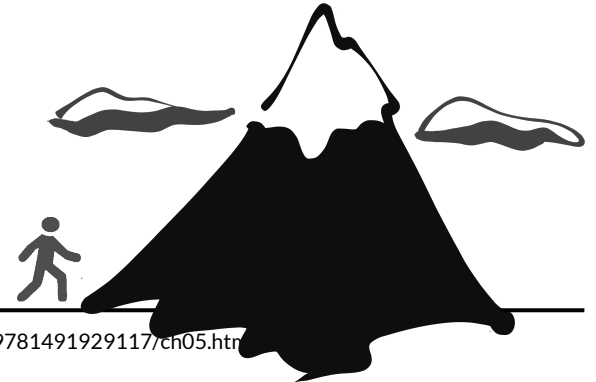
So who's used containers



The problem



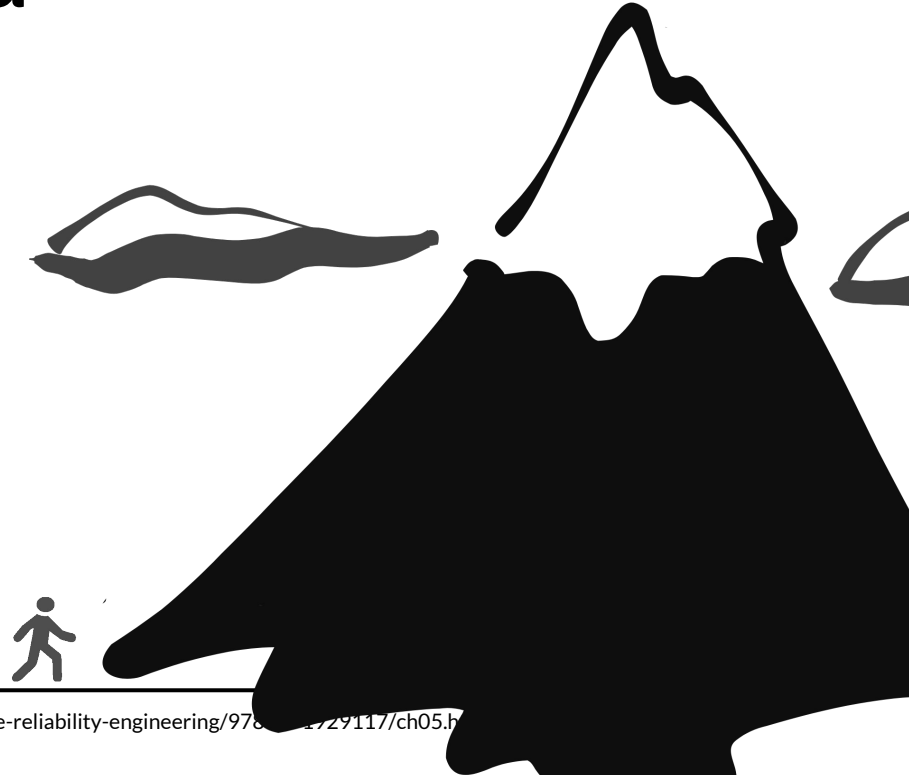
Building a server is hard



Maintaining servers is super hard



Maintaining servers for years is insanely hard



**Introducing change at “scale”
(30+ machines) is damned near
impossible.**



Solution

Containers!

