Configuration Management

Develop Intelligence

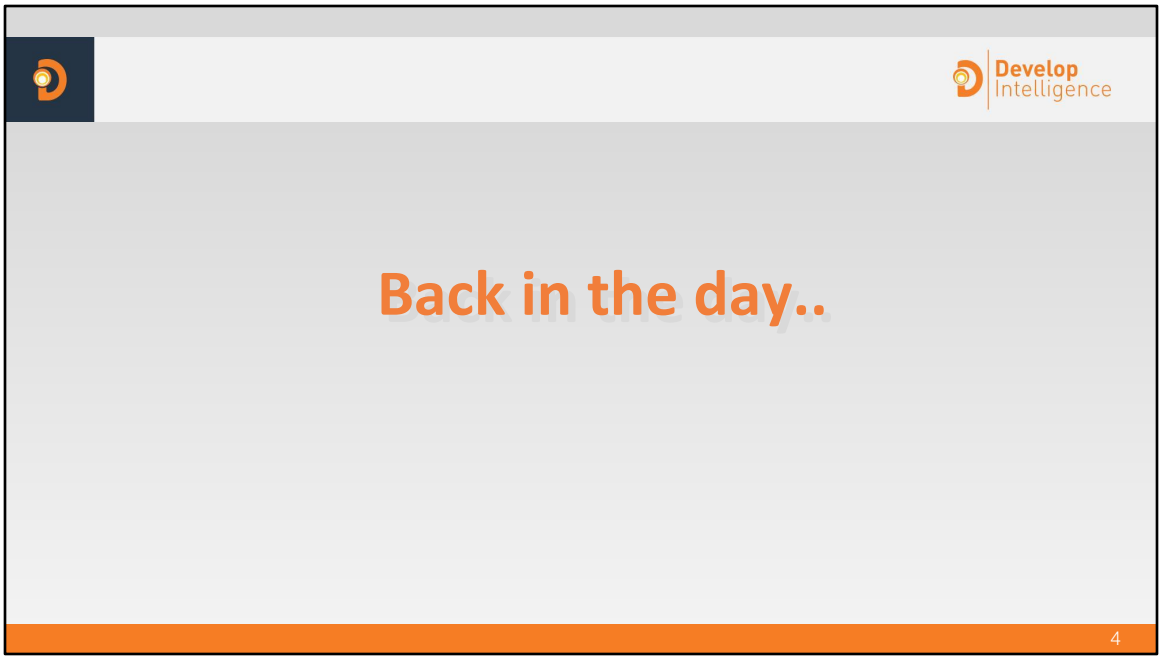
Module Outline

- **Why?**
- How?
- Tool comparison
- Review

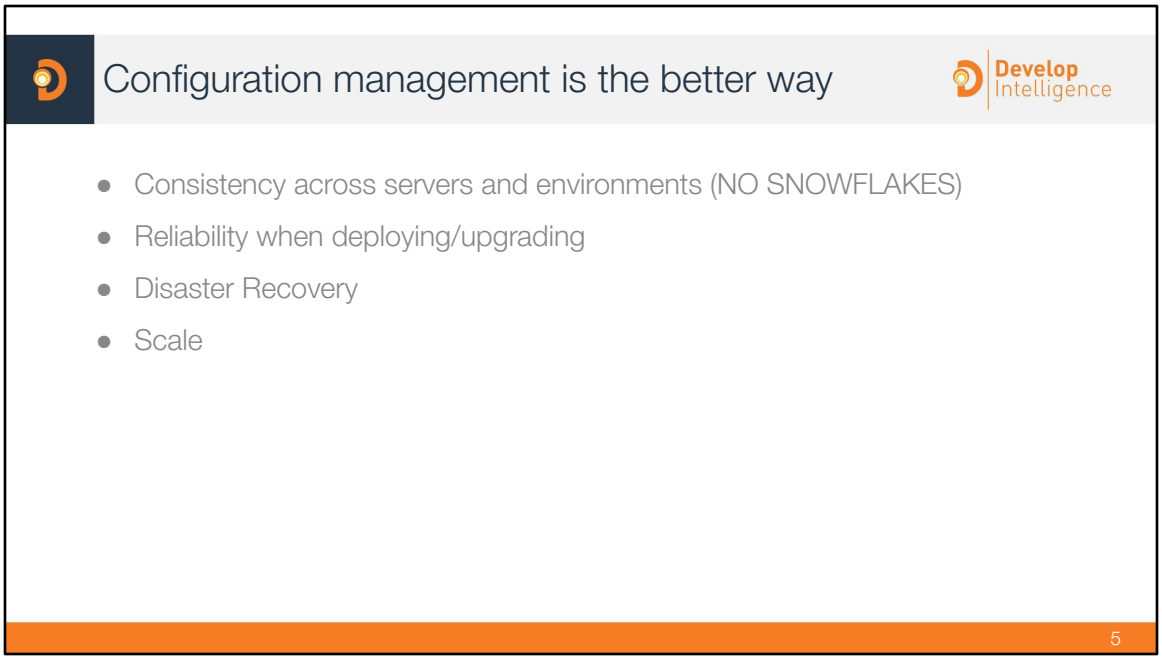
What do we need to do?

