



How to create a new execution environment for Ansible Automation Platform?

🔒 SOLUTION VERIFIED - Updated January 16 2024 at 4:55 PM - English ▼

Environment

- Ansible Automation Platform 2.1+

Issue

- Need to create a new execution environment for the following use cases:
 - Configure kerberos authentication
 - Install ansible collections within the execution environment so they are readily available

Resolution

- The recommended way for building an execution environment for the Ansible Automation Platform would be through the `ansible-builder` tool. Please review the Ansible Builder Official Documentation in order to install it.

The steps listed in this article should be performed on a registered Red Hat Enterprise Linux (RHEL) 8 or 9 server. The server should have access to the appropriate Red Hat repositories in order to obtain any package dependencies that Ansible collections require. The reason for this is that the universal basic images (UBI) will include the host's repositories only if the above conditions are met. If the UBI image is being built on a non-Red Hat Enterprise Linux server or the server is not registered, then only the UBI repositories will be used. Just to give an example, the `redhat.rhv` ansible collection depends on the `qemu-img` package. This package is not available in the UBI repositories; it is only available in the appropriate appstream repository.

If any issues are faced with building an execution environment, then open a support case with Red Hat Support.

Step - Build Execution Environment

- Create a working directory

```
# mkdir -p builder/custom_ee && cd builder/custom_ee
```

- Create the `execution-environment.yml` file with the following contents

```
---
version: 1

build_arg_defaults:
  EE_BUILDER_IMAGE: 'registry.redhat.io/ansible-automation-platform-24/ansible-builder-rhel8:latest'
  EE_BASE_IMAGE: 'registry.redhat.io/ansible-automation-platform-24/ee-minimal-rhel8:latest'

ansible_config: 'ansible.cfg'

dependencies:
  galaxy: requirements.yml
  python: requirements.txt
  system: bindep.txt

additional_build_steps:
  prepend: []
  append: []
```

- Create the `ansible.cfg` file with the following contents

```
[defaults]
```

- Create the `requirements.yml` file with the following contents

```
---
collections: []
```

- Create the `requirements.txt` file
- Create the `bindep.txt` file
- Log into the registry

```
# podman login --log-level debug registry.redhat.io
```

- Use `ansible-builder` to add files to the `context/` directory

```
# ansible-builder create -v 3
```

- See **Optional Steps** section for additional changes that can be made before building the execution environment
- Re-run the above `ansible-builder create` command (see above) if any additional changes were made
- Build the execution environment

```
# podman build -f context/Containerfile context --no-cache -t  
localhost/custom_ee:latest
```

Wait for the build to complete. If any errors occur during the build process, make appropriate adjustments to the above files and re-run the create and build commands.

Step – Add Execution Environment to Automation Hub

- Once the execution environment has been successfully built, its recommended that it be added to Automation Hub.
 - Log into Automation Hub

```
# podman login --log-level debug $AUTOMATION_HUB_FQDN
```

- Tag the image using the fully qualified domain name of Automation Hub

```
# podman tag localhost/custom_ee $AUTOMATION_HUB_FQDN/custom_ee:latest
```

- Push the image to Automation Hub

```
# podman push --log-level debug $AUTOMATION_HUB_FQDN/custom_ee:latest
```

Step – Export Execution Environment and import into awx user's local storage

- This is an optional step if there is a need to add the newly created execution environment to the `awx` user's local image storage.

Note: These steps must be taken by `awx` user. For further information, see this KCS.

- Export the execution environment

```
# podman save -o custom_ee.tar localhost/custom_ee:latest
```

- Copy the tar archive into `/tmp` on all Automation Controllers and Execution Nodes
- Run the following command on each Automation Controller and Execution Node as the `awx` user

```
# podman load -i /tmp/custom_ee.tar
```

Step – Add Execution Environment to list within Automation Controller

- The following steps must be done in the UI for Automation Controller.
 - Go to `Administration -> Execution Environments`
 - Click `Add` button
 - Fill out the Name, Image (e.g `$AUTOMATION_HUB_FQDN/custom_ee:latest`), Pull, and Registry Credential fields in the form
 - Save

Optional Steps

Automation Hub Integration

- Need a way to configure the execution build to download collections from Automation Hub. Additional information on configuring `ansible-galaxy` can be found in the Ansible Galaxy User Guide and Automation Hub Getting Started Documentation.
- Add the following to the `ansible.cfg` file. Be sure to replace the `$AUTOMATION_HUB_FQDN` and `$AUTOMATION_HUB_TOKEN` variables with appropriate values.

```
[galaxy]
server_list = automation_hub, release_galaxy

[galaxy_server.automation_hub]
url=https://$AUTOMATION_HUB_FQDN/api/automation-hub/
token=$AUTOMATION_HUB_TOKEN

[galaxy_server.release_galaxy]
url=https://galaxy.ansible.com/
```

Note: Remove `release_galaxy` from the server list if you only want collections to be downloaded from Automation Hub.

Custom Certificate Authority

- Need a way to add internal custom certificate authority certificates to the execution environment.
 - Create a certificate directory in the `context/` directory

```
# mkdir -p context/certs
```

- Copy all certificates into the newly created directory
- Adjust the `execution-environment.yml` file to include the following commands (see example below)

```
additional_build_steps:
  append:
    - COPY certs/* /etc/pki/ca-trust/source/anchors/
    - RUN update-ca-trust extract
```

Kerberos Authentication

- Need a way to make kerberos authentication work from an execution environment.
 - Add the following to `bindep.txt`

```
krb5-libs [platform:rpm]
krb5-workstation [platform:rpm]
```

- Add a `krb5.conf` file to the `context/` directory

- Modify the `execution-environment.yml` file to include the following commands (see example below)

```
additional_build_steps:
  prepend:
    - COPY krb5.conf /etc/krb5.conf
    - RUN dnf install -y crypto-policies-scripts
    - RUN update-crypto-policies --set DEFAULT:AD-SUPPORT
```

AWS Libraries

- Need a way to add AWS libraries to the execution environment.
 - Add the following to `requirements.txt`

```
botocore>=1.19.0
boto3>=1.16.0
boto>=2.49.0
```

- Add the `community.aws` to to `requirements.yml` (see example below)

```
collections:
  - name: community.aws
```

Remove UBI Repositories

- Need a way to remove the UBI repositories because `dnf` should only be pulling from internal repositories.
 - Add `RUN rm -f /etc/yum.repos.d/ubi.repo` to `context/Containerfile` (see example below on where to add them)

```
...
RUN rm -f /etc/yum.repos.d/ubi.repo
RUN assemble
...
RUN rm -f /etc/yum.repos.d/ubi.repo
RUN /output/install-from-bindep && rm -rf /output/wheels
```

Product(s) Red Hat Ansible Automation Platform

Component podman