Red Hat
Smart Management

Some Assembly Required: Satellite as Code

Somewhere between IaC and GitOps

Josh Swanson
Platform Specialist Solution Architect
joshswanson@redhat.com

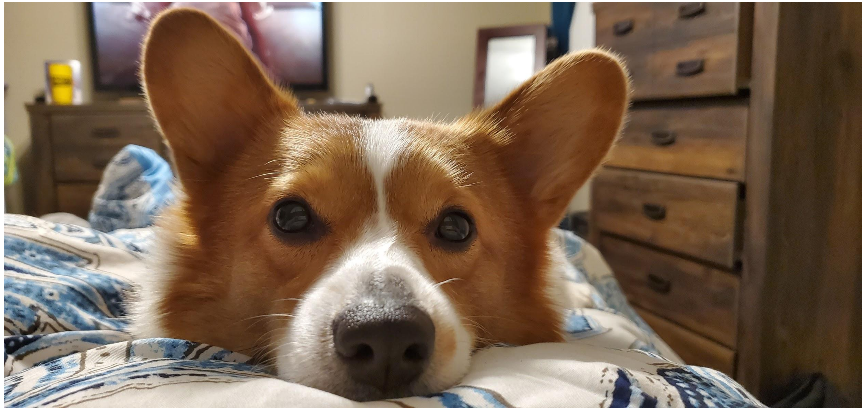
Preso Build Notes

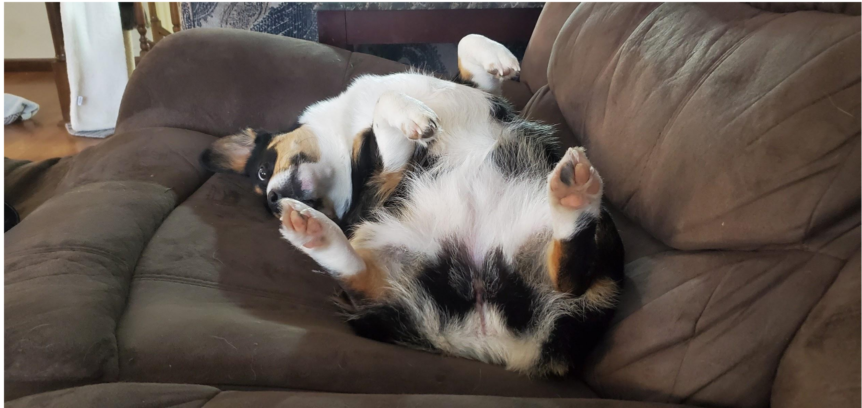
1. Start at 0 (I have a RHEL box)
2. Focus on satellite config sections
3. Preso should be “actionable”
4. Josh’s Wish List
5. Day 2 stuff
6. Creation flow

Red Hat

Josh Swanson

joshswanson@redhat.com





Housekeeping

Ansible Meetup

Want to speak at an ansible meetup? Yes you do!

https://www.meetup.com/Ansible-Minneapolis/

The Goal: Satellite as Code

The Why: Infrastructure as Code and GitOps

What are Infrastructure as Code and GitOps?

https://www.redhat.com/en/blog/helping-you-get-infrastructure-code
https://www.openshift.com/blog/introduction-to-gitops-with-openshift

“IaC is about describing the desired infrastructure in a file, written in a structured manner (code), so that an automation tool or engine can take that description and provision the infrastructure, or reconfigure an already-deployed infrastructure so it matches that description.

You can think of that codified description as something equivalent to a printing press die for a painting - once the die is made, you can create identical copies.”

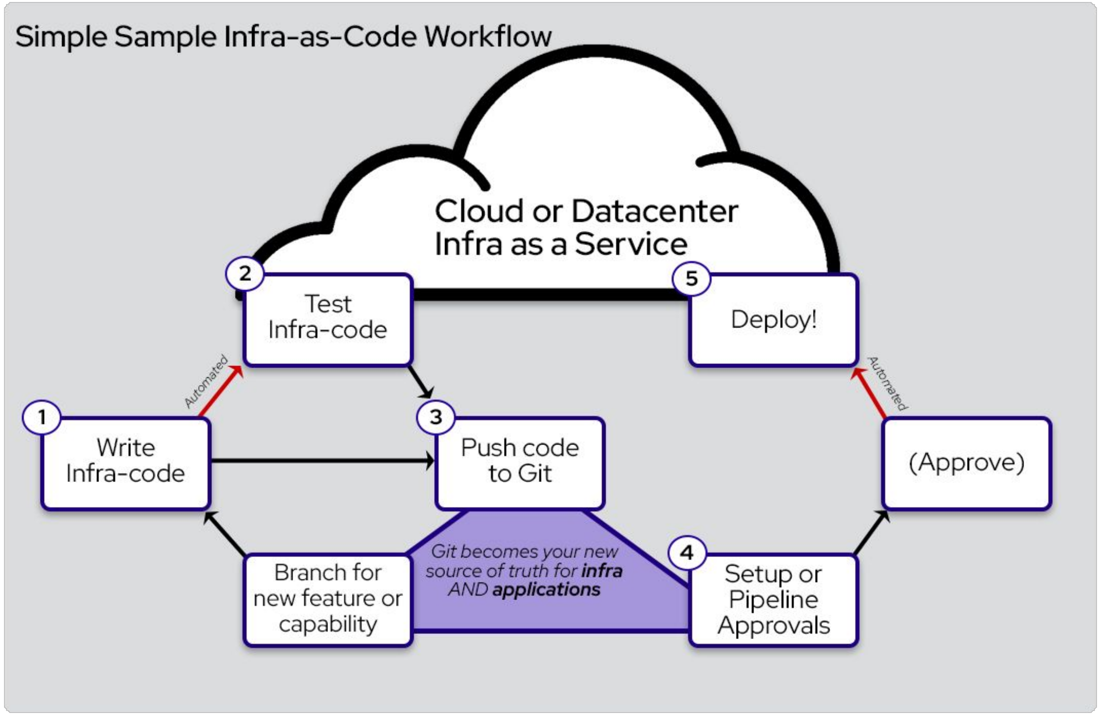
“GitOps in short is a set of practices to use Git pull requests to manage infrastructure and application configurations. Git repository in GitOps is considered the only source of truth and contains the entire state of the system so that the trail of changes to the system state are visible and auditable.

Traceability of changes in GitOps is no novelty in itself as this approach is almost universally employed for the application source code. However GitOps advocates applying the same principles (reviews, pull requests, tagging, etc) to infrastructure and application configuration so that teams can benefit from the same assurance as they do for the application source code.”

Basically: Code defines configuration.

What Does IaC Look Like in the Real World?

https://www.redhat.com/en/blog/helping-you-get-infrastructure-code



Simple Sample Infra-as-Code Workflow

Cloud or Datacenter Infra as a Service

1 Write Infra-code

2 Test Infra-code

3 Push code to Git

4 Setup or Pipeline Approvals

(Approve)

5 Deploy!

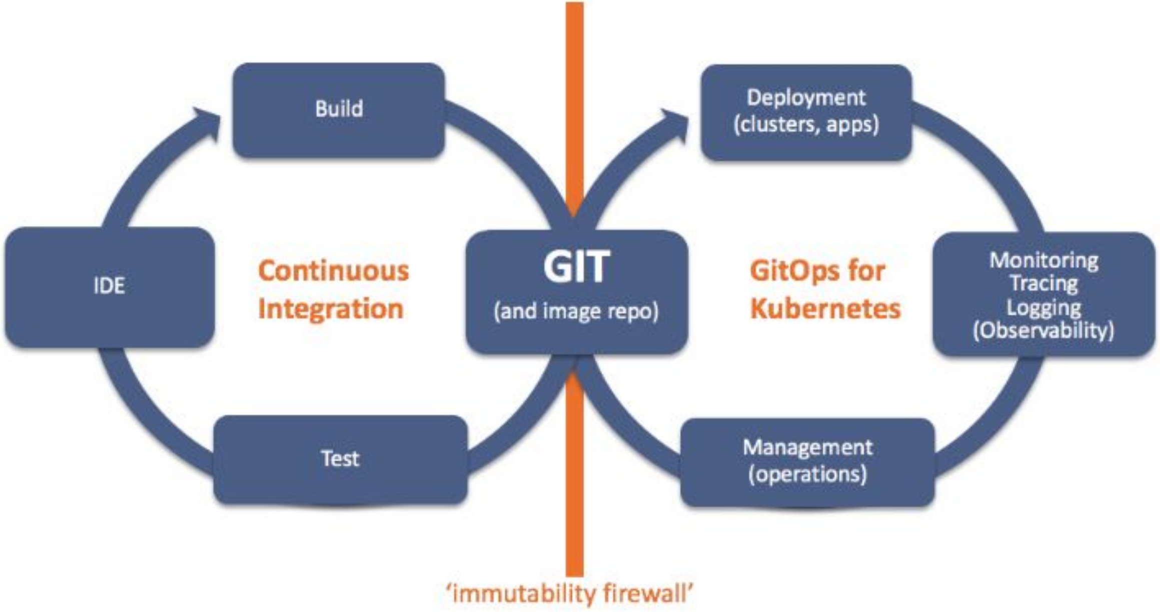
Branch for new feature or capability

*Git becomes your new source of truth for **infra** AND **applications***

Red Hat

What does GitOps Look Like in the Real World?

