ARMY

Information Application Services

Efficiency and Effectiveness Through DevOps

Part II – "To Infinity and Beyond!"

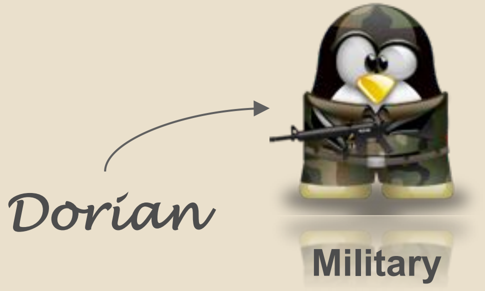
Aidan Beeson
(Linux Architect)

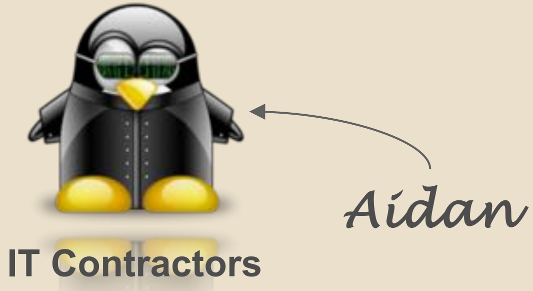
Lt Col Dorian Seabrook
(Head of Operations)

Presentation Overview

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

Information Application Services





100+ Staff

200+ Services

3 Security Domains

Army, Navy & Air Force

Private Cloud Infrastructure Technology

Application Servers: ORACLE WEBLOGIC, Microsoft IIS, APACHE HTTP SERVER

Databases: ORACLE, Microsoft SQL Server, MySQL, PostgreSQL

Operating Systems: redhat LINUX, Windows Server

Virtualisation: vmware, vmware NSX

Networking: CISCO

Compute / Storage: Hewlett Packard Enterprise

DevOps and Continuous Integration

Infrastructure:

ANSIBLE TOWER by Red Hat®

vmware® vRealize Suite

Application Pipeline:

Visual Studio Team Foundation Server

TC

DevOps “Hit List”

Concentrated initially on the Linux based issues:

- Base Operating System (BOS) Updates
  - Get the patches out regularly to all platforms whilst minimising downtime
- Oracle Infrastructure Configuration Change
  - Stop configuration drift between Production, Pre-prod, Dev and Test environments
- Oracle Patching
  - Oracle Critical Patch Updates
  - Oracle Upgrades

The Problem…

Hey Operations,
I need {{ something }}
done on {{ platform }}
in {{ environment }}
…
It's {{ priority }} !

(Where {{priority}} is *always*:
Priority 1
Priority 1+
Priority 1++…)

