RED HAT SUMMIT

ANSIBLE ALL THE THINGS

From traditional to unorthodox, Ansible for Everything

Adam Miller
Principal Software Engineer
Red Hat Summit 2017

Nicolas FANJEAU
Airbus Infrastructure

AGENDA

AGENDA

WHAT WE'RE GOING TO TALK ABOUT TODAY

- Quick intro to Ansible (just in case)
- Why on earth would I want to do all the things with Ansible?
- Automation Tool
- Configuration Management
- Provisioning and Systems Management
- Deployment
- Application Lifecycle Management
- Orchestration
- Command Line Tooling
- Event Based Execution
- Workflow Automation
- CI/CD
- Ansible Container
- Ansible Tower
- Case Study: Airbus

WHAT IS ANSIBLE?

QUICK INTRODUCTION

WAIT, YOU DON'T KNOW WHAT ANSIBLE IS?

Ansible is an automation tool

- Ansible is a simple agentless idempotent **task automation tool**
    - By default, tasks are executed in-order but we can change that if we want.
- **Tasks** are performed via **modules**
- **Tasks** are grouped together via **plays**
    - Also via **roles**, but more on that later
    - A **play** operates on a set of hosts
- **Playbooks** can contain one or many **plays**
    - Can be used with "traditional" configuration management systems
        - There's even a puppet module!

QUICK INTRODUCTION

BEST THING SINCE SLICED BREAD

- Example of an ad-hoc ansible orchestration task
  - **Module:** yum
  - **Arguments:** `pkg=bash state=installed`

```
$ ansible localhost -m yum -a "pkg=bash state=installed"
localhost | SUCCESS => {
    "changed": false,
    "msg": "Nothing to do"
}
```

- What if I wanted to do more than one thing? **Playbooks**!

BUT FIRST... INVENTORY

INVENTORY

KEEPING TRACK OF YOUR MARBLES… ERR SYSTEMS

Inventory to defines hosts and groups of hosts

  - Special "all" group that is implicitly defined as the sum of all hosts in your inventory.
  - Also, “localhost” is a built-in and does not need to be defined

- Example:
  - Below we have a simple inventory with two groups, appservers and webservers.

```
[appservers]
app1.example.com
app1.example.com

[webservers]
webserver1.example.com
webserver2.example.com
```

PLAYBOOKS AND ROLES

PLAYBOOKS

DOING STUFF AND THINGS

Playbooks are a way to combine many tasks, written in YAML, to be carried out against one or many hosts.

```yaml
---
- name: common things to run on all hosts
  hosts: all
  tasks:
    - name: make sure bash is installed
      yum:
        pkg: bash
        state: installed
```

```yaml
- name: webserver-only tasks
  hosts: webservers
  tasks:
    - name: start and enable httpd service
      service:
        name: httpd
        state: started
        enabled: yes
```

INCLUDES

DON’T JUST COPY/PASTE … COWSAY IS WATCHING

Include file defines a set of **tasks** that can be **included** by a **playbook**, this allows sharing sets of tasks without copy/pasting everywhere.

```
enablewebservice.yml
---
- name: start and enable httpd
  service:
    name: httpd
    state: enabled
```

```
webserver.yml
---
- name: Webserver Playbook
  hosts: webservers
  tasks:
    - include: enablewebservice.yml
```

```
 ____________________________
< Don't copy/paste, include! >
 ----------------------------
        \   ^__^
         \  (oo)\_______
            (__)\       )\/\
                ||----w |
                ||     ||
```

Playbooks can also include other playbooks!

ROLES

YOUR MOM WAS RIGHT, IT'S BETTER TO SHARE

Roles are reusable logical groupings of tasks that (normally) define a service

- Role-level subdirs for namespaced variable defaults, files, templates, and handlers
- Can pass variables to roles to modify behavior per-use
- Searched for and/or shared via Ansible Galaxy
    - https://galaxy.ansible.com/

```
---
- name: using myrole
- hosts: webservers
- roles:
  - myrole
```

Typical Role Layout

```
myrole/
├── defaults
│   └── main.yml
├── files
├── handlers
│   └── main.yml
├── tasks
│   └── main.yml
├── templates
└── vars
    └── main.yml
```

WHAT IS ANSIBLE?

USING ANSIBLE FOR EVERYTHING

WHY WOULD I WANT TO DO THAT?