https://www.openshift.com/blog/introduction-to-gitops-with-openshift



Git as the single source of truth of a system's desired state

GitOps Diffs compare desired state with observed state (eg Kubediff, Terradiff, Canary..)

ALL intended operations are committed by pull request, for all environments

ALL diffs between GIT and observed state lead to (auto) convergence using tools like K8s

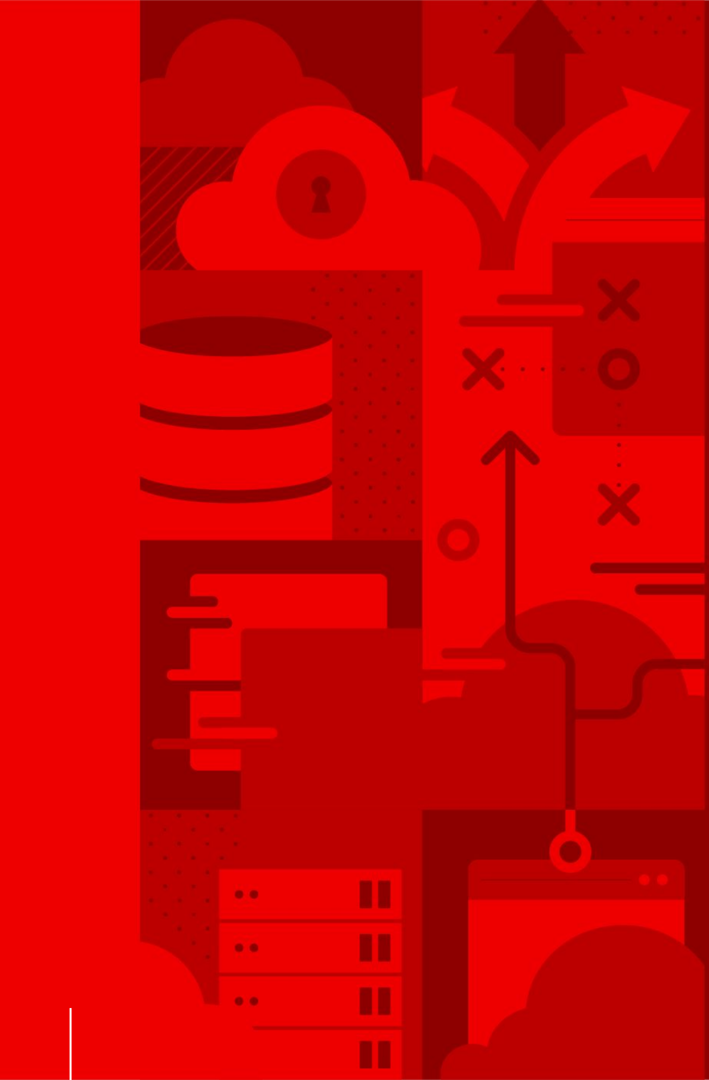
ALL changes are observable, verifiable and audited indisputably, with rollback & D/R

What are the Results of Adopting IaC/GitOps?

https://www.redhat.com/en/blog/helping-you-get-infrastructure-code
https://www.openshift.com/blog/introduction-to-gitops-with-openshift

- **The Automation of Repeated Tasks and Workflows**
  - Content view creation and promotion
  - Testing of newly published content
  - Rotation of credentials used for compute resources
- **The Benefits of Source Control**
  - Clear audit trail
  - Control around what gets deployed
  - Single source of truth
  - Low barrier to entry

Tl;dr: A better experience with Satellite



The How: Ansible and the Foreman Ansible Modules

The Foreman Ansible Modules

What are the Foreman Ansible Modules?

https://theforeman.org/2019/09/automating-foreman-and-katello-with-ansible.html

Foreman Ansible Modules (FAM) are a set of Ansible modules to manage Foreman ;-)

These modules are an evolution from the foreman and katello modules currently present in Ansible itself, as those are deprecated since Ansible 2.8 and are scheduled for removal in 2.12. Due to the use of a Katello (or rather Satellite) specific library, the old modules would not work properly in plain Foreman setups and often lacked features that were not yet present in Red Hat Satellite 6.

Over the course of the past year, the community sat together, cleaned the modules up, created tests and documentation and finally also ported the modules to a Satellite independent library.

How do the Modules Actually Work?

https://theforeman.org/2019/09/automating-foreman-and-katello-with-ansible.html

MAGIC! Well, actually, no, not magic, DOCUMENTATION!

Foreman has a powerful API with rich API documentation. This documentation is generated by the apipie-rails gem, which also provides a machine readable version of said documentation. You’ve probably seen long-ish rake apipie:cache processes when installing Foreman and plugins – that’s the gem re-generating the documentation to match the set of plugins available in your environment.

The modules use a library (apypie) that can parse the machine readable documentation of your instance and generate correct API requests based on that documentation.

Given almost all modules share a lot of common code, there is an abstraction class ForemanAnsibleModule which takes care of the common tasks like establishing an API connection, executing searches and creating/updating/deleting entities. This allows the modules be clean and only contain data/code relevant for their specific task – have a look at the foreman_organization module for a very simple example.

How can the modules be obtained?

Shipped as a collection

- Github
  - git clone https://github.com/theforeman/foreman-ansible-modules.git
- Automation Hub
  - ansible-galaxy collection install redhat.satellite
- Ansible Galaxy
  - ansible-galaxy collection install theforeman.foreman

```
[jswanson@rocinante configure_satellite]$ ansible-galaxy collection install theforeman.foreman
Process install dependency map
Starting collection install process
Skipping 'theforeman.foreman' as it is already installed
```

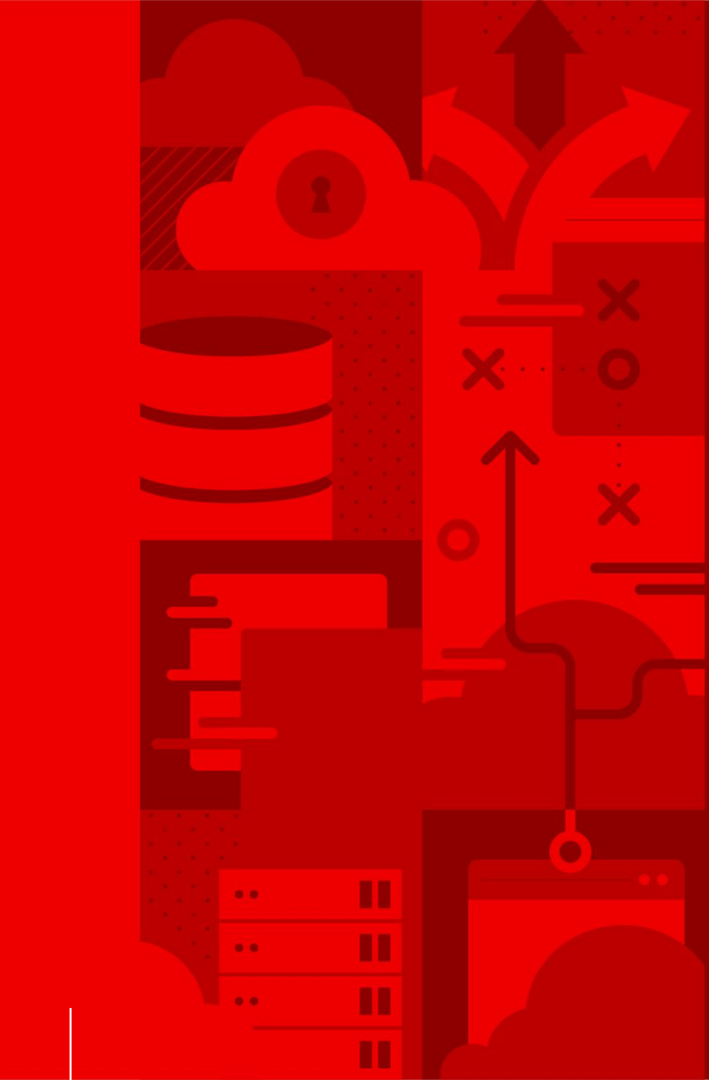
How Can the Modules be Used?