The Problem…

Operations

The Problem…

Documentation

Operations

The Problem…

Documentation

Operations

The Problem…

Documentation

Operations

Installation Media

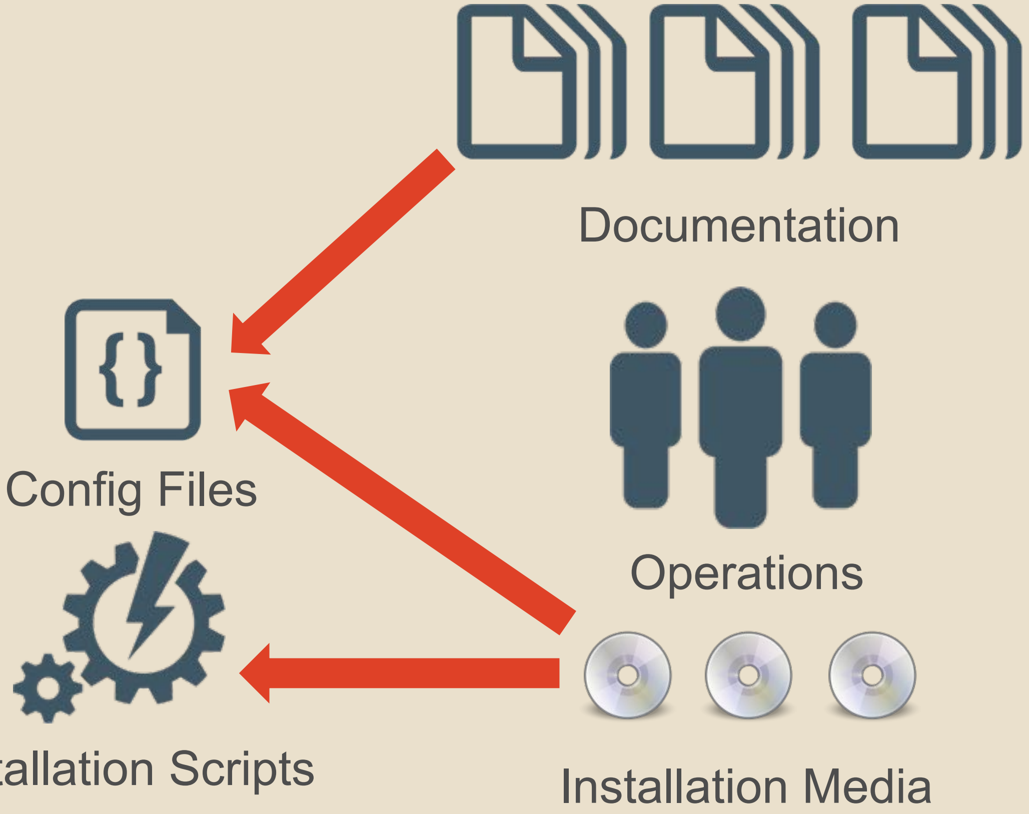
The Problem…

Documentation

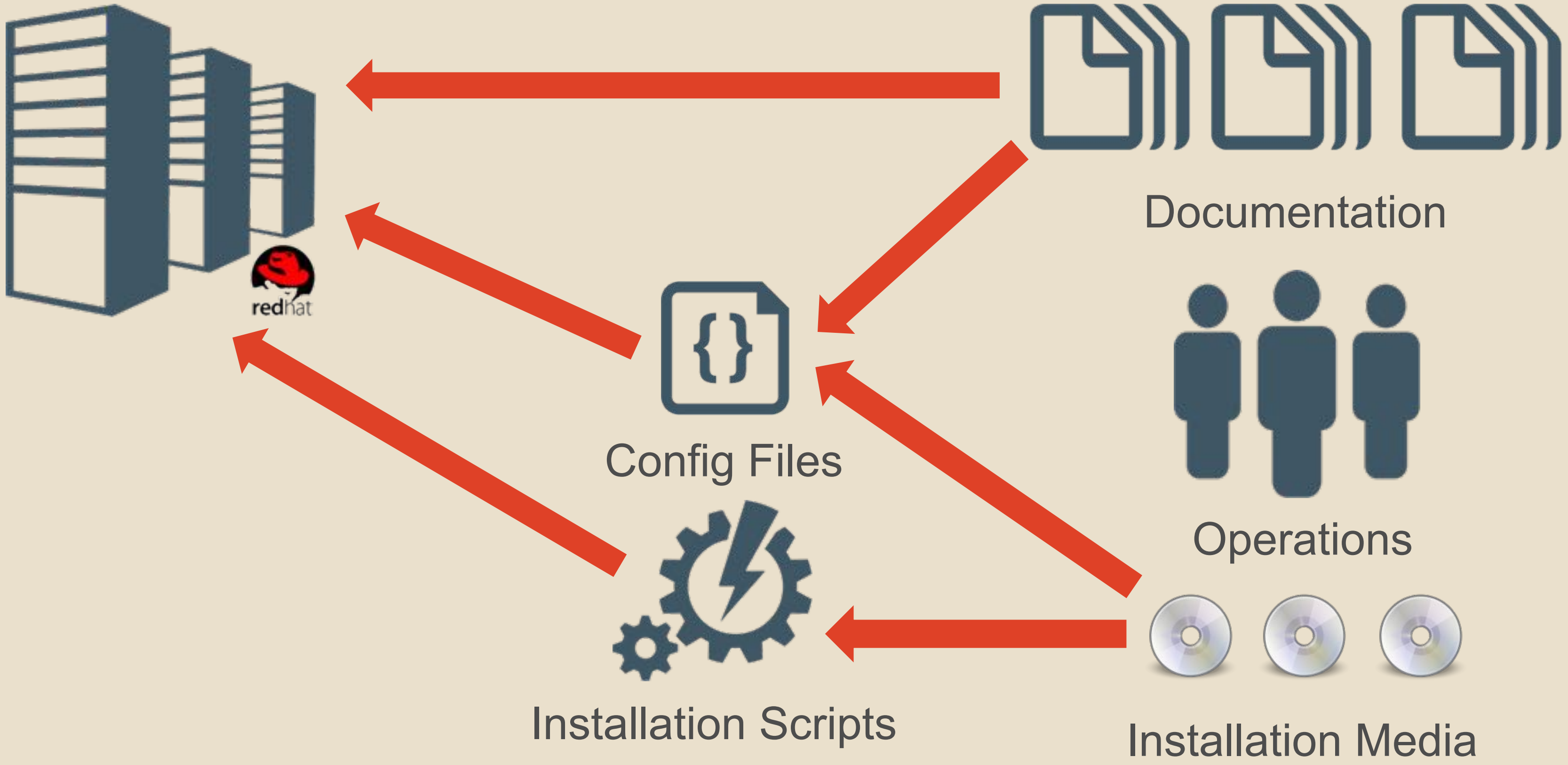
Operations

Installation Media

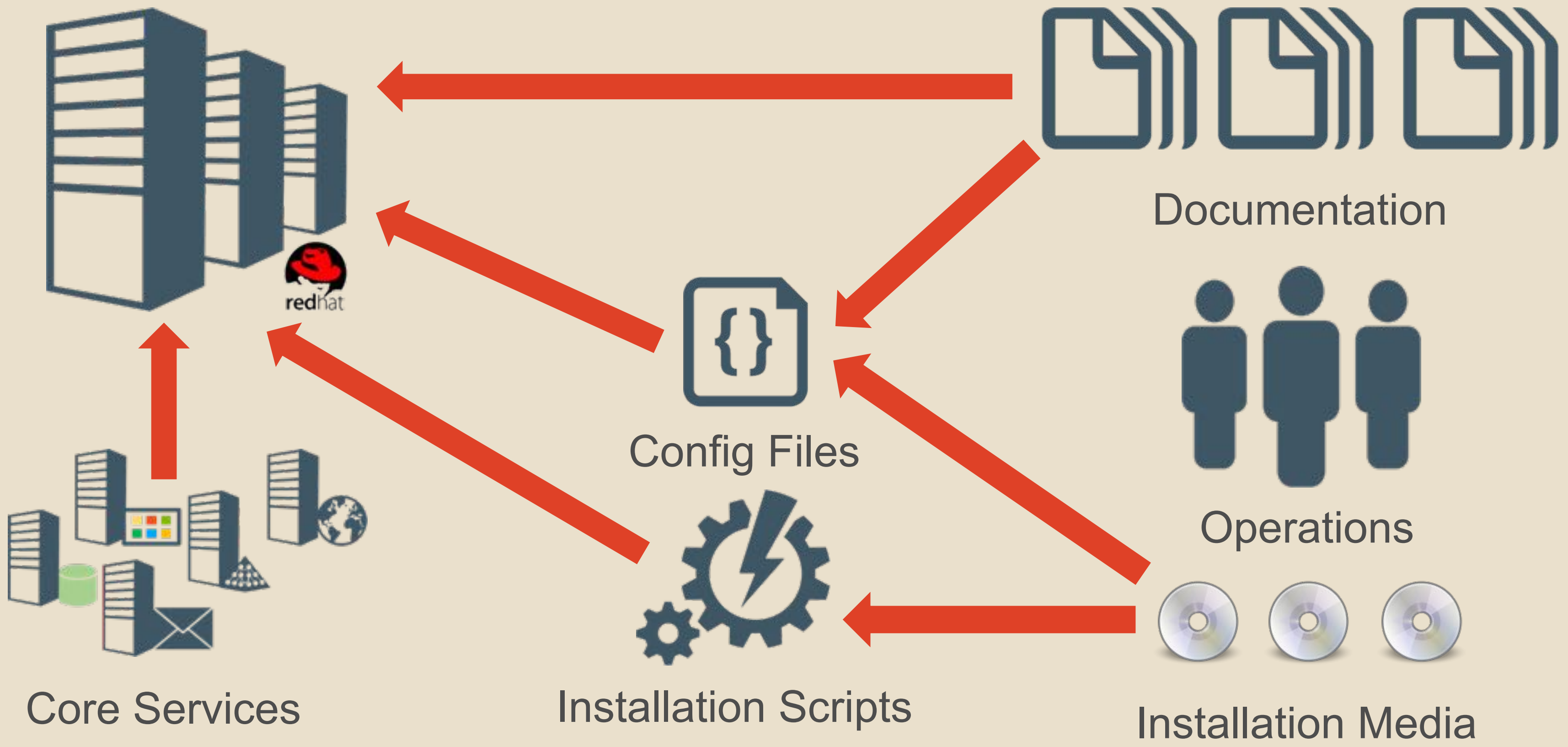
The Problem…


Documentation
Config Files
Operations
Installation Scripts
Installation Media

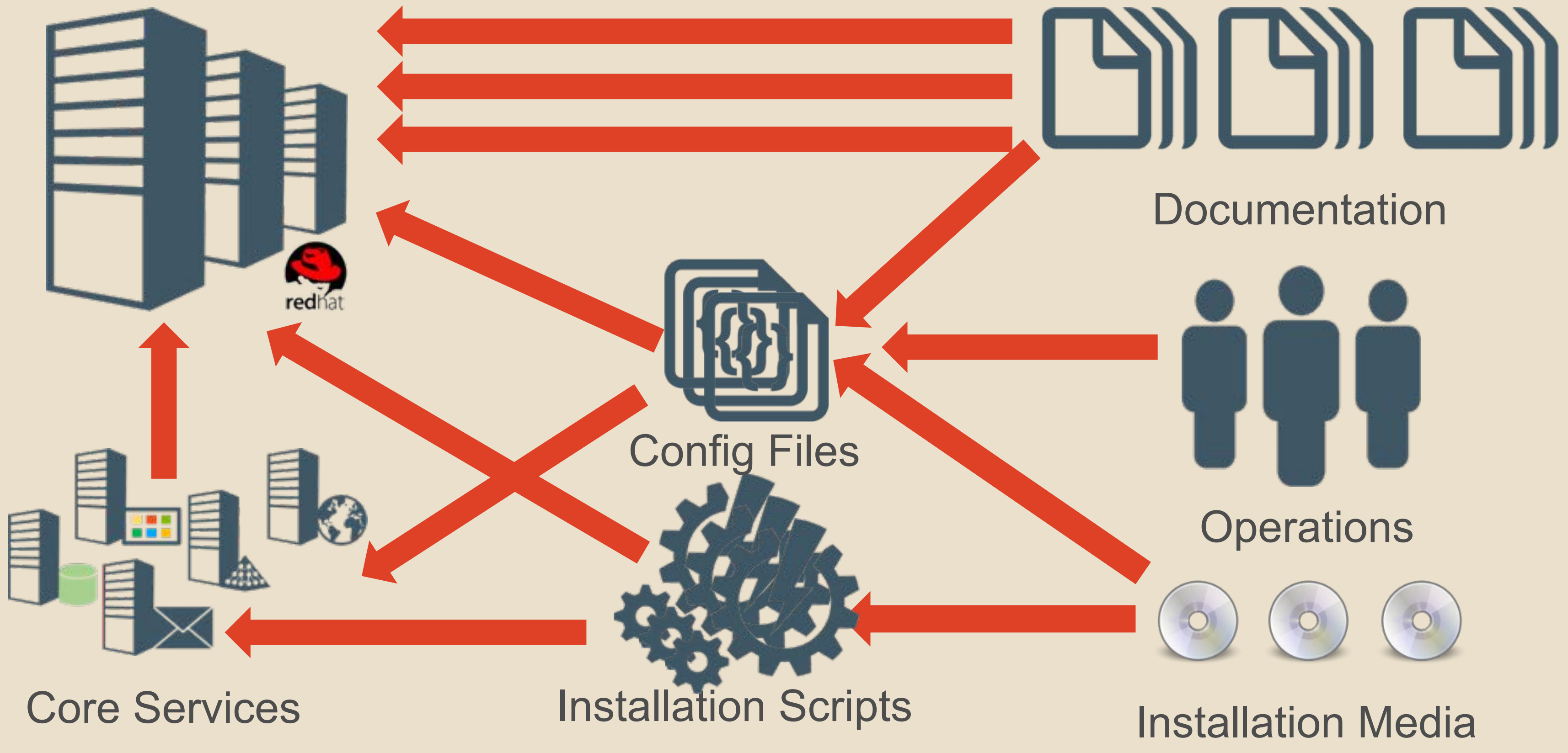
The Problem…


redhat
Documentation
Config Files
Operations
Installation Scripts
Installation Media

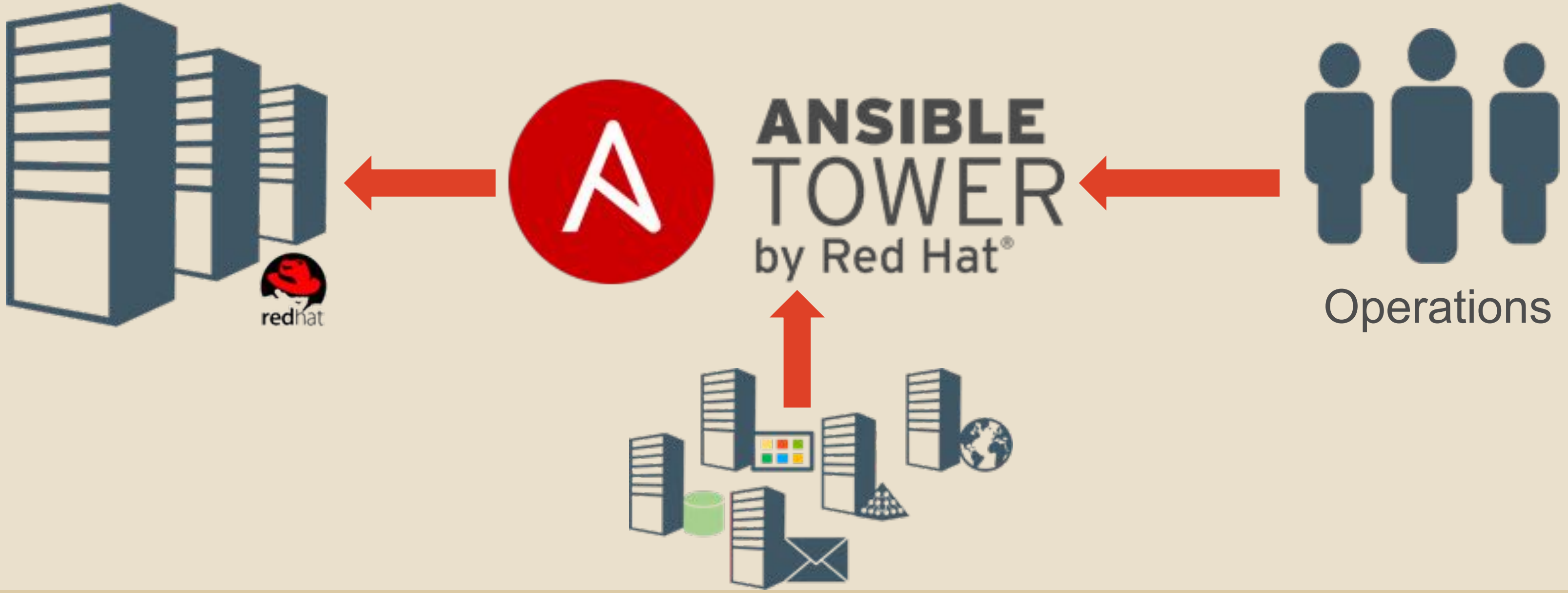
The Problem…


redhat
Documentation
Config Files
Operations
Core Services
Installation Scripts
Installation Media

