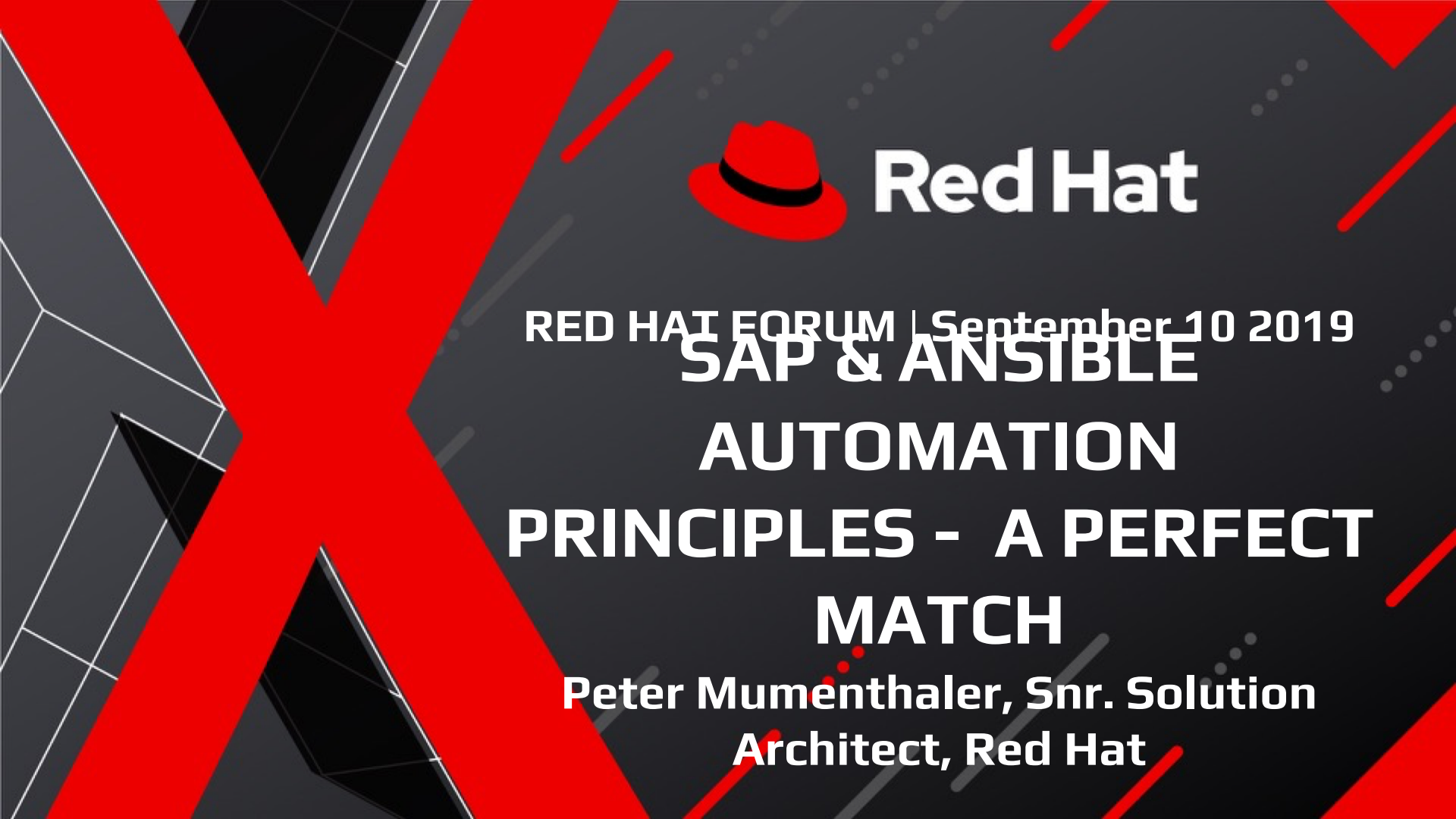


RED HAT FORUM | September 10 2019

SAP & ANSIBLE AUTOMATION PRINCIPLES - A PERFECT MATCH

Peter Mumenthaler, Snr. Solution Architect, Red Hat

Red Hat Enterprise Linux

Everything not automated slows you down

This applies as well to SAP

Peter Mumenthaler
Senior Solution Architect

Red Hat

What we'll be discussing today

- Automation Principles

  Why and how
- Red Hat SAP Offering
- Ansible Automation
- Ansible Automation for SAP
- Ansible Playbook Example