https://theforeman.org/2019/09/automating-foreman-and-katello-with-ansible.html

The foreman-ansible-modules git repository contains instructions how the modules can be installed in your environment and module documentation is available from theforeman.org.

Usually you'll find one module per Foreman entity (Organization, Location, Host Group etc.) or action (Katello Repository Sync, Katello Content Upload, etc).

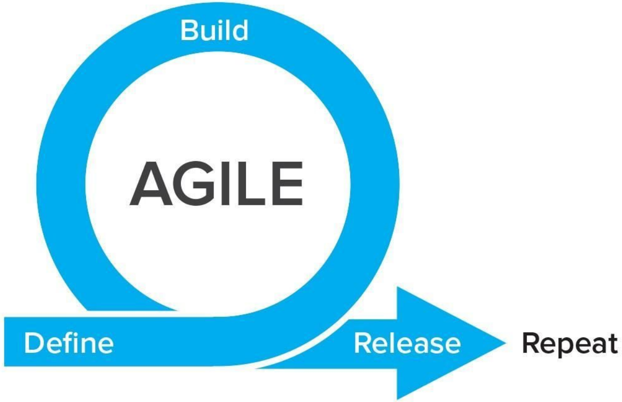
```
[jswanson@rocinante tasks]$ ls -l
/home/jswanson/.ansible/collections/ansible_collections/theforeman/foreman/plugins/modules
total 384
-rw-r--r--. 1 jswanson jswanson 15243 Jul 21 21:20 activation_key.py
-rw-r--r--. 1 jswanson jswanson  2715 Jul 21 21:20 architecture.py
-rw-r--r--. 1 jswanson jswanson  5955 Jul 21 21:20 auth_source_ldap.py
-rw-r--r--. 1 jswanson jswanson  3389 Jul 21 21:20 bookmark.py
-rw-r--r--. 1 jswanson jswanson  3202 Jul 21 21:20 compute_attribute.py
-rw-r--r--. 1 jswanson jswanson  5641 Jul 21 21:20 compute_profile.py
-rw-r--r--. 1 jswanson jswanson 11338 Jul 21 21:20 compute_resource.py
```



Satellite as Code

Take an Iterative Approach

- Start small and grow
  - Satellite doesn’t have to be 100% code on day 1
- Align to agile methodologies/principles
  - Satellite is “just another app”



Satellite as Code

A bit of a teaser...

```
---
satellite:
  admin_username: admin
  admin_password: changeme
  foreman:
    organizations: "{{ lookup('file', 'vars/satellite/organizations.yml') | from_yaml }}"
    locations: "{{ lookup('file', 'vars/satellite/locations.yml') | from_yaml }}"
    domains: "{{ lookup('file', 'vars/satellite/domains.yml') | from_yaml }}"
    subnets: "{{ lookup('file', 'vars/satellite/subnets.yml') | from_yaml }}"
    compute_resources: "{{ lookup('file', 'vars/satellite/compute-resources.yml') | from_yaml }}"
    compute_profiles: "{{ lookup('file', 'vars/satellite/compute-profiles.yml') | from_yaml }}"
    hostgroups: "{{ lookup('file', 'vars/satellite/hostgroups.yml') | from_yaml }}"
    partition_tables: "{{ lookup('file', 'vars/satellite/partition-tables.yml') | from_yaml }}"
    operating_systems: "{{ lookup('file', 'vars/satellite/operating-systems.yml') | from_yaml }}"
  katello:
    - organization_name: xyz_corp
      state: present
      manifest: files/satellite/manifest_xyz_corp_20200901T135443Z.zip
      repo_sync_attempts: 1
      lifecycle_environments: "{{ lookup('file', 'vars/satellite/lifecycle-environments.yml') | from_yaml }}"
      content_views: "{{ lookup('file', 'vars/satellite/content-views.yml') | from_yaml }}"
      composite_content_views: "{{ lookup('file', 'vars/satellite/composite-content-views.yml') | from_yaml }}"
      activation_keys: "{{ lookup('file', 'vars/satellite/activation-keys.yml') | from_yaml }}"
```

Josh’s Satellite Collections

Focus on Satellite configuration

Getting Started

Getting the Code

```
[jswanson@rocinante naps-satellite-demo]$ ansible-galaxy collection install jjaswanson4.setup_rhel_for_satellite
Process install dependency map
Starting collection install process
Installing 'jjaswanson4.setup_rhel_for_satellite:1.0.1' to
'/home/jswanson/.ansible/collections/ansible_collections/jjaswanson4/setup_rhel_for_satellite'

[jswanson@rocinante naps-satellite-demo]$ ansible-galaxy collection install jjaswanson4.install_satellite
Process install dependency map
Starting collection install process
Installing 'jjaswanson4.install_satellite:1.0.0' to
'/home/jswanson/.ansible/collections/ansible_collections/jjaswanson4/install_satellite'

[jswanson@rocinante naps-satellite-demo]$ ansible-galaxy collection install jjaswanson4.configure_satellite
Process install dependency map
Starting collection install process
Installing 'jjaswanson4.configure_satellite:1.0.2' to
'/home/jswanson/.ansible/collections/ansible_collections/jjaswanson4/configure_satellite'
```

Configuration Flow

- **Katello-Independent Components**
  - Organizations
  - Locations
  - Subnets
  - Domains
  - Compute resources
- **Katello Components**
  - Lifecycle environments
  - Content views
  - Activation keys
- **Katello-Dependent Components**
  - Hostgroups
  - Operating systems
  - Provisioning templates
  - Partitioning templates

```
---
- name: configure satellite
  hosts:
    - satellite
  collections:
    - jjaswanson4.configure_satellite
    - theforeman.foreman
  pre_tasks:
    - name: import satellite configuration vars file
      include_vars:
        file: vars/satellite/satellite.yml
      delegate_to: localhost
  tasks:
    - name: include configure_foreman role with katello independent pieces
      include_role:
        name: jjaswanson4.configure_satellite.configure_foreman
    - name: build satellite by organization
      include_role:
        name: configure_katello
      loop_control:
        loop_var: organization
      loop: "{{ satellite.katello }}"
    - name: include tasks to create hostgroup after content is available
      include_role:
        name: configure_foreman
      vars:
        requires_katello_content: true
```

Configuring Foreman

Configuring Foreman

Settings

```
# vars file
---
satellite:
  foreman:
   settings:
      - name: default_download_policy
        value: on_demand
      - name: default_proxy_download_policy
        value: on_demand
      - name: unregister_delete_host
        value: true
```

```
# task file
---
- name: set initial settings
  theforeman.foreman.setting:
    username: "{{ satellite.admin_username }}"
    password: "{{ satellite.admin_password }}"
    validate_certs: false
    server_url: "{{ satellite_url }}"
    name: "{{ setting.name }}"
    value: "{{ setting.value }}"
  loop_control:
    loop_var: setting
  loop: "{{ satellite.foreman.settings }}"
  when:
    - satellite.foreman.settings is defined
```