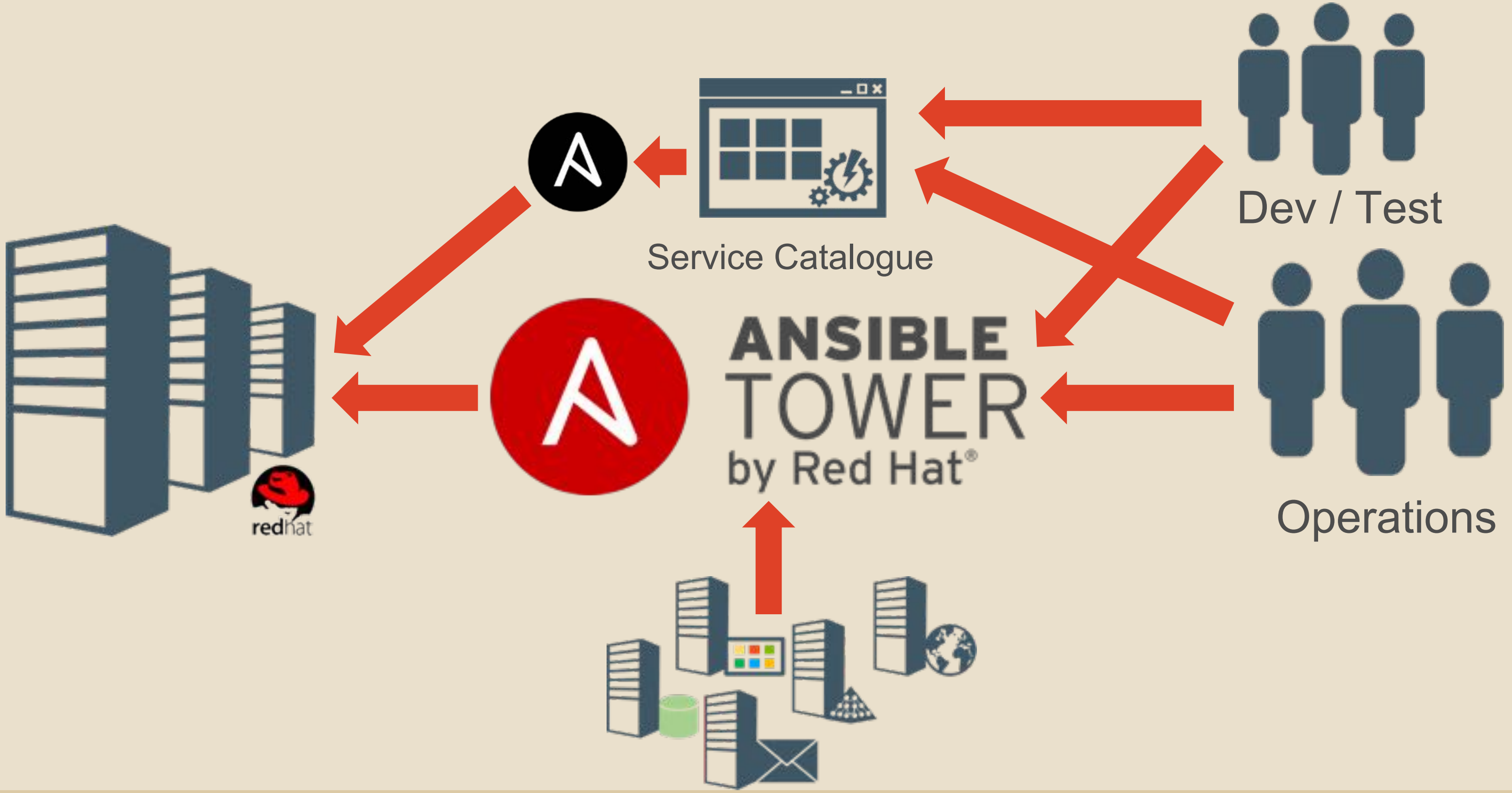
The Problem…


redhat
Documentation
Config Files
Operations
Core Services
Installation Scripts
Installation Media