Containers

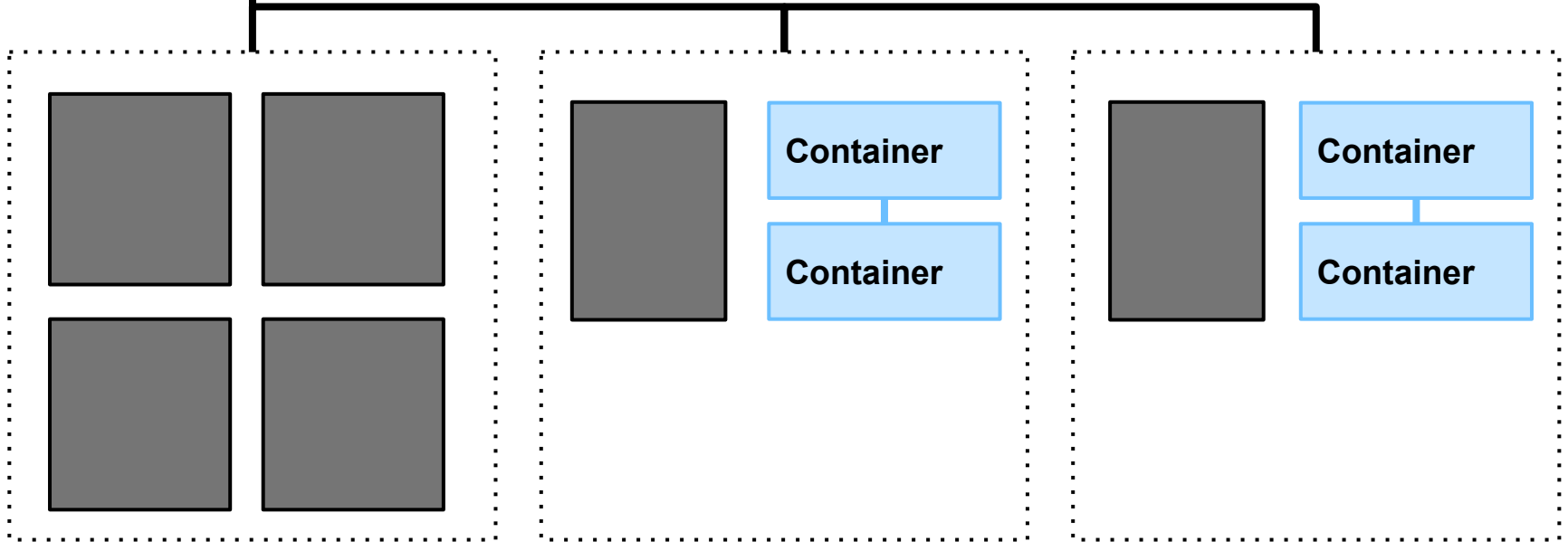
Basically fancy tarballs

- Me

<https://www.opencontainers.org/>

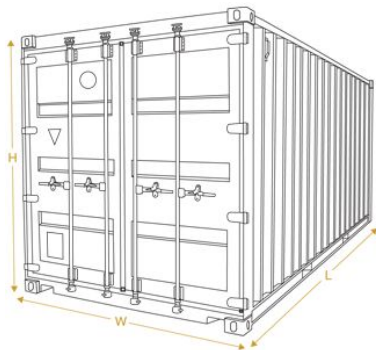


Architecture Diagram



The promise of a container





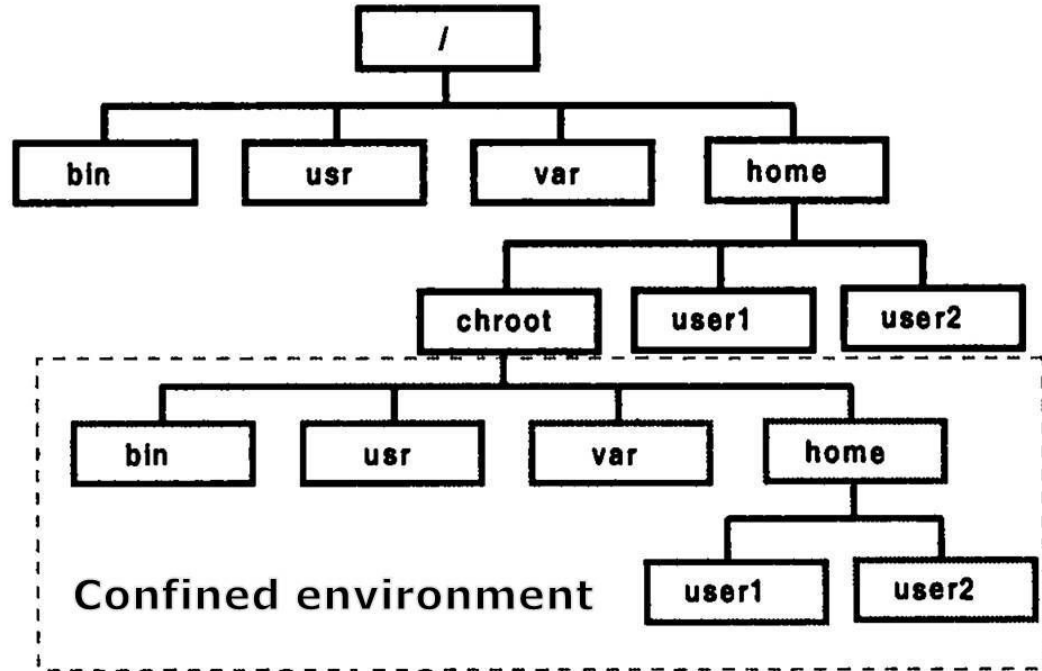
What is a container even

chroot • process namespace • network namespace • control group (cgroup)

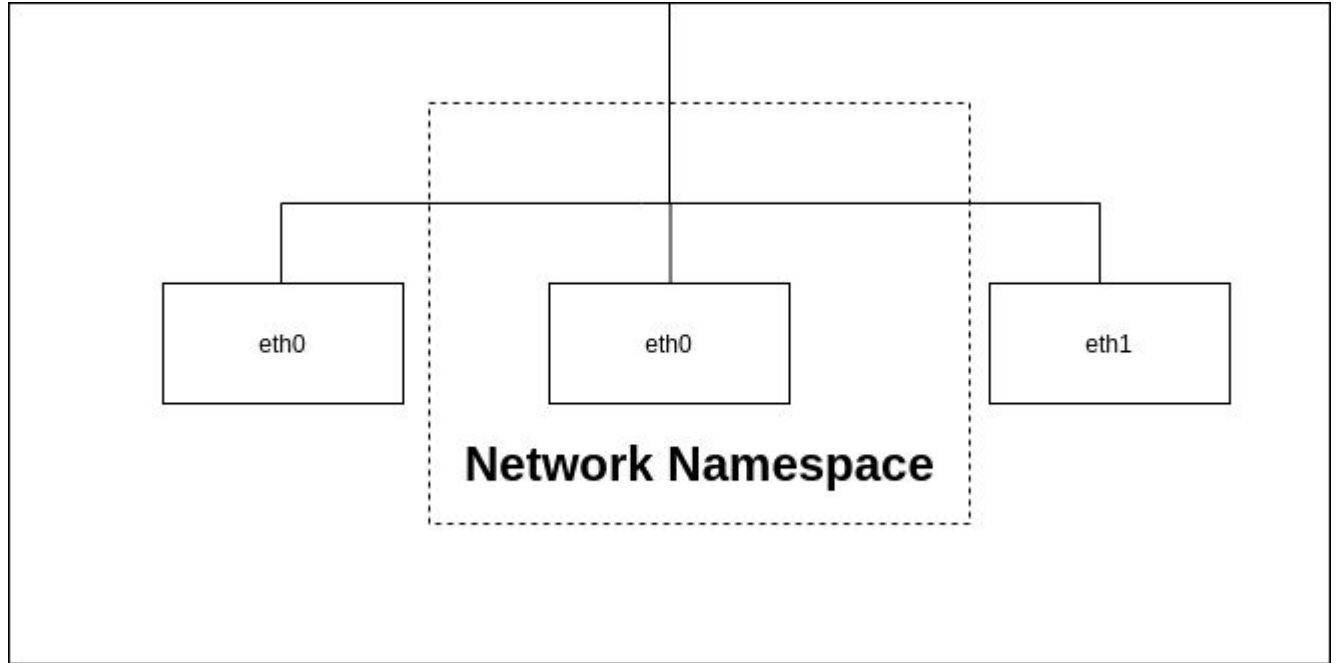
TL, DR - It's basically a “cheap” version of a virtual machine.



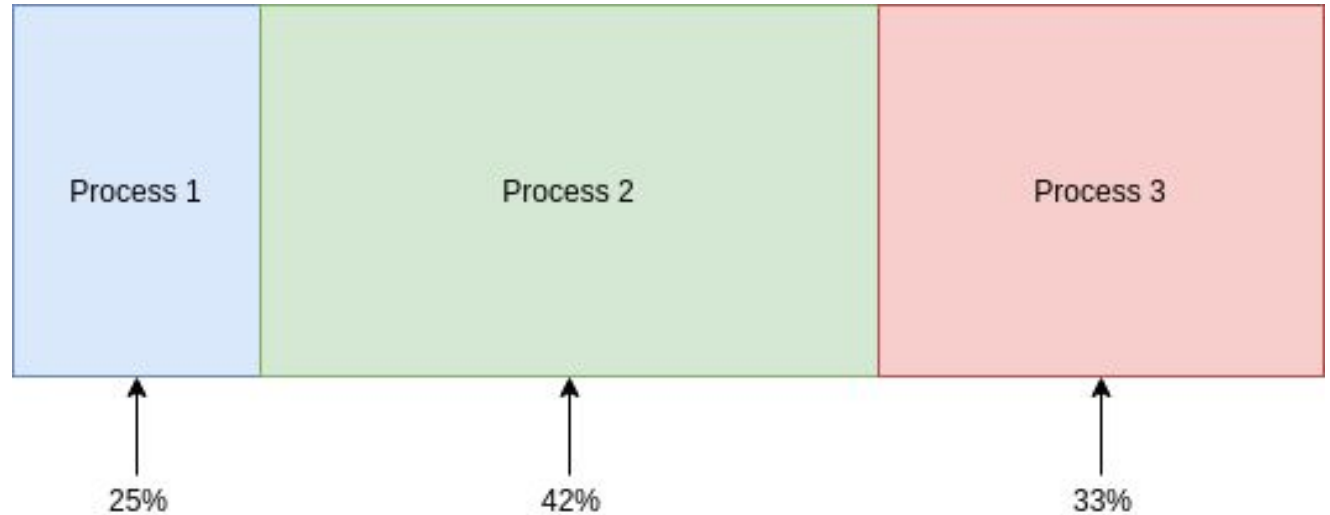
chroot

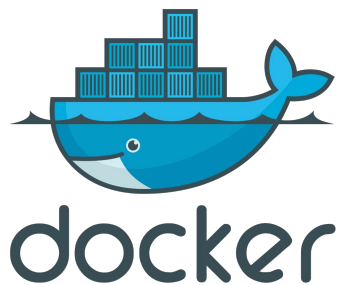


Namespaces



Control Group





A nice wrapper around containers.
Not the only one.

<https://www.docker.com/> · <https://github.com/coreos/rkt> ·
<https://linuxcontainers.org/> · <https://github.com/opencontainers>



Uhhm what do I do
with it



Who has played snake



Fun!



