Red Hat
Ansible Automation Platform

Tech Deep Dive: Ansible Execution Environments

Portable, testable, container-based Ansible control nodes

Shane McDonald
Principal Software Engineer

Adam Miller
Senior Principal Software Engineer

What we'll discuss today

- What are Execution Environments
- Background Technical Information
- Challenges Ansible Content Creators Face Today
- How We're Making Ansible Content Creation Easier In The Future / Moving Forward

Present Day

Develop content

A developer writes Ansible content locally, and installs dependencies directly on their computer.

Account for Production

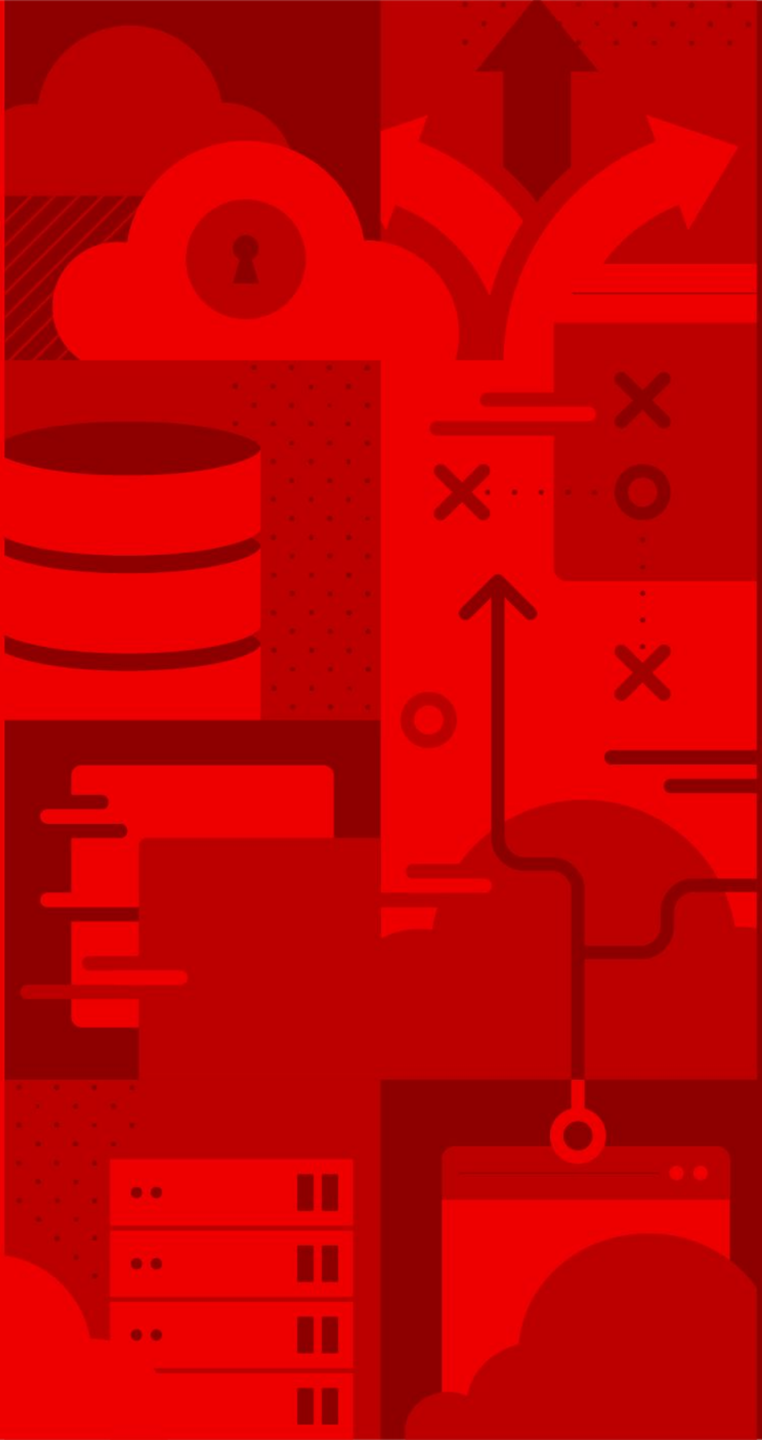
A help desk ticket may be necessary for installing software onto locked-down production systems.