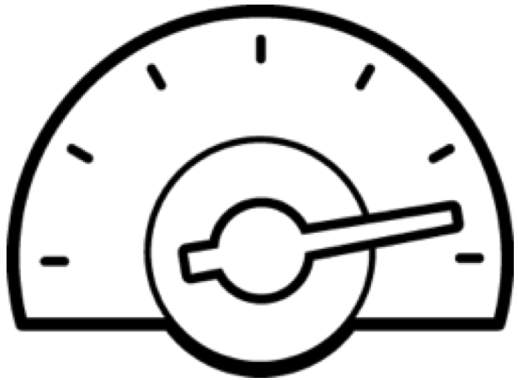
Automation Principles - Why and how

Automation with Version Control

- Auditable
- Changes are traceable
- Compliance aware
- Reporting / Logging

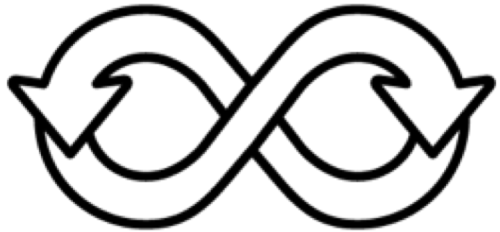
Automation leads to speed

- Reproducible / repeatable
- Standard(s) procedures
- Easy to integrate into changemangement process
- Fast and close to production - DevOps approach



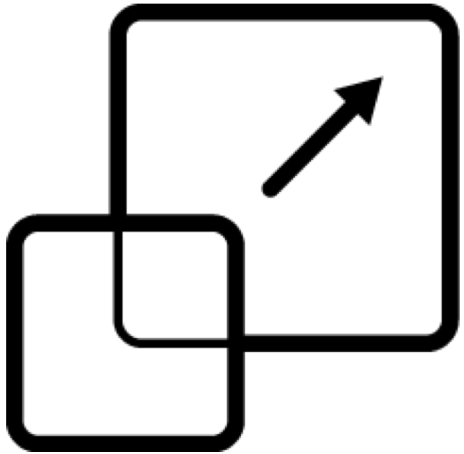
Automation leads to stability

- Less errors because standardizations and unit tests
- Reproducible / repeatable
- Documented
- Always up2date
- You know the state of your IT
- State is always enforced



