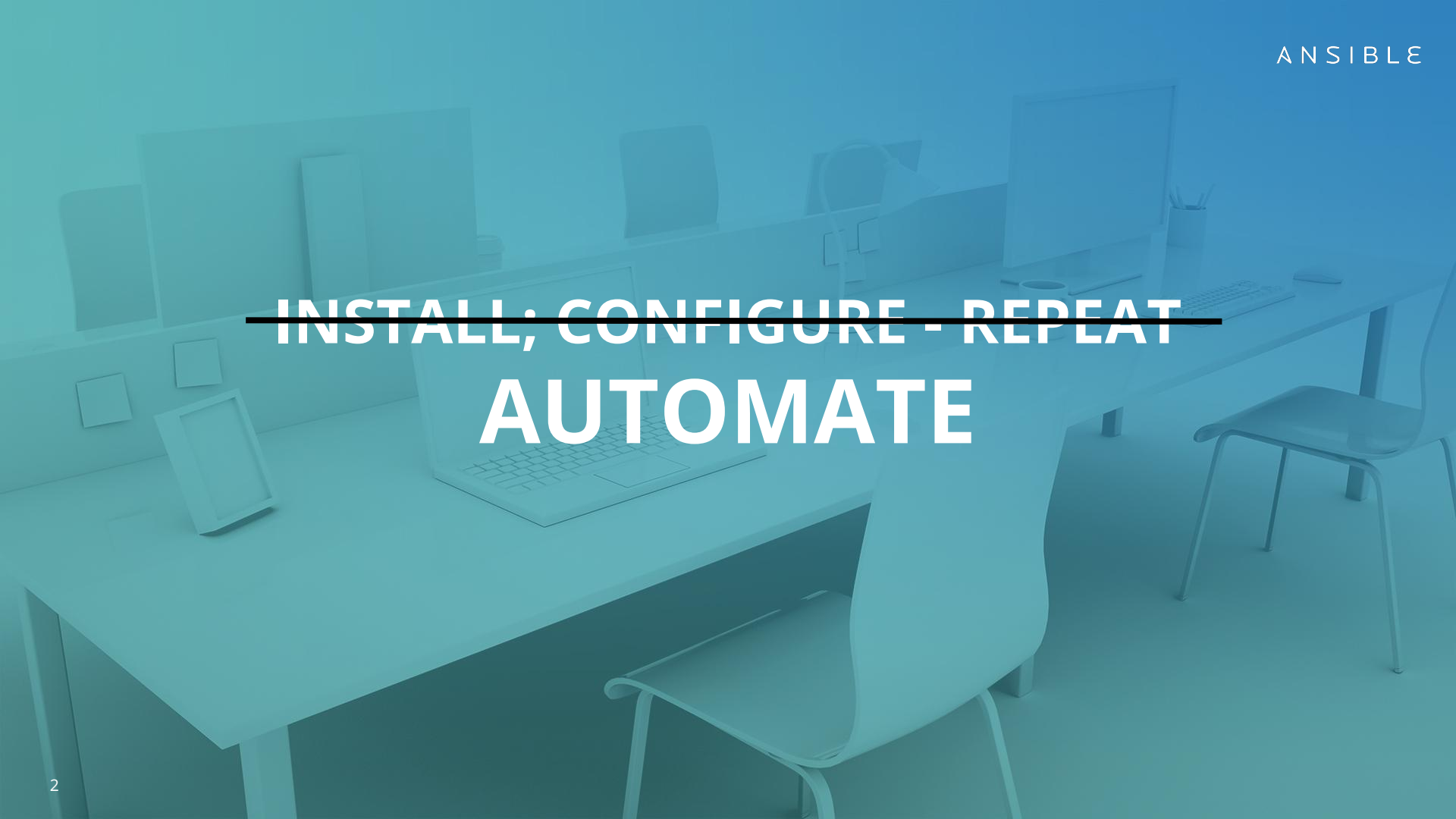
redhat.