Test

Test. Ideally in a staging environment, but this isn't always the case. This might require multiple iterations.

Repeat

This process needs to be repeated each time more content is added which requires new dependencies.



Looking Forward:

Ansible Automation with Container Technology

Overview of Collections

- Collections are a distribution format for Ansible Content that can include:
    - Modules
    - Roles
    - Plugins
        - Connection
        - Inventory
        - Become
        - Lookup
        - etc

Setting Context

Example playbook:

```
 $ cat test.yml
---
- hosts: localhost
  connection: local

  tasks:
    - name: Ensure the myapp Namespace exists.
      redhat.openshift.k8s:
        api_version: v1
        kind: Namespace
        name: myapp
        state: present
```

Setting Context

Example playbook:

```
 $ cat test.yml
---
- hosts: localhost
  connection: local

  tasks:
    - name: Ensure the myapp Namespace exists.
      redhat.openshift.k8s:
        api_version: v1
        kind: Namespace
        name: myapp
        state: present
```

Setting Context

Example playbook:

```
 $ cat test.yml
---
- hosts: localhost
  connection: local

  tasks:
    - name: Ensure the myapp Namespace exists.
      redhat.openshift.k8s:
        api_version: v1
        kind: Namespace
        name: myapp
        state: present
```

Container Recap

Using a pre-existing image from a registry

```
 $ podman pull registry.redhat.io/ubi8/ubi
Trying to pull registry.access.redhat.com/ubi8/ubi...
Getting image source signatures
Copying blob 77c58f19bd6e done
Copying blob 47db82df7f3f done
Copying config a1f8c96997 done
Writing manifest to image destination
Storing signatures
a1f8c969978652a6d1b2dfb265ae0c6c346da69000160cd3ecd5f619e26fa9f3

 $ podman run registry.redhat.io/ubi8/ubi whoami
root
```

Container Recap

Using a pre-existing image from a registry

```
 $ podman pull registry.redhat.io/ubi8/ubi
Trying to pull registry.access.redhat.com/ubi8/ubi...
Getting image source signatures
Copying blob 77c58f19bd6e done
Copying blob 47db82df7f3f done
Copying config a1f8c96997 done
Writing manifest to image destination
Storing signatures
a1f8c969978652a6d1b2dfb265ae0c6c346da69000160cd3ecd5f619e26fa9f3

 $ podman run registry.redhat.io/ubi8/ubi whoami
root
```

Container Recap

Building and running a custom image

```
 $ cat Containerfile
FROM registry.redhat.io/ubi8/ubi

RUN adduser appuser
USER appuser
```

Container Recap

Building and running a custom image

```
 $ cat Containerfile
FROM registry.redhat.io/ubi8/ubi

RUN adduser appuser
USER appuser
```

Container Recap

Building and running a custom image

```
 $ podman build -t my-custom-image .
STEP 1: FROM registry.redhat.io/ubi8/ubi
Getting image source signatures
Copying blob 47db82df7f3f done
Copying blob 77c58f19bd6e done
Copying config a1f8c96997 done
Writing manifest to image destination
Storing signatures
STEP 2: RUN adduser appuser
--> 159547becdd
STEP 3: USER appuser
STEP 4: COMMIT my-custom-image
--> cefe9da2417
cefe9da24171933eea4dadde4757398c71b3ed902c9c88e85dfb7cdaedc03133

 $ podman run my-custom-image whoami
appuser
```

