



DWP Digital and AAP As Code

Rethinking IAC in Isolated AAP Environments

Tom Page
Senior Consultant
Red Hat

Mike Healey
Senior Devops Engineer
DWP

Steve Paynter
Automation Engineer
DWP

%whoami



Tom Page
Senior Consultant, Red Hat



Mike Healey
Senior DevOps Engineer, DWP



Steve Paynter
Automation Engineer, DWP



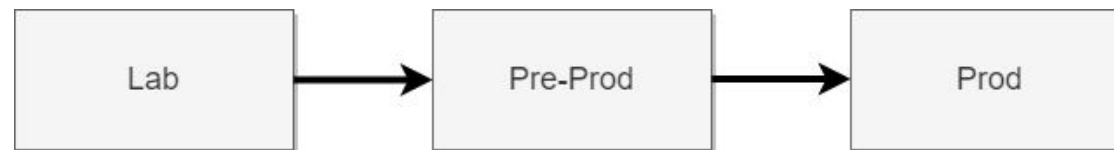
DWP HCS Automation Squad

- Part of Hybrid Cloud Services at DWP Digital
- We support the on-premise instances of AAP, Artifactory and Gitlab
- Offer AAP as a shared service
- We manage all vendor support, licensing and upgrades
- Allows teams to concentrate on their automation use cases
- Ensure a consistent, regularly updated platform for all users
- Manage formalised onboarding of PDUs (business units) into AAP