Ansible is a simple automation tool that can:

- Execute tasks on one or many hosts
- Orchestrate an otherwise complex order of operations, even conditionally based on system facts or variables provided at runtime.
- Custom modules can be written in any programming language with JSON support

Question of the day:

What are you trying to accomplish that could be automated?

USING ANSIBLE FOR EVERYTHING

ANSIBLE ALL THE THINGS!!!!

What are you trying to do?

- Configuration Management?
- Provision Virtual Machines or IaaS instances?
- Test software?
- Automate workflows?
- Continuous Integration / Continuous Deployment?
- Configure hardware switches, routers, and load balancers?
- Replace terrible shell scripts that have survived too long already?
- Other?

ANSIBLE CAN DO ALL OF THAT! (AND MUCH MORE)

ANSIBLE DOES THAT

CONFIGURATION MANAGEMENT

KEEPING THE TRAIN ON THE TRACKS

What is configuration management?

Systems engineering process for establishing and maintaining consistency of a product's performance, functional, and physical attributes with its requirements, design, and operational information throughout its life.

Generally boils down to:

- Managing file content
- Configuration Templating
- System and Service state
- Package Management
- Lifecycle Management

ANSIBLE DOES THAT

OMG, NO WAY?!?!?!

- Service state: `service` module
- Files and configuration modules: `acl archive assemble blockinfile copy fetch file find ini_file iso_extract lineinfile patch replace stat synchronize tempfile template unarchive xattr`
- System state modules: `aix_inittab alternatives at authorized_key beadm capabilities cron cronvar crypttab debconf facter filesystem firewalld gconftool2 getent gluster_volume group hostname iptables java_cert kernel_blacklist known_hosts locale_gen lvg lvol make modprobe mount ohai open_iscsi openwrt_init osx_defaults pam_limits pamd parted ping puppet runit seboolean sefcontext selinux selinux_permissive seport service setup solaris_zone svc sysctl systemd timezone ufw user`
- Package Management modules: `bower bundler composer cpanm easy_install gem maven_artifact npm pear pip apk apt apt_key apt_repository apt_rpm dnf dpkg_selections homebrew homebrew_cask homebrew_tap layman macports openbsd_pkg opkg package pacman pkg5 pkg5_publisher pkgin pkgng pkgutil portage portinstall pulp_repo redhat_subscription rhn_channel rhn_register rpm_key slackpkg sorcery svr4pkg swdepot swupd urpmi xbps yum yum_repository zypper zypper_repository`

More modules being added all the time...

ADVANCED CONFIGURATION MANAGEMENT

THAT LITTLE EXTRA

The following categories of Infrastructure Needs are covered extensively by Ansible modules:

- Clustering
- Commands
- Crypto
- Database
- Files
- Identity
- Inventory
- Messaging
- Monitoring
- Network
- Notification
- Packaging
- Remote Management
- Source Control
- Storage
- System
- Utilities
- Web Infrastructure

PROVISIONING

MAKING SOMETHING FROM NOTHING

What do you want to accomplish?

- Create IaaS compute instances, object stores, or ephemeral resources?
- Provision virtual machines?
- Create storage allocations?
- Set firewall rules?
- Configure highly available load balancers?
- Create VLANs?
- Deploy container orchestration resources?
- Create databases?
- Other?

ANSIBLE CAN DO THAT

WHAT? AGAIN? NO WAY!!

Provisioning support for many IaaS providers:

- Amazon Web Services
- Apache CloudStack
- Centurylink Cloud
- Digital Ocean
- DimensionData
- Google Cloud
- Linode
- Microsoft Azure
- OpenStack
- Rackspace Public Cloud
- Softlayer Webfaction

Datacenter and Virtualization:

- oVirt / RHV
- libvirt resource management
- Joyent SmartOS Virt
- VMWare (VSphere/ESXi)

Storage:

- AIX LVM
- Gluster Volume
- Infinidat
- LVM2
- NetApp
- ZFS

PROVISIONING - CONTINUED

OMG, THIS LIST JUST KEEPS GOING...