Automation leads to scalability

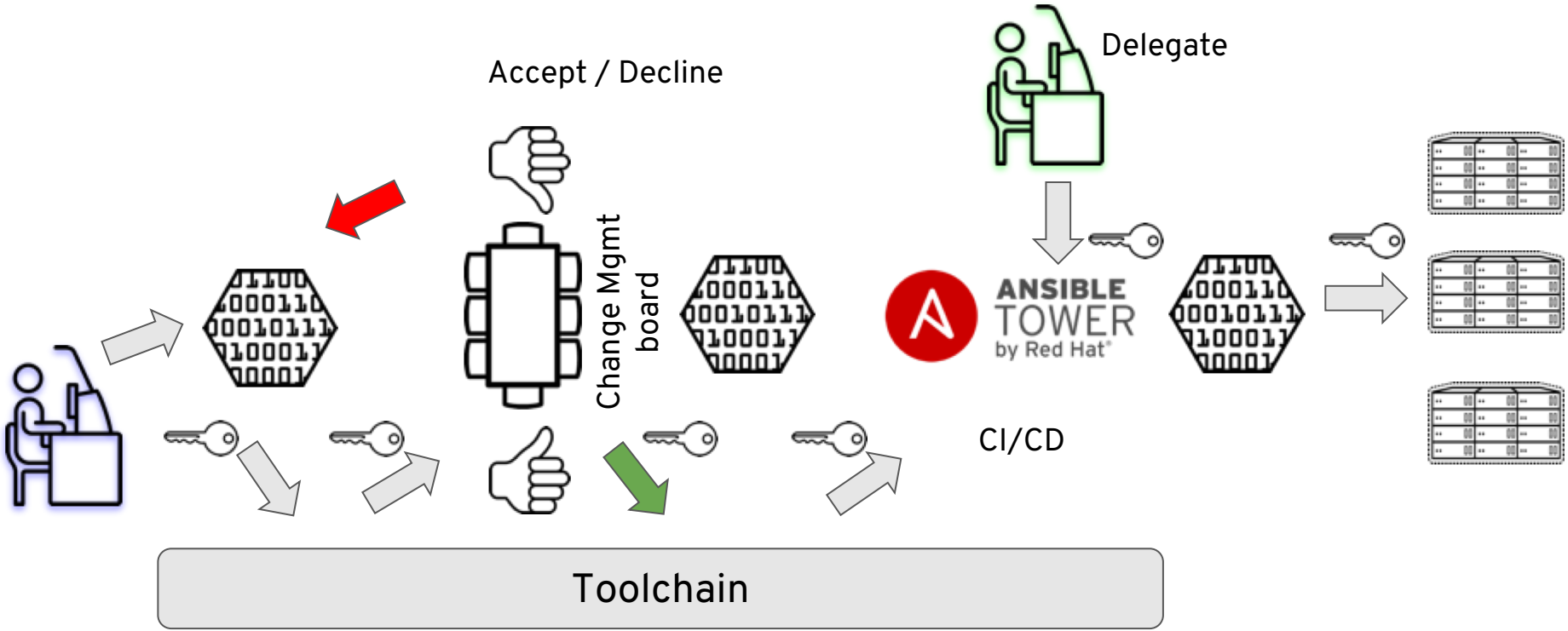
- Define it once - apply n times
- Abstraction of Infrastructure
- Reproducible / repeatable



Automation enables software development approach

- Treat automation like common software development
- Put changes through different stages
- Use peer review
- Use the DevOps principle
- Be agile