Introduction to Ansible Engine and Ansible Tower

Markus Koch

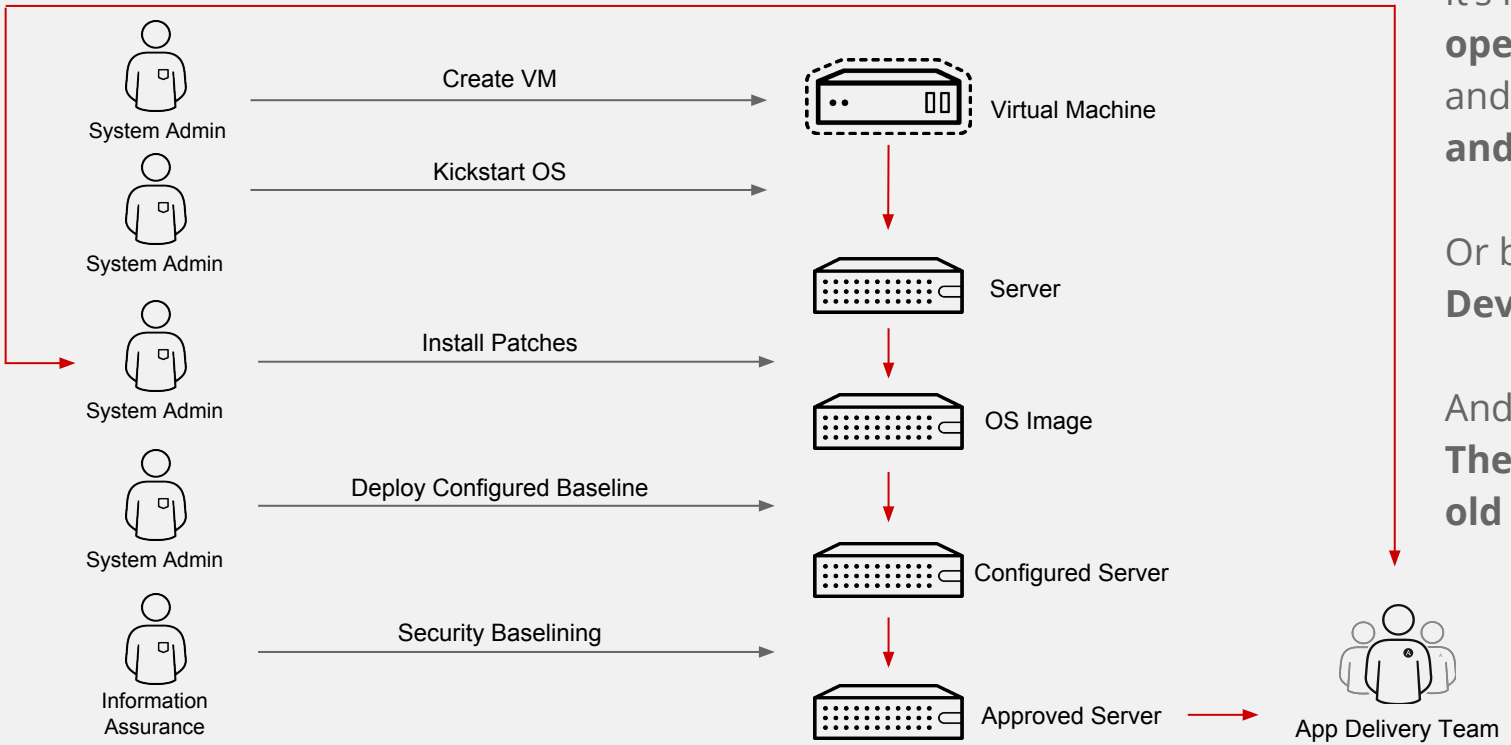
Partner Enablement Manager SAP



~~INSTALL; CONFIGURE - REPEAT~~

AUTOMATE

THE GOOD OLD DAYS…

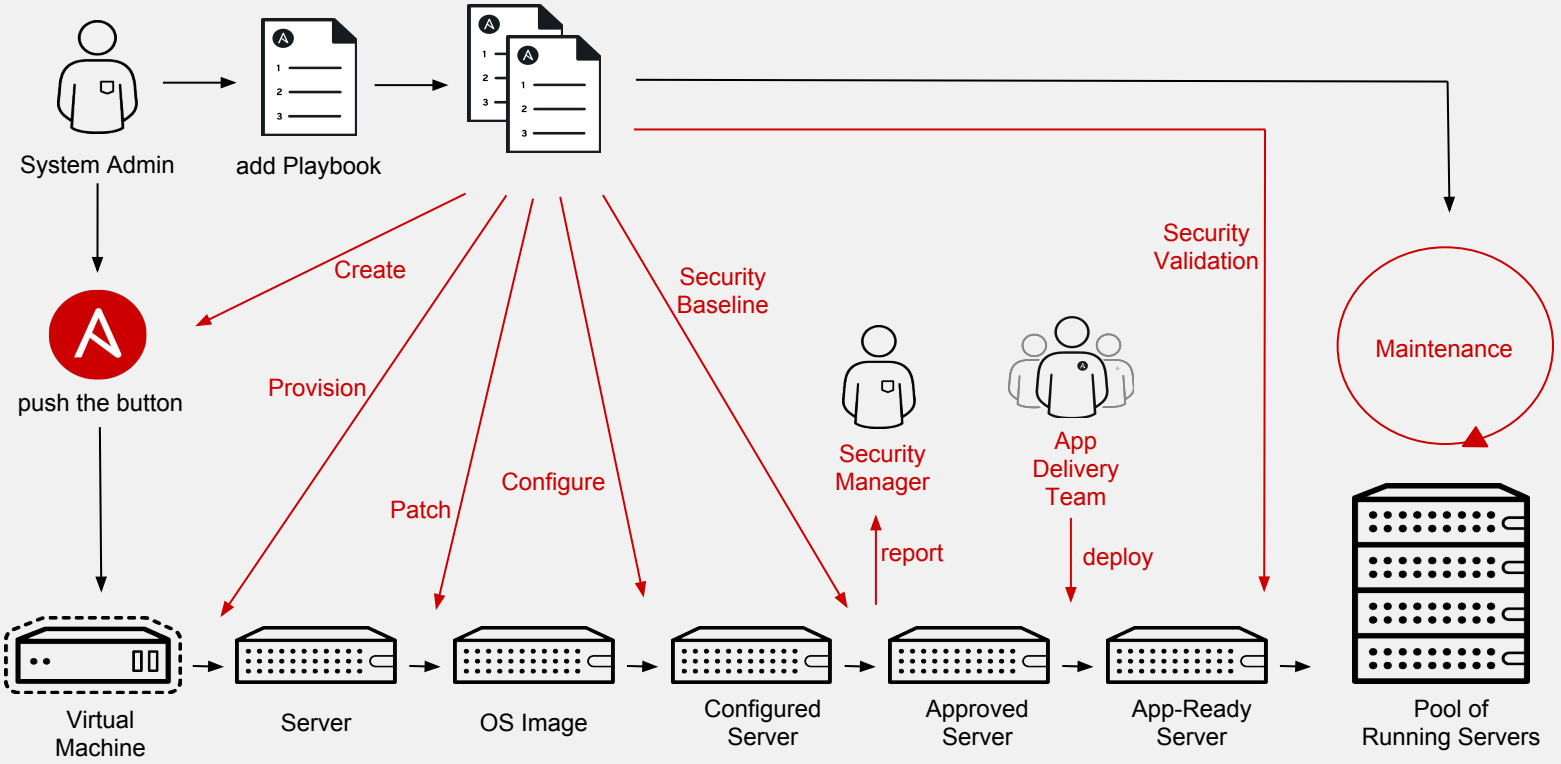


It's hard to **run IT operations this way** and **deliver flexible and agile IT** services.

Or being the **Ops** in **DevOps**.

And yes, you can work **The Cloud** in the **same old way**…

NOTHING ROUTINE SHOULD BE DONE MANUALLY



WHAT IS ANSIBLE AUTOMATION?

The Ansible project is an open source community sponsored by Red Hat. It's also a **simple automation language** that perfectly describes IT application environments in **Ansible Playbooks**.