Networking

- A10 Networks
- Apstra AOS
- Arista EOS
- Avi Networks
- BigSwitch
- Cisco (ASA, IOS/IOS-XR, and NX-OS)
- Cumulus Networks (Cumulus Linux)
- Dell EMC (OS6, OS9, and OS10)
- F5 BigIP
- Fortios Firewall
- JunOS
- Lenovo CNOS
- Netvisor
- Open vSwitch
- Palo Alto Networks PAN-OS
- Nokia SR OS
- VyOS

Databases

- InfluxDB
- Redis
- Riak
- MS-SQL
- MySQL
- Postgresql
- Vertica

PROVISIONING - CONTINUED

SERIOUSLY? MORE STUFF?

Web Infrastructure and Clustering

- Apache HTTPD (module and mod_proxy management)
- Consul
- Django Management
- eJabberd
- htpasswd
- JBoss
- Jenkins (Jobs, Plugin, and Jenkinsfile management)
- Jira
- Kubernetes
- Letsencrypt
- Pacemaker
- Supervisord
- ZooKeeper

DOING THINGS WITH ANSIBLE

DEPLOYMENT

I JUST GIT PUSH TO THE CLOUD, RIGHT?

Software Deployment is the act of making software available on systems; most often, this is a sequence of steps that must be performed in-order. (**In-order task execution anyone**?)

Example:

- Sync some data
- Database schema migration
- Remove systems from load balancer
- Push new code
- Put systems back in load balancer
  - Rinse/Repeat on previously not upgraded set
- Verify services are functional
- Status update

Remember what a Playbook does?

APPLICATION LIFECYCLE MANAGEMENT

DO IT LIVE!

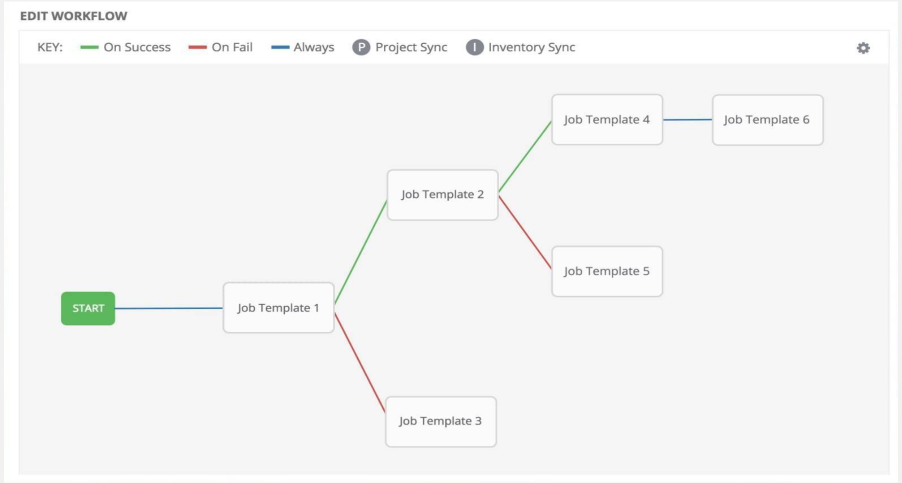
Managing application lifecycle across one or many hosts

- Ansible can orchestrate both simple and complex lifecycle management
- Lifecycle “order of operations” defined in Playbooks
    - Whatever your requirements are
- Plays can execute on different sets of hosts
    - Multiple plays per playbook
- Plays can use varying execution strategies for various requirements
    - Cluster node management
    - Database schema updates
    - etc
- Sky is the limit
    - (something something … cloud)

ORCHESTRATION AND WORKFLOW

AUTOMATION WITH FEELING

Flow controlled automation by data from the environment allowing the automation tasks to make “intelligent” decisions.



COMMAND LINE TOOLING

BUT WHAT ABOUT MY PERL ONE-LINERS?

Make Ansible your new command line tooling API, stop re-inventing the wheel

- Ansible provides a very capable Python API for modules
- Modules can be written in any programming language that understands JSON
- Provides a consistent “UX” for all tasks
- Gives you and your ops team an “on ramp” to scaling your tasks across the infrastructure

```
$ ansible localhost -m my_task -a “arg1=foo arg2=bar”
```

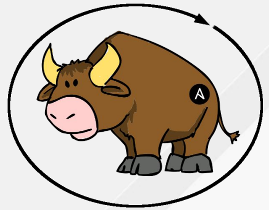
EVENT BASED EXECUTION

COWSAY WHAT?

Ansible can easily integrate with existing infrastructure to perform actions based on events.

