



The Linux & Open Source Company

# Director's Cut

A new role for Ansible in Foreman

Jan Bundesmann, Thorben Denzer

# Agenda

## 1 Prequel

## 2 Storyboard

- Execution Environments
- Content Management
- Lifecycle Management

## 3 Demo

## 4 Critical Review



Linux Platform  
Operations



Infrastructure  
Automation



Container  
Platforms &  
Cloud Solutions



DevOps



Cloud Native  
Solution



## orcharhino

- ▶ Vendor independent datacenter management
- ▶ Patch management
- ▶ Configuration management
- ▶ Pizza



Thorben  
Software Engineer @ ATIX AG



Jan  
Software Quality Engineer @ ATIX AG

# Agenda

## 1 Prequel

## 2 Storyboard

- Execution Environments
- Content Management
- Lifecycle Management

## 3 Demo

## 4 Critical Review

# Every story has its beginning

Our premise: What if we could...

1. use a new Ansible version than the one installed on Foreman?
2. avoid managing Ansible content on the terminal?
3. manage different versions of Ansible content?
4. focus on collections instead of roles?

Audience wishes + consultants' opinions + engineers' concerns

# Agenda

## 1 Prequel

## 2 Storyboard

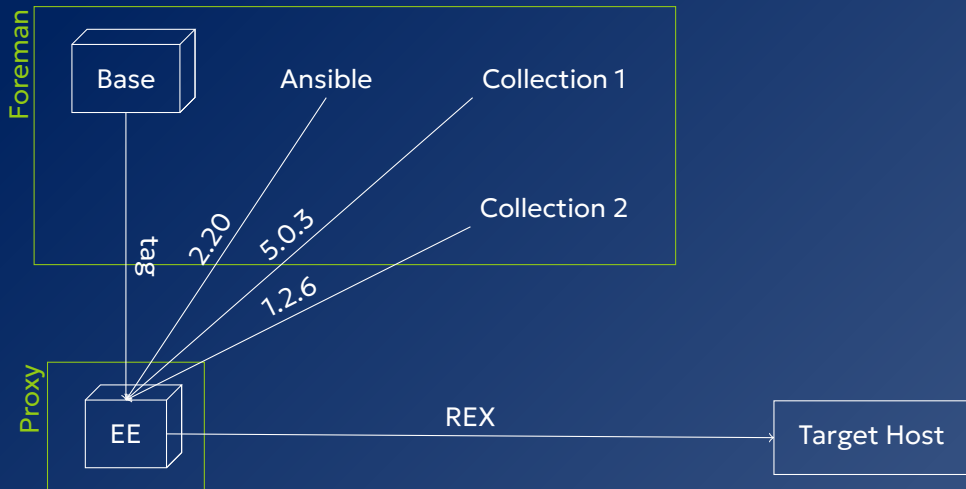
- Execution Environments
- Content Management
- Lifecycle Management

## 3 Demo

## 4 Critical Review



# Design of Ansible Director



# Agenda

## 1 Prequel

## 2 Storyboard

- Execution Environments
- Content Management
- Lifecycle Management

## 3 Demo

## 4 Critical Review

- ▶ Execution Environments encapsulate Ansible and dependencies
- ▶ Re-usable and well-defined
- ▶ ⇒ Checks the first box of our premise!



- ▶ Container image containing Ansible, content and dependencies
- ▶ Implications:
  - ▶ Host filesystem stays (relatively) untouched
  - ▶ Decouples Ansible version from the Smart Proxy Server
  - ▶ Content is isolated between runs
- ▶ `ansible-navigator` spawns a container from the execution environment image
- ▶ Ansible runs inside the container against selected hosts

# Automated container builds in the background

- ▶ Foreman keeps track of Execution Environment inputs and
- ▶ automatically rebuilds a base-image if inputs change
- ▶ → Environments fully reproducible, content-wise
- ▶ Actual execution environment is built prior to Ansible run

- ▶ (Base-) Execution Environments are defined in Foreman
- ▶ Definitions are translated and built on the Smart Proxy Server
- ▶ Built images are stored in Katello (pulp\_container) registry for re-usability