Build

Locally, or in CI build an “image” - a deployable container



Push

Push the image to a “registry” - a place to put these containers until they’re deployed



Deploy

Update the deployment configuration so it uses the new container



Build

```
FROM nginx:latest
```

```
# Define the snake repo in a handy environment variable for easy replacement
ENV SNAKE_REPO="https://github.com/PKief/Snake.git"
```

```
# Install git so we can download the snake repo
RUN apt-get update && \
    apt-get install --yes \
        git
```

```
# Clone the repo to a directory that NGINX can access
RUN git clone ${SNAKE_REPO} /var/www/snake
```

```
# Modify NGINX configuration to serve our snake game
RUN sed --in-place 's/usr/share/nginx/html/var/www/snake/' /etc/nginx/conf.d/default.conf
```



Push

Repository Tags

Compact Expanded



1 - 25 of 78

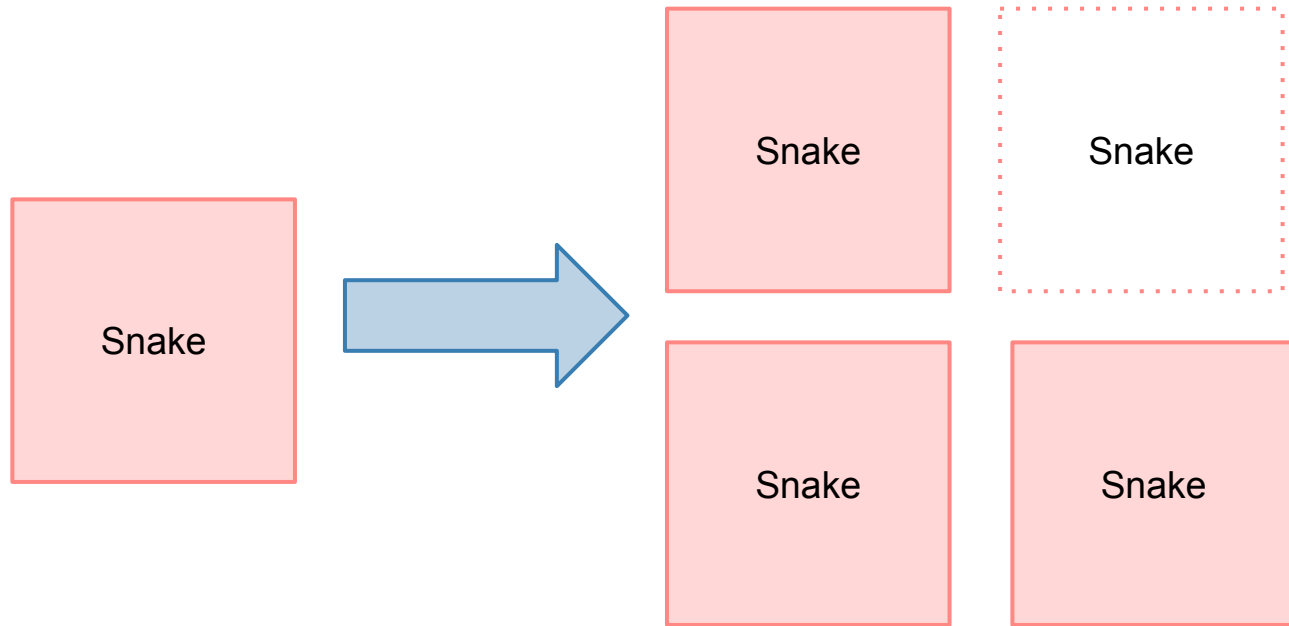


Filter Tags...

TAG	SIGN	LAST MODIFIED ↓	SECURITY SCAN	SIZE	IMAGE
<input type="checkbox"/> 7.1-latest		19 hours ago	13 High	136.6 MB	SHA256 925715a21057
<input type="checkbox"/> 7.0-latest		19 hours ago	15 High	136.3 MB	SHA256 928eb654719e
<input type="checkbox"/> 5.6-latest		19 hours ago	13 High	140.9 MB	SHA256 f2cb2ca6ee21
<input type="checkbox"/> 7.1.15-1		2 months ago	24 High • 45 fixable	184.2 MB	SHA256 c31a08bbf13e
<input type="checkbox"/> ad-hoc-reduce-duplica...		5 months ago	67 High • 143 fixable	167.9 MB	SHA256 68c688918f65



Deploy



Deploy to what?