Container Recap

Building and running a custom image

```
 $ podman build -t my-custom-image .
STEP 1: FROM registry.redhat.io/ubi8/ubi
Getting image source signatures
Copying blob 47db82df7f3f done
Copying blob 77c58f19bd6e done
Copying config a1f8c96997 done
Writing manifest to image destination
Storing signatures
STEP 2: RUN adduser appuser
--> 159547becdd
STEP 3: USER appuser
STEP 4: COMMIT my-custom-image
--> cefe9da2417
cefe9da24171933eea4dadde4757398c71b3ed902c9c88e85dfb7cdaedc03133

 $ podman run my-custom-image whoami
appuser
```

Running Ansible in Containers

First iteration: Container Image for running Ansible

```
 $ cat Containerfile
FROM registry.redhat.io/ubi8/ubi

RUN dnf install -y python3-pip
RUN pip3 install ansible

RUN mkdir -p /ansible # A location to bind-mount our playbooks
WORKDIR /ansible
```

Running Ansible in Containers

First iteration: Container Image for running Ansible

```
 $ cat Containerfile
FROM registry.redhat.io/ubi8/ubi

RUN dnf install -y python3-pip
RUN pip3 install ansible

RUN mkdir -p /ansible # A location to bind-mount our playbooks
WORKDIR /ansible
```

Running Ansible in Containers

First iteration: Container Image for running Ansible

```
 $ cat Containerfile
FROM registry.redhat.io/ubi8/ubi

RUN dnf install -y python3-pip
RUN pip3 install ansible

RUN mkdir -p /ansible # A location to bind-mount our playbooks
WORKDIR /ansible
```

Running Ansible in Containers

First iteration: Container Image for running Ansible

```
 $ cat Containerfile
FROM registry.redhat.io/ubi8/ubi

RUN dnf install -y python3-pip
RUN pip3 install ansible

RUN mkdir -p /ansible # A location to bind-mount our playbooks
WORKDIR /ansible
```

Running Ansible in Containers

First iteration: Container Image for running Ansible

```
 $ cat Containerfile
FROM registry.redhat.io/ubi8/ubi

RUN dnf install -y python3-pip
RUN pip3 install ansible

RUN mkdir -p /ansible # A location to bind-mount our playbooks
WORKDIR /ansible

 $ podman build --tag my-ansible-image .
<snip>
```

Running Ansible in Containers

First attempt

```
 $ podman run -ti -v $HOME/.kube/config:/root/.kube/config:Z \
                  -v $PWD:/ansible:Z \
                  --workdir=/ansible \
                  my-ansible-image \
                  ansible-playbook -i 'localhost,' test.yml -v

PLAY [localhost] ************************************************************

TASK [Ensure the myapp Namespace exists.] ***********************************

An exception occurred during task execution. To see the full traceback, use
-vvv. The error was: ModuleNotFoundError: No module named 'kubernetes'
...
```

Running Ansible in Containers

Second iteration: Container Image for running Ansible

```
 $ cat Containerfile
FROM registry.redhat.io/ubi8/ubi

RUN dnf install -y python3-pip
RUN pip3 install ansible  kubernetes

RUN mkdir -p /ansible # A location to bind-mount our playbooks
WORKDIR /ansible

 $ podman build --tag my-ansible-image .
<snip>
```

Running Ansible in Containers

Second iteration: Container Image for running Ansible

```
 $ cat Containerfile
FROM registry.redhat.io/ubi8/ubi

RUN dnf install -y python3-pip
RUN pip3 install ansible kubernetes

RUN mkdir -p /ansible # A location to bind-mount our playbooks
WORKDIR /ansible

 $ podman build --tag my-ansible-image .
<snip>
```

Running Ansible in Containers

First attempt

```
 $ podman run -ti -v $HOME/.kube/config:/root/.kube/config:Z \
                  -v $PWD:/ansible:Z \
                  --workdir=/ansible \
                  my-ansible-image \
                  ansible-playbook -i 'localhost,' test.yml -v

PLAY [localhost] ************************************************************

TASK [Ensure the myapp Namespace exists.] ***********************************

An exception occurred during task execution. To see the full traceback, use
-vvv. The error was: ModuleNotFoundError: No module named 'openshift'
...
```