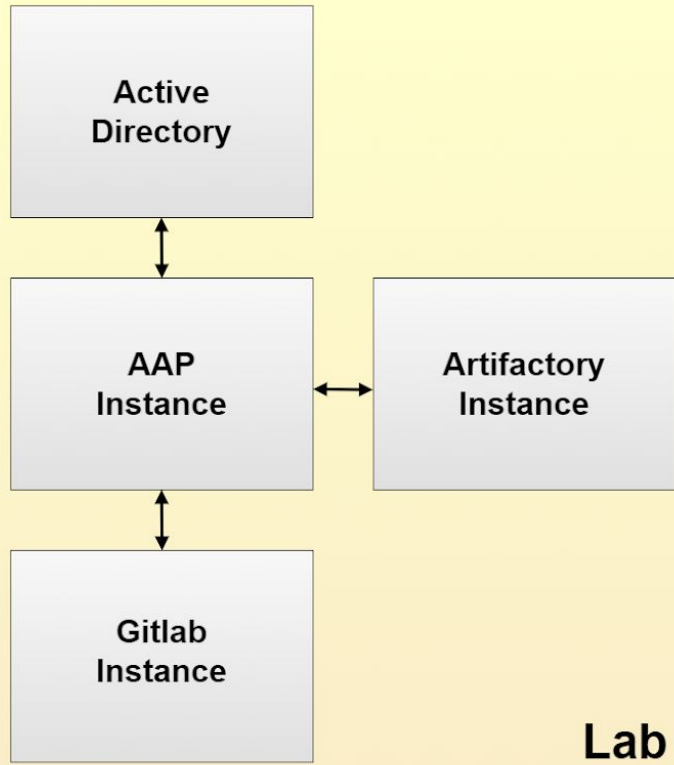


The Environment

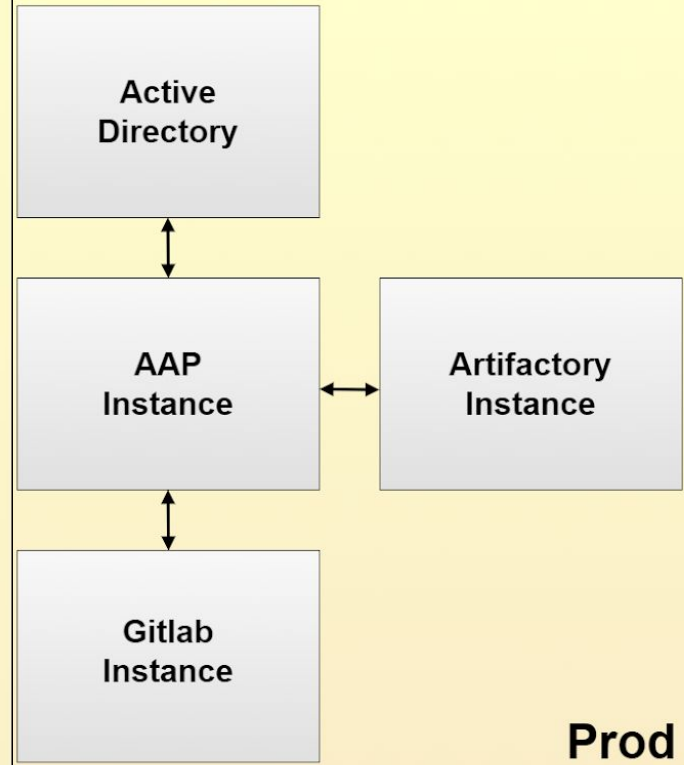
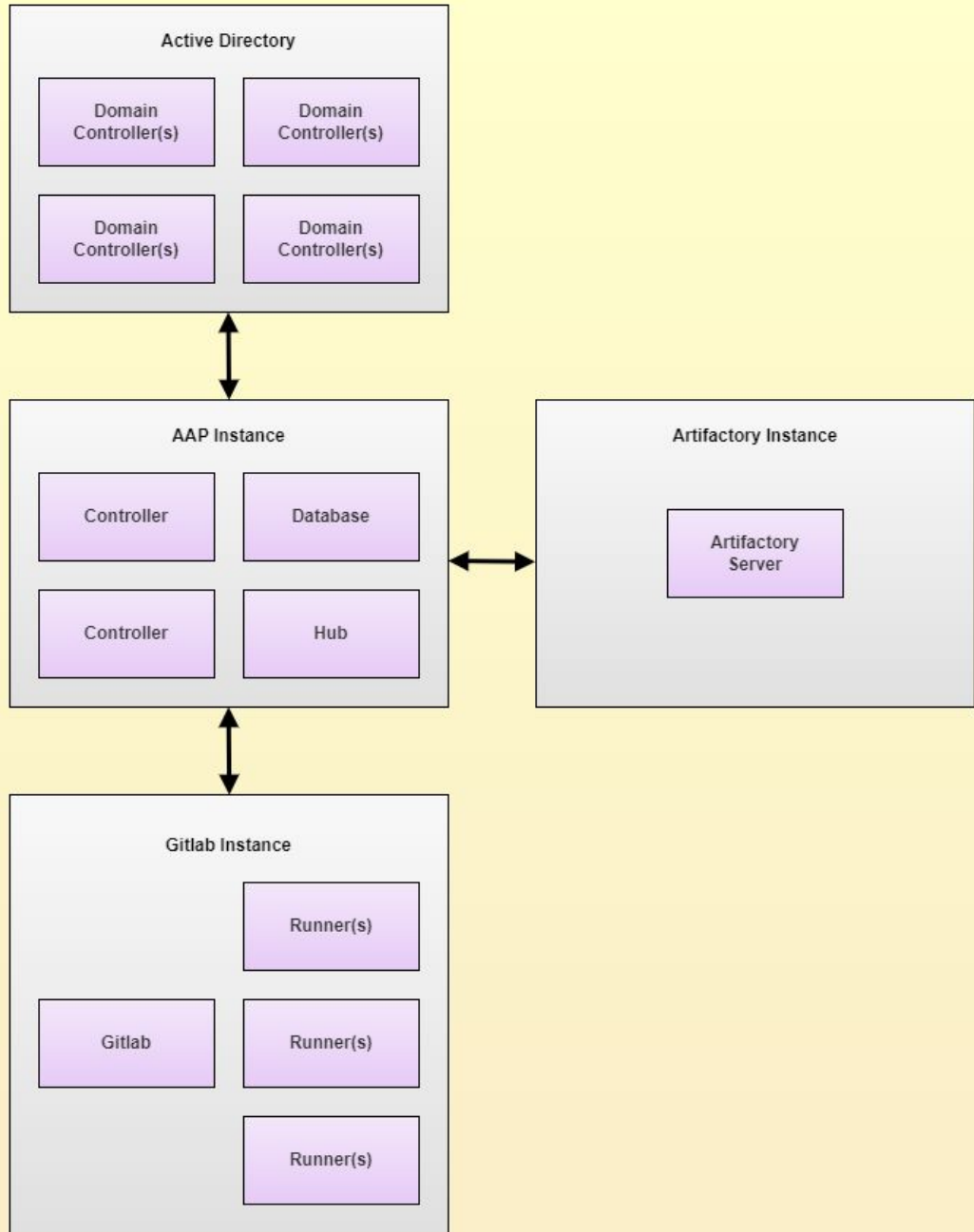
- We support three environments: lab, pre-production and production
- The lab has Internet access, but pre-prod and prod do not
- Code is developed and tested in the lab before being pushed to the other environments
- Each environment has an identical deployment of AAP, Gitlab and Artifactory
- The environments are on separate, physically segregated networks.