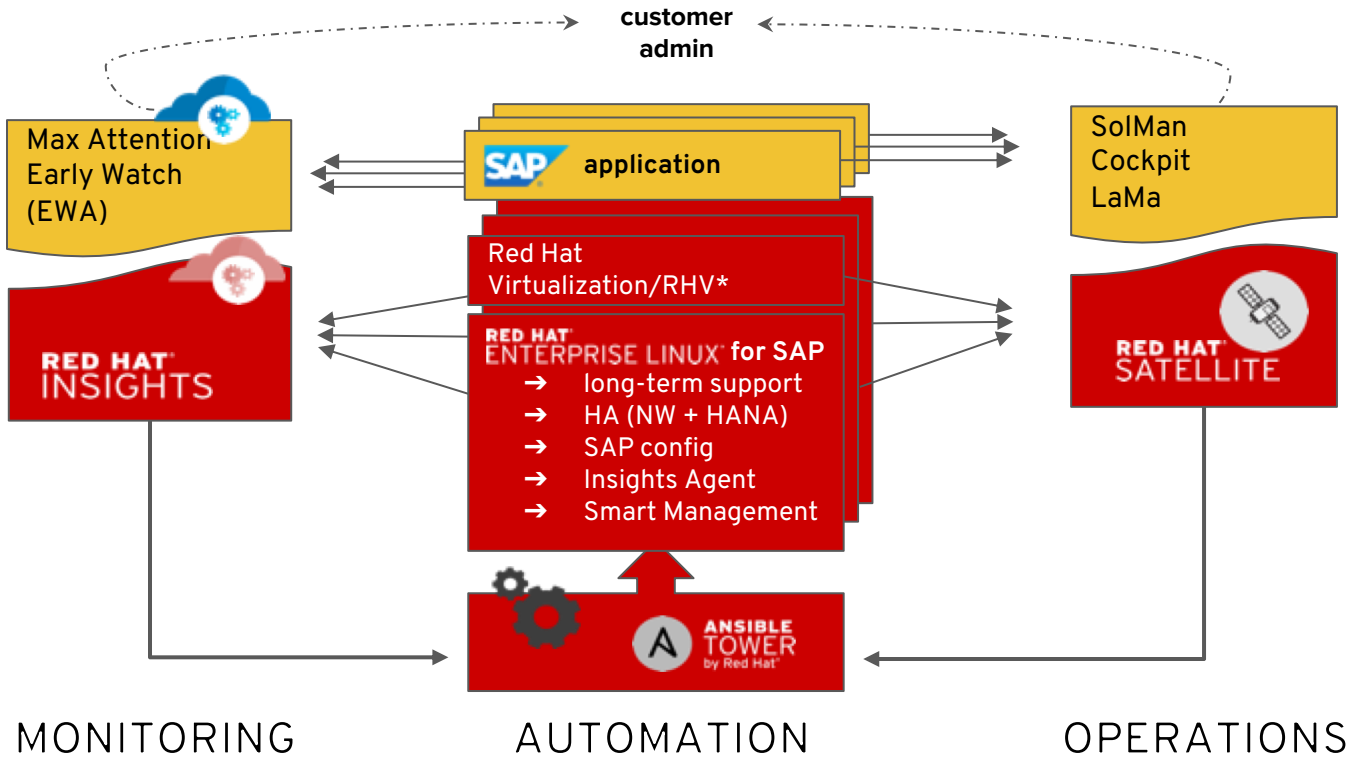
Automation enables software development approach



Red Hat

Red Hat SAP Offering

Match of SAP and Red Hat solutions



Red Hat

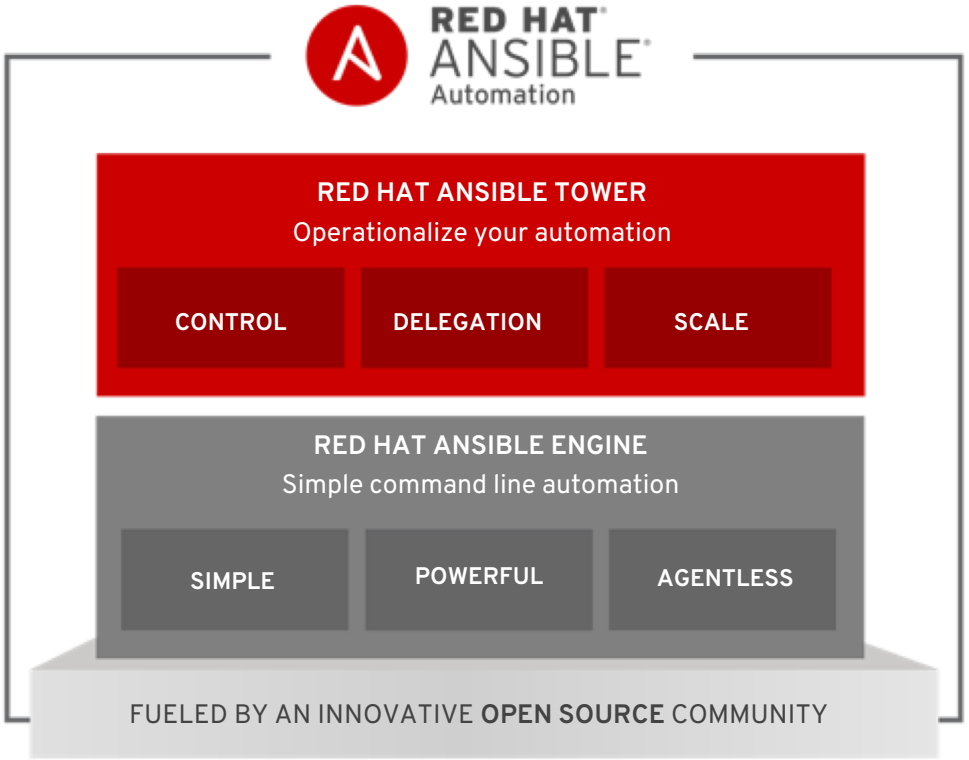
Ansible Automation

What is Ansible Automation?