Running Ansible in Containers

Second iteration: Container Image for running Ansible

```
 $ cat Containerfile
FROM registry.redhat.io/ubi8/ubi

RUN dnf install -y python3-pip
RUN pip3 install ansible kubernetes  openshift

RUN mkdir -p /ansible # A location to bind-mount our playbooks
WORKDIR /ansible

 $ podman build --tag my-ansible-image .
<snip>
```

Running Ansible in Containers

Second iteration: Container Image for running Ansible

```
 $ cat Containerfile
FROM registry.redhat.io/ubi8/ubi

RUN dnf install -y python3-pip
RUN pip3 install ansible kubernetes  openshift

RUN mkdir -p /ansible # A location to bind-mount our playbooks
WORKDIR /ansible

 $ podman build --tag my-ansible-image .
<snip>
```

Running Ansible in Containers

A minimally functional example

```
 $ podman run -ti -v $HOME/.kube/config:/root/.kube/config:Z
                  -v $PWD:/ansible:Z \
                  --workdir=/ansible \
                  my-ansible-image \
                  ansible-playbook -i 'localhost,' test.yml

PLAY [localhost] ***************************************************************************************

TASK [Ensure the myapp Namespace exists.] **************************************************************
changed: [localhost]

PLAY RECAP
********************************************************************************************************
localhost        : ok=1    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

The Solution: Execution Environments

Develop content

A developer writes Ansible content locally, using container technology to create portable automation runtimes.

Share and Distribute

Containers enable developers to share pre-packaged environments that can be tested and promoted to production.

Accelerate Operations

Streamline development and deployment operations, by simplifying and automating antiquated processes.

General Workflow

- Build with Ansible Builder

General Workflow

- Build with Ansible Builder
- Test with Ansible Navigator

General Workflow

- Build with Ansible Builder
- Test with Ansible Runner
- Use in production
  - Spring release of Ansible Automation Platform will be able to use Execution Environments to run jobs in Ansible Tower

Ansible Builder

Ansible Builder is a CLI tool that aids in the creation of Execution Environments.

- Produces portable, self-contained environments for executing Ansible.
- Compatible with Podman and Docker

```
 $ cat execution-environment.yml
---
version: 1
dependencies:
  galaxy: requirements.yml
 $ cat requirements.yml
---
collections:
 - redhat.openshift
```

```
$ ansible-builder build --tag my-exec-env
...
STEP 9: COMMIT my-exec-env
--> 10338eb3b88
10338eb3b886a154c8307b969774cffdfb9a86bebf7a937b52bee297b8615aa2
Complete! The build context can be found at: ./context
```

```
$ podman run my-exec-env ansible-galaxy collection list

# /usr/share/ansible/collections/ansible_collections
Collection           Version
-------------------- -------
redhat.openshift     0.1.0
```

```
 $ podman run my-exec-env pip list installed | grep -e openshift -e kubernetes
kubernetes          11.0.0
openshift           0.11.2
```

```
$ ls -1 context
bindep_combined.txt
bindep_output.txt
Containerfile
introspect.py
requirements_combined.txt
requirements.yml
```

```
$ cat context/Containerfile
FROM quay.io/ansible/ansible-runner:devel

ADD requirements.yml /build/

RUN ansible-galaxy role install -r /build/requirements.yml \
    --roles-path /usr/share/ansible/roles
RUN ansible-galaxy collection install -r /build/requirements.yml \
    --collections-path /usr/share/ansible/collections

ADD bindep_output.txt /build/
RUN dnf -y install $(cat /build/bindep_output.txt)

ADD requirements_combined.txt /build/
RUN pip3 install --upgrade -r /build/requirements_combined.txt
```