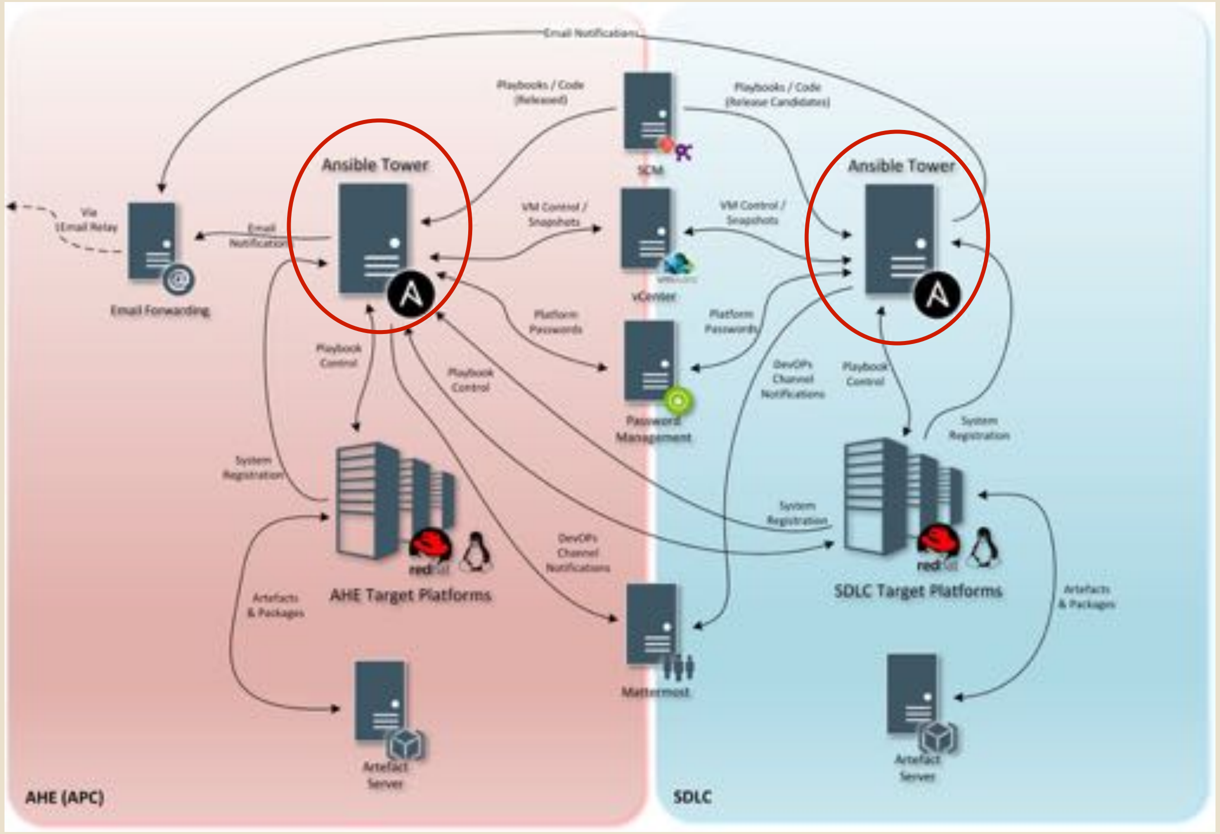
The Solution



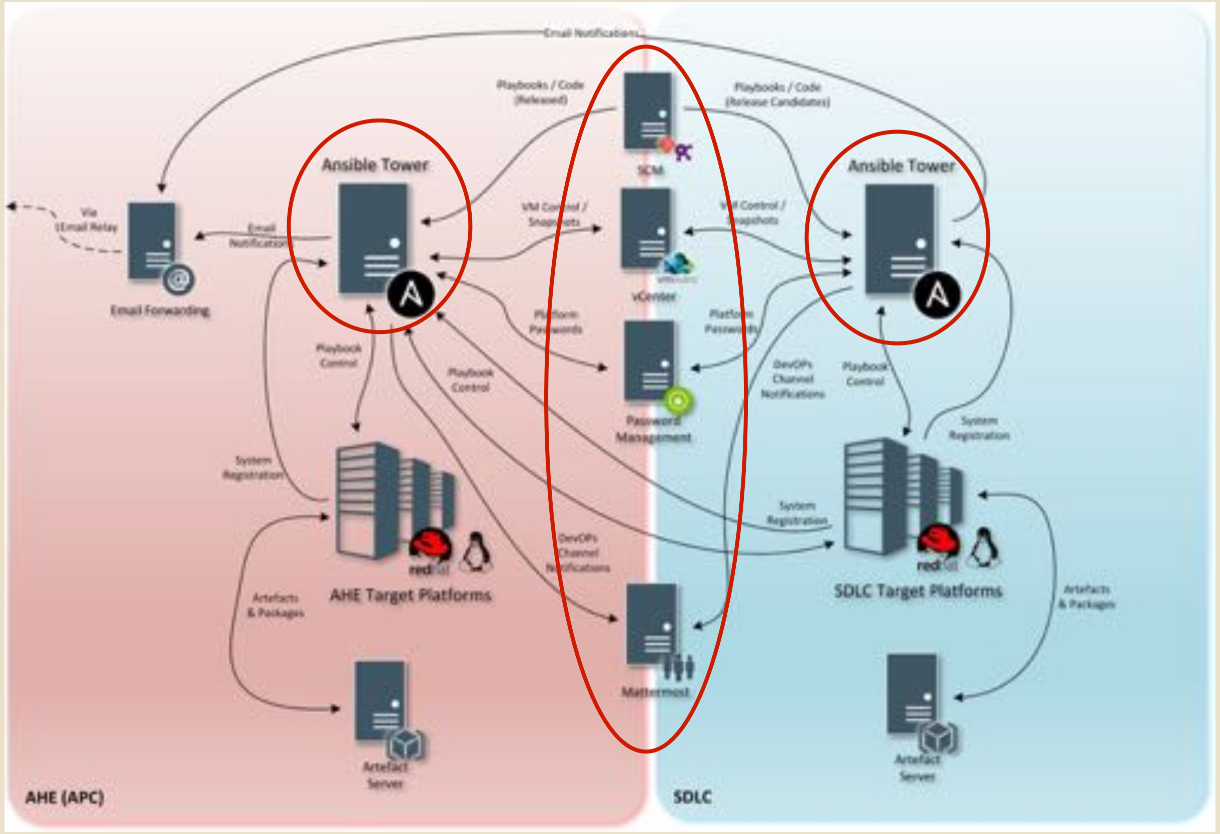
The Solution



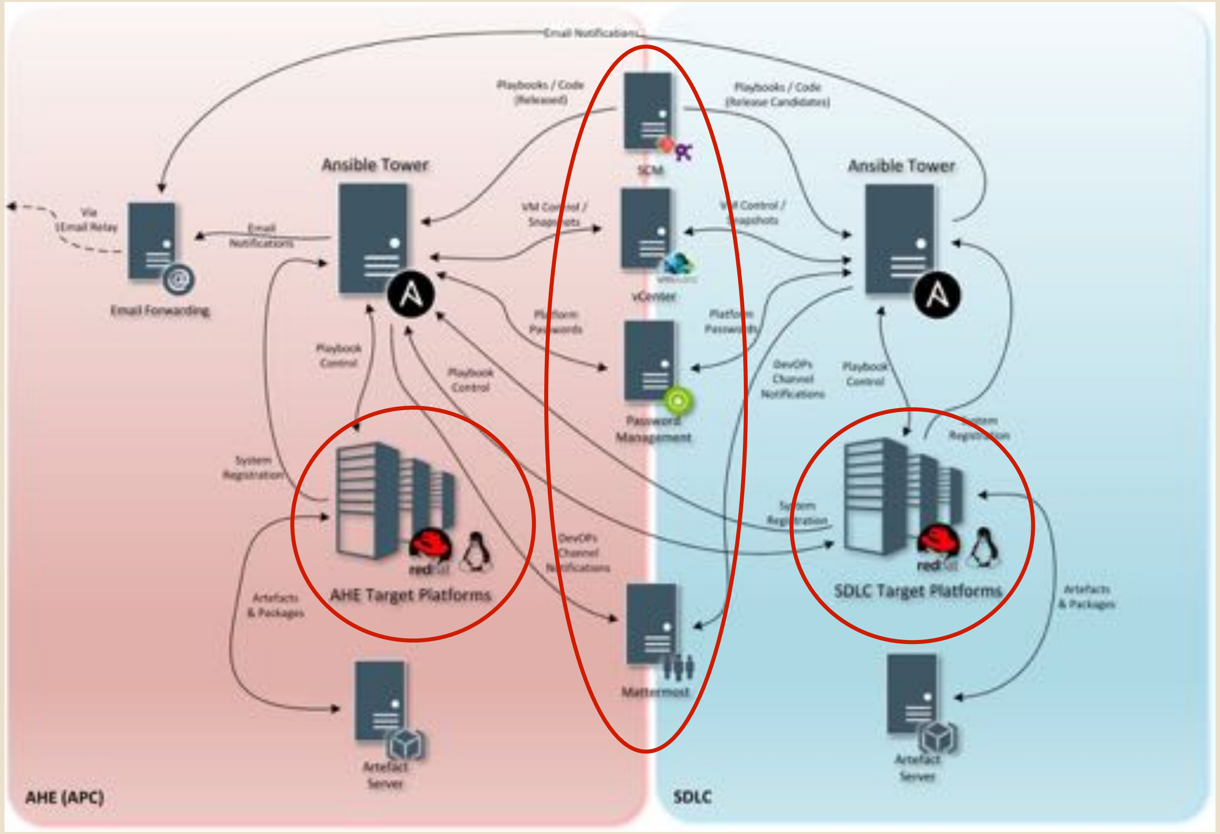
The Army Private Cloud



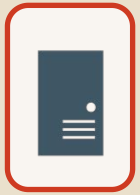
The Army Private Cloud



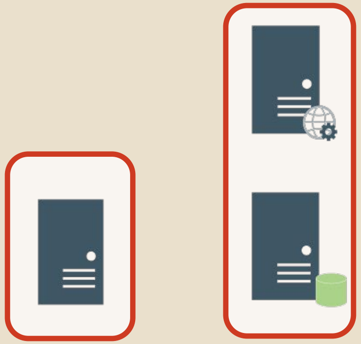
The Army Private Cloud



Herding Cats

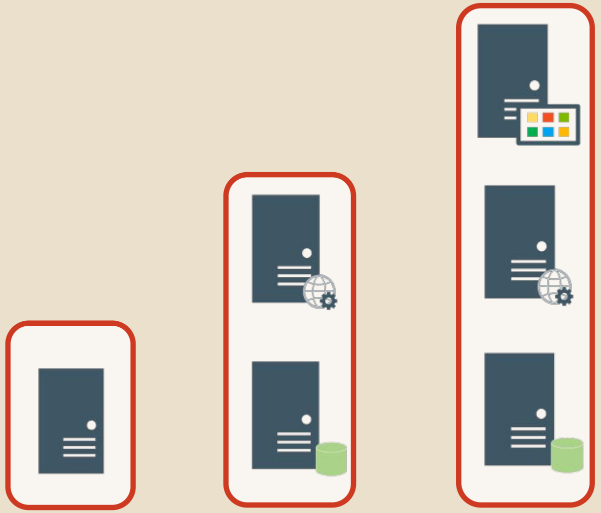


Herding Cats



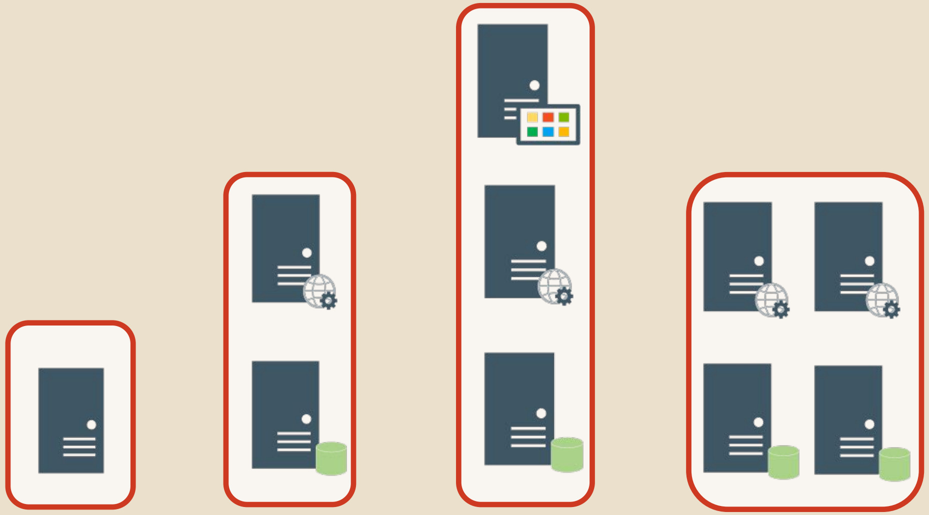
Dual Server Platform

Herding Cats



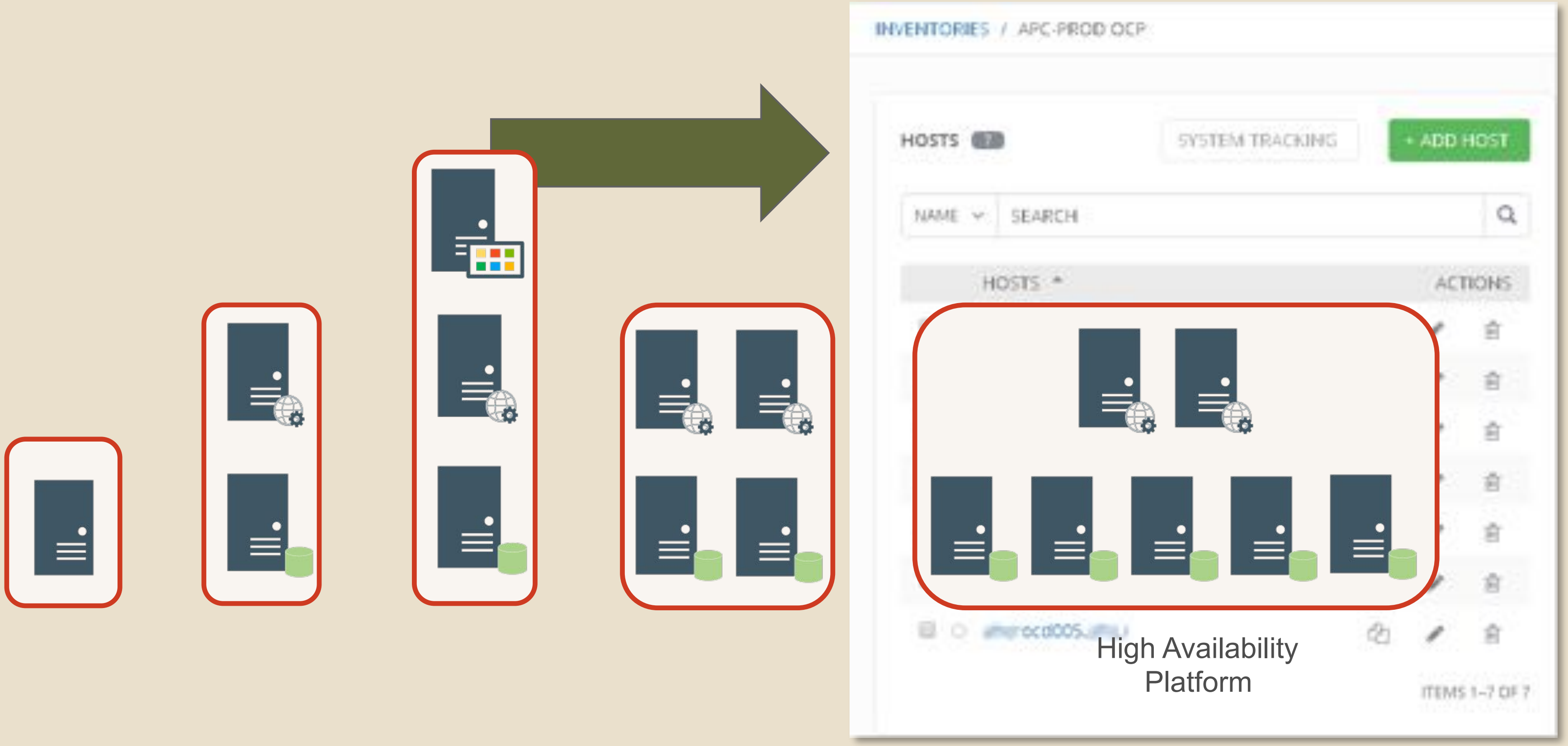
Multi Tier Platform

Herding Cats



High Availability Platform

Herding Cats



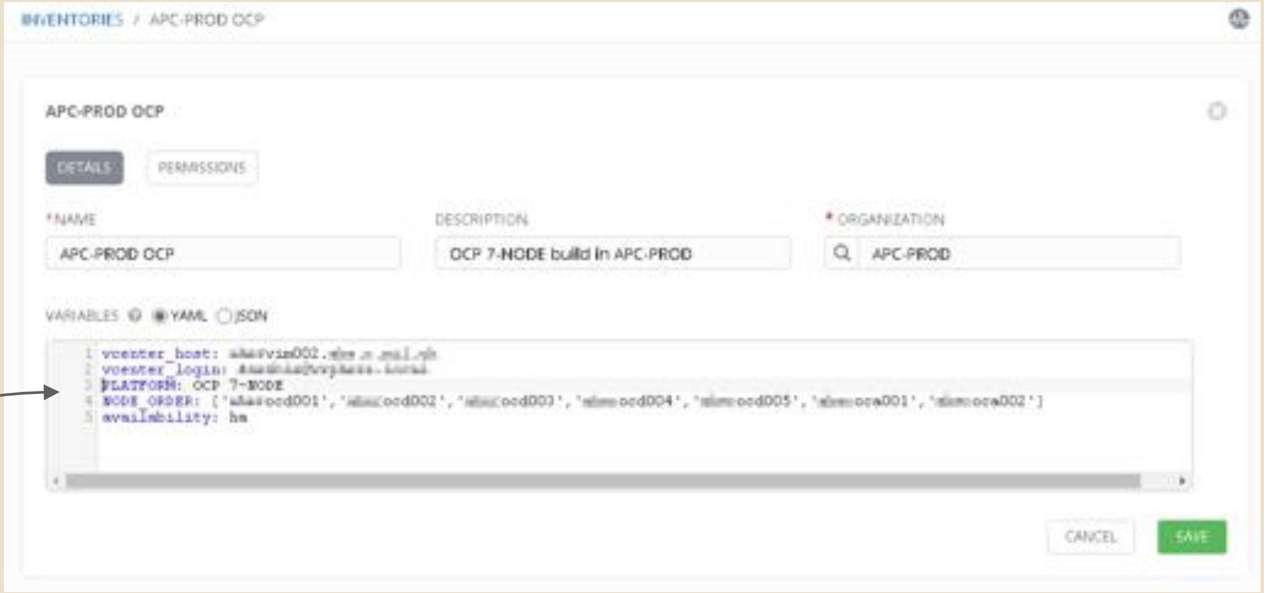
High Availability Platform

Static Inventories

You’re doing what?!? Why don’t you just get them dynamically from the hypervisor?

