

CM SMACKDOWN: BATTLE OF THE CONFIGURATION MANAGERS

POWERSHELL & DEVOPS GLOBAL SUMMIT, 2021

ADIL LEGHARI

AGENDA

- Introductions: Adil & Config. Mgrs.
- DSC, Puppet, Ansible:
 - Overview
 - Pro's & Con's
- Code Walk-Throughs & Demos
- Conclusions & Use Cases

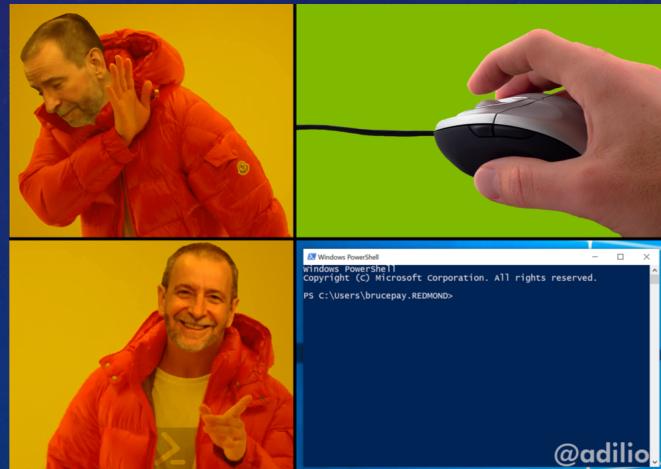




INTRODUCTIONS: ADIL



- Adil Leghari
- 15-year SysAdmin, passionate about PowerShell
- Resident PowerShell Sticker Artist
- Sr. Solutions Engineer, Chocolatey Software



WHY USE A CONFIGURATION MANAGER?

- Declaratively state your configuration
- Idempotence
- Speak the Domain-Specific Language (DSL) of tool
- Let Configuration Manager tooling simply "make it so"
- Managing Configuration Drift (automatically)

WHAT IS YOUR CORE BUSINESS?

- “Build what differentiates you, and buy what doesn’t.”
 - -Jeffrey Snover
- Stages of Automation
 1. Manual maintenance
 2. Writing code to manage Infrastructure
 3. Declare and let the tool “make it so”
- Cost of engineering hours vs. OPEX (TCO is often lower)



