

devops Practices & Tools

Or How to Deal with Buzzword Bingo

Who am I and why am I talking to you

- Josh Zimmerman - Library Technology Group
- Organizer of Madison devops meetup
<https://www.meetup.com/Madison-Devops/>
- Organizer of DevOps Days Madison (CFP and registration open!)
<http://www.devopsdaysmsn.org>
- Helping to resurrect and organize Containers/Cloud Native - Madison
<https://www.meetup.com/Containers-Madison/>
- Has spoken at a few devopsdays events.
- Etc.

Today's Rough Agenda

- What is devops?
- Cultural Practices
 - Just Culture
 - Continuous Learning/Improvement
 - Diversity/Inclusion
- Tools/Tech
 - Continuous Integration/Testing
 - Provisioning
 - Configuration Management
 - Orchestration
 - Secrets Management
 - Containers
 - What containers actually are
 - Container orchestration
 - Monitoring and observability

What is this buzzword and why
should I care?

devops is primarily a cultural
movement.

devops seeks to align the
incentives of organizational units

Empathy is key.

However, all of this doesn't mean that it is not about tools.

Tools that were built for one culture may not work as well for another culture.

You need to understand the context for which tools were intended to properly use them.

So, what does all of this mean to
us at the UW?

Cultural Concepts

Just Culture

People who make mistakes are
victims too.

The goal is for the organization
to share accountability and to
learn from failure.

Kaizen / Continuous Improvement

1. In consideration of a vision or direction...
2. Grasp the current condition.
3. Define the next target condition.
4. Move toward that target condition iteratively, which uncovers obstacles that need to be worked on.

Diversity & Inclusion

Diversity means squat if people
are not heard or leave the
organization.

Tech!

Testing and Continuous Integration

Continuous Integration is more than just running your tests on a CI server.

Deploy small changes more
often.

Testing/CI Tools to look at

- Rule of thumb: Use the testing tools that the community around your language/tool use to test it. (junit, rspec, etc.)
- Acceptance testing: Inspec, serverspec, beaker-rspec
- CI servers: No CI server is going to be perfect. At the end, you still need to know what kind of tests you want to run, and if you're not using a cloud based CI, you may need to make some architecture decisions. Jenkins, Gitlab, Go CD are great for on prem. Travis CI and Circle CI are great hosted solutions.

Configuration Management

CM tools are not just for
operations folk.

If you're using containers, they're still running on a server that needs to be configured.

CM tools:

- Every CM tool vendor will tell you, “It doesn’t matter which tool you use, what’s more important is that you are using one of them.” They all have pros and cons, but what really matters is that you’re addressing the problems that CM tools solve.
- Puppet
- Chef
- Ansible
- Salt

Orchestration

For any task that needs to be done multiple times, it should be automated.

Orchestration tools

- Consider using whatever goes best in the ecosystem of tools you're using, if a tool exists in that ecosystem.
- Ansible's paradigm is kind of doing CM over orchestration, and can be leveraged for orchestration.
- Rundeck seems cool but I haven't used it.
- An okay but not great option is always a half decent UI to call shell scripts.
- ChatOps is best Ops.

Provisioning

Provisioning tools:

- Image management: HashiCorp's Packer is awesome if you're doing virtual servers of any flavor.
- Infrastructure Provisioning:
 - HashiCorp's Terraform - Great for AWS, Azure, GCP
 - Cloud Formation - AWS only, less effort than terraform, but not as good IMO.
 - Puppet's Razor - For bare metal
 - etc.

Secrets Management

Secret management tools:

- Lowest barrier to entry is using a tool specific solution.
- HashiCorp's Vault - Arguably the best free/open source option. Super high learning cliff. Can be leveraged in conjunction with several other tools. Built in support to do things like credential rolling while also letting your tools update the servers that need the credentials. Enterprise options available.
- CyberArk Conjur - Works with a variety of tools out of the box. Much easier to use out of the box. Pay for solution only.

Containers

What the heck is a container
anyways?

Containers on their own don't do much for you in production. Use a container orchestration tool.

Monitoring, Metrics and Observability

Your applications are more
important than your servers.

Your applications cannot be monitored by checking if a process is running.

You need to understand what
the normal state(s) of your
application is/are.

You cannot do that without
people who either wrote or are
the local expert on the
application.

Checks/monitoring/metrics
should be written to things
important to the organization.

Logs should be structured and
you should be collecting metrics.

Observability > Monitoring

Monitoring, metrics and observability tools:

- Monitoring/metrics tools: Nagios, Sensu, Statsd, Graphite/Graphana, Prometheus
- Logging: ELK, Splunk
- Observability tools: Honeycomb, OpenTracing, Zipkin

Things that you should check out/a few citations:

- Books: The Phoenix Project, Effective DevOps, The DevOps Handbook, Just Culture: Balancing Safety and Accountability, Toyota Kata, Continuous Delivery
- Papers: State of DevOps Report
- Podcasts: DevOps Cafe, The Ship Show, Arrested DevOps
- Blogs:
 - <https://ericchiang.github.io/post/containers-from-scratch/>
 - <https://blog.jessfraz.com/post/containers-zones-jails-vms/>

If you want to ask questions, let's chat after or drop me an e-mail.