Execution environments



MyEE	
Base-Image URL	registry.fedoraproject.org/fedora:42
Ansible version ⓘ	2.20.0

# Agenda

## 1 Prequel

## 2 Storyboard

- Execution Environments
- Content Management
- Lifecycle Management

## 3 Demo

## 4 Critical Review

- ▶ Manage both Ansible Roles and Collections
- ▶ Multi-version support for Ansible content
- ▶ Add Ansible content to organizations
- ▶ Simplified Ansible content management:
  - ▶ Importing and deletion via web-UI
  - ▶ Import from Ansible Galaxy and Git

## Select source of Ansible content



### Ansible Galaxy

Import Ansible collections and roles from any source that implements the Ansible Galaxy API.

Supported sources include:

- **Ansible Galaxy** - <https://galaxy.ansible.com>
  - The official community repository
- **Private Galaxy servers** - Self-hosted instances based on [Pulp](#)
- **Content views** - [Content Views](#) - Repositories of type "ansible collection" which are published in a CV



### Git repository

Import Ansible collections and roles from a git repository.

**Note:** The repository must be sufficiently decorated with a `galaxy.yml` file.

Supported sources:

Any server implementing the git API



### Y(A)ML File

#### Bulk Import Ansible Content

Import multiple Ansible collections and roles efficiently using a Y(A)ML file. Supports a subset of the official ansible-galaxy requirements format.

Supported sources:

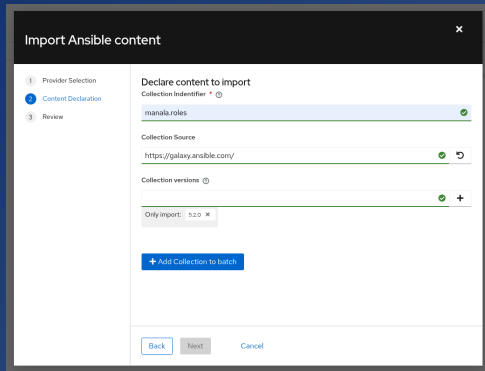
Ansible Galaxy API-enabled servers  
Git repositories



- ▶ Any galaxy-API compliant server
  - ▶ <https://galaxy.ansible.com>
  - ▶ Pulp with pulp\_ansible
  - ▶ Katello Products

Side note:


- ▶ Granular importing only possible for collections
- ▶ Roles are downloaded completely





Import Ansible content



1 Provider Selection  
2 Content Declaration  
3 Review


Declare content to import


Collection Identifier 


manala.roles 

Collection Source 

https://galaxy.ansible.com/  

Collection versions 

 +

Only import: 5.2.0 

+ Add Collection to batch

Back Next Cancel

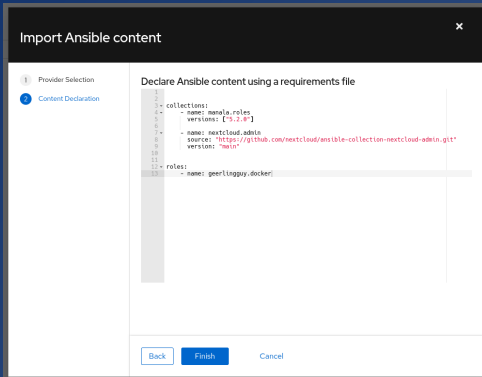
- ▶ Any (public) git-repository as source
- ▶ Collections must have necessary metadata
- ▶ Any valid git reference may be used
  - ▶ “Dynamic” references (branches/pulls) are treated differently

The screenshot shows a web-based interface for importing Ansible content. The dialog is titled "Import Ansible content" and has a close button (X) in the top right corner. On the left, there is a sidebar with three steps: "1 Provider Selection", "2 Content Declaration" (which is highlighted with a blue circle), and "3 Review". The main area is titled "Declare content to import" and contains the following elements:

- Repository URL:** A text input field containing "https://github.com/nextcloud/ansible-collection-nextcloud-admin.git". To the right of the input is a green checkmark icon and a button labeled "Inspect repository".
- Git reference:** A dropdown menu with a plus icon and a refresh icon. Below it are three tabs: "Commit / Manual Input" (selected), "Branch", and "Tag".
- Users:** A section with a header "Users" and a list of users.
- Git references:** A section with a header "Git references" and a list of references. The first reference is "main", which has a green checkmark and a plus icon to its right.
- Buttons:** At the bottom of the main area is a blue button labeled "+ Add Collection to batch". At the bottom of the dialog are three buttons: "Back", "Next" (disabled), and "Cancel".