Ansible Engine is a **supported product** built from the Ansible community project.

```
---
- name: install and start apache
  hosts: web
  become: yes
  vars:
    http_port: 80

  tasks:
  - name: httpd package is present
    yum:
      name: httpd
      state: latest

  - name: latest index.html file is present
    copy:
      src: files/index.html
      dest: /var/www/html/

  - name: httpd is started
    service:
      name: httpd
          state: started
```


ANSIBLE
v1 - Set config file to use on boot
1. Write multiple configuration files
• For each environment/region
2. Inspect metadata on boot and use the matching config file
v1 - Set config file to use on boot
1. Write multiple configuration files
• For each environment/region
2. Inspect metadata on boot and use the matching config file
24,000+
Stars on GitHub
950+
Ansible modules
400,000+
Downloads a month

WHY ANSIBLE?

ANSIBLE

SIMPLE

Human readable automation

No special coding skills needed

Tasks executed in order

Usable by every team

Get productive quickly

POWERFUL

App deployment

Configuration management

Workflow orchestration

Network automation

Orchestrate the app lifecycle

AGENTLESS

Agentless architecture

Uses OpenSSH & WinRM

No agents to exploit or update

Get started immediately

More efficient & more secure

THE ANSIBLE WAY

CROSS PLATFORM

Agentless support for all major OS variants, physical, virtual, cloud and network devices.

HUMAN READABLE

Perfectly describe and document every aspect of your application environment.

PERFECT DESCRIPTION OF APPLICATION

Every change can be made by Playbooks, ensuring everyone is on the same page.

VERSION CONTROLLED

Playbooks are plain-text. Treat them like code in your existing version control.

DYNAMIC INVENTORIES

Capture all the servers 100% of the time, regardless of infrastructure, location, etc.

ORCHESTRATION PLAYS WELL WITH OTHERS

Every change can be made by Playbooks, ensuring everyone is on the same page.

WHAT CAN I DO WITH ANSIBLE?

Automate the deployment and management of your entire IT footprint.

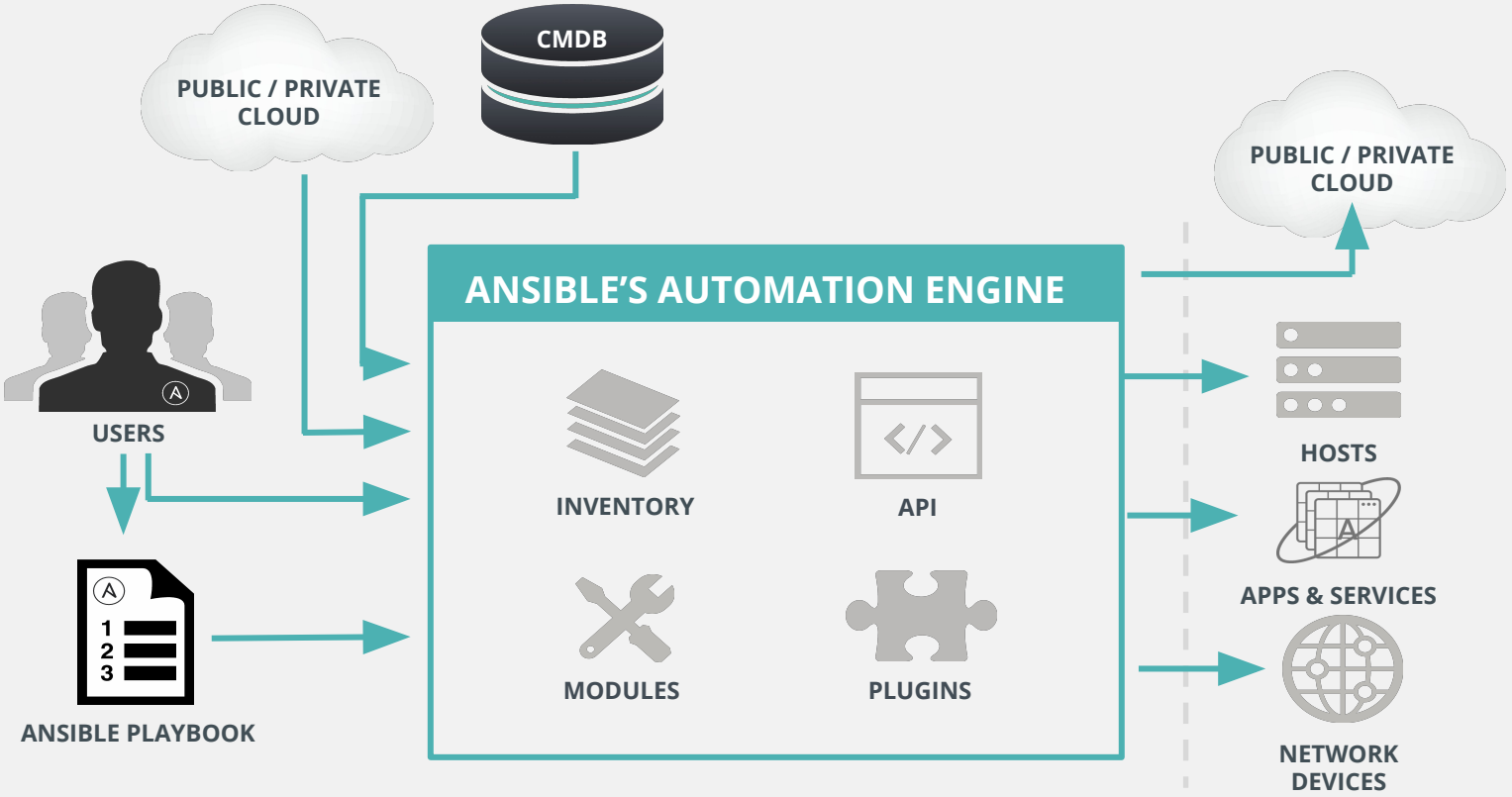
Do this...