- Example: `loopabull`
  - Events in the infrastructure spawn messages on the bus
  - `loopabull` listens on the bus, waiting for a “routing key” that it cares about (message topic)
  - Message payload is injected into Ansible playbooks as variables, allowing for decisions to be made based on message contents

```
+-----------------+       +---------------+
|                 |       |               |
|     Events      +------>|  Looper       |
|                 |       |   (plugin)    |
|                 |       |               |
+-----------------+       +---------------+
                                     |
          +-------------------+      |
          |                   |      |
          |                   |      |
          |     Loopabull     +<---+
          |   (Event Loop)    |
          |                   |
          +---------+---------+
                    |
                    V
        +-----------+----------+
        |                      |
        |   ansible-playbook   |
        |                      |
        +----------------------+
```



CONTINUOUS INTEGRATION

THERE IS ONLY ZUUL ... (BUT ALSO OTHER STUFF)

Brief story of OpenStack Zuul and Jenkins Job Builder

- OpenStack CI System (Zuul) - http://status.openstack.org/zuul/
  - 2,000+ jobs-per-hour
    - single-use OpenStack VMs -> create and destroy 2K+ VMs per hour
  - 1731 git repositories to perform gating on
  - Spread across 7 public OpenStack clouds and 4 private OpenStack clouds
    - Hybrid cloud anyone?
- OpenStack wanted to not fiddle with XML for Jenkins Jobs
- Jenkins Job Builder (YAML) was created
- Jenkins Performance issues ran into…
- No more Jenkins, automatically convert JJB YAML into Ansible Playbooks
- Future: Migrate entirely away from JJB, make it all Ansible!

MORE CONTINUOUS INTEGRATION

THE OTHER STUFF

Fedora Taskotron - https://taskotron.fedoraproject.org/

- CI for the entire Fedora Linux Distribution
- “Tasks” definitions originally in YAML
- Tasks for every RPM, ISO, VM Image, Container, etc in the distro
- Automated reporting to the Fedora Updates System (Bodhi)
- Migration from Taskotron YAML to Ansible Playbooks

ANSIBLE CONTAINER

END THE DOCKERFILE MADNESS

Using Ansible playbooks to build you container images

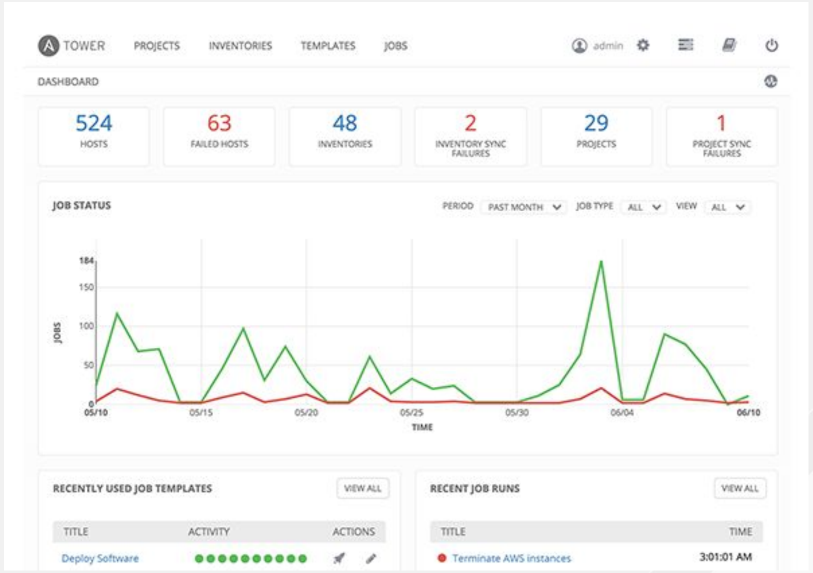
- Stop chaining together shell commands in Dockerfiles
- Create containers the same way you deploy to servers
- **roles** == services, build your containers using **roles**
  - Making single-purpose (microservice) containers easy
- Deploy to Container Orchestration Platforms
  - Currently Supports OpenShift and Kubernetes

ANSIBLE TOWER

PRETTY GRAPHS!

The definitive Ansible Centralized Management Portal

- Role Based Access Control
- Centralized Logging, History Visualizations
- Multi-Playbook Workflow Orchestration
- Playbook and System Auditing (System Tracking)
- Self-Service Automation
    - Sanitized form-based playbook runs
- Integrated Notifications (ChatOps, etc)
- REST API
- ... and much much more!



