Information Application Services

Using Ansible to Enable DevOps Efficiency and Effectiveness

Lt Col Dorian Seabrook
(Head of Operations)

Aidan Beeson
(Linux Architect)

Presentation Overview

https://www.flickr.com/photos/uk-forces-afghanistan

DevOps Culture

https://github.com/gchq/BoilingFrogs

Building a product is a lot like a combat mission. A team of *skilled people* operate in conditions of *high uncertainty*; a commander sets *clear outcomes* with some *guiding principles*; but we *expect the unexpected*; and, we're *trained to take best action*, responding to new information as the situation unfolds. - @jonnyscheider

https://www.mindtheproduct.com/2017/09/understanding-design-thinking-lean-agile-work-together/

Information Application Services

Military

Civil Servants

IT Contractors

100+ Staff

200+ Services

3 Security Domains

Army, Navy & Air Force

Users: Army and Defence

~10

~300,000

Training Recruits

Families

Injured

Regular

Reserves

Veterans

Clouds

Public Cloud

rackspace

Official Sensitive
Private Cloud

Secret
Private Cloud

Army Hosting Environments

Private Cloud Technology

Virtualisation:

Operating Systems:

Development:

Applications and Services

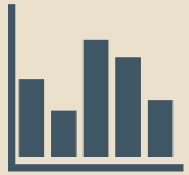
HR

Geo Mapping

Training

CRM

ERP



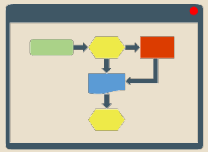
Analytics

ARMY

Hosting Environment

Secure Comms

Legal



Planning

Logistics

Data Warehouse

Accounting

Issues to be resolved

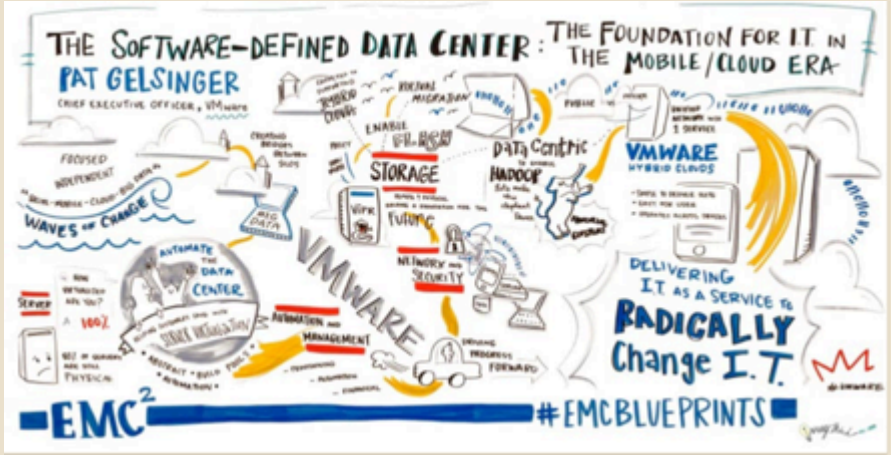
Management Issues:

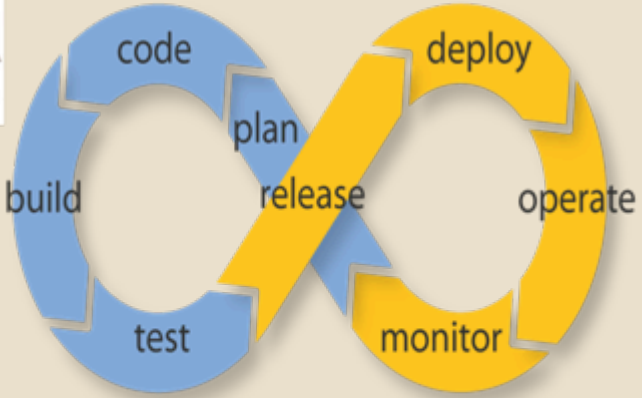
- Time to Deploy
- Platform Configuration Control
- Disruption to Users
- Documentation

Technical Issues:

- Supporting existing code
- How to deal with platforms
- Controlling deployment to complex platforms
- Making it simple for support staff
- Getting results quickly

Drivers for DevOps

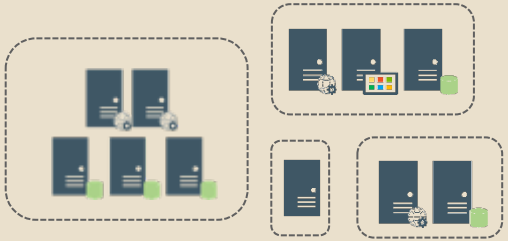




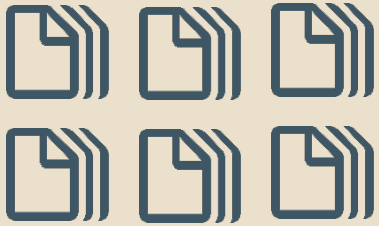
Endless Possibilities: DevOps can create an infinite loop of release and feedback for all your code and deployment targets.