```
$ cat context/Containerfile
FROM quay.io/ansible/ansible-runner:devel

ADD requirements.yml /build/

RUN ansible-galaxy role install -r /build/requirements.yml \
    --roles-path /usr/share/ansible/roles
RUN ansible-galaxy collection install -r /build/requirements.yml \
    --collections-path /usr/share/ansible/collections

ADD bindep_output.txt /build/
RUN dnf -y install $(cat /build/bindep_output.txt)

ADD requirements_combined.txt /build/
RUN pip3 install --upgrade -r /build/requirements_combined.txt
```

```
$ cat context/Containerfile
FROM quay.io/ansible/ansible-runner:devel

ADD requirements.yml /build/

RUN ansible-galaxy role install -r /build/requirements.yml \
    --roles-path /usr/share/ansible/roles
RUN ansible-galaxy collection install -r /build/requirements.yml \
    --collections-path /usr/share/ansible/collections

ADD bindep_output.txt /build/
RUN dnf -y install $(cat /build/bindep_output.txt)

ADD requirements_combined.txt /build/
RUN pip3 install --upgrade -r /build/requirements_combined.txt
```

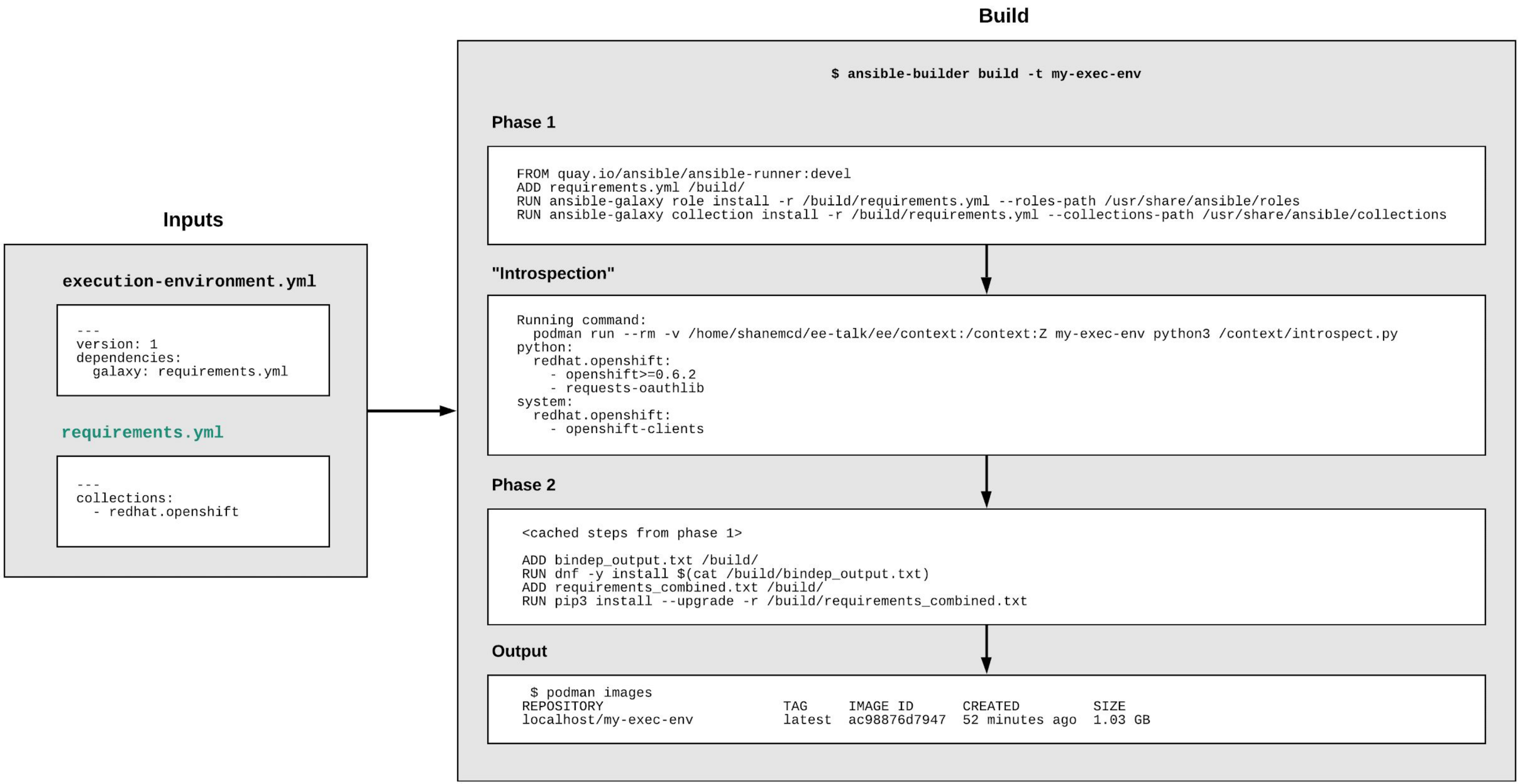
```
$ cat context/Containerfile
FROM quay.io/ansible/ansible-runner:devel