524	63	48	2	29	1
HOSTS	FAILED HOSTS	INVENTORIES	INVENTORY SYNC FAILURES	PROJECTS	PROJECT SYNC FAILURES

JOB STATUS

PERIOD: PAST MONTH · JOB TYPE: ALL · VIEW: ALL

RECENTLY USED JOB TEMPLATES — VIEW ALL

TITLE	ACTIVITY	ACTIONS
Deploy Software	●●●●●●●●●●	

RECENT JOB RUNS — VIEW ALL

TITLE	TIME
● Terminate AWS instances	3:01:01 AM

RED HAT SUMMIT

ANSIBLE @ Airbus

Automation from End2End

Nicolas FANJEAU
Airbus Infrastructure
Red Hat Summit 2017

redhat.

Airbus

55,000
Employees

€45,8billion
Annual revenue*

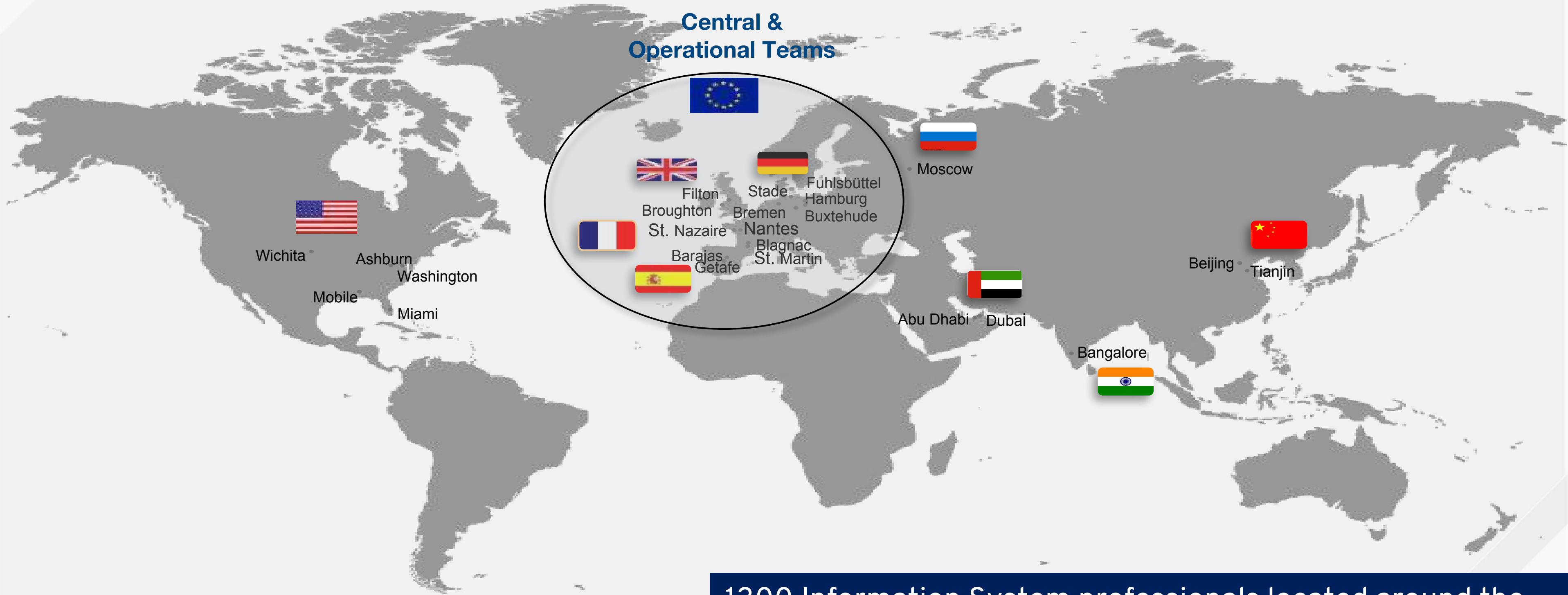
10yrs
Backlog

400
Operators

Passion

Our global workforce is united by a passion for aviation and restless desire to create better ways to fly