Where it all started

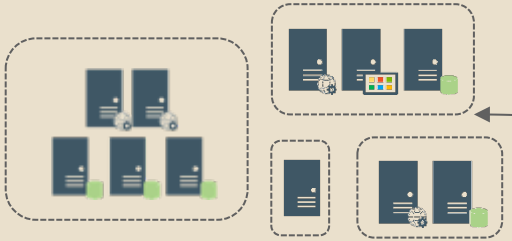
Pre-DevOps



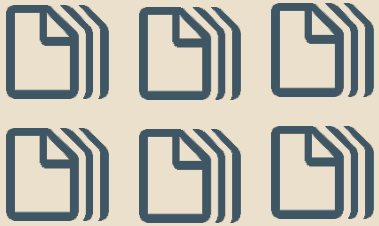
Secured OS Build.
Customisations baked into the ISO.



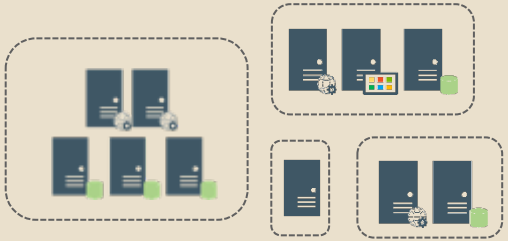
Pre-DevOps



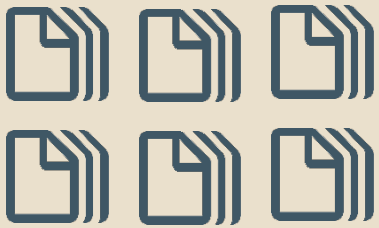
Several platform types of differing configurations



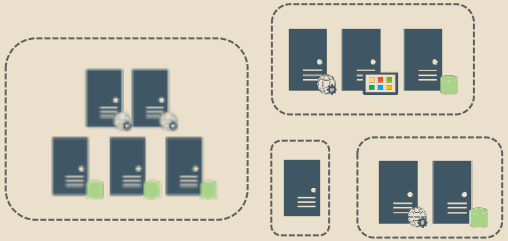
Pre-DevOps



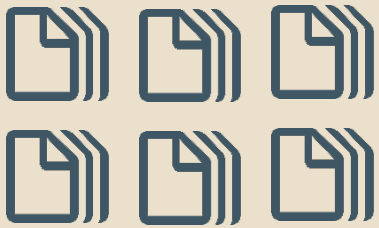
Lots of documentation detailing the platform build and maintenance steps



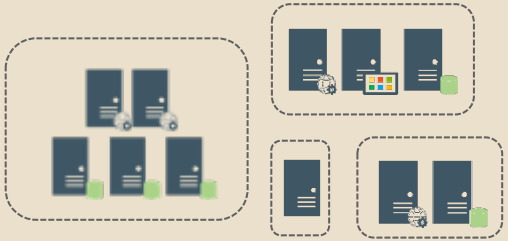
Pre-DevOps



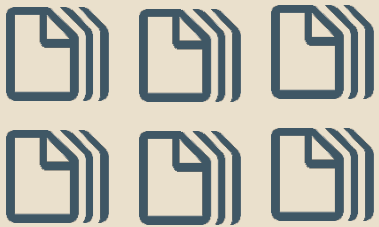
“Mostly” automated build and patching scripts, different scripts for different tasks.