Re-Thinking IAC in Isolated Environments



Lab



Prod



Challenges of PDU Onboarding

- Each PDU has their own org, projects and repos within AAP and Gitlab
- Each org has RBAC AD groups and service accounts
- Each project needs a Gitkey creating
- Addition of any 'shift left' requests the PDU may have also require AD groups creating
- Steps must be replicated exactly for each PDU in each environment they wish to utilise

What is Controller Configuration as Code?

“Config as Code (CaC) separates configuration from the application code”

What?

- Describing the state of Controller infrastructure configuration
- Structured format
- Automation tooling can process description and configure

Why?

- Increase speed of deployment
- Repeatability
- Error Reduction
- Consistency across environments
- Prevention of configuration drift

Ansible Validated Content

Introducing the 'infra' namespace of collections



infra.controller_configuration

A collection of roles to manage Ansible Controller wrapping the ansible.controller modules



infra.ah_configuration

Collection of modules and roles to configure Private Automation Hub



infra.aap_utilities

Ansible Collection for automated deployment of AAP and other objects for general use



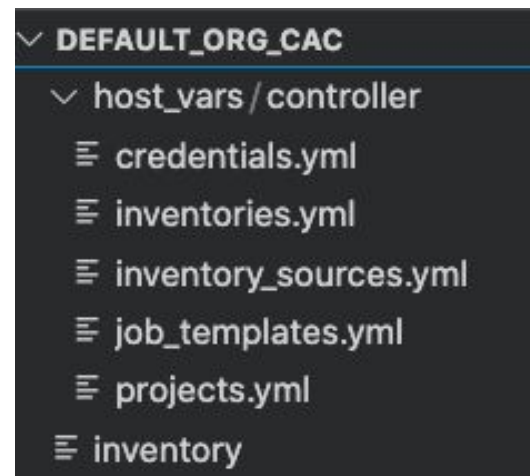
infra.ee_utilities

Collection including roles and tools which can be useful for managing Ansible Execution Environments.