Information & Communication Technology



1300 Information System professionals located around the world wherever Airbus operates.

Airbus IT Infrastructure

Suppliers

- 106 000 users
- 21 000 PCs
- 33 000 mailboxes

Airbus

- 96 000 users
- 61 000 PCs
- 77 000 mailboxes
- 6 600 printers
- 75 000 fixed phones
- 32 400 mobile phones

Airbus Group

- 94 000 users
- 5 000 PCs
- 34 000 mailboxes
- 600 mobile phones

Customers

- 72 000 users

TOTAL

- 368 000
- 87 000
- 144 000
- 6 600
- 75 000
- 33 000

433 000 network ports

5 000 WiFi access points

13 000 Servers

17 petabytes on storage

19 billions transactions per year on SAP

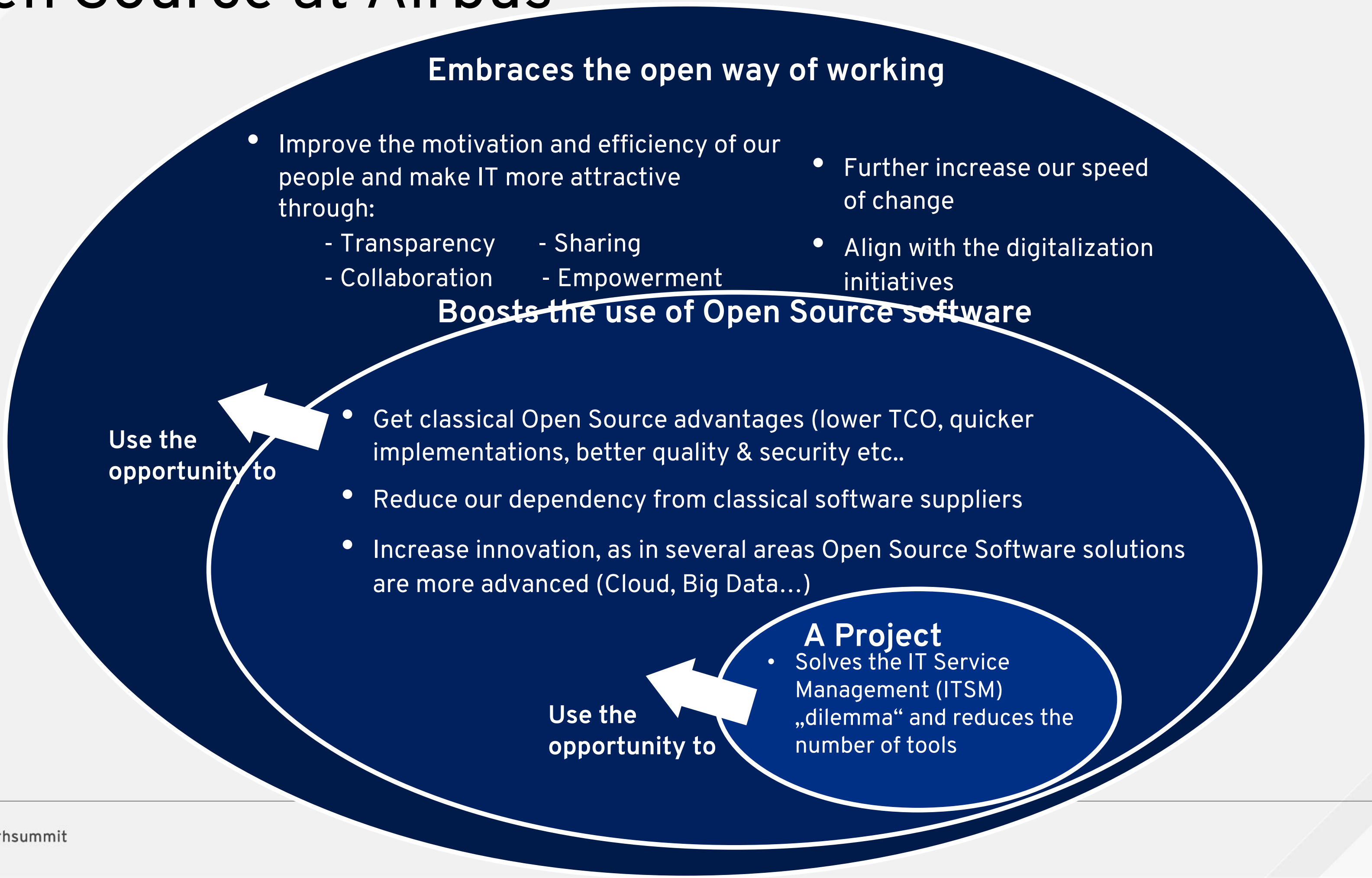
1,2 petaFLOPS on High performance computing

4 200 MIPS on Mainframe

Data to end 2015

redhat.