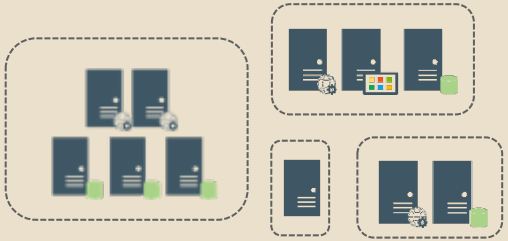
Pre-DevOps

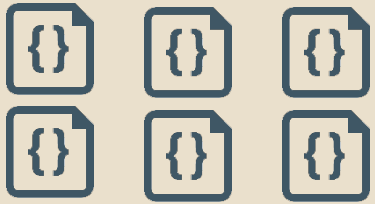


Lots of media for Building and patching

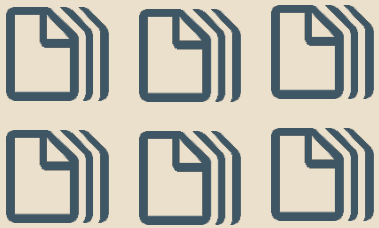


Pre-DevOps

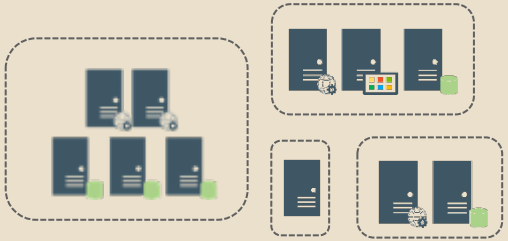


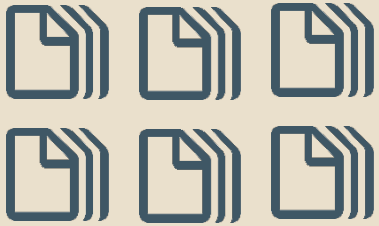


Spreadsheets containing system build information, infrastructure and application passwords