Ansible Automation is the enterprise **framework** for automating across IT operations.

Ansible Engine runs Ansible Playbooks, the automation **language** that can perfectly describe an IT application infrastructure.

Ansible Tower allows you **scale** IT automation, manage complex deployments and speed productivity.



Red Hat

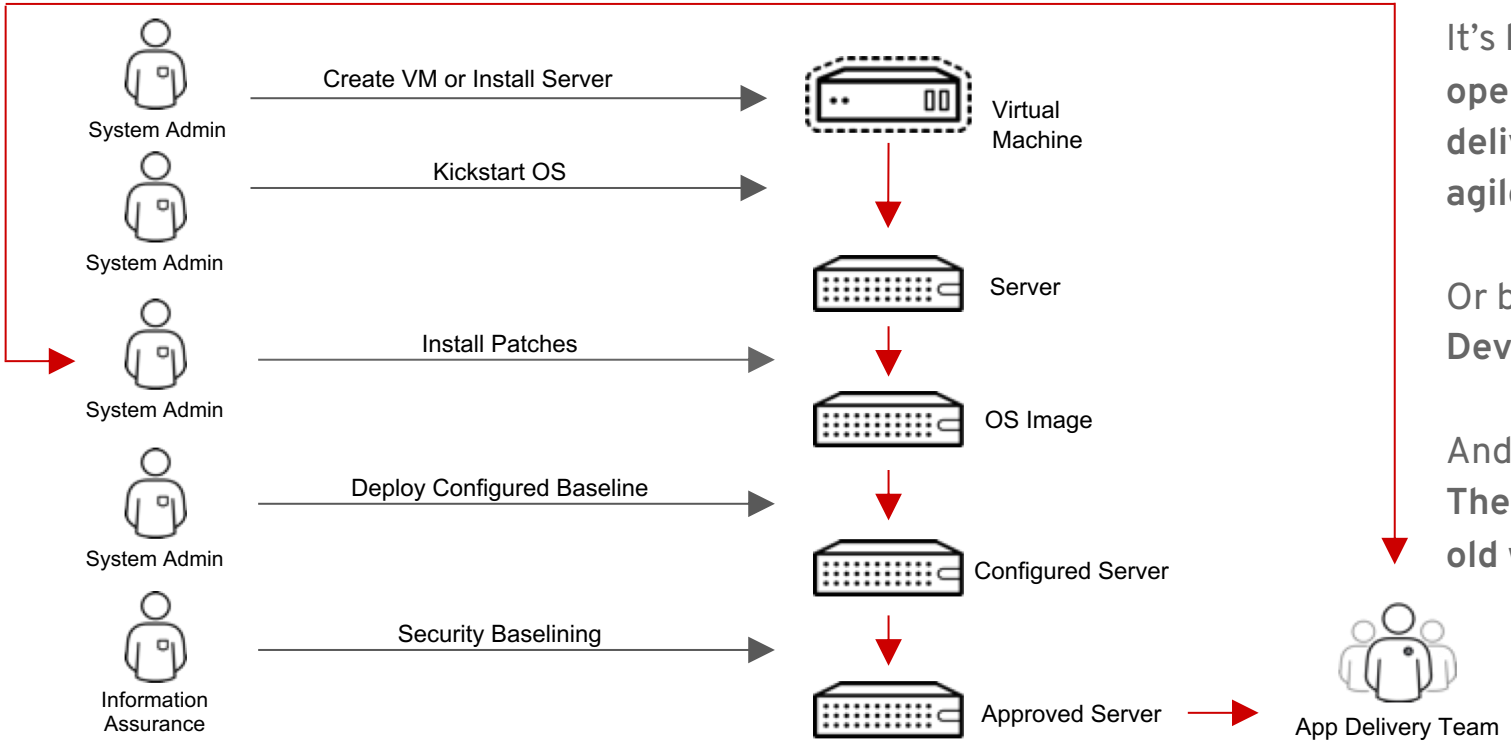
Ansible Automation for SAP

Ansible automation: faster & more secure deployments

- **Automated system provisioning** using configuration management
  - Set up a SAP (HANA) instance including best practices and tuning within less than 10 min.
  - reduce implementation time e.g. for 6 node HANA scale-out environment from 7 to 3 days
- **Fast, controlled and reproducible roll out of changes** into production
  - CI/CD and SOE for SAP HANA Infrastructure enables regular security updates in production environment
  - Avoid configuration drifts between staging / production environments
- **Zero Maintenance Downtime**
  - Make the complex process trustworthy
  - Reduce planned downtime from 100 h to less than 1 min
- **Integration** with existing management tools
  - Bare-Metal-as-a-Service