Ansible Tower– Inventory Variables

Small number of standard inventory variables per platform type



APC-PROD OCP

*NAME: APC-PROD OCP

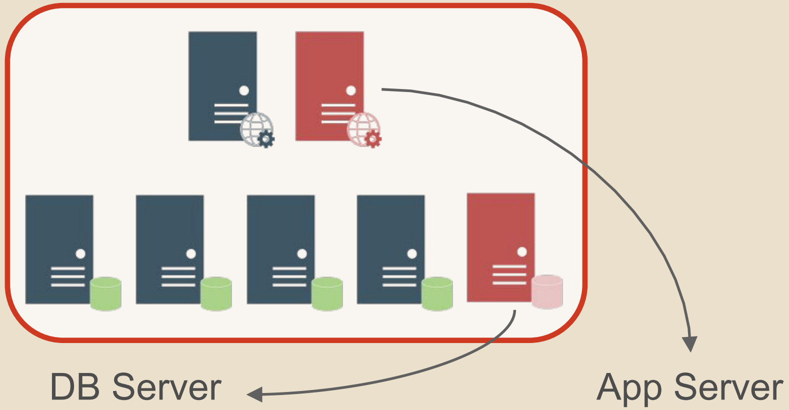
DESCRIPTION: OCP 7-NODE build in APC-PROD

*ORGANIZATION: APC-PROD

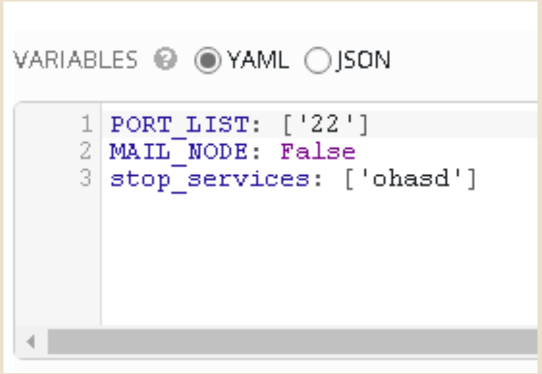
VARIABLES [✓ YAML] JSON

```
vcenter_host: [redacted]
vcenter_login: [redacted]
PLATFORM: OCP 7-NODE
NODE_ORDER: ['[redacted]ocd001','[redacted]ocd002','[redacted]ocd003','[redacted]ocd004','[redacted]ocd005','[redacted]oca001','[redacted]oca002']
availability: ha
```

Ansible Tower– Host Variables

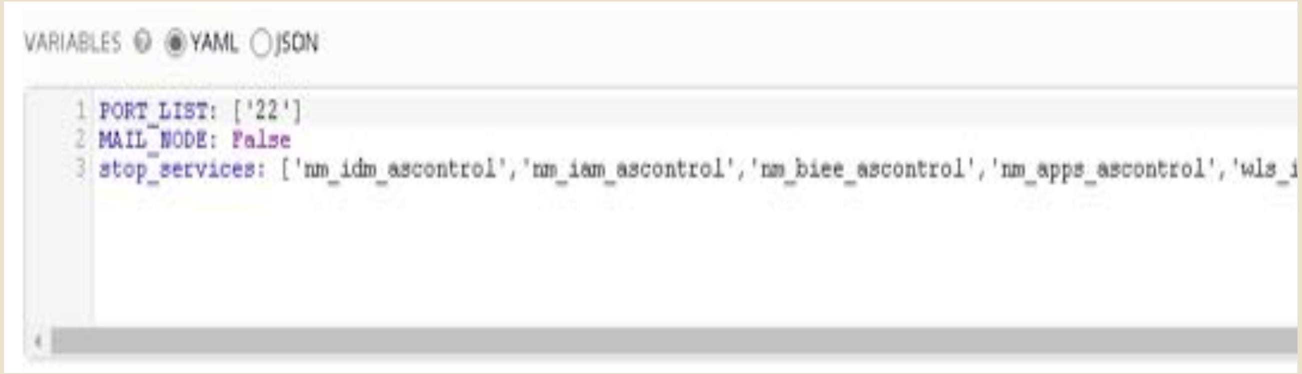


Standardised host variables, based on the platform type, are applied automatically at platform deployment