Pre-DevOps



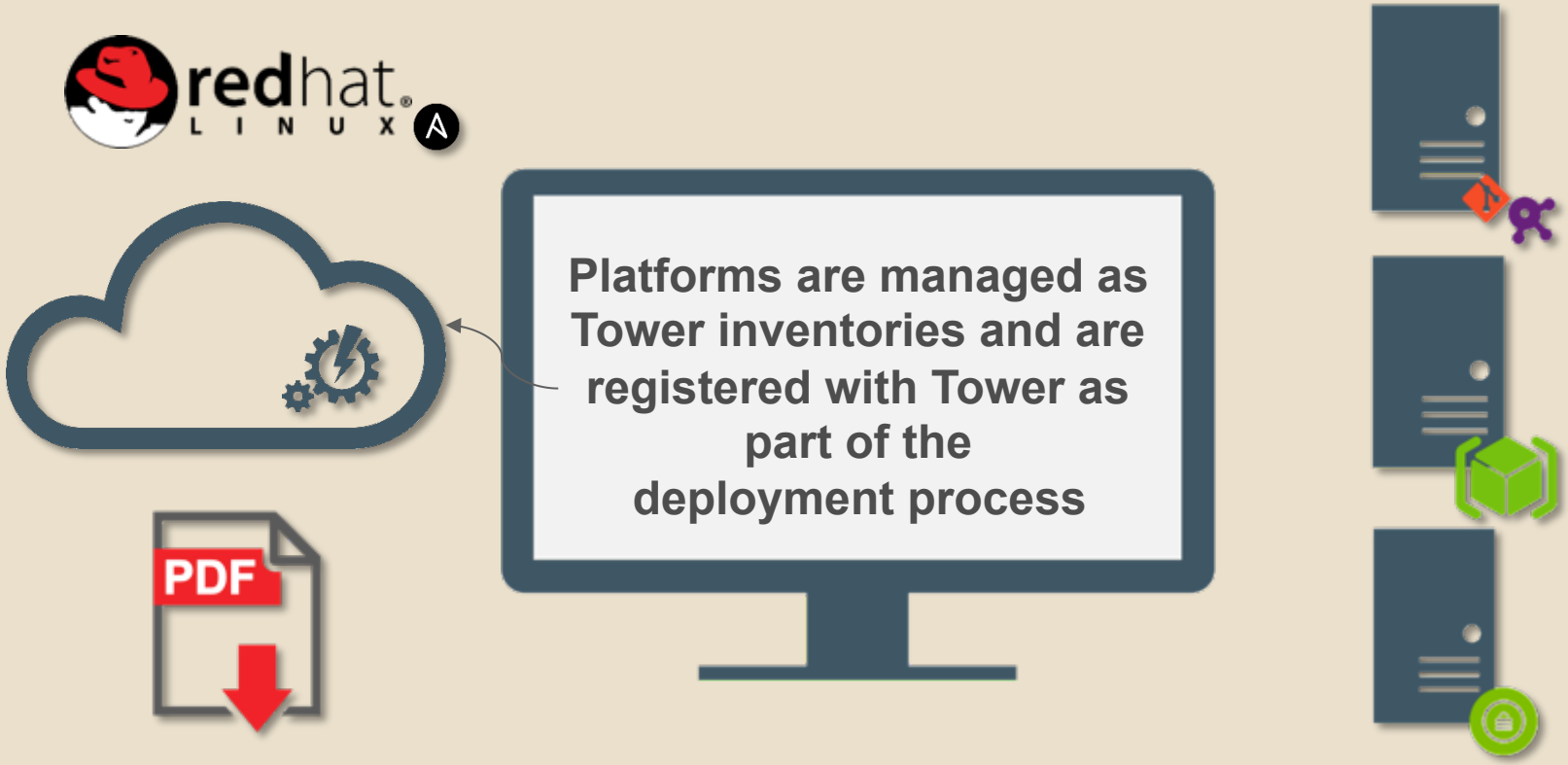


Post-DevOps

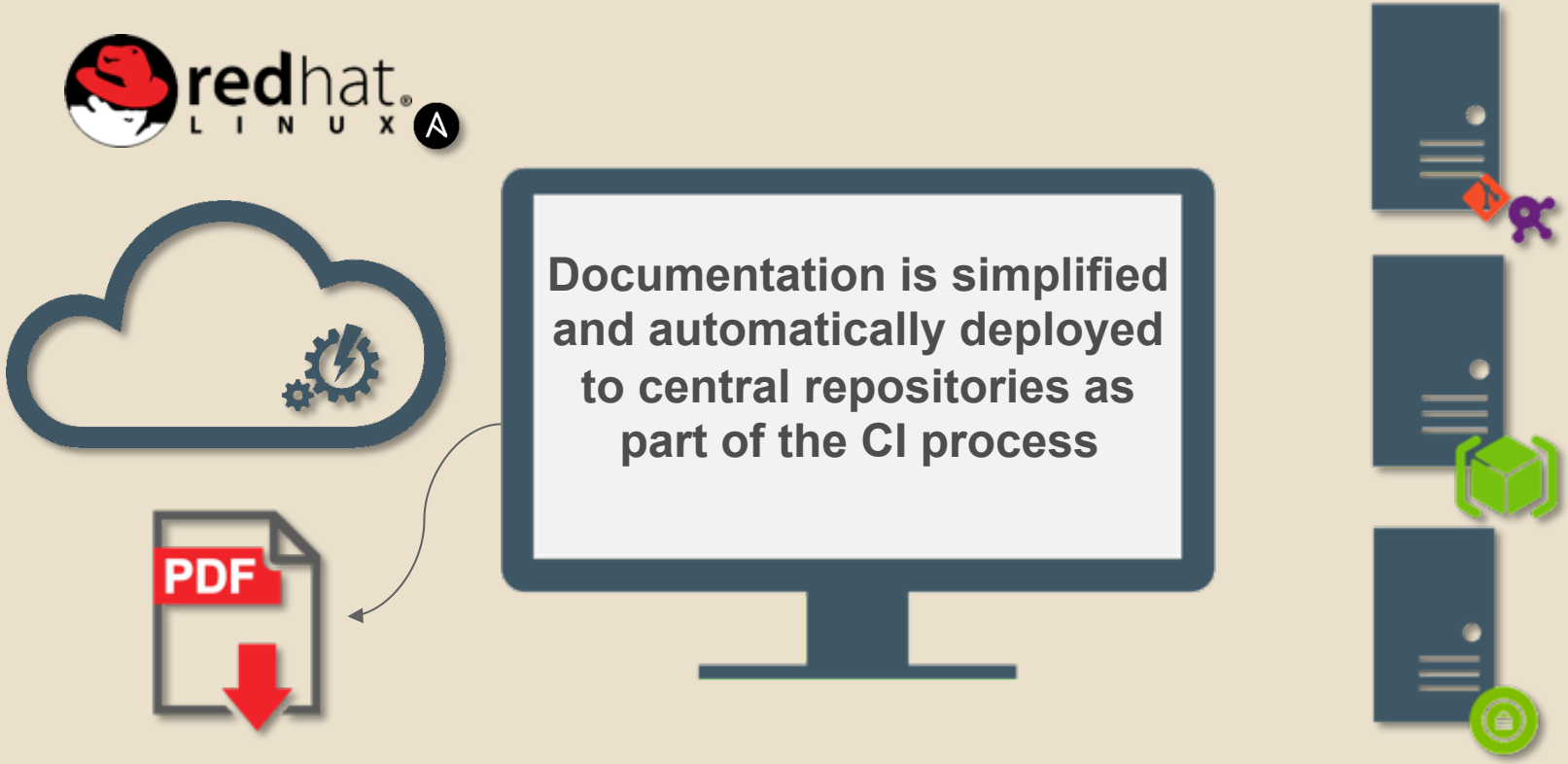
Post-DevOps


redhat.
LINUX
OS Customisations are now performed by playbooks, not baked into the Base OS ISO