Configuration Management	Orchestration	Application Deployment	Provisioning	Continuous Delivery	Security and Compliance

On these...

Firewalls	Load Balancers	Applications	Containers	Clouds
Servers	Infrastructure	Storage	Network Devices	**And more...**

HOW ANSIBLE WORKS



PLAYBOOK EXAMPLE: INSTALL & CONFIGURE APACHE

```
---
- name: install and start apache
  hosts: all
  vars:
    http_port: 80
    max_clients: 200
  become_user: root
  tasks:
  - name: install httpd
    yum: pkg=httpd state=latest
  - name: write the apache config file
    template: src=/srv/httpd.j2 dest=/etc/httpd.conf
  - name: start httpd
    service: name=httpd state=running
```

PLAYBOOK EXAMPLE: AWS CLOUD DEPLOYMENT

```
- hosts: localhost
  connection: local
  gather_facts: False
  tasks:
    - name: Provision a set of instances
      ec2:
         key_name: my_key
         group: test
         instance_type: t2.micro
         image: "{{ ami_id }}"
         wait: true
         exact_count: 5
         count_tag:
            Name: Demo
         instance_tags:
            Name: Demo
      register: ec2

    - name: Add all instance public IPs to dynamic host group
      add_host: hostname={{ item.public_ip }} groups=ec2hosts
      with_items: "{{ ec2.instances }}"
```

PLAYBOOK EXAMPLE: AWS CLOUD DEPLOYMENT

```
- hosts: ec2hosts
  name: configuration play
  user: ec2-user
  gather_facts: true
  tasks:
     - name: Check NTP service
       service:
          name: ntpd
          state: started
```

SAP HANA DEPLOYMENT WITH ANSIBLE

```
yum install @base xfsprogs libaio net-tools bind-utils gtk2 libicu xulrunner tcsh
sudo libssh2 expect cairo graphviz iptraf-ng krb5-workstation krb5-libs libpng12
ntp ntpdate nfs-utils lm_sensors rsyslog openssl098e openssl
PackageKit-gtk3-module libcanberra-gtk2 libtool-ltdl xorg-x11-xauth numactl
```

```
- name: install required packages
    yum: state=latest name={{ item }}
    with_items:
      - chrony
      - xfsprogs
      - libaio
      - net-tools
      - bind-utils
          ...
      - numactl
      - tuned-profiles-sap-hana
```

SAP HANA DEPLOYMENT WITH ANSIBLE

```
systemctl stop numad
systemctl disable numad
systemctl status numad
```

```
- name: disable numad
  service: name=numad state=stopped enabled=no
```

SAP HANA DEPLOYMENT WITH ANSIBLE

```
setenforce 0
sed -i 's/SELINUX=enforcing/SELINUX=permissive/'  /etc/selinux/config
sestatus
```

```
- name: disable selinux
  selinux: state=disabled
```