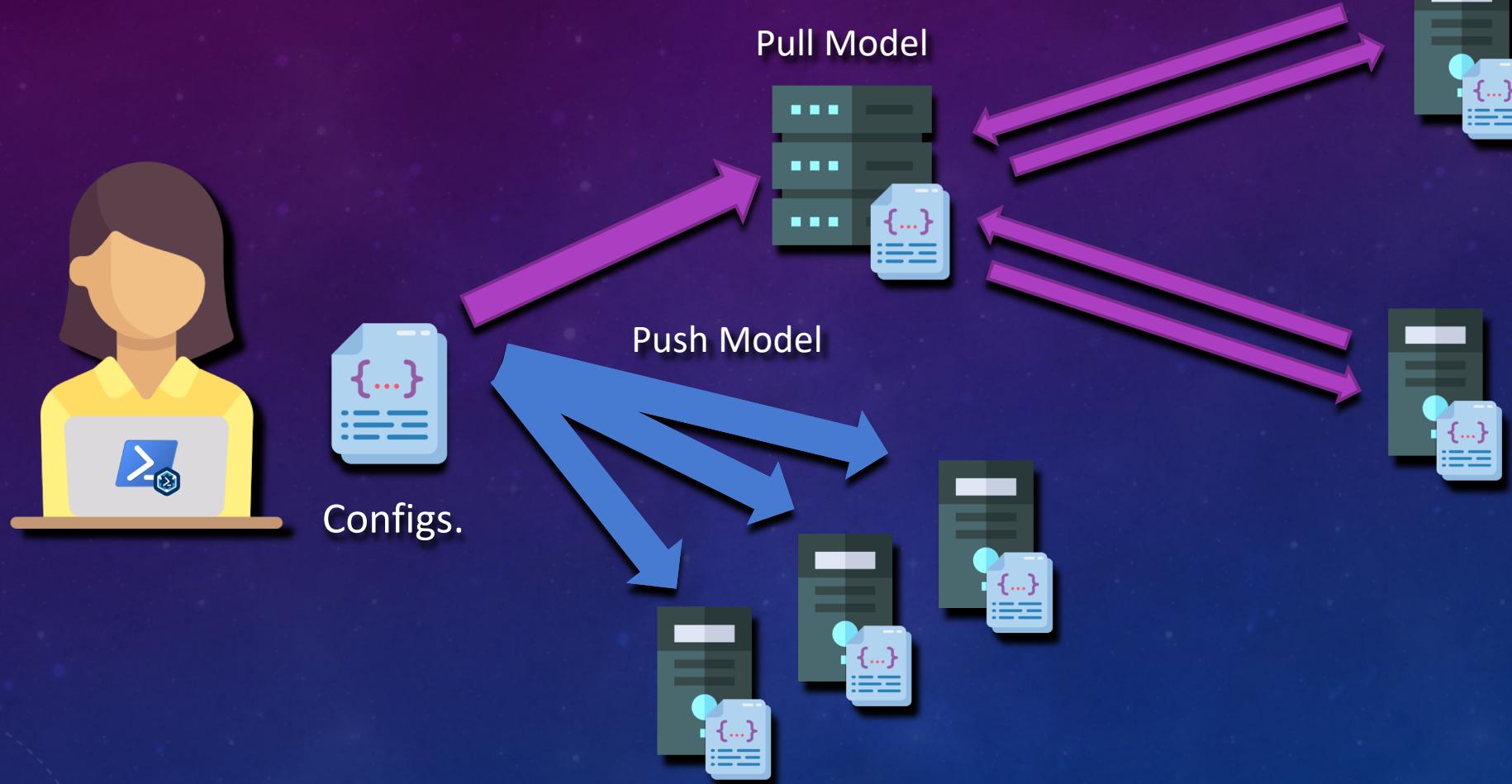
HONOURABLE MENTIONS

- Chef
- SaltStack
- CFEengine
- Microsoft Endpoint Manager (SCCM & Intune)

DSC OVERVIEW

- Management & Maintenance Platform (PowerShell, built-in)
- Configuration-as-Code: Configure, Deploy, Manage
- Declarative; Idempotent-by-Nature
- 3 Components:
 - Configurations
 - Resources
 - Local Configuration Manager (LCM)

DSC OVERVIEW



#PSHSummit

DSC: PRO'S

- You get to write in PowerShell! 😊
 - Familiarity with Language
 - Assists in Troubleshooting
 - DSC Resource in PowerShell Gallery
- Native/Built-in to the OS
- Azure Automation + PS DSC Extension
- Extensible
 - DIY Resources
 - Composite Resources (granular mgmt.)

DSC: CON'S

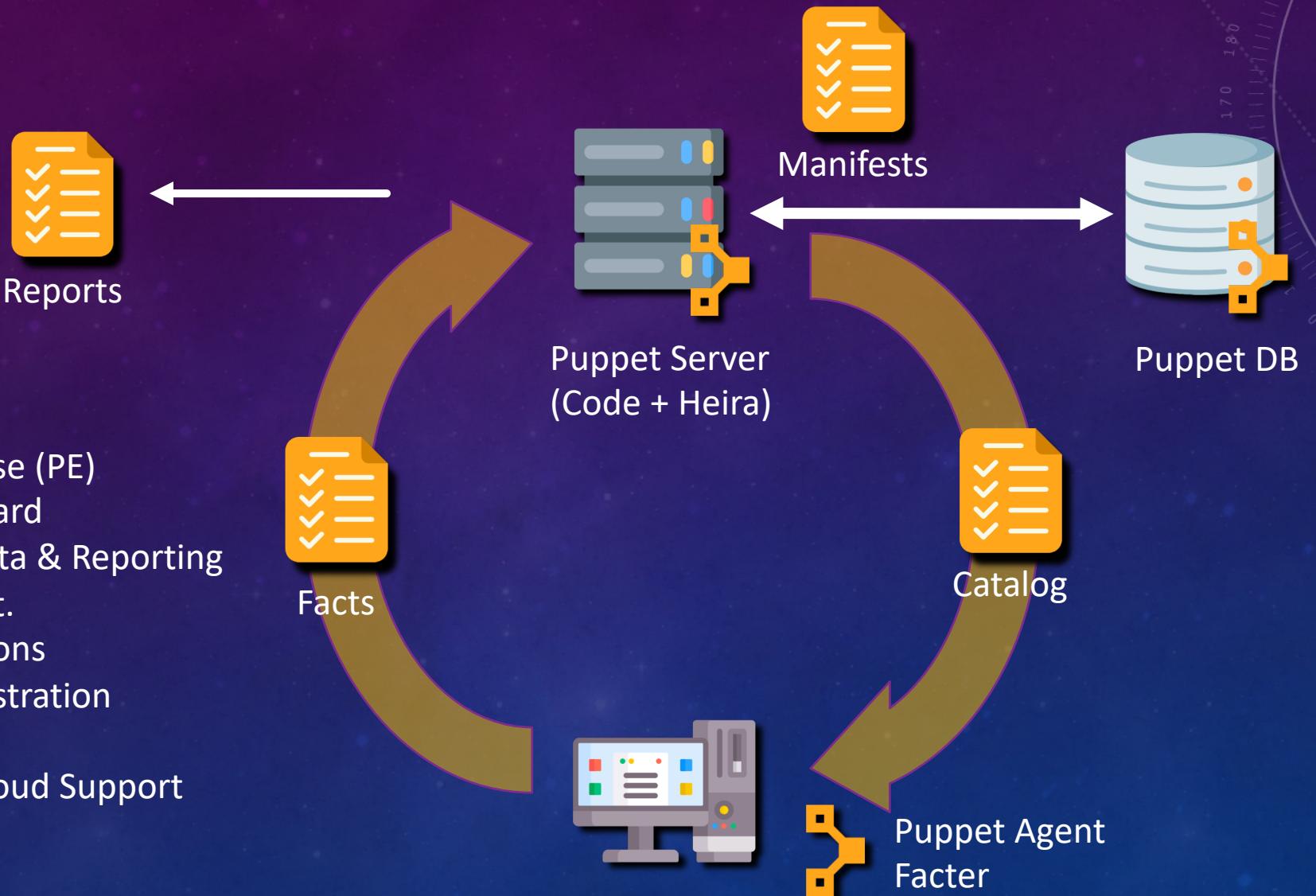
- You may have to write a LOT of PowerShell 😂
 - Custom Resources
 - Composite Configurations & Resources
 - Hierarchical / Data-Driven Config.
- Heavy customization can effect sustainability
- Move to Class-Based Resources (compatibility)
- Lacks some native functionality in other solutions
 - RBAC
 - API-level Integrations
 - Robust Server