PDF

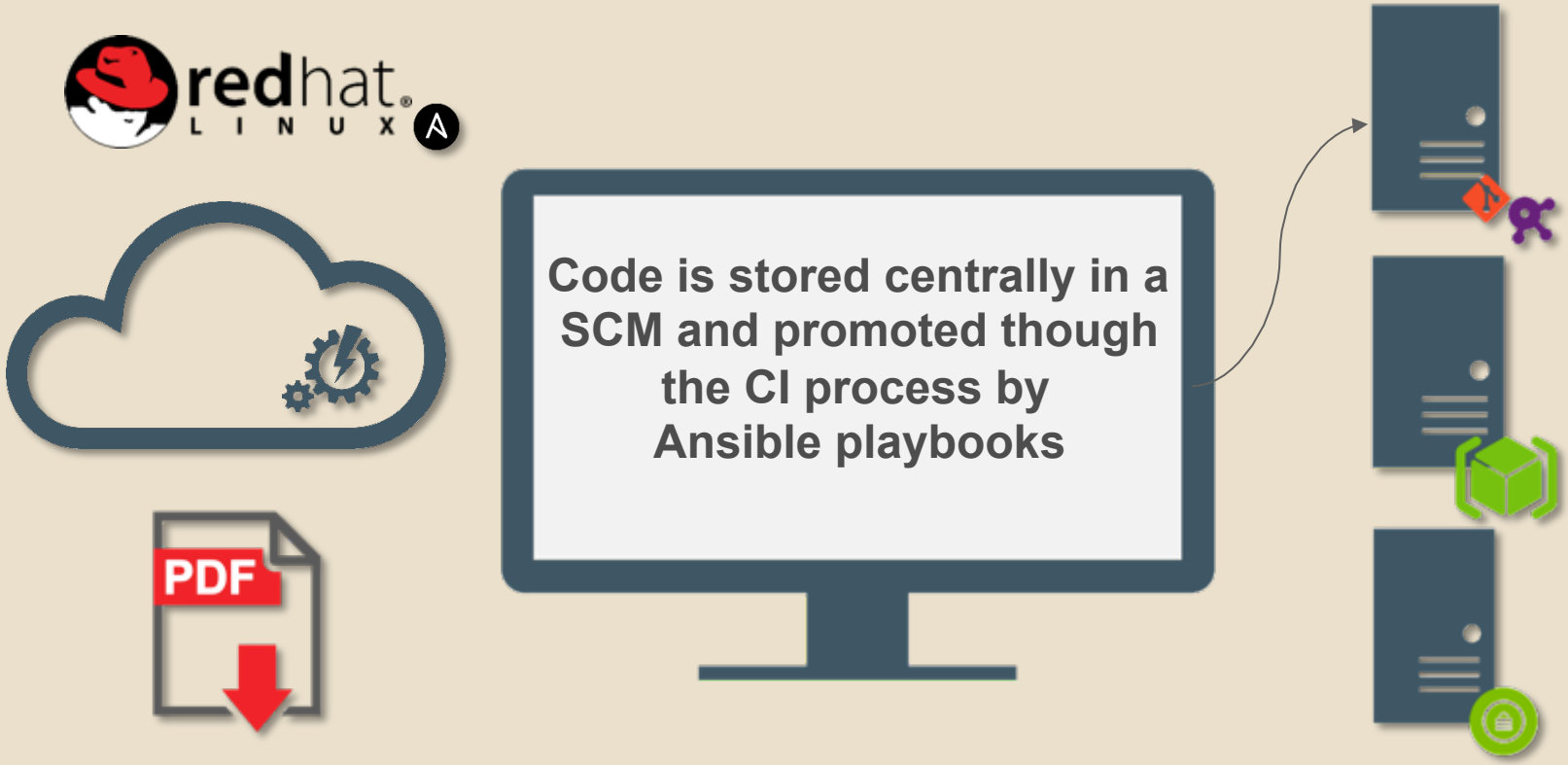
Post-DevOps


redhat.
LINUX
Platforms are managed as Tower inventories and are registered with Tower as part of the deployment process
PDF