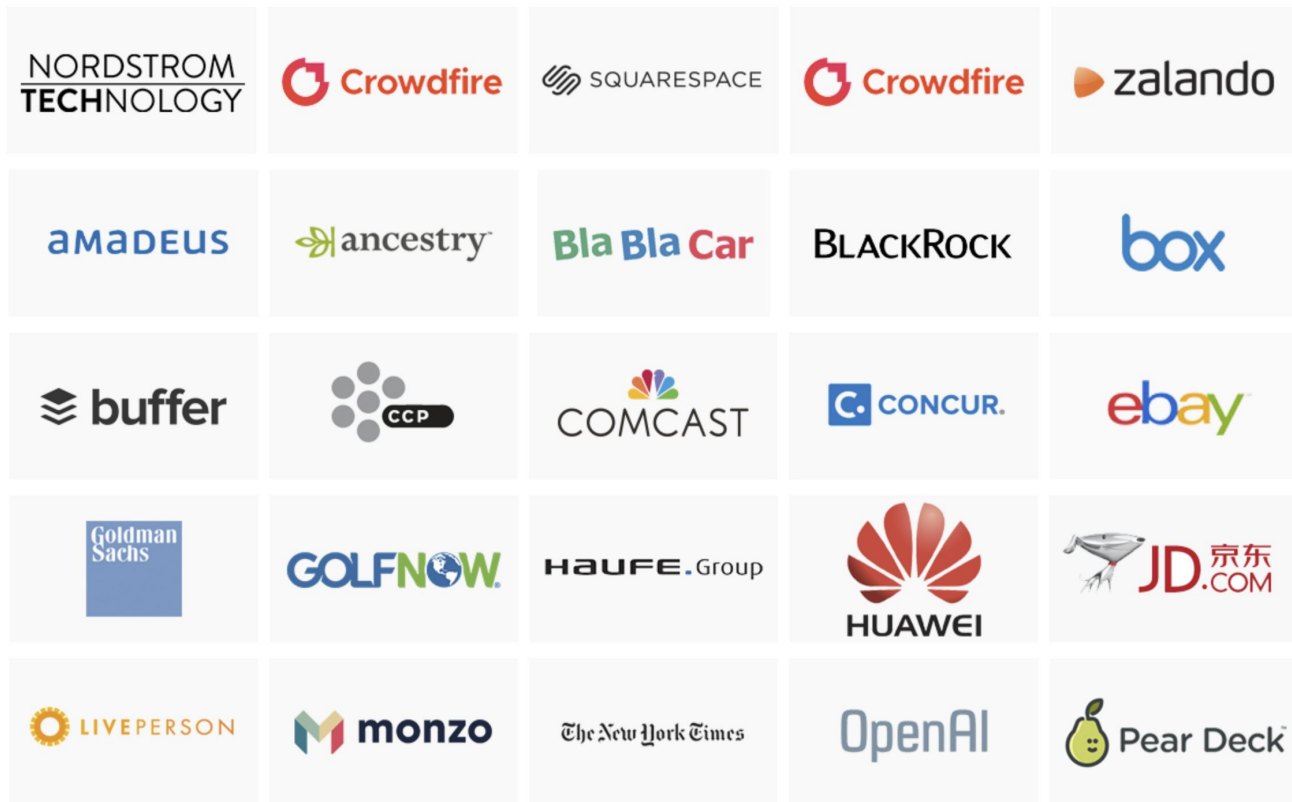


Take a bunch of machines,
make them a single logical
machine

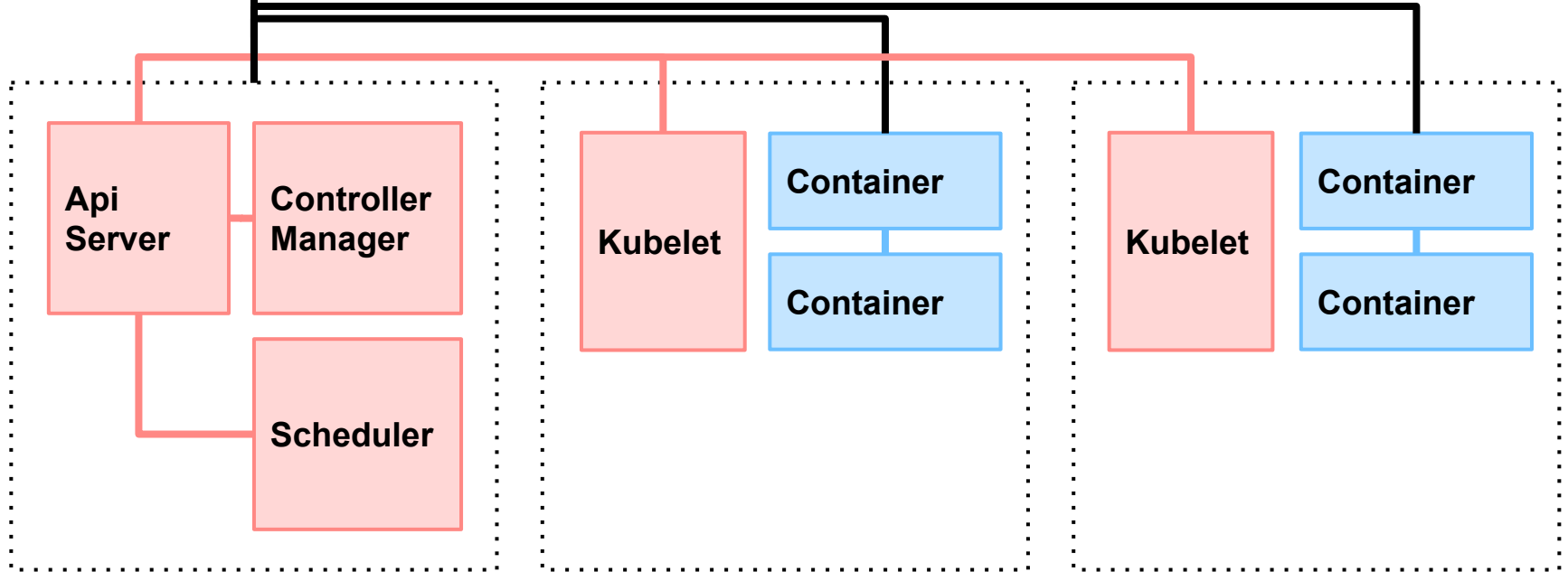


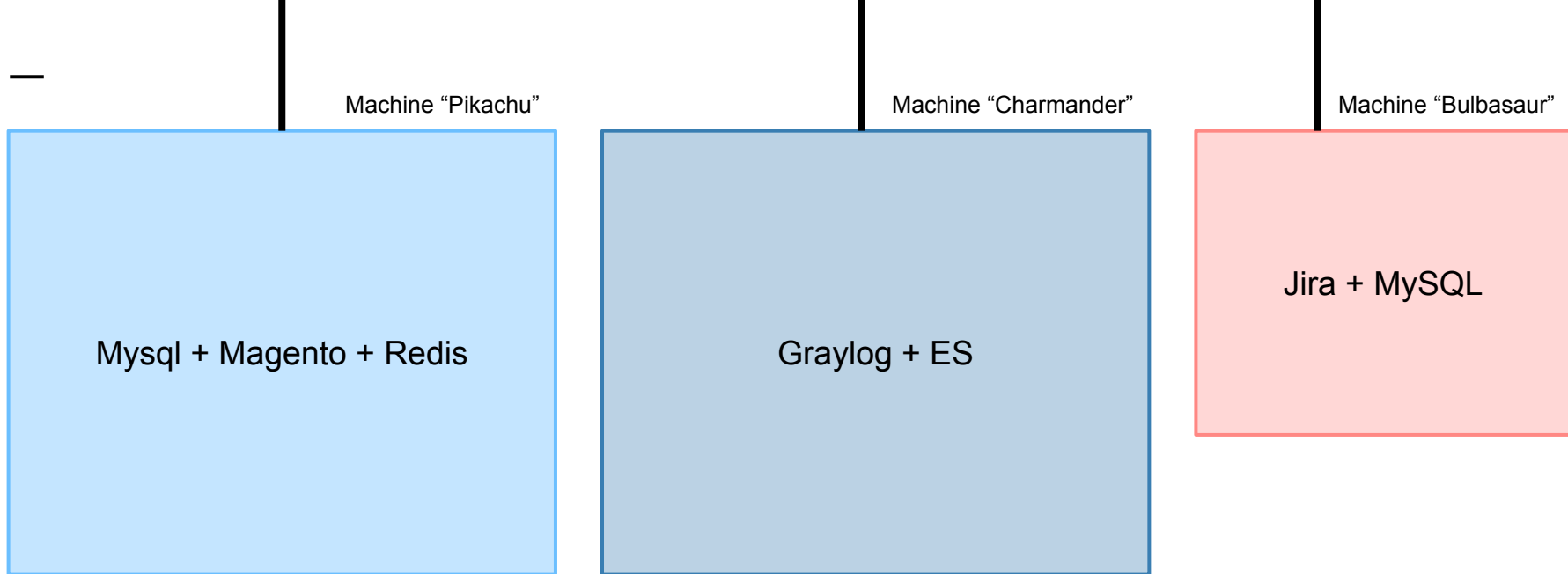
<https://kubernetes.io/docs/concepts/overview/what-is-kubernetes/> ·
<https://www.linux.com/news/learn/chapter/Intro-to-Kubernetes/2017/3/what-kubernetes> ·
<https://cloudplatform.googleblog.com/2016/09/bringing-Pokemon-GO-to-life-on-Google-Cloud.html> ·
<https://kubernetes.io/case-studies/>

Who Uses It?