Configuring Foreman

Organizations

```
# vars file
---
satellite:
  foreman:
   organizations:
      - name: org_1
        initial_organization: true
      - name: org_2
      - name: org_3
```

```
# task file
---
- name: configure satellite organizations
  theforeman.foreman.organization:
    username: "{{ satellite.admin_username }}"
    password: "{{ satellite.admin_password }}"
    validate_certs: false
    server_url: "{{ satellite_url }}"
    name: "{{ organization.name }}"
    state: "{{ organization.state | default(omit) }}"
  loop_control:
    loop_var: organization
  loop: "{{ satellite.foreman.organizations }}"
  delegate_to: "{{ delegate_host }}"
```

Configuring Foreman

Locations

```
# vars file
---
satellite:
  foreman:
   locations:
      - name: loc_1
        initial_location: true
        organizations:
          - name: org_1
          - name: org_2
          - name: org_3
      - name: loc_2
        organizations:
          - name: org_2
      - name: loc_3
        organizations:
          - name: org_2
          - name: org_3
```

```
# task file
---
- name: configure satellite locations
  theforeman.foreman.location:
    username: "{{ satellite.admin_username }}"
    password: "{{ satellite.admin_password }}"
    validate_certs: false
    server_url: "{{ satellite_url }}"
    organizations: "{{ location |
json_query('organizations[*].name') | list }}"
    name: "{{ location.name }}"
    state: "{{ location.state | default(omit) }}"
  loop_control:
    loop_var: location
  loop: "{{ satellite.foreman.locations }}"
  delegate_to: "{{ delegate_host }}"
```

Configuring Foreman

Domains

```
# vars file
---
satellite:
  foreman:
   domains:
      - name: domain1.internal.lcl
        description: default dns domain
        organizations:
          - name: org_1
          - name: org_3
        locations:
          - name: loc_1
          - name: loc_3
      - name: domain2.internal.lcl
        description: secondary dns domain
        organizations:
          - name: org_2
        locations:
          - name: loc_2
```

```
# task file
---
- name: configure domains
  theforeman.foreman.domain:
    username: "{{ satellite.admin_username }}"
    password: "{{ satellite.admin_password }}"
    validate_certs: false
    server_url: "{{ satellite_url }}"
    organizations: "{{ domain | json_query('organizations[*].name')
| list }}"
    locations: "{{ domain | json_query('locations[*].name') | list
}}"
    name: "{{ domain.name }}"
    state: "{{ location.state | default(omit) }}"
  loop_control:
    loop_var: domain
  loop: "{{ satellite.foreman.domains }}"
  when:
    - satellite.foreman.domains is defined
  delegate_to: "{{ delegate_host }}"
```

Configuring Foreman

Subnets

```
# vars file
---
satellite:
  foreman:
   subnets:
      - name: test-subnet-192.168.0.0_24
        network: 192.168.0.0
        mask: 255.255.255.0
        gateway: 192.168.0.1
        dns_primary: 192.168.0.10
        dns_secondary: 192.168.1.11
        domains:
          - name: domain1.internal.lcl
        organizations:
          - name: org_1
          - name: org_3
        locations:
          - name: loc_1
          - name: loc_3
```

```
# task file
- name: configure subnets
...
    organizations: "{{ subnet | json_query('organizations[*].name')
| list }}"
    locations: "{{ subnet | json_query('locations[*].name') | list
}}"
    domains: "{{ subnet | json_query('domains[*].name') | list  }}"
    name: "{{ subnet.name }}"
    state: "{{ subnet.state | default(omit) }}"
    from_ip: "{{ subnet.from_ip | default(omit) }}"
    to_ip: "{{ subnet.to_ip | default(omit) }}"
    boot_mode: "{{ subnet.boot_mode | default(omit) }}"
    dhcp_proxy: "{{ subnet.dhcp_proxy | default(omit) }}"
    tftp_proxy: "{{ subnet.tftp_proxy | default(omit) }}"
    dns_proxy: "{{ subnet.dns_proxy | default(omit) }}"
    template_proxy: "{{ template_proxy | default(omit) }}"
    vlanid: "{{ subnet.vlanid | default(omit) }}"
    mtu: "{{ subnet.mtu | default(omit) }}"
    network: "{{ subnet.network }}"
    mask: "{{ subnet.mask }}"
    gateway: "{{ subnet.gateway }}"
...
```

Configuring Foreman

Compute Resources

```
# vars file
---
satellite:
  foreman:
    compute_resources:
      - name: example_vcenter
        provider: vmware
        provider_params:
          url: vcenter.domain1.interna.lcl
          user: provisioning@vsphere.local
          password: "{{ lookup('file',
'/tmp/vcenter-password') }}"
          datacenter: dc1
        organizations:
          - name: org_1
          - name: org_2
        locations:
          - name: loc_1
          - name: loc_3
```

```
# task file
---
- name: configure compute resources
  theforeman.foreman.compute_resource:
    username: "{{ satellite.admin_username }}"
    password: "{{ satellite.admin_password }}"
    validate_certs: false
    server_url: "{{ satellite_url }}"
    organizations: "{{ compute_resource |
json_query('organizations[*].name') | list }}"
    locations: "{{ compute_resource |
json_query('locations[*].name') | list }}"
    name: "{{ compute_resource.name }}"
    state: "{{ compute_resource.state | default(omit) }}"
    provider: "{{ compute_resource.provider }}"
    provider_params: "{{ compute_resource.provider_params }}"
  loop_control:
    loop_var: compute_resource
  loop: "{{ satellite.foreman.compute_resources }}"
  when:
    - satellite.foreman.compute_resources is defined
  delegate_to: "{{ delegate_host }}"
```

Configuring Foreman

Compute Profiles

```
# vars file
---
satellite:
  foreman:
   compute_profiles:
      - name: general-vm
        compute_resource: example_vcenter
        vm_attrs:
          cpus: 2
          corespersocket: 2
          memory_mb: 2048
          cluster: general
          path: /Datacenters/dc1/vm
          guest_id: rhel7_64Guest
          interface_attributes:
            0:
              type: VirtualVmxnet3
              network: virtaul-machines
          volumes_attributes:
            0:
              size_gb: 100
              datastore: datastore1
```

```
# task file
---
- name: configure compute_profiles
  theforeman.foreman.compute_profile:
    username: "{{ satellite.admin_username }}"
    password: "{{ satellite.admin_password }}"
    validate_certs: false
    server_url: "{{ satellite_url }}"
    name: "{{ compute_profile.name }}"
    state: "{{ compute_profile.state | default(omit) }}"
    compute_attributes:
      - compute_resource: "{{ compute_profile.compute_resource }}"
        vm_attrs: "{{ compute_profile.vm_attrs }}"
  loop_control:
    loop_var: compute_profile
  loop: "{{ satellite.foreman.compute_profiles }}"
  when:
    - satellite.foreman.compute_profiles is defined
  delegate_to: "{{ delegate_host }}"
```

Configuring Foreman

Provisioning Templates

```
# vars file
---
satellite:
  foreman:
   provisioning_templates:
      - name: kickstart-rhel
        file: /tmp/kickstartrhel.erb
        organizations:
          - name: org_1
          - name: org_2
        locations:
          - name: loc_1
```

```
# task file
---
- name: configure provisioning templates
  theforeman.foreman.provisioning_template:
    username: "{{ satellite.admin_username }}"
    password: "{{ satellite.admin_password }}"
    validate_certs: false
    server_url: "{{ satellite_url }}"
    name: "{{ provisioning_template.name }}"
    state: "{{ provisioning_template.state | default(omit) }}"
    file_name: "{{ provisioning_template.file }}"
    organizations: "{{ provisioning_template |
json_query('organizations[*].name') | list }}"
    locations: "{{ provisioning_template |
json_query('locations[*].name') | list }}"
  loop_control:
    loop_var: provisioning_template
  loop: "{{ satellite.foreman.provisioning_templates }}"
  when:
    - satellite.foreman.provisioning_templates is defined
  delegate_to: "{{ delegate_host }}"
```