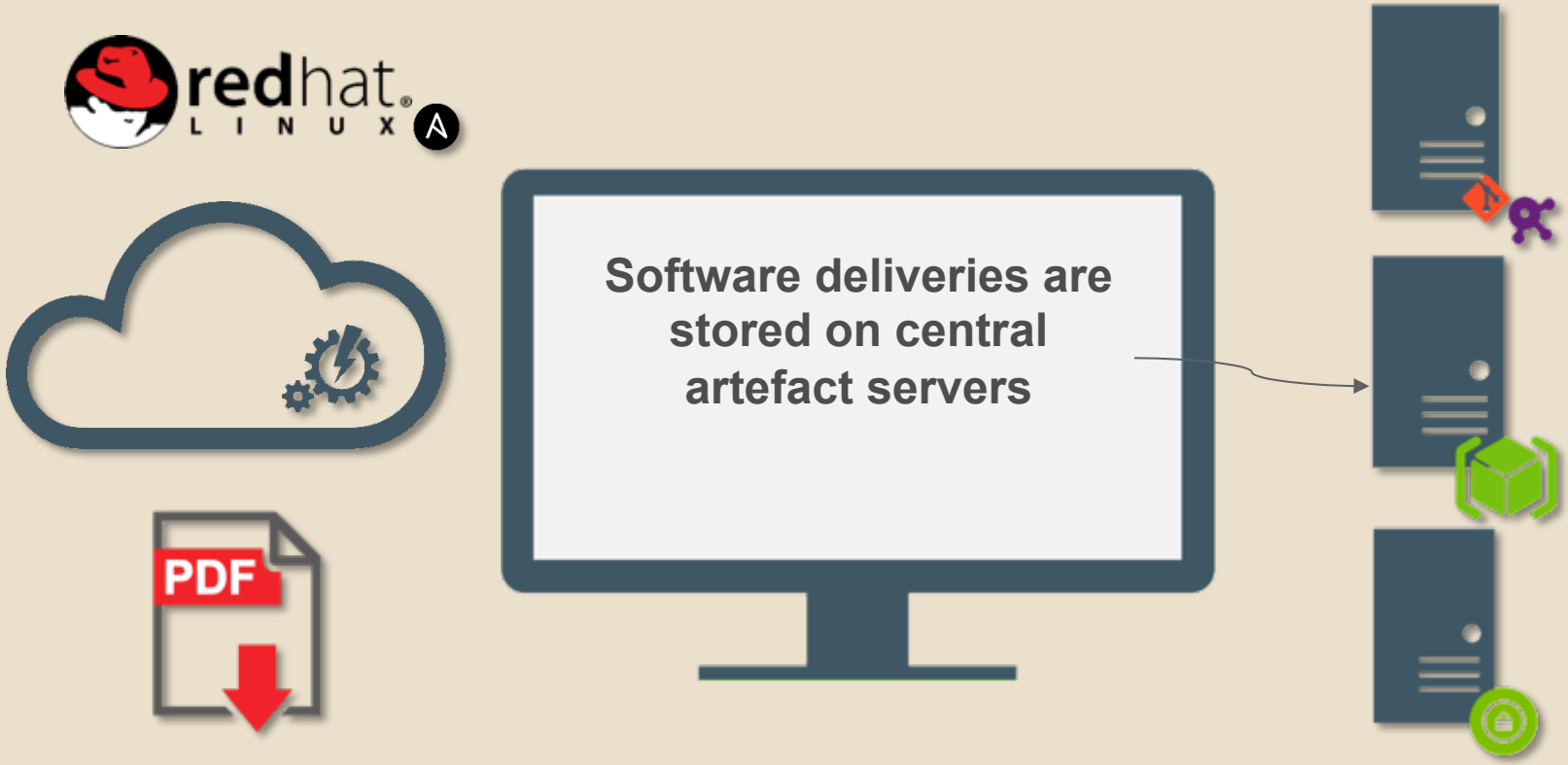
Post-DevOps


redhat.
LINUX
Documentation is simplified and automatically deployed to central repositories as part of the CI process
PDF