ADD requirements.yml /build/

RUN ansible-galaxy role install -r /build/requirements.yml \
    --roles-path /usr/share/ansible/roles
RUN ansible-galaxy collection install -r /build/requirements.yml \
    --collections-path /usr/share/ansible/collections

ADD bindep_output.txt /build/
RUN dnf -y install $(cat /build/bindep_output.txt)

ADD requirements_combined.txt /build/
RUN pip3 install --upgrade -r /build/requirements_combined.txt
```



Inputs

execution-environment.yml

```
---
version: 1
dependencies:
  galaxy: requirements.yml
```

requirements.yml

```
---
collections:
  - redhat.openshift
```

Build

```
$ ansible-builder build -t my-exec-env
```

Phase 1

```
FROM quay.io/ansible/ansible-runner:devel
ADD requirements.yml /build/
RUN ansible-galaxy role install -r /build/requirements.yml --roles-path /usr/share/ansible/roles
RUN ansible-galaxy collection install -r /build/requirements.yml --collections-path /usr/share/ansible/collections
```

"Introspection"

```
Running command:
  podman run --rm -v /home/shanemcd/ee-talk/ee/context:/context:Z my-exec-env python3 /context/introspect.py
python:
  redhat.openshift:
    - openshift>=0.6.2
    - requests-oauthlib
system:
  redhat.openshift:
    - openshift-clients
```

Phase 2

```
<cached steps from phase 1>

ADD bindep_output.txt /build/
RUN dnf -y install $(cat /build/bindep_output.txt)
ADD requirements_combined.txt /build/
RUN pip3 install --upgrade -r /build/requirements_combined.txt
```

Output

```
 $ podman images
REPOSITORY                        TAG     IMAGE ID      CREATED         SIZE
localhost/my-exec-env             latest  ac98876d7947  52 minutes ago  1.03 GB
```

Ansible Runner

How Ansible Automation Platform runs Ansible.

Python API

Enabling direct integration with various components within the Ansible Automation Platform, Red Hat Products, and Partner Integrations.

Command Line Utility

Interactive usage to mimic methodologies provided by Ansible Automation Platform as well as the CLI of Ansible Base.

Standard Execution

Provides a single unified and opinionated strategy for executing automation jobs.

Execution Environments

Provides the runtime for Ansible Execution Environments. Ansible Builder builds them, Ansible Runner runs them.

Documentation

https://ansible-builder.readthedocs.io

https://ansible-runner.readthedocs.io/en/latest/execution_environments.html

Asking Questions and Providing Feedback

GitHub:

- https://github.com/ansible/ansible-builder
- https://github.com/ansible/ansible-runner

Freenode:

- #ansible-builder
- #ansible-runner
- #ansible-awx
- #ansible

Mailing List: https://groups.google.com/g/awx-project