VARIABLES [✓ YAML] JSON

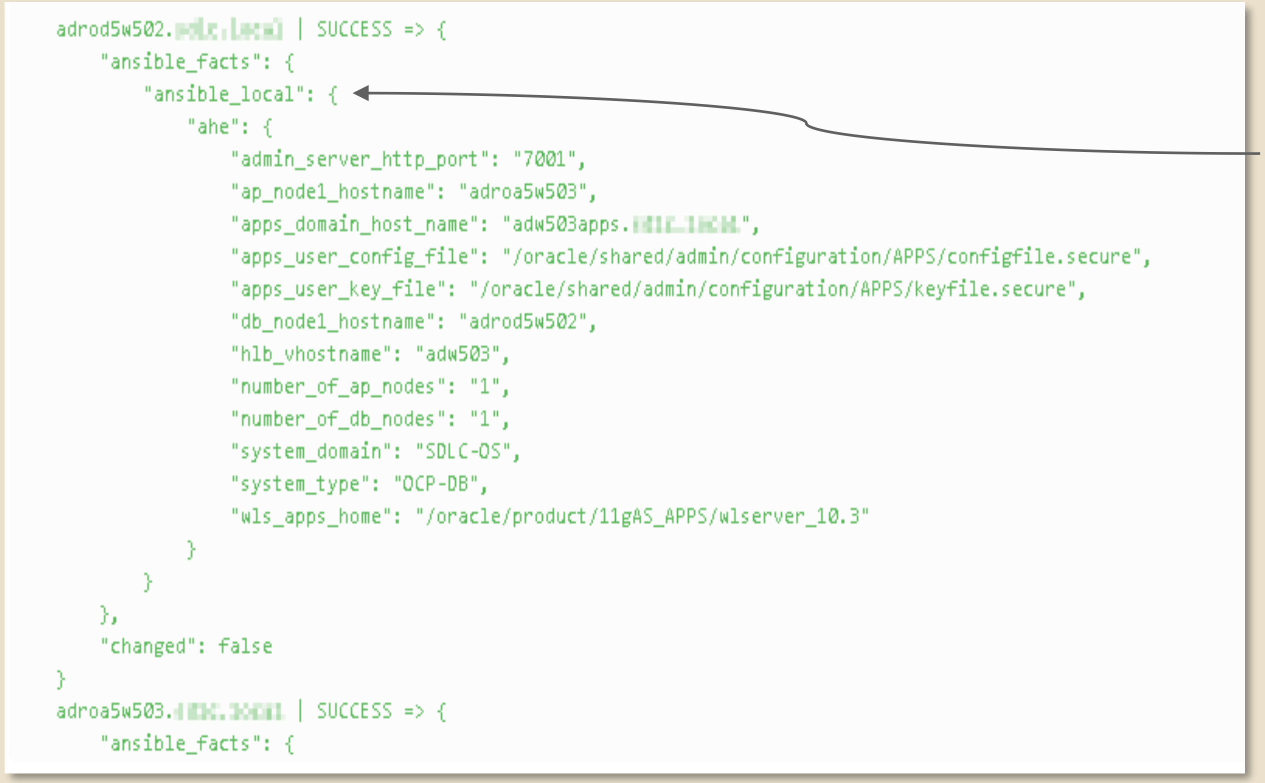
```
PORT_LIST: ['22']
MAIL_NODE: False
stop_services: ['ohasd']
```



VARIABLES [✓ YAML] JSON

```
PORT_LIST: ['22']
MAIL_NODE: False
stop_services: ['nm_idm_ascontrol','nm_iam_ascontrol','nm_biee_ascontrol','nm_apps_ascontrol','wls_i
```

Custom Facts



```
adrod5w502.[redacted] | SUCCESS => {
    "ansible_facts": {
        "ansible_local": {
            "ahe": {
                "admin_server_http_port": "7001",
                "ap_node1_hostname": "adroa5w503",
                "apps_domain_host_name": "adw503apps.[redacted]",
                "apps_user_config_file": "/oracle/shared/admin/configuration/APPS/configfile.secure",
                "apps_user_key_file": "/oracle/shared/admin/configuration/APPS/keyfile.secure",
                "db_node1_hostname": "adrod5w502",
                "hlb_vhostname": "adw503",
                "number_of_ap_nodes": "1",
                "number_of_db_nodes": "1",
                "system_domain": "SDLC-OS",
                "system_type": "OCP-DB",
                "wls_apps_home": "/oracle/product/11gAS_APPS/wlserver_10.3"
            }
        }
    },
    "changed": false
}
adroa5w503.[redacted] | SUCCESS => {
    "ansible_facts": {
```

Custom facts provide host information that is not stored in the inventory

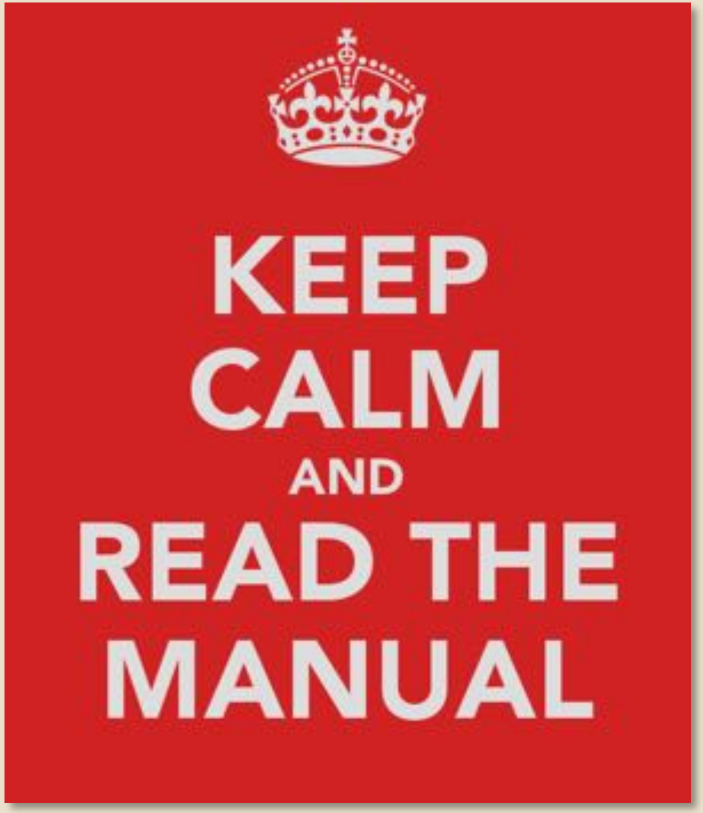
Ansible - Keeping It Simple

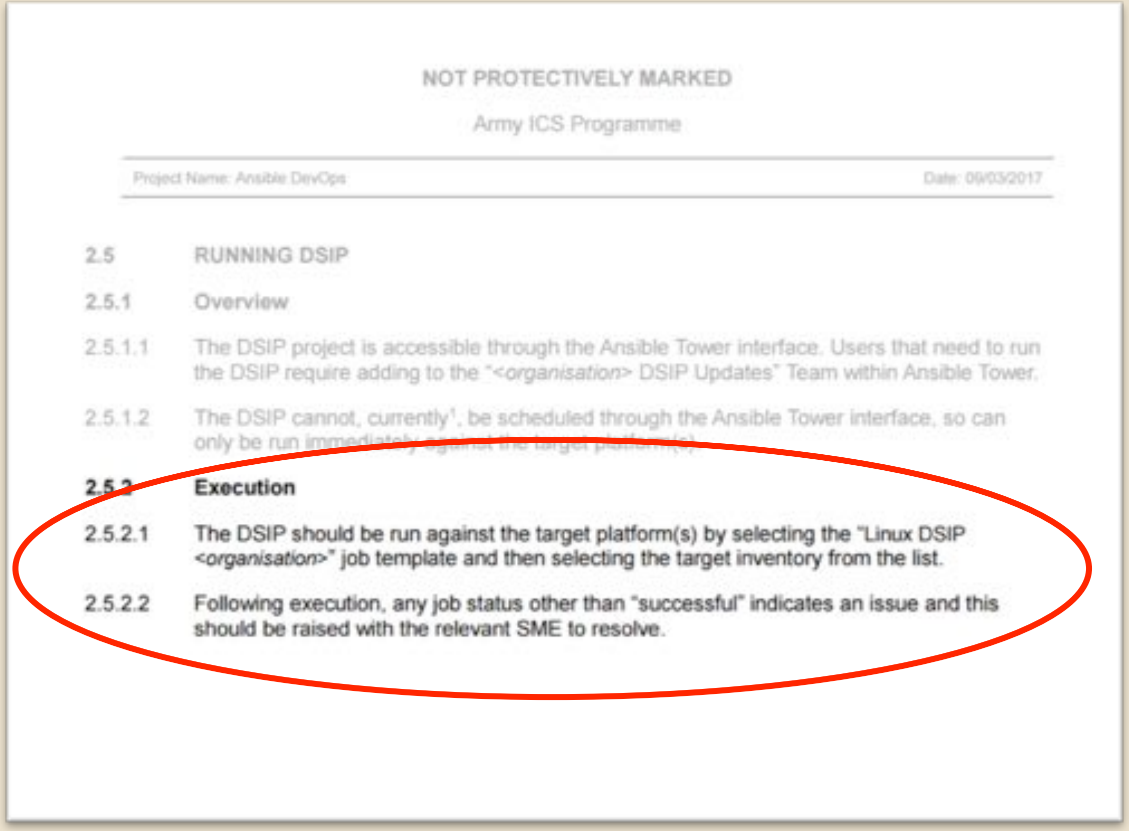
“KISS is a British Army mnemonic meaning Keep It Simple, Stupid. It doesn’t mean to say that soldiers are stupid, but suggests that under stress, when a plan contains ambiguities or is difficult to understand, it will lead to misunderstandings.”