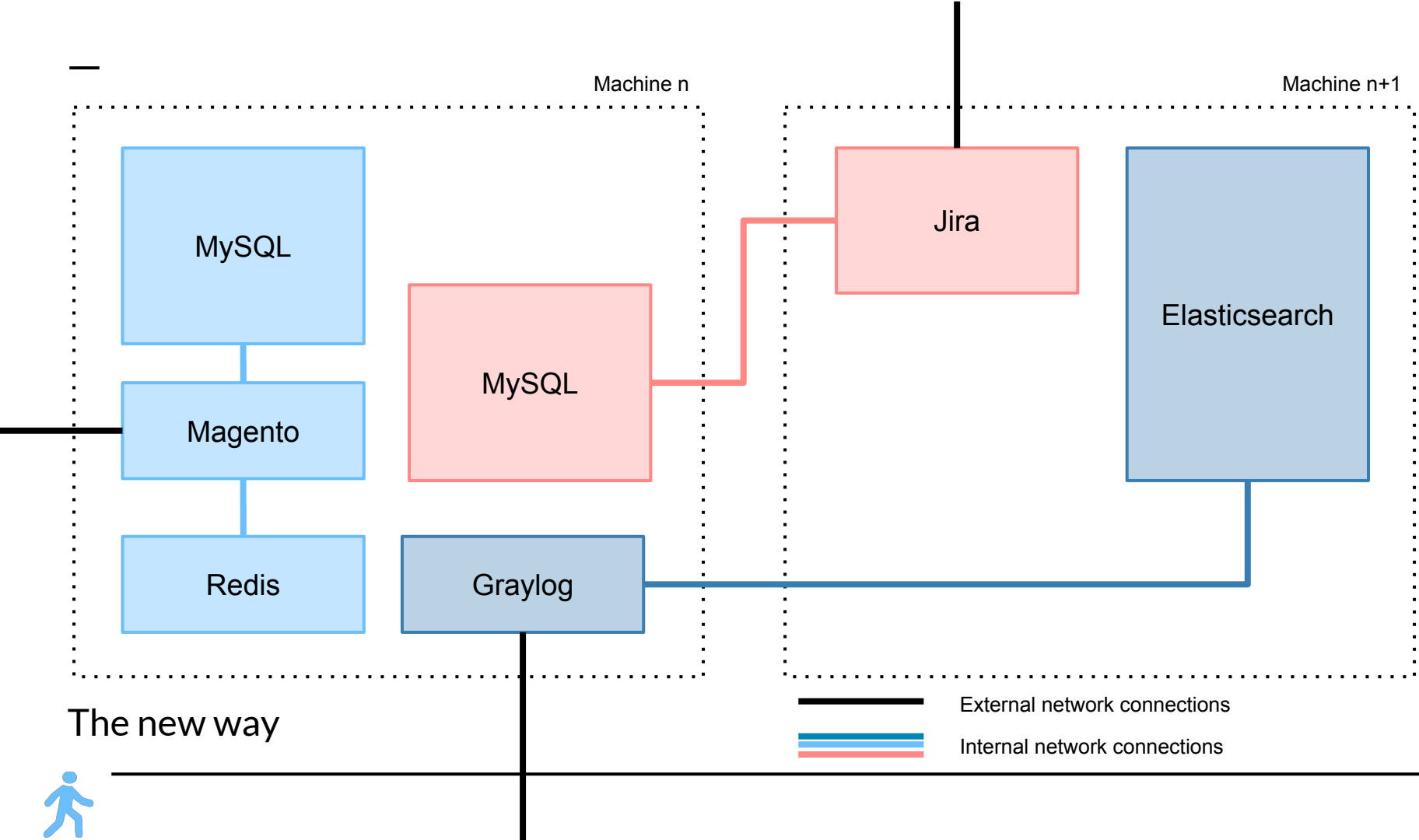
Architecture Diagram





The old way





**Ah hm how does this
happen?**



Controller-manager

Looks at the what the cluster is now, and what it should be, and makes changes so that what it is becomes what it should be.

Scheduler

Decides which workloads should be run on which machine.

Kubelet

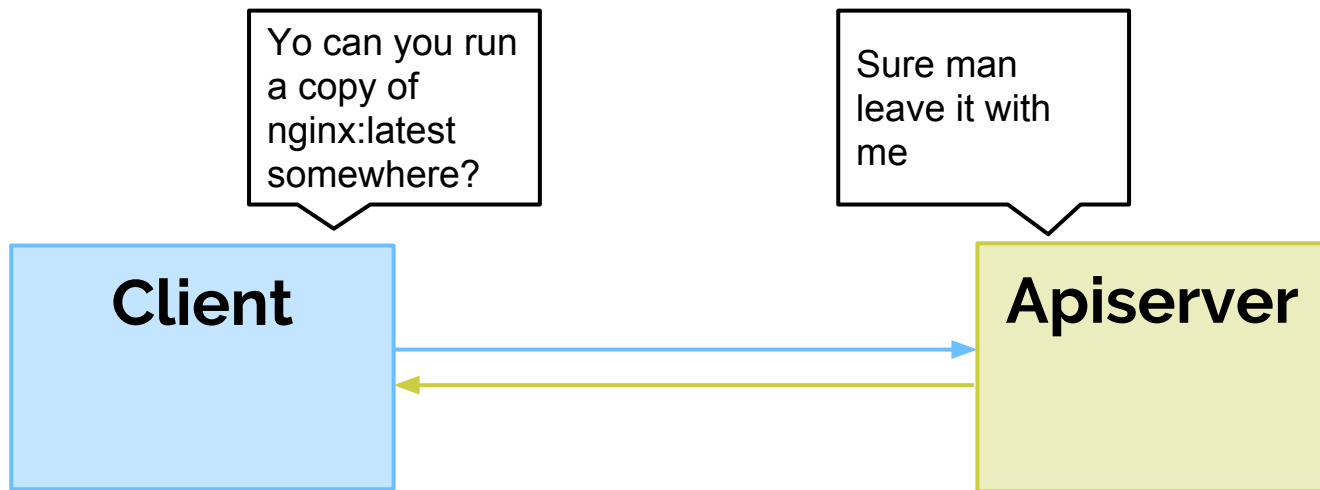
Runs and reports on workloads.

Apiserver

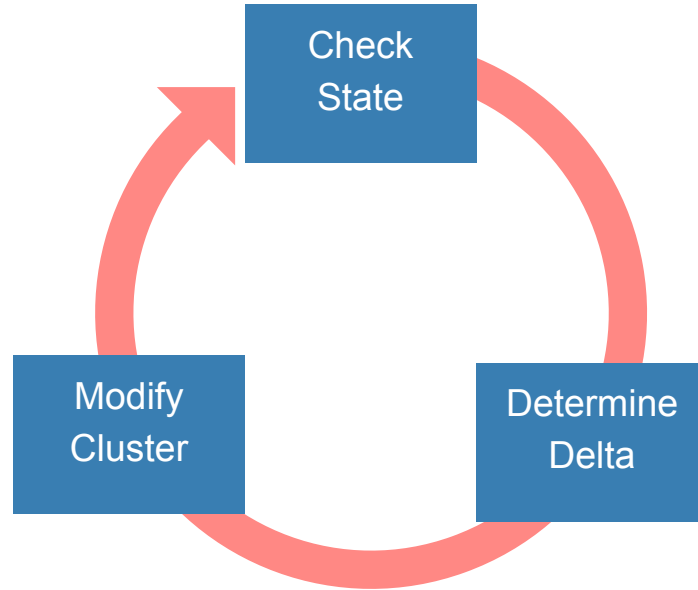
Handles communications between the various pieces



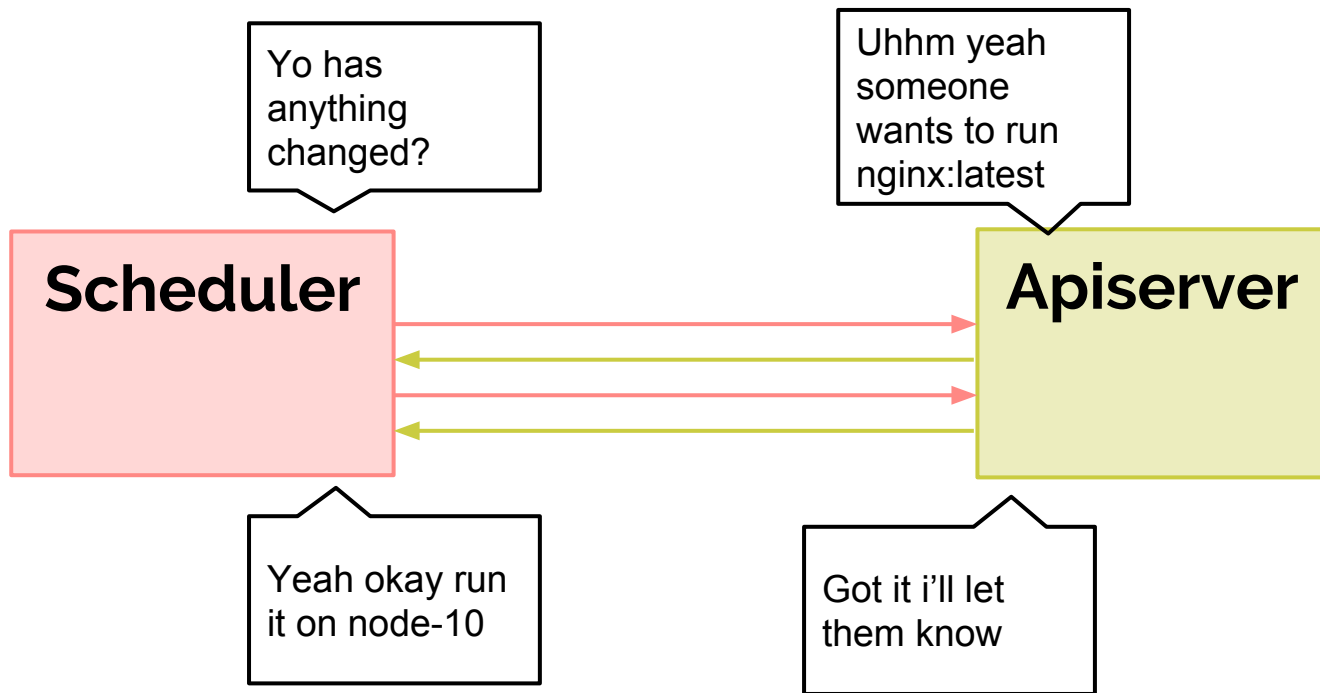
“Please cluster make this happen”