# YAML to combine those

- ▶ Editor in UI
  - ▶ Functionality exposed via API endpoint
- ▶ Subset of requirements.yaml; source can be
  - ▶ Galaxy
  - ▶ Git repository
  - ▶ Versions can be declared
    - ▶ Ignored for Roles
    - ▶ Multiple Versions allowed for collections



Import Ansible content

1 Provider Selection

2 Content Declaration

Declare Ansible content using a requirements file

```
1 collections:
2   - name: ansible.builtin
3     version: ["9.2.0"]
4   - name: nextcloud.adrole
5     source: "https://github.com/nextcloud/ansible-collection-nextcloud-admin.git"
6     version: "main"
7 roles:
8   - name: geerlingguy.docker
```

Back Finish Cancel

# Agenda

## 1 Prequel

## 2 Storyboard

- Execution Environments
- Content Management
- Lifecycle Management

## 3 Demo

## 4 Critical Review

# Similar to Katello environments, yet fully decoupled

- ▶ Library: Default meta-environment containing everything
- ▶ LCEs bundle a subset of content + Execution Environment
- ▶ Paths define “flow” of content
- ▶ First environment in path can manage arbitrary content
- ▶ Later environments receive content by promotion
- ▶ Content is “snapshotted” on promotion
  - ▶ Snapshots...
    - ▶ ...are atomic and immutable
    - ▶ ...use hashes to identify differences
    - ▶ ...are created and destroyed automatically
- ▶ Special treatment of dynamic git references:
  - ▶ Dynamic references can be synced before use
  - ▶ References are pinned to latest commit on promotion

# Agenda

## 1 Prequel

## 2 Storyboard

- Execution Environments
- Content Management
- Lifecycle Management

## 3 Demo

## 4 Critical Review



# Agenda

## 1 Prequel

## 2 Storyboard

- Execution Environments
- Content Management
- Lifecycle Management

## 3 Demo

## 4 Critical Review



## 1. Unit tests for “services”

- ▶ Running locally
- ▶ Planned
  - ▶ GitHub workflows
  - ▶ `.gitlab-ci.yml` publicly available

## 2. Installability tests

## 3. Systems tests via robottelo

- ▶ API / CLI / UI functionality
- ▶ End-to-end tests how to act on target hosts

Criterion	foreman_ansible	foreman_ansible_director
Maturity	well established	newcomer
Requirements	foreman	katello, container runtime
Ansible content	roles	roles, collections
Content management	manually	automatic
Versions	as Foreman Proxy	as you like
Organizations	same content	distinct content
Lifecycles	in theory	inherent feature

# Achievements and future objectives

- ▶ Add authentication / secret management
- ▶ Playbooks
- ▶ Address hosts in multiple subnets
- ▶ Increase test coverage
- ▶ Support for offline installations
- ▶ (Rollbacks)
- ▶ Production ready approx. Q1 2026

- ▶ 16k lines of code
  - ▶ 9000 lines of TypeScript
  - ▶ 7000 lines of Ruby
- ▶ Planned end of 2024; In dev since 2025-01-01
- ▶ ~40 DynFlow actions
- ▶ 2 Repositories
  - ▶ [https://github.com/ATIX-AG/foreman\\_ansible\\_director/](https://github.com/ATIX-AG/foreman_ansible_director/)
  - ▶ [https://github.com/ATIX-AG/smart\\_proxy\\_ansible\\_director/](https://github.com/ATIX-AG/smart_proxy_ansible_director/)