Configuring Foreman

Partition Tables

```
# vars file
---
satellite:
  foreman:
   partition_tables:
      - name: RHEL7 Kickstart Partition Table
        file:
/tmp/rhel7_kickstart_partition_table.erb
        organizations:
          - name: org_1
          - name: org_2
        locations:
          - name: loc_1
          - name: loc_2
          - name: loc_3
```

```
# task file
---
- name: configure partition templates
  theforeman.foreman.partition_table:
    username: "{{ satellite.admin_username }}"
    password: "{{ satellite.admin_password }}"
    validate_certs: false
    server_url: "{{ satellite_url }}"
    name: "{{ partition_table.name }}"
    state: "{{ partition_table.state | default(omit) }}"
    file_name: "{{ partition_table.file }}"
    organizations: "{{ partition_table |
json_query('organizations[*].name') | list }}"
    locations: "{{ partition_table | json_query('locations[*].name')
| list }}"
  loop_control:
    loop_var: partition_table
  loop: "{{ satellite.foreman.partition_tables }}"
  when:
    - satellite.foreman.partition_tables is defined
  delegate_to: "{{ delegate_host }}"
```

Configuring Foreman

Example .erb file

```
<%#
name: RHEL7 Kickstart Partition Table
snippet: false
model: Ptable
os_family: Redhat
organizations:
- org_1
locations:
- loc_1
%>
zerombr
clearpart --all --initlabel
part /boot --fstype xfs --size=1024
part swap --size=4096
part pv.01 --size=1000 --grow --ondisk=sda
volgroup os pv.01
logvol / --vgname=os --fstype=xfs --size=4096 --name=root
logvol /var --vgname=os --fstype=xfs --size=4096 --name=var
logvol /tmp --vgname=os --fstype=xfs --size=4096 --name=tmp
```

Configuring Katello

Configuring Katello

A more “order-oriented” workflow

1. Get subscriptions onto satellite
2. Setup lifecycle environments
3. Enable Red Hat repositories
4. Setup content credentials
5. Setup sync plan
6. Setup custom products
7. Setup custom repositories
8. Sync content
9. Create content views
10. Create composite content views
11. Promote content views
12. Setup activation keys

…And do it by organization for multi-org satellites

Configuring Katello

Red Hat Manifest

```
# vars file
---
satellite:
 katello:
    - organization_name: org_1
      state: present
      manifest:
/tmp/manifest_demo-satellite01-org_1_20200722T18430
6Z.zip
```

```
# task file
---
- name: upload manifest
  theforeman.foreman.subscription_manifest:
    username: "{{ satellite.admin_username }}"
    password: "{{ satellite.admin_password }}"
    validate_certs: false
    server_url: "{{ satellite_url }}"
    organization: "{{ organization.organization_name }}"
    state: present
    manifest_path: "{{ organization.manifest }}"
  when:
    - organization.manifest is defined
  delegate_to: "{{ delegate_host }}"
```

Configuring Katello

Lifecycle Environments

```
# vars file
---
satellite:
 katello:
    - organization_name: org_1
      lifecycle_environments:
        - name: test
          description: wild west
          prior: Library
        - name: stage
          description: dont test in prod
          prior: test
        - name: prod
          description: big leagues
          prior: stage
```

```
# task file
---
- name: configure lifecycle environments
  theforeman.foreman.lifecycle_environment:
    username: "{{ satellite.admin_username }}"
    password: "{{ satellite.admin_password }}"
    validate_certs: false
    server_url: "{{ satellite_url }}"
    organization: "{{ organization.organization_name }}"
    name: "{{ lifecycle_environment.name }}"
    prior: "{{ lifecycle_environment.prior }}"
    description: "{{ lifecycle_environment.description |
default(omit) }}"
  loop_control:
    loop_var: lifecycle_environment
  loop: "{{ organization.lifecycle_environments }}"
  when:
    - organization.lifecycle_environments is defined
  delegate_to: "{{ delegate_host }}"
```

Configuring Katello

Red Hat Repositories

```
# vars file
---
satellite:
 katello:
    - organization_name: org_1
      content_views:
        - name: cv-server-rhel8
          repos:
            - repo: Red Hat Enterprise Linux 8 for
x86_64 - BaseOS (RPMs)
              product: Red Hat Enterprise Linux for
x86_64
              releasever: 8
            - repo: Red Hat Satellite Tools 6.6 for
RHEL 8 x86_64 (RPMs)
              product: Red Hat Enterprise Linux for
x86_64
              content_view: cv-server-rhel8
```

```
# task file
---
- name: configure Red Hat repos
  theforeman.foreman.repository_set:
    username: "{{ satellite.admin_username }}"
    password: "{{ satellite.admin_password }}"
    validate_certs: false
    server_url: "{{ satellite_url }}"
    organization: "{{ organization.organization_name }}"
    name: "{{ repo.1.repo }}"
    product: "{{ repo.1.product }}"
    repositories:
    - releasever: "{{ repo.1.releasever | default(omit) }}"
      basearch: "{{ repo.1.basearch | default(omit) }}"
  when:
    - repo.1.content_credential is not defined
  loop_control:
    loop_var: repo
  loop: "{{ organization.content_views | subelements('repos') }}"
  delegate_to: "{{ delegate_host }}"
```

Configuring Katello

Sync Plan

```
# vars file
---
satellite:
 Katello:
   -    organization: org_1
        sync_plan: daily
```

```
# task file
---
- name: set Red Hat products to sync plan
  theforeman.foreman.product:
    username: "{{ satellite.admin_username }}"
    password: "{{ satellite.admin_password }}"
    validate_certs: false
    server_url: "{{ satellite_url }}"
    organization: "{{ organization.organization_name }}"
    name: "{{ product }}"
    sync_plan: "{{ organization.sync_plan }}"
  loop_control:
    loop_var: product
  with_items:
    - "{{ redhat_products.stdout_lines }}"
  delegate_to: "{{ delegate_host }}"
```

Configuring Katello

Custom Products

```
# vars file
---
satellite:
 katello:
    - organization_name: org_1
      content_views:
        - name: cv-epel-rhel8
          repos:
            - repo: Extra Packages for Enterprise
Linux 8 Repository
              repo_url:
http://download.fedoraproject.org/pub/epel/8/Everyt
hing/x86_64
              product: Extra Packages for
Enterprise Linux 8
              content_credential:
https://dl.fedoraproject.org/pub/epel/RPM-GPG-KEY-E
PEL-8
              content_credential_name: cc-epel8
```

```
# task file
---
- name: configure custom products
  theforeman.foreman.product:
    username: "{{ satellite.admin_username }}"
    password: "{{ satellite.admin_password }}"
    validate_certs: false
    server_url: "{{ satellite_url }}"
    organization: "{{ organization.organization_name }}"
    name: "{{ product.1.product }}"
    gpg_key: "{{ product.1.content_credential_name }}"
    sync_plan: daily
  when:
    - product.1.product is defined
    - product.1.content_credential_name is defined
  loop_control:
    loop_var: product
  loop: "{{ organization.content_views | subelements('repos') }}"
  delegate_to: "{{ delegate_host }}"
```

Configuring Katello

Sync Repos

```
# vars file
---
satellite:
  katello:
    - organization_name: org_1
      repo_sync_attempts: 3
```

```
# task file
---
- include_tasks: sync-repos.yml
  loop: "{{ range(0, organization.repo_sync_attempts, 1) | list }}"

---
- name: sync all repos
  theforeman.foreman.repository_sync:
    username: "{{ satellite.admin_username }}"
    password: "{{ satellite.admin_password }}"
    validate_certs: false
    server_url: "{{ satellite_url }}"
    organization: "{{ organization.organization_name }}"
    product: "{{ product }}"
  loop_control:
    loop_var: product
  loop: "{{ all_products.stdout_lines }}"
  async: 999999
  poll: 0
  delegate_to: "{{ delegate_host }}"
  tags:
    - sync_repos
```