Develop Intelligence

- Deploy our software onto servers
  - Install dependencies
  - Create configuration files
  - Register with external systems
  - Configure the system to start our application
  - etc...



Back in the day, or even still at a lot of organizations, servers are managed by hand. That means that whenever new software needs to be installed or updated, someone logs into the machine and runs the necessary commands. This can very quickly become difficult as the sysadmin may not remember or have properly documented what has been installed on a server, requiring forensics whenever something needs to be changed. This is also a big issue if a server needs to be recreated from scratch or more servers created for scaling. You can very easily end up with servers that should be identical with different libraries installed, causing applications running on those servers to behave differently.

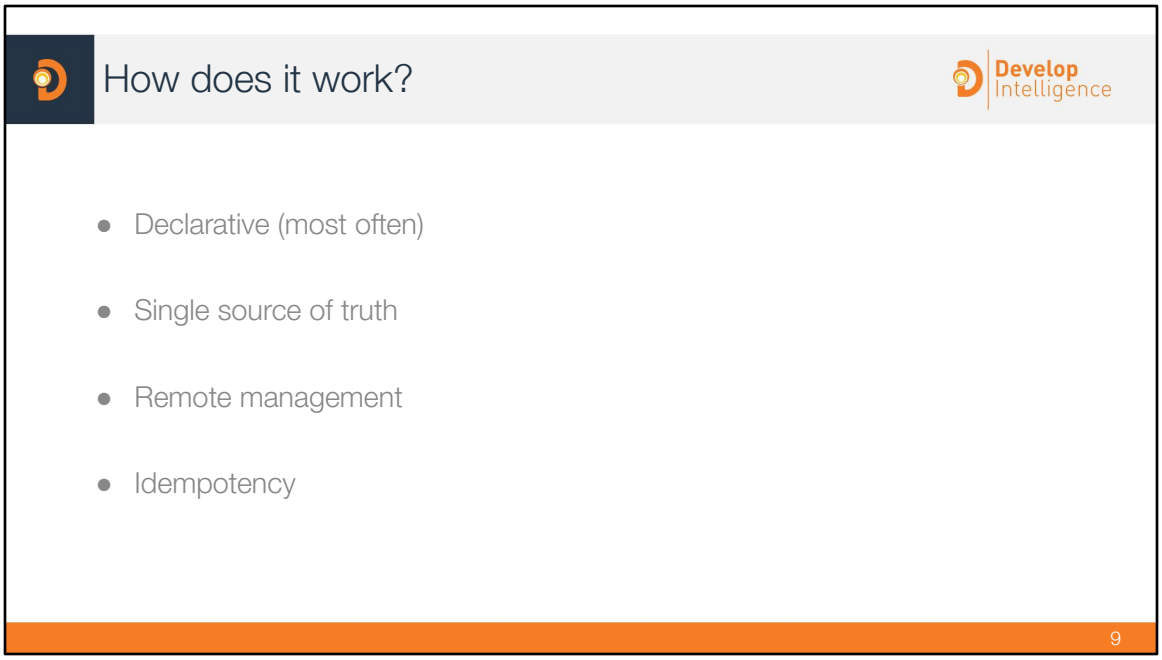


Back in the day, or even still at a lot of organizations, servers are managed by hand. That means that whenever new software needs to be installed or updated, someone logs into the machine and runs the necessary commands. This can very quickly become