Open Source at Airbus



Embraces the open way of working

- Improve the motivation and efficiency of our people and make IT more attractive through:
  - Transparency
  - Sharing
  - Collaboration
  - Empowerment
- Further increase our speed of change
- Align with the digitalization initiatives

Use the opportunity to

Boosts the use of Open Source software

- Get classical Open Source advantages (lower TCO, quicker implementations, better quality & security etc..
- Reduce our dependency from classical software suppliers
- Increase innovation, as in several areas Open Source Software solutions are more advanced (Cloud, Big Data…)

Use the opportunity to

A Project

- Solves the IT Service Management (ITSM) „dilemma“ and reduces the number of tools

Our needs

Functional	Solution
Application	Maintenance
Infrastructure	Robust
Entry in Service	Library
Maintenance	Integrated
Job Scheduler	Scalable
Public Cloud	Deployment
Linux	High Availability
Private Cloud	Secure
Windows	Interoperable
Plug and play	Cost
Reporting	Segregation
	Agent less

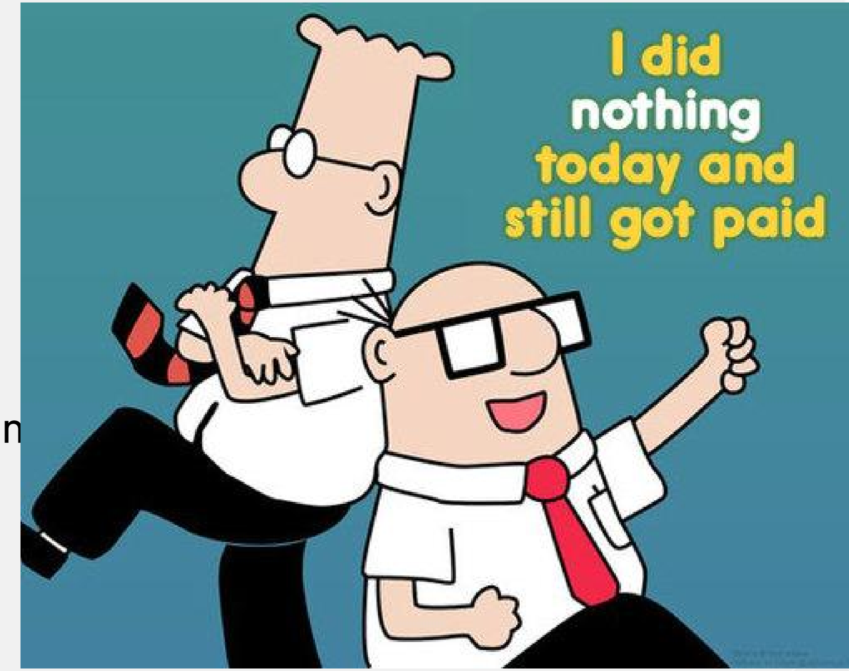
Automation as Self Service

EXPECTATIONS

- Reduce time and cost to deploy application
- Move to DevOps philosophy
- Give back the responsibility to Application Owner
- Simplify process

SOLUTION

- Propose customer oriented service for Automation
- Develop the service for and with the customers
- Propose tailored solutions to all customers via a catalogue of services
  - Awareness on Automation
  - Training : Platform usage, How to implement Playbook
  - Playbook On Demand, conversion of Install Manual to Playbook
  - eLearning, User Manual, Best practices



From the PoC to the Project

PoC

- Objective is to evaluate the solution
- Test the deployment of 5 applications (Win & Linux) with 6 automation solutions