Red Hat

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...**

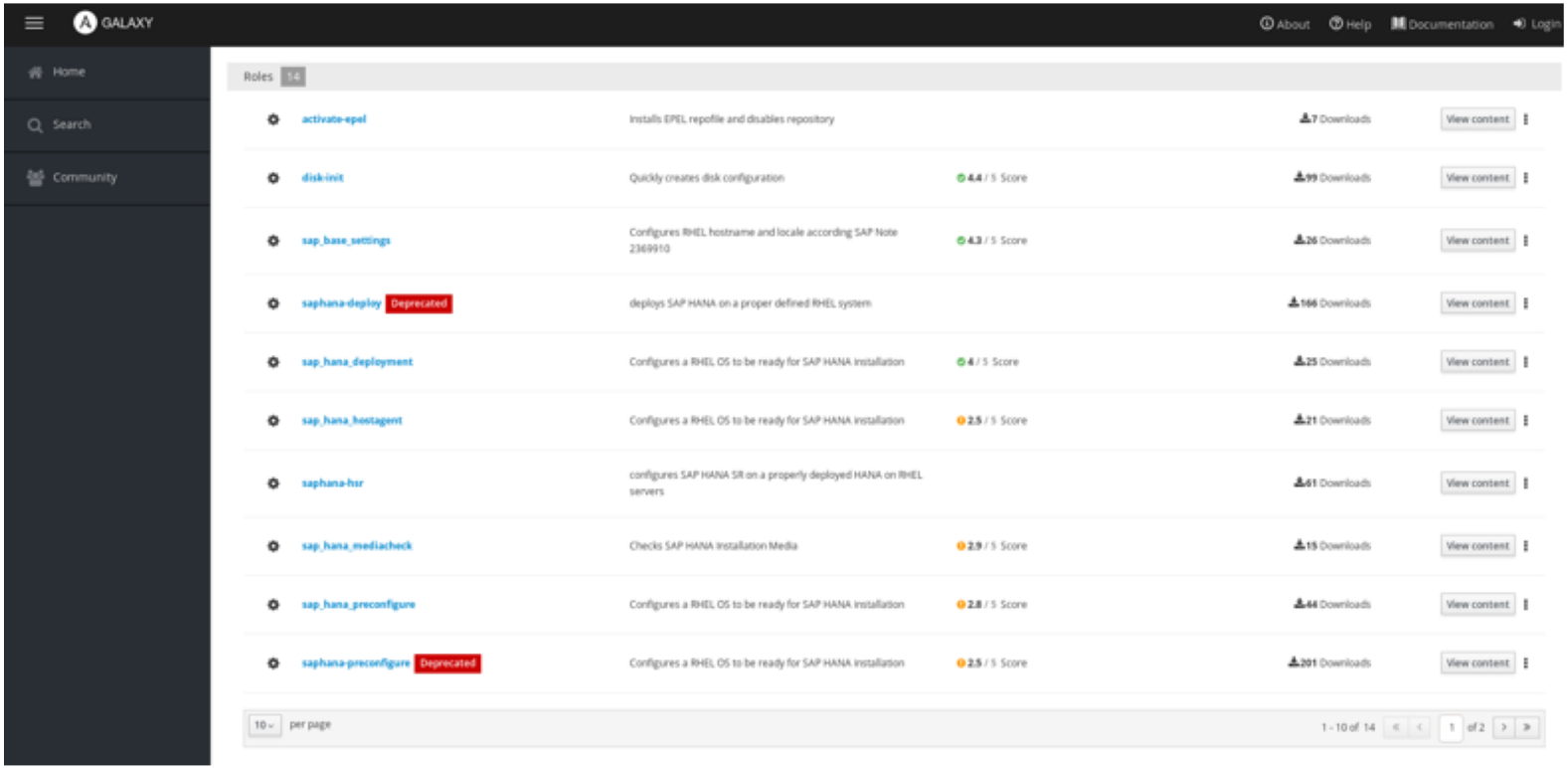
Red Hat

SAP HANA standard installation process

- On Site construction
- Installation & Configuration
- Installation & Configuration
- Validation & Customization
- Maintenance & Updates

Standard processes scream for automation!

Don’t reinvent the wheel: SAP HANA playbooks on Ansible Galaxy



Red Hat

Ansible Playbook Example

Automation workflow overview

Software enablement

Subscribe the systems to correct repositories

Set correct time/date

Configure the correct ntp servers

Prepare File Systems/Disks

Create Logical Volumes and filesystems and mount them accordingly

Preconfigure SAP (HANA)

Set correct hostname kernel parameters, DNS, etc.

Install SAP HANA Express

Install SAP HANA and start

Red Hat

Automation playbook example

Branch: **master** ▾ | New pull request | Find File | Clone or download ▾

pmumenthaler added missing vars — Latest commit `7fbd56d` on Feb 6

group_vars	add ansible files	7 months ago
host_vars	add config for hana0.example.com	7 months ago
README.md	first commit	7 months ago
ansible.cfg	add ansible files	7 months ago
install-hana.yml	added missing vars	7 months ago
inventory	add ansible files	7 months ago

Automation playbook example (snippets)


```yaml
- name: Install SAP HANA
  hosts: all
  become: yes

  vars:
            ######################################################
            # Default Subscription Information for HANA Servers
            # used in: mk-ansible-roles.rhn-subscribe

            satellite_server: satellite.example.com