Configuring Katello

Content Views

```
# vars file
---
satellite:
  katello:
    - organization_name: org_1
      content_views:
       - name: cv-server-rhel8
          repos:
            - repo: Red Hat Enterprise Linux 8 for
x86_64 - BaseOS (RPMs)
              product: Red Hat Enterprise Linux for
x86_64
              releasever: 8
            - repo: Red Hat Satellite Tools 6.6 for
RHEL 8 x86_64 (RPMs)
              product: Red Hat Enterprise Linux for
x86_64
              content_view: cv-server-rhel8
```

```
# task file
---
- name: push task file to create content view based on number of
repos
  template:
    src: templates/content-view.yml.j2
    dest: "{{ role_path }}/tasks/satellite-configuration/{{
organization.organization_name }}/content-views/{{ content_view.name
}}.yml"
  loop_control:
    loop_var: content_view
  loop: "{{ organization.content_views }}"
  delegate_to: "{{ delegate_host }}"

- name: include tasks to create content views
  include_tasks: "{{ role_path }}/tasks/satellite-configuration/{{
organization.organization_name }}/content-views/{{ content_view.name
}}.yml"
  loop_control:
    loop_var: content_view
  loop: "{{ organization.content_views }}"
```

Configuring Katello

Templated Ansible Task File

```
- name: configure content view {{ content_view.name }}
  theforeman.foreman.content_view:{% raw %}
    username: "{{ satellite.admin_username }}"
    password: "{{ satellite.admin_password }}"
    validate_certs: false
    server_url: "{{ satellite_url }}"
    organization: "{{ organization.organization_name }}"
{% endraw %}
    name: {{ content_view.name }}
{% if content_view.repos is defined %}
    repositories:
{% for repo in content_view.repos %}
      - name: {{ repo.repo | regex_replace('\\(','') | regex_replace('\\)','') + (' ' + repo.basearch if repo.basearch is defined else '') + (' ' + (repo.releasever | string) if
repo.releasever is defined else '') }}
        product: {{ repo.product }}
{% endfor %}
{% endif %}
  register: publish_content_view{% raw %}
  delegate_to: "{{ delegate_host }}"
{% endraw %}

{% if content_view.filters is defined %}
{% for content_view_filter in content_view.filters %}
- name: configure filter {{ content_view_filter.name }} for {{ content_view.name }}
  theforeman.foreman.content_view_filter:{% raw %}
    username: "{{ satellite.admin_username }}"
    password: "{{ satellite.admin_password }}"
    validate_certs: false
    server_url: "{{ satellite_url }}"
    organization: "{{ organization.organization_name }}"
{% endraw %}
...
```

Configuring Katello

Content Views

```
# vars file
---
satellite:
  katello:
    - organization_name: org_1
     composite_content_views:
        - name: composite-rhel8
          description: rhel 8 packages
          auto_publish: false
          component_content_views:
            - name: cv-server-rhel8
            - name: cv-ansible-rhel8
            - name: cv-appstream-rhel8
            - name: cv-supplementary-rhel8
            - name: cv-kickstart-rhel8.2
```

```
# task file
---
- name: push task file to create composite content view
  template:
    src: templates/composite-content-view.yml.j2
    dest: "{{ role_path }}/tasks/satellite-configuration/{{
organization.organization_name }}/content-views/{{
composite_content_view.name }}.yml"
  loop_control:
    loop_var: composite_content_view
  loop: "{{ organization.composite_content_views }}"
  delegate_to: "{{ delegate_host }}"

- name: include tasks to create composite content views
  include_tasks: "{{ role_path }}/tasks/satellite-configuration/{{
organization.organization_name }}/content-views/{{
composite_content_view.name }}.yml"
  loop_control:
    loop_var: composite_content_view
  loop: "{{ organization.composite_content_views }}"
```

Configuring Katello

Templated Ansible Task File

```
- name: create composite content view {{ composite_content_view.name }}
  theforeman.foreman.content_view:{% raw %}
    username: "{{ satellite.admin_username }}"
    password: "{{ satellite.admin_password }}"
    validate_certs: false
    server_url: "{{ satellite_url }}"
    organization: "{{ organization.organization_name }}"
{% endraw %}
    name: {{ composite_content_view.name }}
    composite: true
    components:
{% for component_content_view in composite_content_view.component_content_views %}
      - content_view: {{ component_content_view.name }}
{% if component_content_view.content_view_version is defined %}
        content_view_version: {{ component_content_view.content_view_version }}
{% else %}
        latest: true
{% endif %}
{% endfor %}
  delegate_to: localhost
  register: publish_content_view

- name: include tasks to publish new version of {{ composite_content_view.name }}
  include_tasks: "{{ role_path }}/tasks/publish-content-view.yml"
  vars:
    content_view: {{ composite_content_view.name }}
  when:
    - publish_content_view is defined
    - publish_content_view.changed
```

Configuring Katello

Content Views

```
# vars file
---
satellite:
  katello:
    - organization_name: org_1
     composite_content_views:
        - name: composite-rhel8
          description: rhel 8 packages
          auto_publish: false
          component_content_views:
            - name: cv-server-rhel8
            - name: cv-ansible-rhel8
            - name: cv-appstream-rhel8
            - name: cv-supplementary-rhel8
            - name: cv-kickstart-rhel8.2
```

```
# task file
---
- name: promote version 1.0 of a composite content view
  theforeman.foreman.content_view_version:
    username: "{{ satellite.admin_username }}"
    password: "{{ satellite.admin_password }}"
    validate_certs: false
    server_url: "{{ satellite_url }}"
    organization: "{{ organization.organization_name }}"
    content_view: "{{ composite_content_view.name }}"
    current_lifecycle_environment: Library
    lifecycle_environments: "{{ organization |
json_query('lifecycle_environments[*].name') | list }}"
    force_promote: true
  when:
    - composite_content_view.latest_version == '1.0'
    - composite_content_view | json_query('environments[*].name') |
join() == 'Library'
  loop_control:
    loop_var: composite_content_view
  loop: "{{ content_view_info.resources }}"
  delegate_to: "{{ delegate_host }}"
```

Configuring Katello

Content Views

```
# vars file
---
satellite:
  katello:
    - organization_name: org_1
      activation_keys:
        - name: ak-virtual
        - name: ak-epel8
          subscriptions:
            - subscription_name: Extra Packages for
Enterprise Linux 8
        - name: ak-rhel8-prod
          lifecycle_environment: prod
          content_view: composite-rhel8
          release_version: 8
```

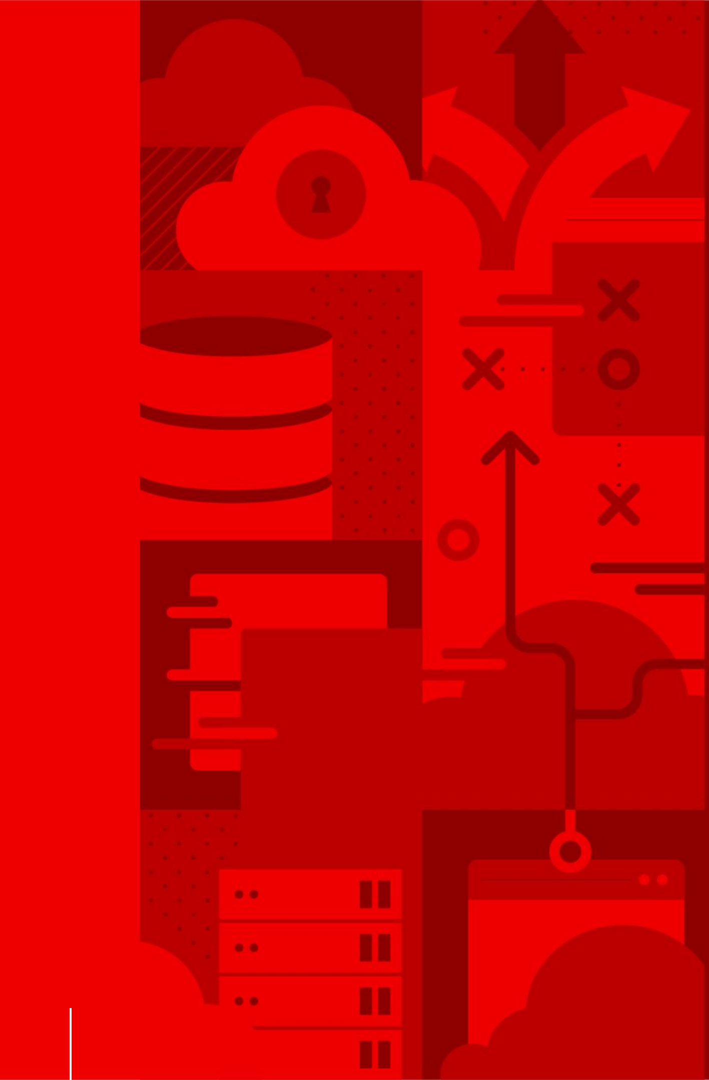
```
# task file
---
- name: push task file to create activation key
  template:
    src: templates/activation-key.yml.j2
    dest: "{{ role_path }}/tasks/satellite-configuration/{{
organization.organization_name }}/activation-keys/{{
activation_key.name }}.yml"
  loop_control:
    loop_var: activation_key
  loop: "{{ organization.activation_keys }}"
  delegate_to: "{{ delegate_host }}"

- name: include tasks to create activation keys
  include_tasks: "{{ role_path }}/tasks/satellite-configuration/{{
organization.organization_name }}/activation-keys/{{
activation_key.name }}.yml"
  loop_control:
    loop_var: activation_key
  loop: "{{ organization.activation_keys }}"
```