Control Loop



“Please cluster make this happen”



Task Runners

node-10

Kubelet

Apiserver

Yo what
should I be
doing exactly?

Uhh please run
1x nginx:latest

Okay yah sure
1 sec
...

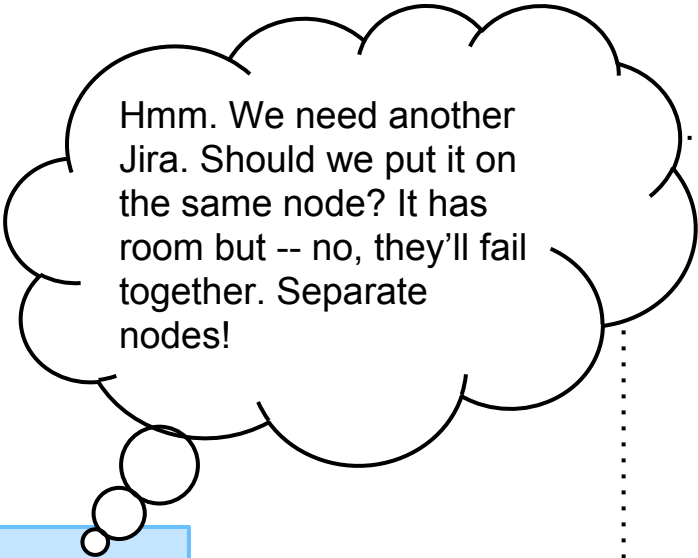
Done.
Anything else?



**Great! But what
does this buy us**



—



Hmm. We need another Jira. Should we put it on the same node? It has room but -- no, they'll fail together. Separate nodes!