difficult as the sysadmin may not remember or have properly documented what has been installed on a server, requiring forensics whenever something needs to be changed. This is also a big issue if a server needs to be recreated from scratch or more servers created for scaling. You can very easily end up with servers that should be identical with different libraries installed, causing applications running on those servers to behave differently.

Configuration management

Develop Intelligence

- Define EVERYTHING in code
  - code can be versioned, shared, reviewed, and modified
- Prevent configuration drift (manual changes will be clobbered)
- A single script to managed 100s or 1000s of machines
- If anything is lost, you can recreate it quickly



Declarative = Say what you want not how to get it. I want 5 machines running version 2 of X library. Once you tell the tool, it figures out how to make it happen.

Idempotent = no matter how many times you tell the tool you want something, it will be in the same state.

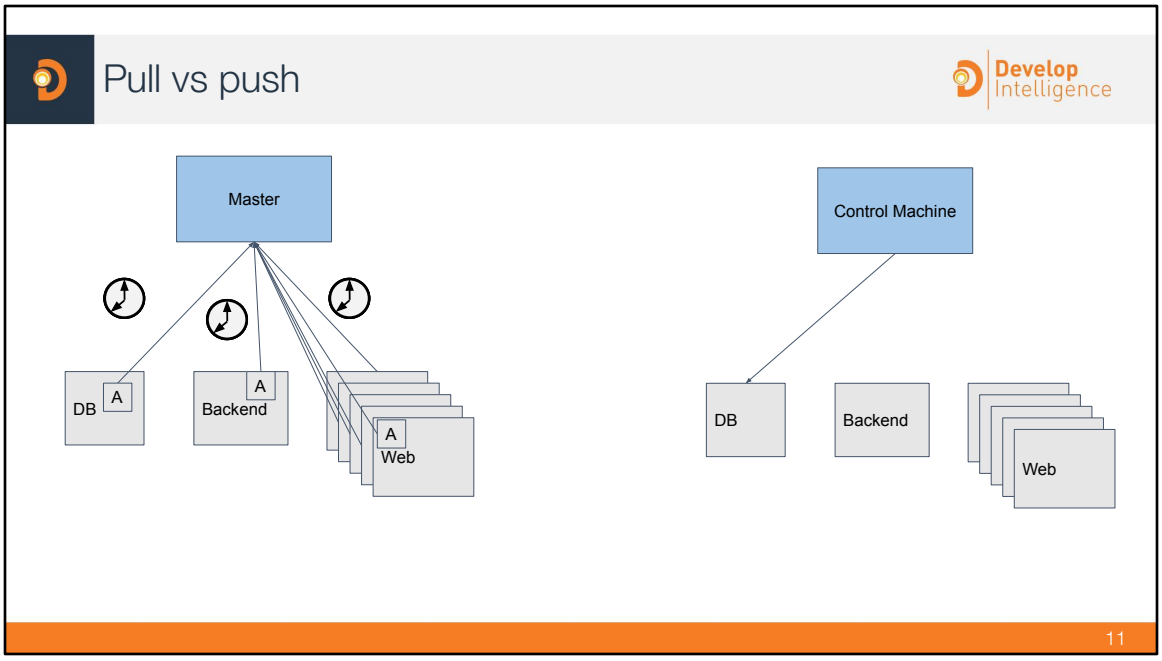
Remote management = Logging into each and every machine directly is unnecessary, manage them from a central location through remote connections.