Configuring Katello

Templated Ansible Task File

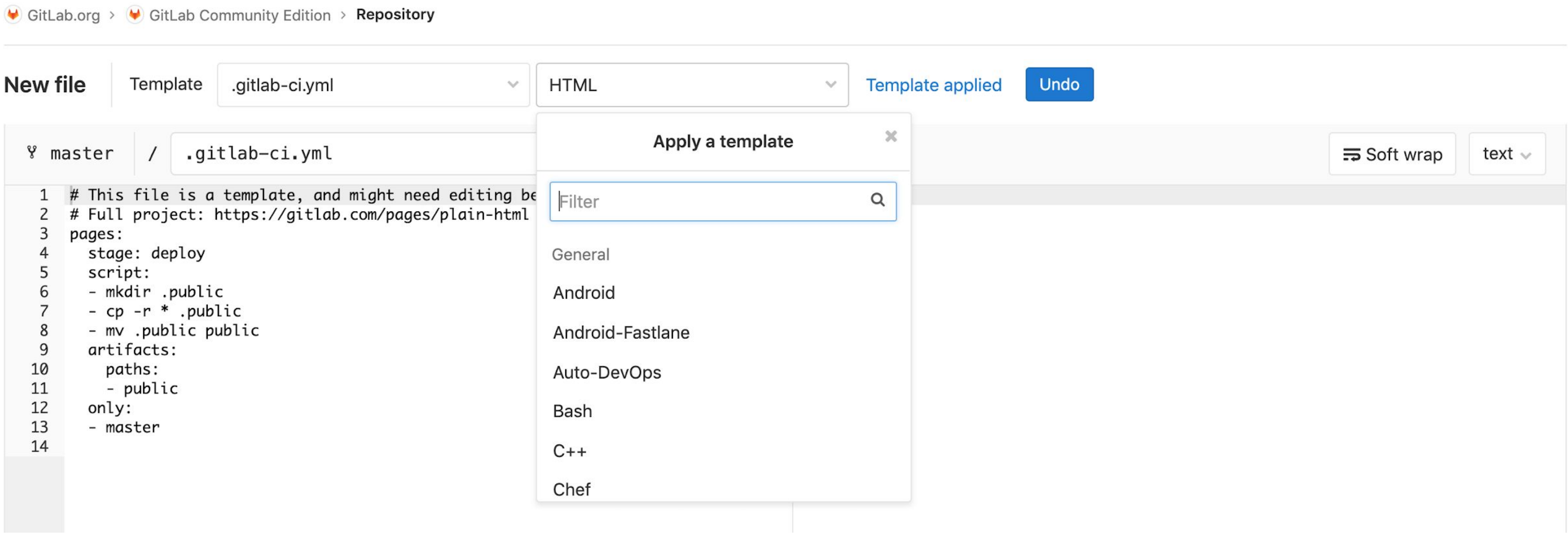
```
- name: create activation key {{ activation_key.name }}
  theforeman.foreman.activation_key:{% raw %}
    username: "{{ satellite.admin_username }}"
    password: "{{ satellite.admin_password }}"
    validate_certs: false
    server_url: "{{ satellite_url }}"
    organization: "{{ organization.organization_name }}"
{% endraw %}
    name: {{ activation_key.name }}
    lifecycle_environment: {{ activation_key.lifecycle_environment | default('Library') }}
    content_view: {{ activation_key.content_view | default('Default Organization View') }}
{% if activation_key.host_collections is defined %}
    host_collections:
{% for host_collection in activation_key.host_collections %}
      - {{ host_collection.name }}
{% endfor %}
{% endif %}
{% if activation_key.subscriptions is defined %}
    subscriptions:
{% for subscription in activation_key.subscriptions %}
{% if subscription.subscription_name is defined %}
      - name: "{{ subscription.subscription_name }}"
{% endif %}
{% if subscription.pool_id is defined %}
      - pool_id: {{ subscription_id }}
{% endif %}
{% endfor %}
{% endif %}
...
```



Bringing It All Together

Gitlab CI/CD

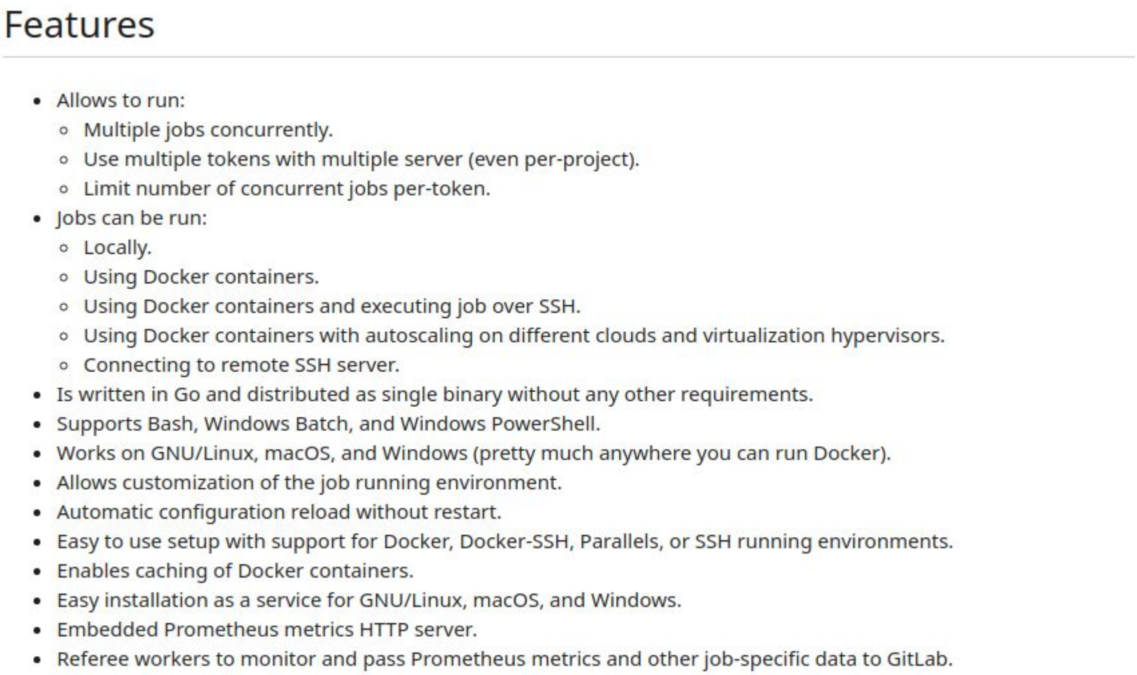
https://docs.gitlab.com/ce/ci/



```
# This file is a template, and might need editing be
# Full project: https://gitlab.com/pages/plain-html
pages:
  stage: deploy
  script:
  - mkdir .public
  - cp -r * .public
  - mv .public public
  artifacts:
    paths:
    - public
  only:
  - master
```