Scheduler

Machine n

Jira

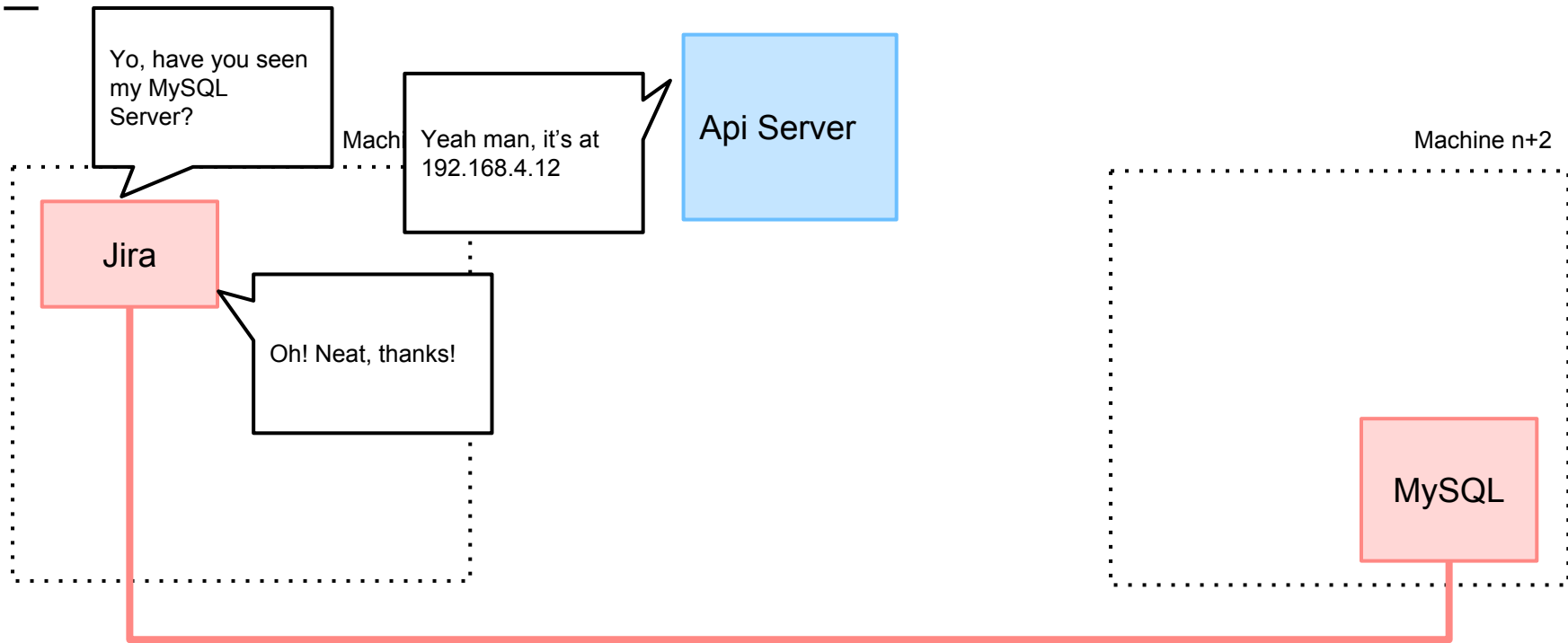
Machine n+2

Jira

MySQL

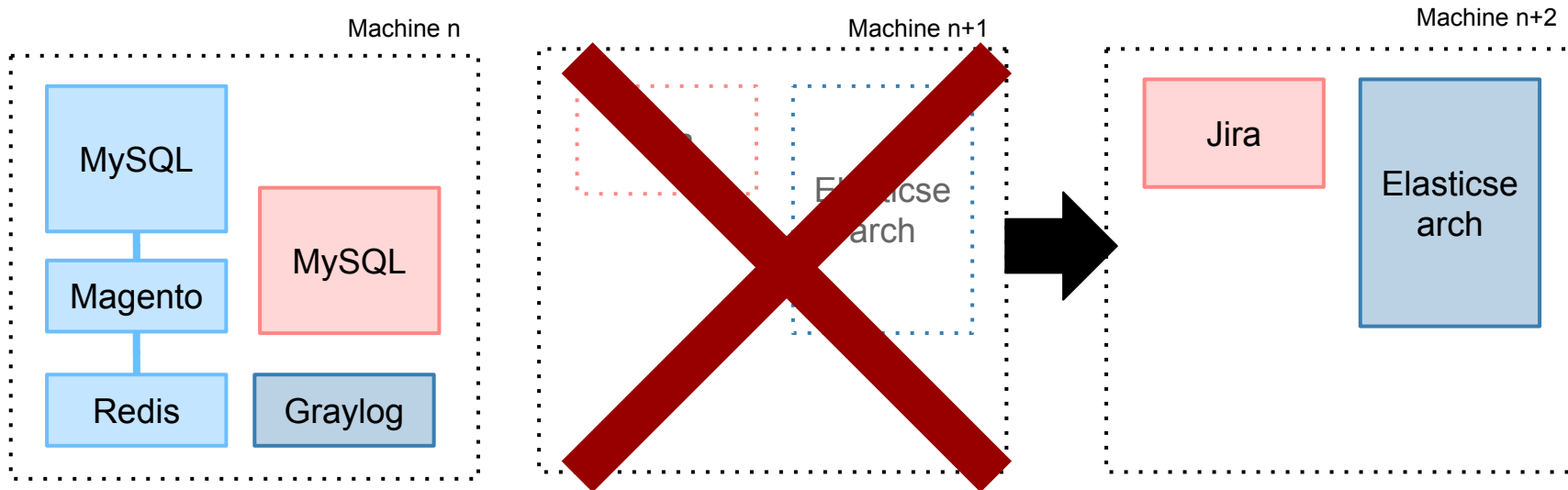
Software Scheduling





Service Discovery





Automated Failure Recovery



—

NGINX

NGINX

NGINX



NGINX

NGINX

NGINX

NGINX

NGINX

NGINX

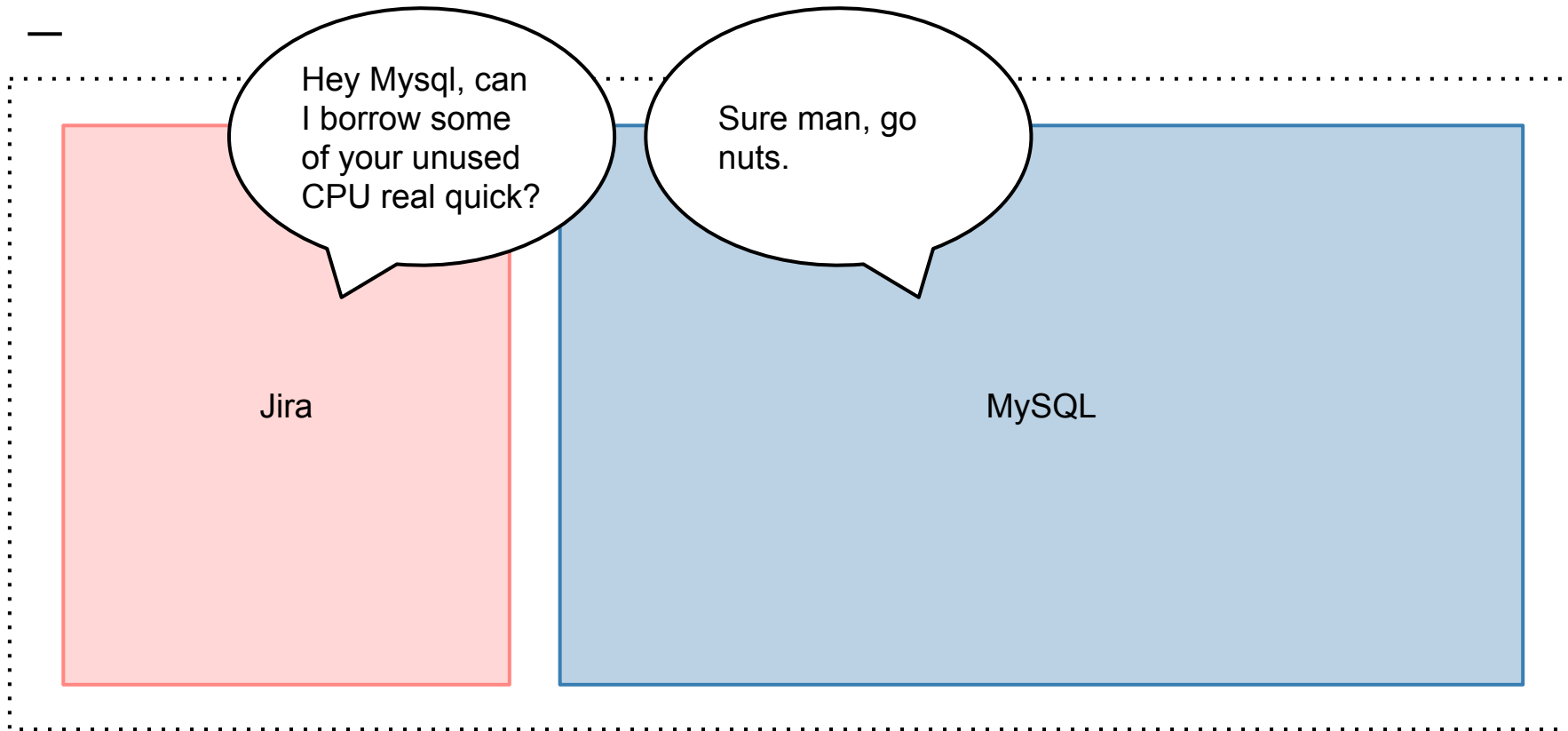
NGINX

NGINX

NGINX

Horizontal Scalability

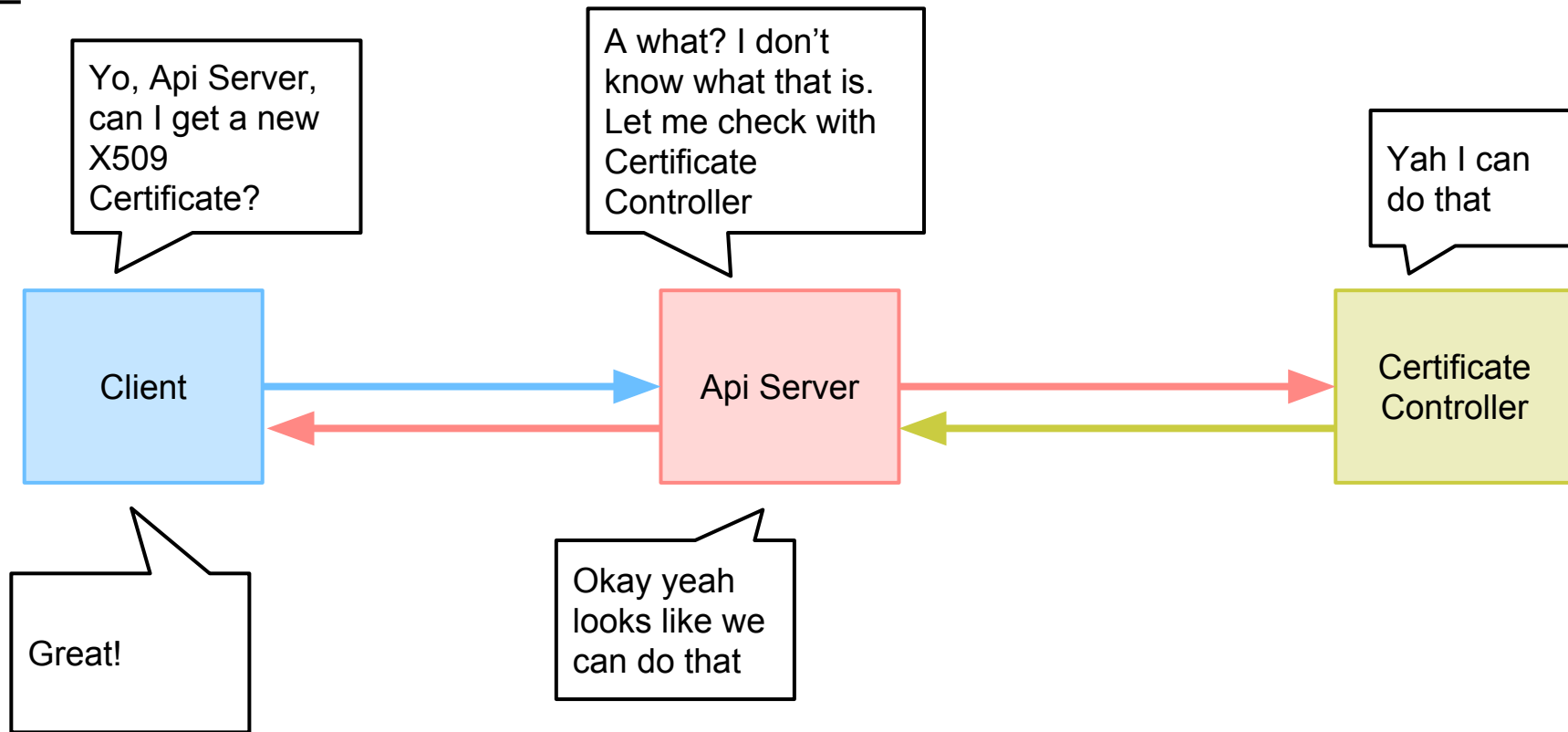




Application Bursting



—



Extensibility



In Summary

Failure Handling · Smart Application Scheduling · Horizontal Scaling · Service Discovery · Application Bursting · Scheduled Jobs · Batch Jobs · Canary Deployments · Rollable Deployments · Health Checking · Managed Configuration · Managed Secrets · Managed Storage · Monitoring · Log Aggregation · Extensible API



Profit?



<https://kubernetes.io/docs/api-reference/v1/definitions/>



Since Squarespace moved to Kubernetes, in conjunction with modernizing its networking stack, deployment time has been reduced by almost 85%. ... Because of that, "productivity time is the big cost saver".

<https://kubernetes.io/case-studies/squarespace/>





The impact has been considerable: With Kubernetes, the company has experienced a 90% cost savings on Elastic Load Balancer, which is now only used for their public, user-facing services. Their EC2 operating expenses have been decreased by as much as 50%.

<https://kubernetes.io/case-studies/crowdfire/>





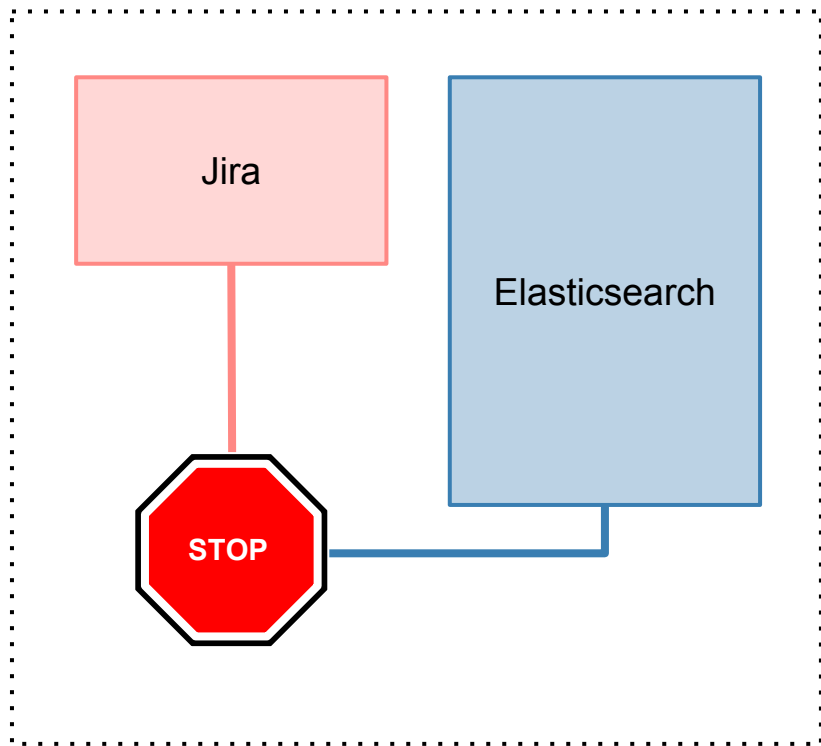
- 20 percent of web tools that account for more than 40 percent of web traffic now run on Kubernetes
- A 25-node cluster that keeps up with each new Kubernetes release
- Thousands of lines of old code have been deleted, thanks to Kubernetes