Single source of truth = the manifests you define that are passed into the configuration management tool are the authoritative source of truth because that is the only thing the CM tool is acting upon.

Comparing Tools

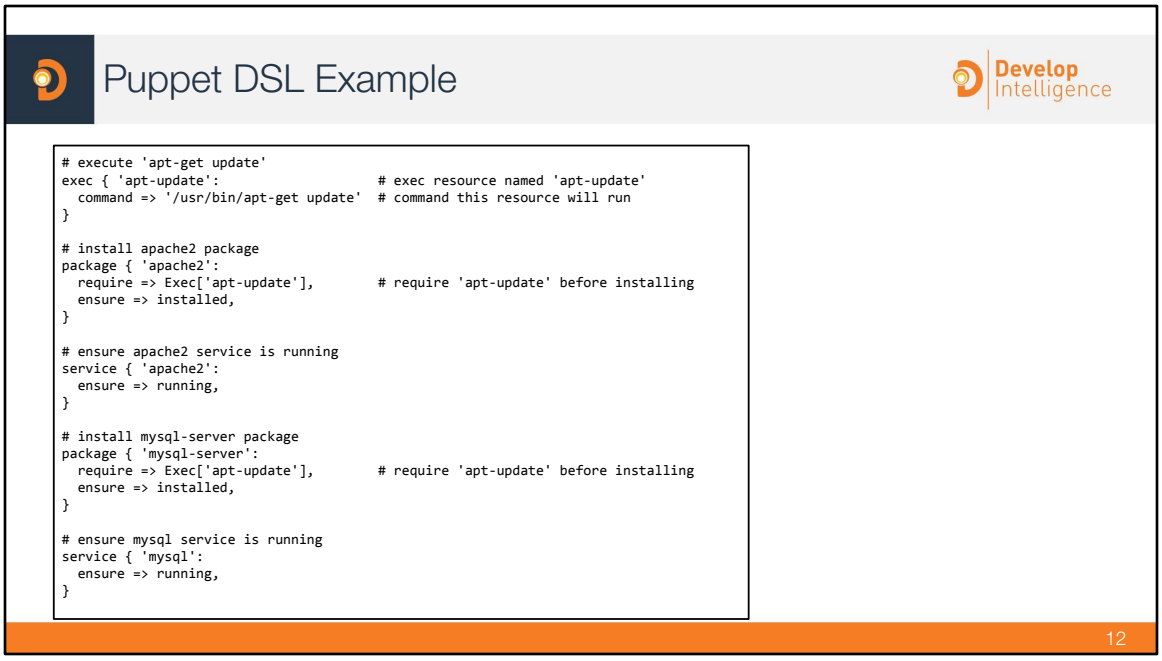
Develop Intelligence

	Puppet	Chef	SaltStack	Ansible
Initial release	2005	2009	2011	2012
Language	Custom DSL	Customer DSL (Based on Ruby)	YAML	YAML
Pull/Push	Pull	Pull	Push	Push
Communication	Agent	Agent	SSH	SSH & WinRM (and other options)



Pull:
* Every machine must check in periodically to get it's desired state
* Requires an agent to check in with a central server
* If you have a lot of servers, this can cause a lot of traffic going to the central server. Scaling bottleneck
+ Configuration is enforced periodically (desired state may not change but the actual state may have, I.E. someone logged into the server and disabled a service)

Push:
* Desired state can be pushed out to specific machines when the desired state changes
* Does not require an agent if standard protocols are used



This puppet manifest shows the Puppet DSL. It does not resemble any other programming language. But you can see how the **desired state is declared** instead of the steps defined to get to the desired state.

Which of the following are advantages of a push over a pull based CM system?

Develop Intelligence

1. Any changes made on the servers that do not align with the declared desired state will be overwritten on the next update cycle.
2. There is no bottleneck
3. No agents to manage

Additional Resources

Develop Intelligence

- Tool comparisons
  https://www.intigua.com/blog/puppet-vs.-chef-vs.-ansible-vs.-saltstack