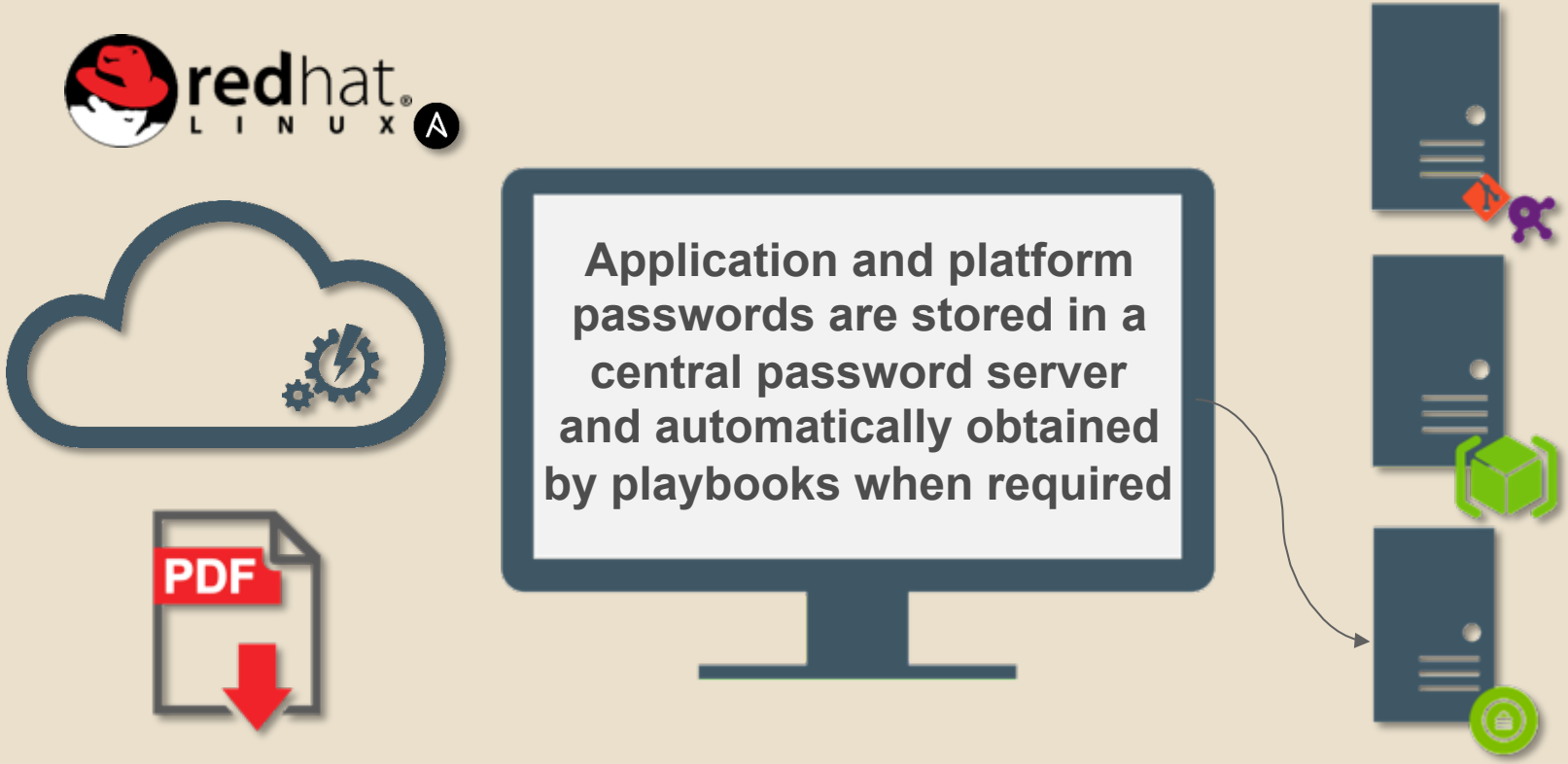
Post-DevOps


redhat.
LINUX
PDF
Code is stored centrally in a SCM and promoted though the CI process by Ansible playbooks

Post-DevOps


redhat.
LINUX
PDF
Software deliveries are stored on central artefact servers

Post-DevOps


redhat.
LINUX
PDF
Application and platform passwords are stored in a central password server and automatically obtained by playbooks when required

Why Ansible & Ansible Tower?

