The Open Sourcing of Infrastructure
Elizabeth K. Joseph, Developer Advocate

- Developer Advocate at Mesosphere working on DC/OS, Apache Mesos
- 15+ years working in open source communities
- 10+ years in Linux systems administration and engineering roles
- Author of *The Official Ubuntu Book* and *Common OpenStack Deployments*
The [recent] history of infrastructure

(from a highly opinionated, open source view)
“To make a server, first add...”
And so rose the proprietary world of software

With proprietary Unix and Windows-based platforms, the stage was set for the golden age of proprietary software in the 1990s and into the 2000s.
Linux was an upstart, at best seen as “cheap Unix”

Lots of FUD around open source
I liked it anyway.
So I got a junior Linux systems administrator job!
Some of the topics during a seminar I spoke at in the 00s

- What is Free/Open Source Software (FOSS)?
- How & Why Linux and FOSS can Deliver Business Results
- Managing FOSS: Thousands of Alternatives - How To Choose?
- Using Open Source Web Applications to Produce Business Results
Turning point: LAMP stack
Flood of changes to how we interact with software

Reluctance to be locked-in by a vendor
Greater concern over security
Wanted the ability to fix bugs ourselves
Learned that innovation is stifled when software is developed in isolation
Flood of changes to how we use software

Downtime becoming [considerably more] unacceptable
Increase in reliance upon scaling and automation
Transition from server “pets” to “cattle”
Larger focus on data (retention, speed)
“[T]he seeds of the future were found in free software and the Internet rather than in the now-establishment technologies offered by Microsoft.”

Tim O’Reilly, *What's the Future and Why It's Up to Us*
Open source is now ubiquitous
Developers are using, developing on, contributing to, and sharing open source software!

Operations is using and developing on open source software.
When I left my ops job, I left my tools behind

CC BY 2.0 https://www.flickr.com/photos/90692443@N05/8239219385/
Time to open source ops stuff!
Configuration management
- Puppet Modules
- Chef Cookbooks
- Ansible Playbooks

Open application definitions
- DC/OS Universe Catalog
- Juju Charms

Full disk images
- Dockerhub and other container registries
Welcome to the present!
The Cloud.

Including IaaS, PaaS, SaaS...
What were some of the reasons for going open source in the first place?

- Security
- Ability to diagnose and fix bugs without vendor intervention
- Increased control over our data and services
- Avoiding vendor lock-in
“Most people just consume the cloud without thinking ... many users are sinking cost into infrastructure that is not theirs, and they are giving up data and information about themselves without thinking.”

Edward Snowden, OpenStack Summit, May 9, 2017
Let’s think.
Is the service I’m using adhering to open standards, or am I locked in?
What is my recourse if the service vendor goes out of business
...or is bought by a competitor?
Does the vendor have a history of communicating clearly and honestly with their customers about downtime, security, etc?
Does the vendor respond to bugs and feature requests?
Will the vendor use our data in a way that I’m not comfortable with?
(or worse, isn’t allowed by your own customer agreements)
Initial costs may be low, but do you have a plan to handle long term, growing costs?
You *could* consider all these things and acknowledge them as acceptable risks.

Many organizations do!

Just make sure you are *actually, seriously* considering the risks.
Or look again to Open Source

Various infrastructure technologies are available:

- OpenStack
- Kubernetes*
- Docker swarm mode*
- DC/OS with Apache Mesos*

...and more in the future with a constantly growing ecosystem!

* Can be used in the cloud or on premises
**EXAMPLE:**
**DC/OS Architecture Overview**

### Services & Containers

<table>
<thead>
<tr>
<th>HDFS</th>
<th>Jenkins</th>
<th>Marathon</th>
<th>Cassandra</th>
<th>Flink</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spark</td>
<td>TensorFlow</td>
<td>Kafka</td>
<td>MongoDB</td>
<td>Your App</td>
</tr>
</tbody>
</table>

### DC/OS

<table>
<thead>
<tr>
<th>Container Orchestration</th>
<th>Security &amp; Governance</th>
<th>Monitoring &amp; Operations</th>
<th>User Interface &amp; Command Line</th>
</tr>
</thead>
</table>

**ANY INFRASTRUCTURE**

- Physical Servers
- Virtual Servers
- Private Cloud
- Public Cloud Providers (Google, AWS, Azure)
Open source tooling can be platform agnostic, all you need is some kind of Linux install, this gives you:

- Opportunity to use multiple cloud platforms at once, or in-house components of your infrastructure
- Ability to migrate between cloud and in-house platforms as cost, performance or other metrics change
- Gives your developers choice about where their applications run

I leave you with: Hybrid Cloud
“Why open source should be the first choice for cloud-native environments” article

https://opensource.com/article/17/8/open-sourcing-infrastructure

Questions?

Elizabeth K. Joseph
Twitter: @pleia2
Email: lyz@princessleia.com