

Bringing Ansible to an existing infrastructure



A N S I B L E

James Nelson – jsnelson@vacourts.gov

VM Workshop 2023

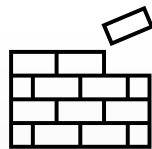
What is Ansible?

- set of Python programs and libraries to make everyday sysadmin tasks easier
- uses YAML configuration, other config structures exist but YAML is standard
- allows easy, repeatable scripting of system actions



Why do we need another piece of tech?!?

- Ansible is impressively cross-platform - the only tooling written for a specific Linux distro is for the package manager
- It can be used as a piece of middleware - it can call existing automation or scripting, or be called by higher level scripting, fairly easy
- "UNIX tools" model - individual components that do one thing well, taken to a higher abstraction level



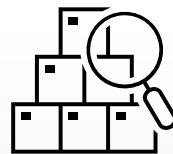
The problem with Internet tutorials

- Ansible tutorials are full of "here's how to install X" content
- "Day one" install and setup covered, "day two" operations, not so much
- Modern containerized applications make this less of an issue for some software since you can just rebuild the hosting containers
- Not everything is that clean and even containerized apps need to be poked



Step one: the inventory

- The first thing you need - represent your infrastructure in a way that makes sense to you - environments / application clusters / etc
- use inventory groups to make your life suck less, groups by OS, datacenter, whatever organizational structure you see the need for
- don't be afraid to move things around, trust your source control (you're using source control, right?)
- You can use the inventory to store relevant details for provisioning, both on a server as well as a group level



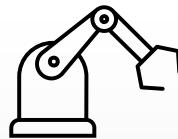
Step two: start small, fix boring problems first

- NFS mounts, the perfect problem for our environment
- through templating and default variables, can enforce specific `/etc/mounts` options on every server
- central location for all your nfs mounts in a large environment, add mounts as you find them and verify they're still needed
- part of generating these Ansible automation jobs is that you are forced to audit the existing configurations to verify them and can find all kinds of surprises



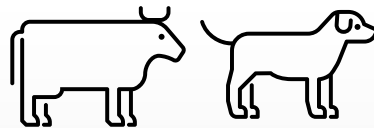
Step three: Automate your job (almost) away

- After getting one job sorted out and working comfortable, you can take the configuration you already have and add another
- use roles to organize your configuration and setup, applying those roles to existing servers as you get them scienced out
- Server provisioning: VMWare and KVM are very well supported through Ansible libraries, Feilong and zPRO offers similar utility to z/VM.



Step four: Servers as livestock vs. servers as pets

- Once you have automated server provisioning worked out, you start hitting a philosophical choice, and one driven by containerized workloads
- Traditional UNIX servers lived for years and years, gathering a tremendous weight of fixes and one-off mods that made migration kind of spicy
- Cloud computing, and Kubernetes, have forced a very different philosophy, where the computing platform can be ephemeral and divorced from the requirements of persistent storage
- Ansible can work with either paradigm, and it is a great stepping stone for documenting existing for an OS migration / application containerizing



Step five: Window cleaning and wire wrangling

- Ansible can also talk to Windows servers! Both through WinRM as well as ssh (natively available on Server 2019+, installable on 2016)
- Chocolatey package manager for deploying Windows software
- AD integrations as well as tooling for IIS and Windows services like on Linux
- The best way to admin a Windows box is to never use Remote Desktop
- Network infrastructure and storage systems also have Ansible interfaces, allowing automation to reach the individual switchport level
- Provisioning volumes on storage arrays if vendor has Ansible integration



Step six: How I learned to stop worrying and become the Borg

- As more things were sent my way, they got tossed into automation - Tomcat deployments, Websphere upgrades, SMB share mounting, more application deployments
- The more information you include in your inventory and groups, the more powerful Ansible becomes
- Reporting becomes very easy with Ansible, the same tooling that allows you to deploy to a hundred servers at once can also run your data gathering
- Becoming automation-first, considering every service request as potentially automatable

Things Ansible will not do easily (or at all)

- Sorry, VSE folks, there's nothing available right now - the VSE Connector tooling could be used with Jython but nothing is available
- If you aren't running DIRMAINT your z/VM automation options are limited - it's zPRO or nothing, and to my knowledge there's no Ansible module
- DB2 also has no available Ansible tooling, especially for operations.
- Anything point-and-click GUI based takes a disturbing amount of work to automate, in house apps should get a CLI / REST API for automation
- It's overkill for a small environment – for a few servers, it's overly elaborate



Thanks for your time!

- Ansible is a large topic with a lot of flexibility, and takes time to get good with
- Good luck with your automation adventures!