Victor Newman, Made to Measure Problem Solving

Efficiency and Effectiveness Through DevOps

Ansible - Keeping It Simple





NOT PROTECTIVELY MARKED

Army ICS Programme

Project Name: Ansible DevOps Date: 09/03/2017

2.5 RUNNING DSIP

2.5.1 Overview

2.5.1.1 The DSIP project is accessible through the Ansible Tower interface. Users that need to run the DSIP require adding to the "*<organisation>* DSIP Updates" Team within Ansible Tower.

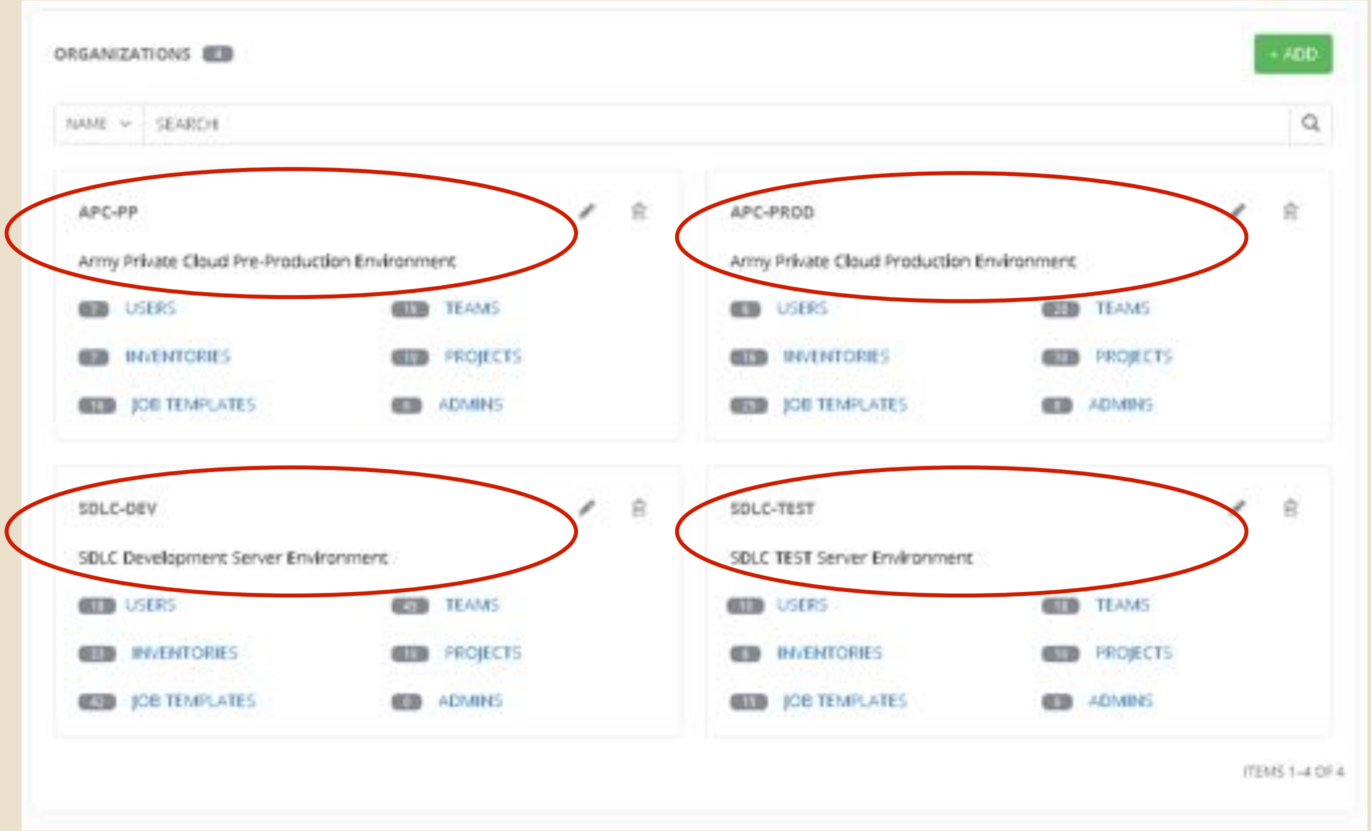
2.5.1.2 The DSIP cannot, currently[1], be scheduled through the Ansible Tower interface, so can only be run immediately against the target platform(s).

2.5.2 Execution

2.5.2.1 The DSIP should be run against the target platform(s) by selecting the "Linux DSIP *<organisation>*" job template and then selecting the target inventory from the list.

2.5.2.2 Following execution, any job status other than "successful" indicates an issue and this should be raised with the relevant SME to resolve.