Maps well to our different platforms

Powerful GUI

Agentless

Good technology support

Use of Ansible has led to

- Automated process is the easiest way for changes to be made.
- Changes are tested through automation and rolled out to the relevant servers.
- Minimal configuration drift across the pipeline.
- Changes are unattended.
- Errors made visible.

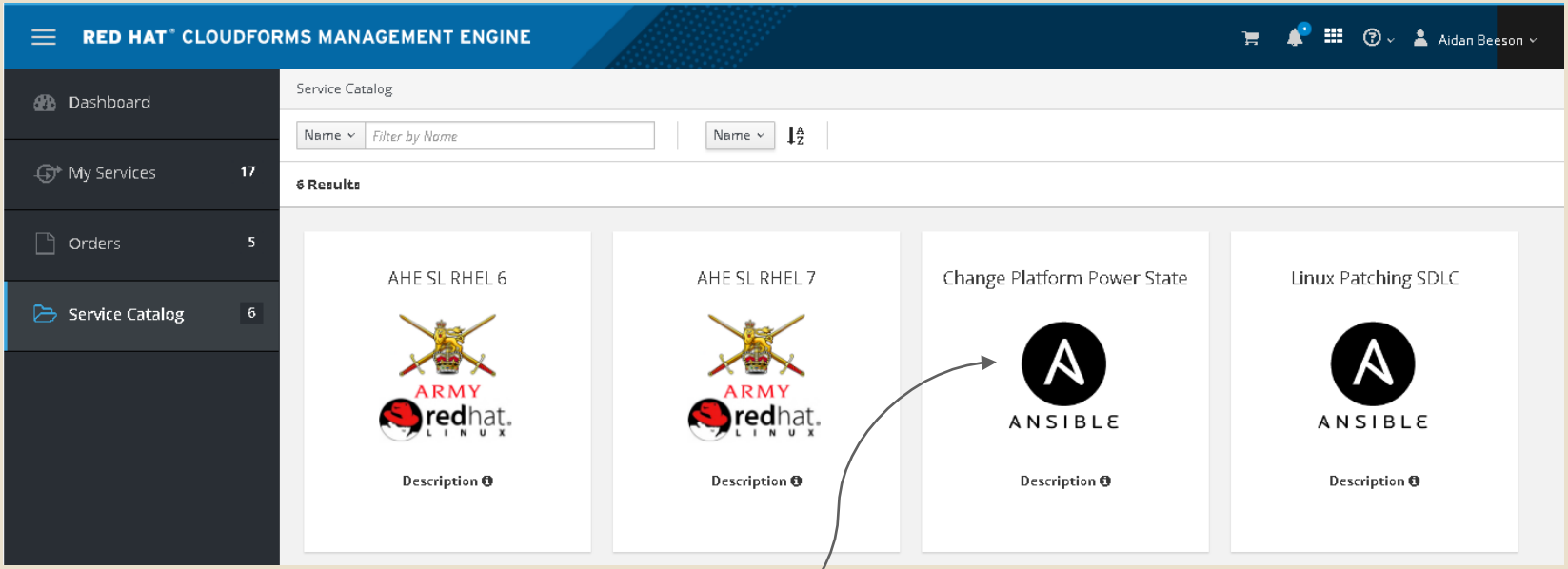
Time Savings

- Urgent changes:
  - Previously: days -> weeks
  - Now: less than a day
- Out of hours, fully automated installs:
  - Previously: ½ day+ of in-hours planned downtime
  - Now: Zero in-hours planned downtime, no “unforced errors”
- Test Team:
  - Previously: Patch application, test and rollback/retest took days -> weeks
  - Now: Application out-of-hours, removes days of effort from test cycle
- Development Team:
  - Previously: Dev platforms massively behind Production (sometimes years!)
  - Now: Less time investigating code issues due to system config differences

Cost Savings

- IAS Team:
  - Less 1st Line Support time required to manage systems and system patching.
  - Development and Test Teams spend more time developing and testing new content rather than troubleshooting infrastructure issues or loading patches.
  - Less “DBA Fiddling” – priceless!
- Customers / End Users:
  - No planned “in hours” disruption to service.
  - More new functionality delivered.

Future



CloudForms can run Ansible jobs natively or run them through Tower

Final Thought

“Legacy” no longer refers to the platform itself, but the way you deliver and maintain the platform