            reg_activation_key: sap-hana
            reg_organization_id: RHPDS_Demo
            reg_server_insecure: yes

            reg_osrelease: 7.4
            repo_reset: true
            repositories:
               - rhel-7-server-e4s-rpms
               - rhel-sap-hana-for-rhel-7-server-e4s-rpms

            ######################################################
            # Default Timeserver settings
            # used in: rhel-system-roles.timeserver

            ntp_servers:
                    - hostname: 0.rhel.pool.ntp.org
                      iburst: yes
                    - hostname: 1.rhel.pool.ntp.org
                      iburst: yes
                    - hostname: 2.rhel.pool.ntp.org
                      iburst: yes
                    - hostname: 3.rhel.pool.ntp.org
                      iburst: yes
```

```yaml
            # SAP Precoonfigure role
            # SAP-Media Check
            install_nfs: "tower.example.com:/export"
            installroot: /install/hxe
            installversion: "HANA_EXPRESS_20"
            hana_installdir: "{{ installroot + '/' + installversion }}"

            hana_pw_hostagent_ssl: "MyS3cret!"
            id_user_sapadm: "30200"
            id_group_shm: "30220"
            id_group_sapsys: "30200"
            pw_user_sapadm_clear: "MyS3cret!"


  roles:
            - { role: mk-ansible-roles.subscribe-rhn }
            - { role: linux-system-roles.timesync }
            - { role: mk-ansible-roles.disk-init }
            - { role: mk-ansible-roles.saphana-preconfigure }
            - { role: mk-ansible-roles.saphana-deploy }
```