Result

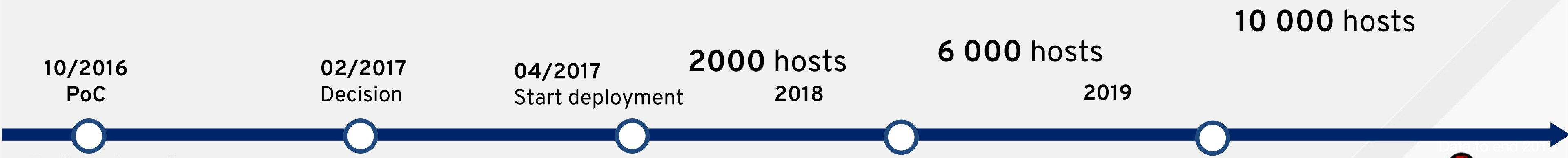
- Despite missing functionalities of Tower vs Competitors, Tower finish first one based on the criteria matrix
- Deployed in Production during the PoC for two critical applications for
  - Release deployment
  - Job scheduler

Key Figures

PoC on 100 Hosts

Target 10 000 Hosts

First deployment 6 months



Target

- **Hosts**
  - Windows 9 400   W2k8   4100 W2k12 3500
  - Linux   5 900 RHEL5 1800   RHEL6 2500   RHEL7 70
  - Unix   3 600
- **Deployed of dedicated Tower infrastructure depending of**
  - Location of the Data Center

    Germany, France, United Kingdom & Spain…
  - Environment

    Integration, Validation, Production, DMZ, Public Cloud…
- **Common architecture base on**
  - Tower, Cluster of two nodes
  - PostgreSQL, Cluster of two nodes
  - Virtual Machine, RHEL 7

Key Figures

Applications   2000

Users   1 000

Deployment Infra 2 months

Next, Automation from End to End

Full automation from the request to the delivery

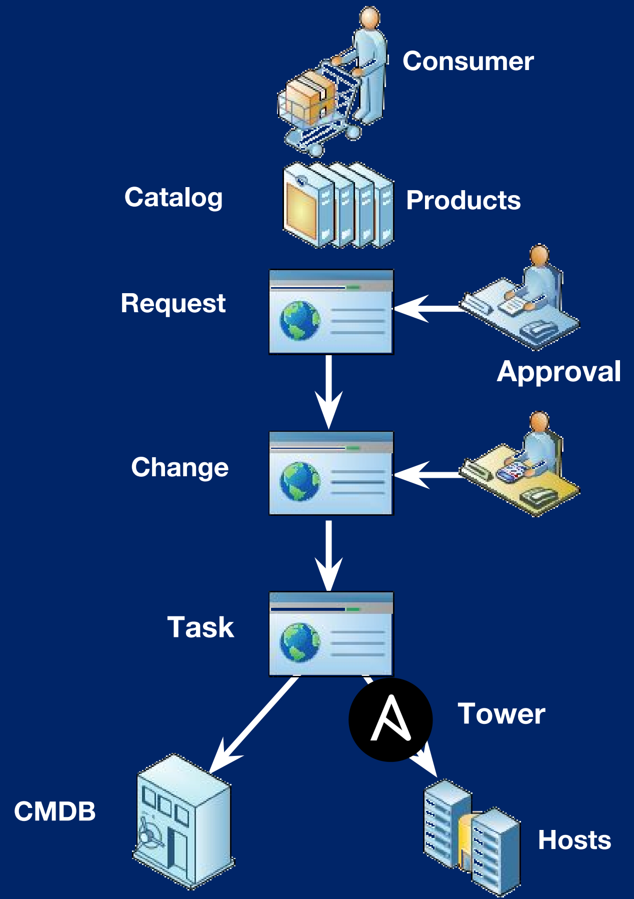
Be user centric and inforce self service usage

Propose a single catalogue and point to aggregate all the products

Use the Tower CLI

Fully integrated with ITSM tool to avoid to data duplication and interfaces

In line with the ITIL best practices



Key Success Factors

Open Source is a key solution to ensure innovative application and quick delivery

Involvement of customers in the development of the solution is a key of the success

A lot of communication & change support to get users adopt the situation

Self-service is the requirement to reach customers' satisfaction and meet company's objectives

RED HAT SUMMIT

THANK YOU

plus.google.com/+RedHat

facebook.com/redhatinc

linkedin.com/company/red-hat

twitter.com/RedHatNews

youtube.com/user/RedHatVideos

ADAM MILLER

maxamillion

maxamillion

@TheMaxamillion

RED HAT
SUMMIT

LEARN. NETWORK.
EXPERIENCE
OPEN SOURCE.