Keeping It Simple - Access Control



ORGANIZATIONS 4

+ ADD

NAME SEARCH

APC-PP

Army Private Cloud Pre-Production Environment

7	USERS	15	TEAMS
7	INVENTORIES	19	PROJECTS
16	JOB TEMPLATES	8	ADMINS

APC-PROD

Army Private Cloud Production Environment

6	USERS	28	TEAMS
15	INVENTORIES	28	PROJECTS
25	JOB TEMPLATES	8	ADMINS

SDLC-DEV

SDLC Development Server Environment

18	USERS	45	TEAMS
33	INVENTORIES	18	PROJECTS
42	JOB TEMPLATES	0	ADMINS

SDLC-TEST

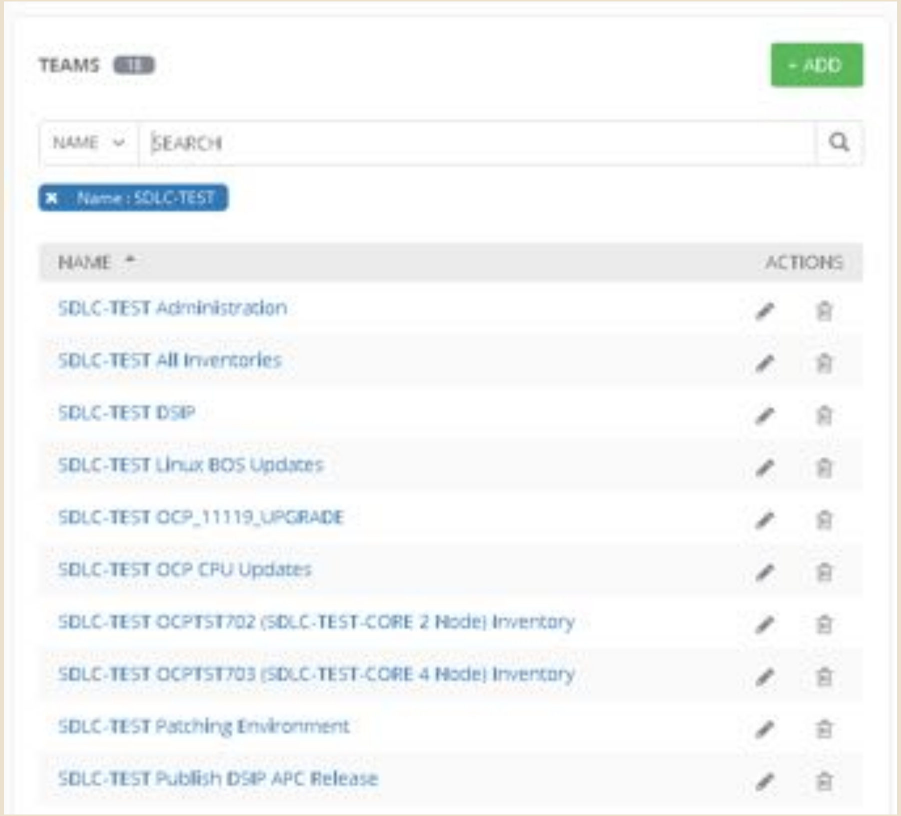
SDLC TEST Server Environment

10	USERS	18	TEAMS
6	INVENTORIES	19	PROJECTS
11	JOB TEMPLATES	8	ADMINS

ITEMS 1-4 OF 4

Efficiency and Effectiveness Through DevOps

Keeping It Simple - Access Control



TEAMS 11

+ ADD

NAME SEARCH

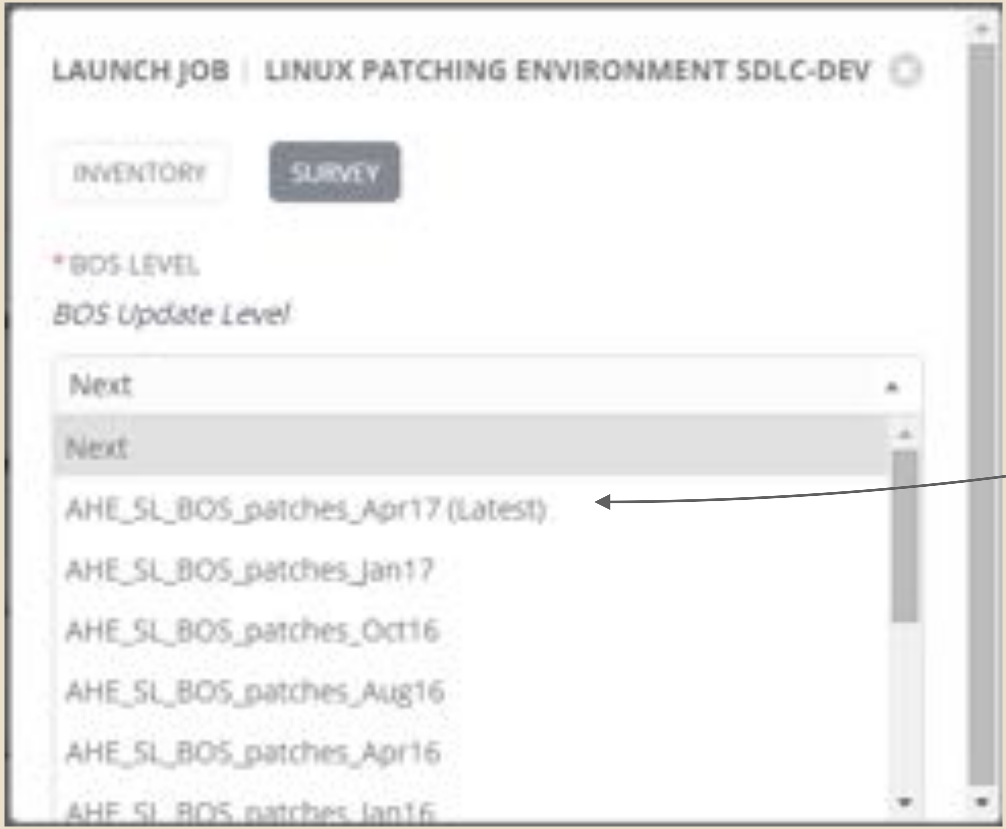
Name : SDLC-TEST

NAME	ACTIONS
SDLC-TEST Administration	
SDLC-TEST All Inventories	
SDLC-TEST DSIP	
SDLC-TEST Linux BOS Updates	
SDLC-TEST OCP_11119_UPGRADE	
SDLC-TEST OCP CPU Updates	
SDLC-TEST OCPTST702 (SDLC-TEST-CORE 2 Node) Inventory	
SDLC-TEST OCPTST703 (SDLC-TEST-CORE 4 Node) Inventory	
SDLC-TEST Patching Environment	
SDLC-TEST Publish DSIP APC Release	

Teams used for fine grained access control to jobs and inventories.

Teams get automatically created when we add new projects or inventories to an organisation.

Keeping It Simple - Surveys



LAUNCH JOB | LINUX PATCHING ENVIRONMENT SDLC-DEV

* BOS LEVEL

BOS Update Level

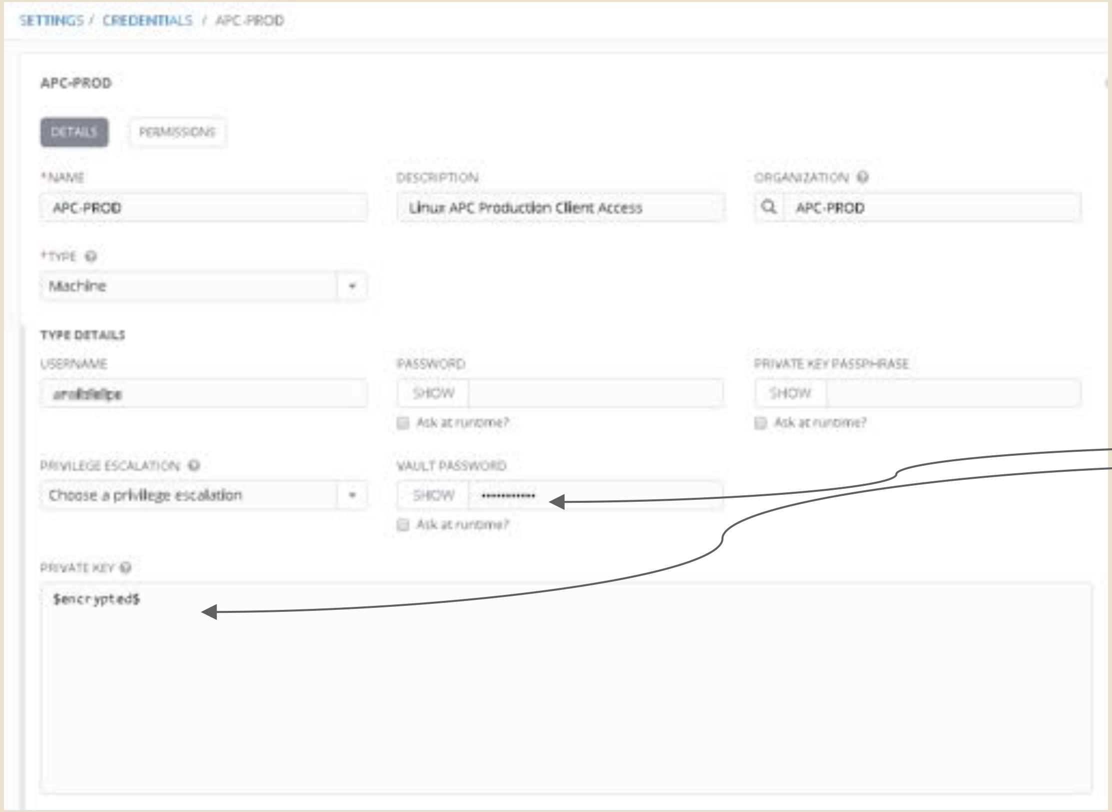
Next

- Next
- AHE_SL_BOS_patches_Apr17 (Latest)
- AHE_SL_BOS_patches_Jan17
- AHE_SL_BOS_patches_Oct16
- AHE_SL_BOS_patches_Aug16
- AHE_SL_BOS_patches_Apr16
- AHE_SL_BOS_patches_Jan16

Survey options are automatically regenerated using REST calls to Tower when changes are detected

Can you keep a secret?

Ansible Tower - Credentials


SETTINGS / CREDENTIALS / APC-PROD

APC-PROD

DETAILS | PERMISSIONS

*NAME: APC-PROD	DESCRIPTION: Linux APC Production Client Access	ORGANIZATION: APC-PROD
*TYPE: Machine		

TYPE DETAILS

USERNAME: [illegible]	PASSWORD: SHOW — Ask at runtime?	PRIVATE KEY PASSPHRASE: SHOW — Ask at runtime?
PRIVILEGE ESCALATION: Choose a privilege escalation	VAULT PASSWORD: SHOW •••••••••••• — Ask at runtime?	

PRIVATE KEY

```
$encrypted$
```

Vault password and private key are never visible in the UI.

Ansible – Custom Modules

Ansible – Custom Modules

Custom Python module makes REST calls to password server

```
- name: Get required passwords from Password Mgt Server
  ss_secret:
     secret_name="IDM_WLS_PASSWORD, BIEE_WLS_PASSWORD"
     platform="{{ansible_local.ahe.hlb_vhostname|upper}}"
     username="{{SS_USERNAME}}"
     password="{{SS_PASSWORD}}"
     domain="{{SS_DOMAIN}}"
  register: ss_result
  no_log: true
```

Returned variables can then be used in plays

```
vars:
  IDM_WLS_PASSWORD: "{{ss_result.IDM_WLS_PASSWORD}}"
  BIEE_WLS_PASSWORD: "{{ss_result.BIEE_WLS_PASSWORD}}"
```

Ansible Vault

Tower-cli / REST interface

Our Ansible Tower installation and configuration is

100%

Software Defined

**…Software Installation, Projects, Inventories,
Job Templates, Access Controls, Surveys, Credentials…**

Securing the Base OS

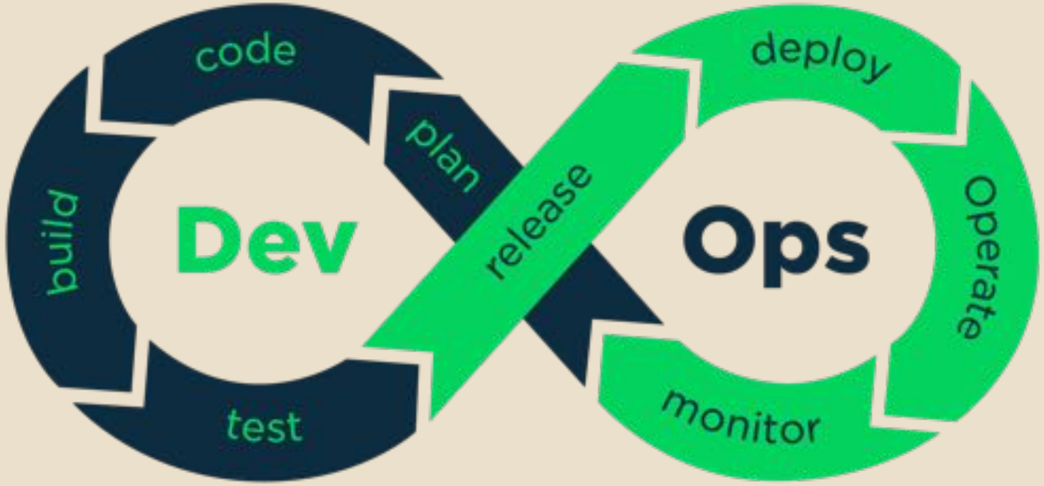
- Red Hat Common Criteria EAL4+ configuration
- cc-config-rhel71 package
- Kickstart install
- https://access.redhat.com/errata/RHEA-2016:2104

- DoD (DISA) STIGs
- https://iase.disa.mil/stigs/os/unix-linux/Pages/red-hat.aspx

- CIS Benchmarks
- https://www.cisecurity.org/cis-benchmarks/

- SCAP (https://scap.nist.gov/)
- Open-SCAP (https://www.open-scap.org/tools/scap-workbench/)

Q&A



Question and Answer Session