Automation playbook example (snippets)

```
- name: create disk partitions
  shell: |
   if pvdisplay -C '{{ item.key }}' ; then
       echo '{{ item.key }}' in use
       exit 0
   else
      sgdisk --zap {{ item.key }}
      /sbin/pvcreate -f '{{ item.key }}'
      exit 90
   fi
  with_dict: "{{ disks }}"
  register: pvcreate_result
  changed_when: pvcreate_result.rc == 90
  failed_when: pvcreate_result.rc > 0 and pvcreate_result.rc != 90
  tags:
    - disk_init
```

```
- name: create filesystems
  filesystem:
    dev:  /dev/{{ item.value.vol}}/{{ item.key }}
    fstype: "{{ item.value.fstype | default('xfs')}}"
    force: no
    opts: "{{ item.value.fsopts | default(omit) }}"
  with_dict: "{{ logvols }}"
  ignore_errors: True
  tags:
    - disk_init
  when: logvols is defined

- name: mount and make fstab entries
  mount:
    name: "{{ item.value.mountpoint }}"
    fstype: "{{ item.value.fstype | default('xfs')}}"
    opts: defaults
    passno: 4
    src: "/dev/{{ item.value.vol }}/{{ item.key }}"
    state: mounted
  with_dict: "{{ logvols }}"
  tags:
    - disk_init
  when: logvols is defined
```

Automation playbook example (snippets)

```
- name: create sap sid user
  user:
    name: "{{ item.value.hana_sid|lower }}adm"
    uid: "{{ item.value.id_user_sidadm }}"
    group: "sapsys"
    groups: "{{ item.value.hana_sid|lower }}shm"
    home: "/usr/sap/{{ item.value.hana_sid|upper }}/home"
    password: "{{ item.value.pw_user_sidadm|password_hash('sha512') }}"
  with_dict: "{{ instances }}"

# Check what is already installed
- name: get installed SAP instances
  shell: |
    # return nothing if uninstalled (rc=127) or not running (rc=1)
    # example output:
    # Inst Info : RH1 - 10 - hana1-repl - 749, patch 418, changelist 1816226
    if /usr/sap/hostctrl/exe/saphostctrl -function Ping > /dev/null 2>&1; then
       echo " - - -"
    else
       /usr/sap/hostctrl/exe/saphostctrl -function ListInstances | cut -d":" -f2-
    fi
  register: sap_hana_deployment_register_instancelist
  changed_when: false
```

Automation playbook example (snippets)

```
- name: execute unattended installation (logfile /var/log/hana_install.log )
  shell: ./hdblcm --configfile=/tmp/hana_install_{{ item.value.hana_sid|lower}}.cfg -b --remote_execution=saphostagent
  args:
    chdir: "{{ sap_hana_installdir }}/"
    creates: "/usr/sap/{{ item.value.hana_sid|upper }}/HDB{{ item.value.hana_instance_number }}/exe/hdbdaemon"
  with_dict: "{{ instances }}"
  when: deployment_instance is defined and deployment_instance|bool == true
```

Thank you

Red Hat is the world's leading provider of enterprise open source software solutions. Award-winning support, training, and consulting services make
Red Hat a trusted adviser to the Fortune 500.


linkedin.com/company/red-hat


youtube.com/user/RedHatVideos


facebook.com/redhatinc


twitter.com/RedHat

Red Hat

RED HAT FORUM | SEPTEMBER 10 2019

NEXT GENERATION SOLUTION FOR RED HAT OPENSHIFT CONTAINER PLATFORM

Kiril Petsev | Solution Architect | HPE

Peter Reichmuth | Senior Storage Consultant | HPE