Gitlab Runner

https://docs.gitlab.com/runner/

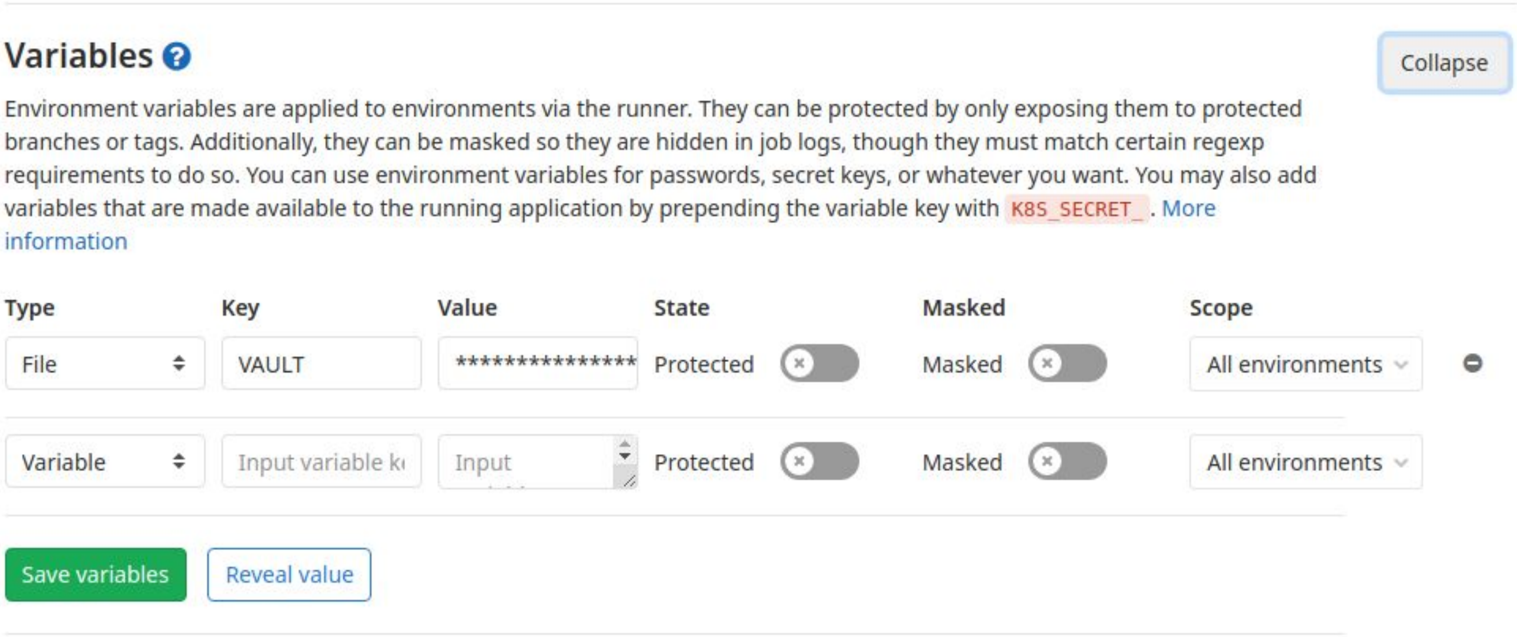


Features

- Allows to run:
  - Multiple jobs concurrently.
  - Use multiple tokens with multiple server (even per-project).
  - Limit number of concurrent jobs per-token.
- Jobs can be run:
  - Locally.
  - Using Docker containers.
  - Using Docker containers and executing job over SSH.
  - Using Docker containers with autoscaling on different clouds and virtualization hypervisors.
  - Connecting to remote SSH server.
- Is written in Go and distributed as single binary without any other requirements.
- Supports Bash, Windows Batch, and Windows PowerShell.
- Works on GNU/Linux, macOS, and Windows (pretty much anywhere you can run Docker).
- Allows customization of the job running environment.
- Automatic configuration reload without restart.
- Easy to use setup with support for Docker, Docker-SSH, Parallels, or SSH running environments.
- Enables caching of Docker containers.
- Easy installation as a service for GNU/Linux, macOS, and Windows.
- Embedded Prometheus metrics HTTP server.
- Referee workers to monitor and pass Prometheus metrics and other job-specific data to GitLab.

Gitlab Secrets

https://docs.gitlab.com/charts/installation/secrets.html

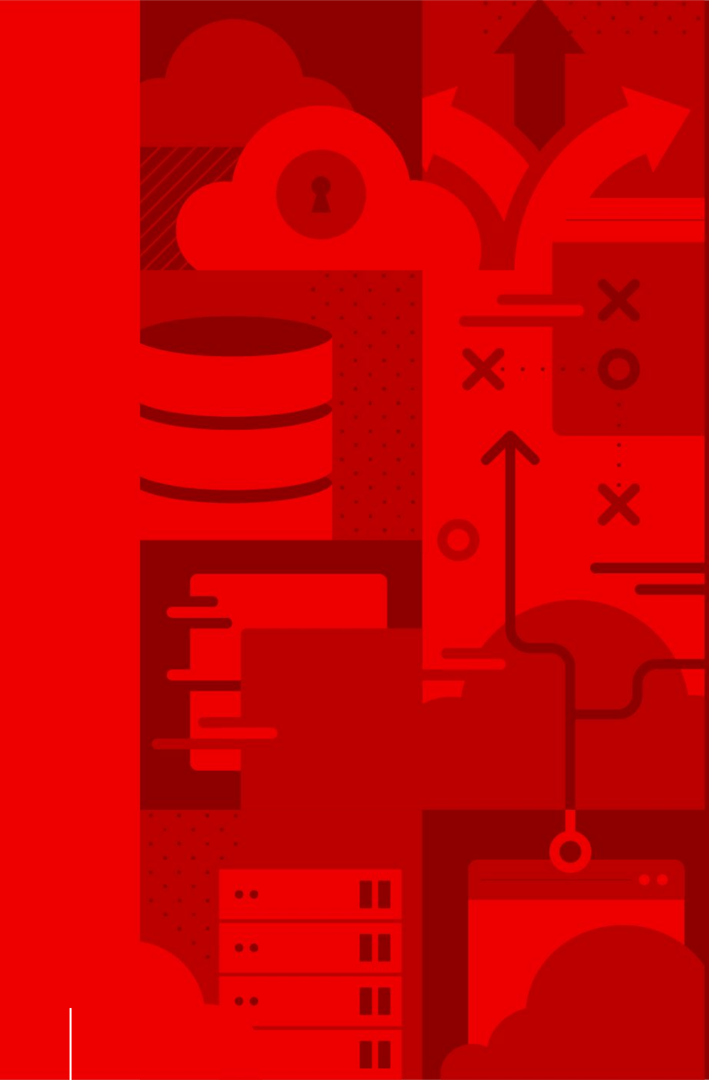


Variables

Environment variables are applied to environments via the runner. They can be protected by only exposing them to protected branches or tags. Additionally, they can be masked so they are hidden in job logs, though they must match certain regexp requirements to do so. You can use environment variables for passwords, secret keys, or whatever you want. You may also add variables that are made available to the running application by prepending the variable key with `K8S_SECRET_`. More information

Type	Key	Value	State	Masked	Scope
File	VAULT	***************	Protected	Masked	All environments
Variable	Input variable k	Input	Protected	Masked	All environments

Demo



What’s Next

What’s Next

Josh’s Wishlist

- Get the code out there
    - Drive more collaboration
    - More testing (Avoid “well it works for me”)
    - More “eyes on” to identify bugs
    - Adapt to more use cases
- Continue to add more features
    - OpenSCAP?
    - Host collections
    - Host groups
- Have a more “official” distribution channel
    - More official branding?
    - Automation hub?
    - ...Hopefully down the road

Resources

Useful Links

- The Foreman Ansible Modules:
  - https://theforeman.org/plugins/foreman-ansible-modules/
  - https://github.com/theforeman/foreman-ansible-modules
- Josh's Collections:
  - https://galaxy.ansible.com/jjaswanson4/setup_rhel_for_satellite
  - https://galaxy.ansible.com/jjaswanson4/install_satellite
  - https://galaxy.ansible.com/jjaswanson4/configure_satellite
  - https://github.com/jjaswanson4/setup_rhel_for_satellite
  - https://github.com/jjaswanson4/install_satellite
  - https://github.com/jjaswanson4/configure_satellite

Thank you!

in linkedin.com/company/red-hat

▶ youtube.com/user/RedHatVideos

f facebook.com/redhatinc

twitter.com/RedHat