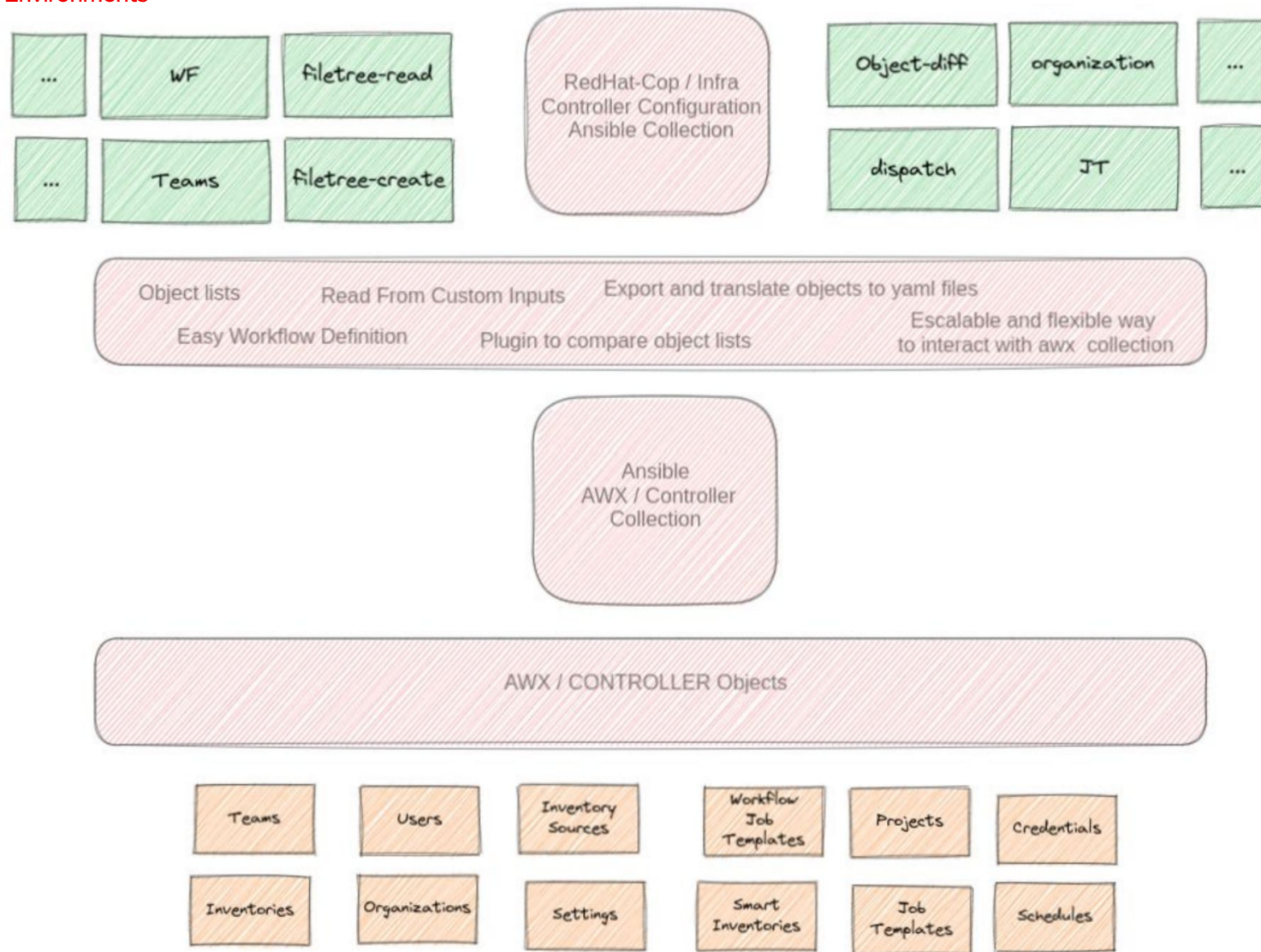
infra.controller_configuration

- Enhances the modules from the certified *ansible.controller* collection
- Adds structure to how to define objects at scale
- Offers roles for each object type
- Additional 'helper' roles as opinionated overall procedure

```
---  
controller_projects:  
  - name: Demo project  
    organization: Default  
    scm_branch: master  
    scm_type: git  
    scm_update_on_launch: true  
    scm_url: https://github.com/org/demo.git
```

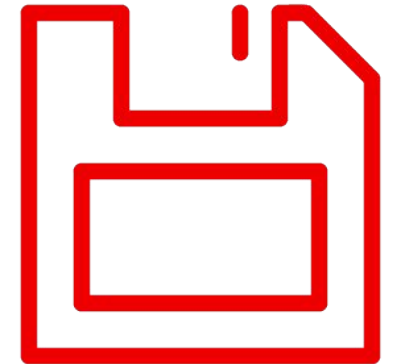


Re-Thinking IAC in Isolated Environments



Using Config as Code for DR

- By defining all the Controller objects as code we can restore the system
- Blank installation of system should be captured as code anyway
- Everything else is captured as code and can be recovered



Caveats:

- Can't back up any actions that have been taken on Controller (e.g. job launches and logs)
- May not want to store credentials as code (even encrypted)
- Need some way to kick off automation into Controller. Perhaps a manually run seed job?

Benefits of IAC PDU Onboarding

- Ensures each PDU gets the same, consistent experience using the products we manage and all essential functionality is already in place
- Deployment across all three environments is automated
- Process is iterative and granular, only changes and additions to an inventory file are performed
- A web-hook from Gitlab to AAP triggers the org building template automatically on approval of merge request

Why IAC as Code?

- Use the experience within the team to develop processes to maintain and migrate code across the three environments
- Ensures consistent build standards in each environment
- Minimise toil when developing new functionality
- Use of labels can restrict deployments to specific environments
- Use of environmental variables removes unpicking associated with hard coding
- Easy to onboard an entirely new environment if we need to

Standards

- Resource Naming Standards

```
CRD_   # Credential
CCRD_  # Custom Credential
EE_    # Execution Environment
PRO_   # Project
INV_   # Inventory
ISRC_  # Inventory Source
TEM_   # Job Template
WOR_   # Workflow
```

- No Whitespace , Use Underscores
- No Special Characters
- Create Standard Repository
- Align All Environments



Our Development Process

- Develop in POC
- Curated collections and REST API
- Manually create initial Job Template
- Mandate EE, name (TEM_POC), description & labels
- Squad Collaboration
- Mandate standards
- Document