PUPPET OVERVIEW

- Configuration Management tool (c. 2005)
- Client-Server architecture
- Written in Ruby, some C++/Clojure
- Puppet DSL (based on Ruby); PDK & VS Code Extension
- OSS (Apache); Puppet Enterprise
- Platform-Agnostic (Linux, Unix-like, Windows)
- Declarative & Idempotent; Manifests + Facts = Catalogs + Reports

PUPPET



Puppet Enterprise (PE)

- Web Dashboard
- Advanced Data & Reporting
- RBAC / AD int.
- API/Integrations
- Better Orchestration
- CD for PE
- Container/Cloud Support
- Support

PUPPET PRO'S

- Mature Toolset
- Better Scalability (agent)
- Great for Complex Environments
- Hierarchical Configuration
- Cross-Platform / Hybrid support
- Advanced Enterprise Features (PE)
- Great for Change Management / Testing
- Supports Ad-Hoc Executions (Bolt)

PUPPET CON'S

- Complicated Toolset
- Ruby DSL 😬
- Some Customization Required
- Heavier Footprint
- Higher Initial Cognitive Load
- Advanced Features (Dashboard/API) in PE (\$)

ANSIBLE OVERVIEW

- OSS Config. Mgr. (c. 2012)
- Enterprise Version: Ansible Automation Platform (formerly Tower)
- Agentless and Cross-Platform
 - SSH on Linux/Unix
 - WinRM on Windows
- Written in Python
 - Windows Modules are PowerShell-based
- Declarative & Idempotent
- YAML DSL (Playbooks), Hosts, Roles, Modules

ANSIBLE



Playbook
Modules
Collections
Hosts
Roles



Ansible
Automation
Platform

Web Dashboard
API / Integrations
RBAC
Inventory
Scheduling
Reporting



ANSIBLE PRO'S

- Agentless 😊
- Native Remoting and Code (WinRM & PowerShell)
- Ease of Setup
- Pipeline-Ready
- Supports Ephemeral Container Use
- Supports Majority of Networking Appliances
- Advanced Feature set with Ansible Automation Platform

ANSIBLE CON'S

- Agentless, so not always Scalable 😬
- Limited to Resources on Control Node / Server
- YAML is Space-Sensitive (VS Code & VIM Linting)
- Requires Linux Control Node (WSL / Container)
- Advanced Feature set with Ansible Automation Platform (\$)



DEMO



ANSIBLE RESOURCES

- Ansible for DevOps by Jeff Geerling
 - <https://www.ansiblefordevops.com/>
- Become Ansible by Josh Duffney
 - <https://gumroad.com/l/become-ansible>



PUPPET RESOURCES

- Puppet Forge:
 - <https://forge.puppet.com/>
- Puppet Learning VM
 - <https://puppet.com/try-puppet/puppet-learning-vm/>
- Puppet Community Slack
 - <https://slack.puppet.com/>

DSC RESOURCES

- The DSC Book:
 - <https://leanpub.com/the-dsc-book>
- DSC Community:
 - <https://dsccommunity.org/>
- DSC Community Slack
 - <https://aka.ms/PSSlack>

CONCLUSIONS

- NOT Exhaustive
- NO Winners & Losers
- Depends Entirely on Your Use Case
- Some Suggestions Based on My Research...

SUGGESTIONS BASED ON USE CASE

- Simple Configs. with PowerShell, Windows, Low-Cost (Azure)
 - DSC
- Advanced Tooling & Configurations, Cross-Platform, Large Fleet
 - Puppet
- Cross-Platform, Simplicity, Up & Running Quickly, Pipeline-Friendly
 - Ansible
- Servers & Workstations, Azure-AD joined, Remote Work Force
 - Microsoft Endpoint Manager (MECM & Intune)



THANK YOU!

- Please Feel Free to Reach Out!
 - Twitter: @adilio
 - Github: @adilio
 - LinkedIn: /in/adilio
 - PowerShell Discord:
 - aka.ms/PSDiscord