<https://kubernetes.io/case-studies/wikimedia/>



Security Implications





Multitenancy



Applications only see other applications they should*

More tools!



Helm

A tool for managing
pre-configured Kubernetes
resources.

Kind of like apt-get or yum for
Kubernetes. It makes the complexity
manageable.

<https://github.com/kubernetes/helm> · <https://github.com/kubernetes/charts> ·
<https://kubernetes.io/docs/concepts/containers/kubernetes-manifest-extractor/>



GKE

Google Container Engine

Kubernetes as a service, on top of the already brilliant Google Cloud platform.

<https://cloud.google.com/container-engine/>



I didn't get what you just said

There are lots of other people in this area. Check out “Kelsey Hightower”, “Jess Frazelle”, “Carter Morgan”, “Timothy Hockin”



More Halp

- Pets versus Cattle: <https://blog.engineyard.com/2014/pets-vs-cattle>
- Udacity Course: <https://www.udacity.com/course/scalable-microservices-with-kubernetes--ud615> (Free!)
- The Docker Book: <https://www.dockerbook.com/>
- Infrastructure as Code book: <http://shop.oreilly.com/product/0636920039297.do>
- Site Reliability Engineering book: <https://landing.google.com/sre/book/index.html> (Free!)



A note about hiring me

Everything goes through Sitewards.

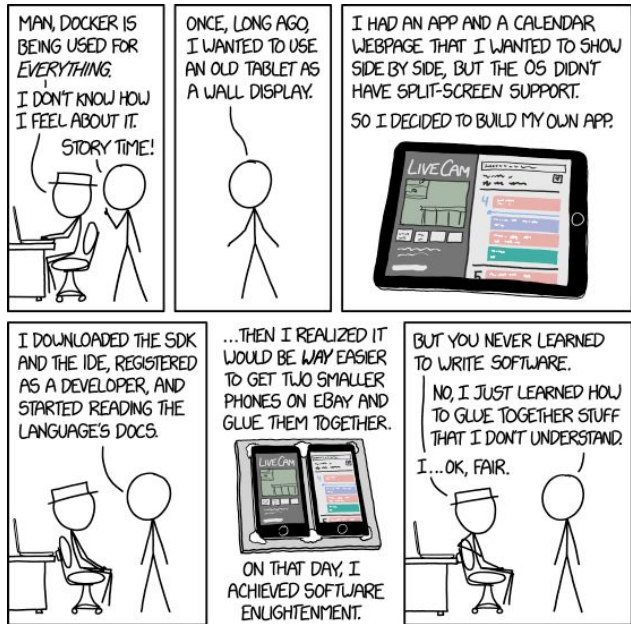
Contact them, and if there is time they'll scheduled it in for me.

The logo for Sitewards, featuring a stylized gold 'S' icon above the word 'Sitewards' in a gold script font.

Sitewards



Questions?



<https://xkcd.com/1988/>

Everything is available at the following link:

<https://github.com/andrewhowdencom/talk-using-docker-in-prod>

Find the rest of my contact information at

<https://www.andrewhowden.com/>