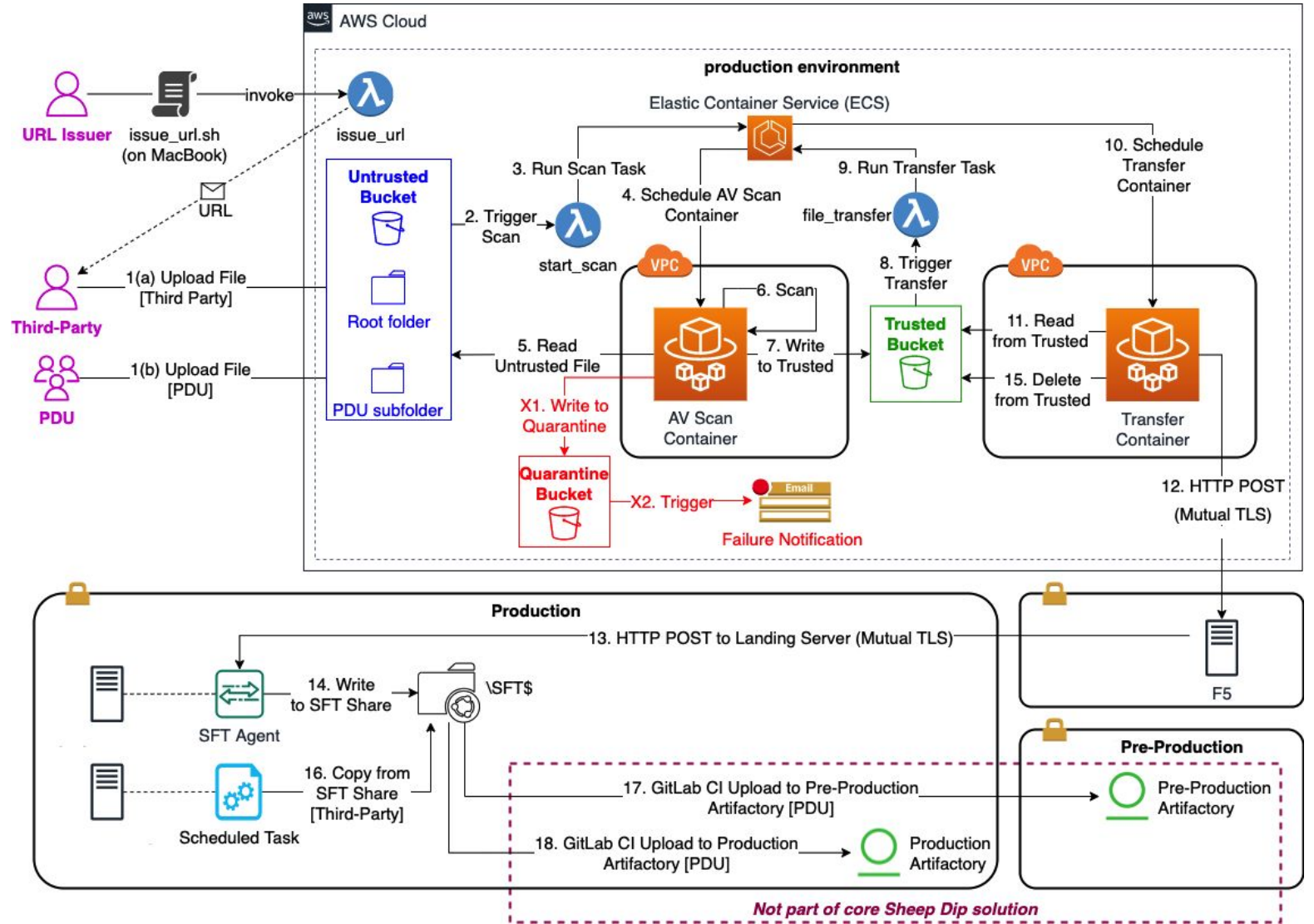
IAC Automation

- Job Template – Gather or Deploy
- Sanitise data
- Check for surveys/ schedules/Shift Left
- Renders IAC file with Jinja2
- Push file to Gitlab

Isolated Environment Deployment

Gitbundle and Sheepdip

- Gitbundle feature
- Entire History of Repository
- Set tag, automation ran.
- Bundled stored in Artifactory
- Transfer files from the lab using "Sheepdip"



DWP Digital and AAP As Code

Rethinking IAC in Isolated AAP Environments

DEMO

Q&A

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.



facebook.com/ansibleautomation



twitter.com/ansible



linkedin.com/company/ansible/



youtube.com/user/RedHatVideos