kyndryl

Kyndryl Resiliency Orchestration

Automated Discovery, Deployment, and Configuration Service (AD2C Service) Installation Guide

Version 8.4.9.0

kyndryl

DISCLAIMER

Kyndryl believes that the information in this publication is accurate as of its publication date. The information is subject to change without notice.

COPYRIGHT

TRADEMARK INFORMATION

Kyndryl and the Kyndryl logo are trademarks or registered trademarks of Kyndryl, Inc. in many jurisdictions worldwide. Other product and service names included herein may be trademarks of Kyndryl or other companies.

Not all offerings are available in every country in which Kyndryl operates. This program is licensed under the terms of the license agreement accompanying the Program. Please read the “Terms of Use” for this offering before using this program. By using the program, you agree to the terms.

kyndryl

Revision History

Document Version	Revision Date	Sections Updated	Pages Updated	Supported Product Version
1.0	March 2022	All sections	NA	8.2.9
1.1	June 2022	Updated the section heading from “Pre-installation Procedures” to “Pre-Installation (Internet-based or Online) Reference JIRA: RO-43258	Page 6	8.3.0
		Updated Step 2 in the section “Install Python3.x and jpype1” Reference JIRA: RO-43258	Page 8	8.3.0
		Added a new section “Pre-Installation (Internet-free or Offline) Reference JIRA: RO-43258	Page 8	8.3.0
		Updated the Product name from “DSS” to “AD2C Service” in all sections. Reference JIRA: RO-43370	Multiple sections	8.3.0
1.2	December 2022	Updated the Podman commands and included details regarding Source and Output in the section “Install AD2C Service on RHEL 8.0 using Podman”.	Page 16 and Page 17	8.3.6

kyndryl

		Reference JIRA: RO-51314		
Document Version	**Revision Date**	**Sections Updated**	**Pages Updated**	**Supported Product Version**
1.3	January 2023	Removed all references to the installation of Python, pip, and jpype1 on the RO Server machine.	Multiple sections	8.3.7
		Mentioning and detailing Podman options before mentioning or detailing Docker options		
		Introduced rigorous structure to the description of the installation process. Introduced comprehensive Installation Steps covering the two alternative scenarios of Podman and Docker. The Installation Steps have appropriate clickable “go to” references that give additional details.		
		Replaced RHEL 7.0 with RHEL 7.x and RHEL 8.0 with RHEL 8.x	13,17	
		Replaced references to Box locations with corresponding Sharepoint locations	16	
		Defined and used the Character Style MyMonospacedCharStyle		
8.3.11.0	May 2023	Modified Installation Steps	Multiple	8.3.x
8.4.0.0	June 2023	Updated Installation Steps	Multiple	8.3.x

kyndryl

8.4.0.0	June 2023	Updated example in section To download AD2C Image -> Online mode	11	
8.4.1.0	July 2023	Updated Installation Steps	Multiple	8.3.x
8.4.2.0	August 2023	Added Uninstallation of AD2C Steps	16	8.3.x
8.4.3.0	September 2023	Updated Pre-requisite section with AD2C Version Matrix	9	8.3.x
8.4.4.0	October 2023	Updated Hardware Requirements with Note.	11	8.4.x
8.4.7.0	January 2024	**Added the section** AD2C Custom Security Certificate Generation	12	8.4.x
8.4.8.0	February 2024	Updated the install steps with a Note.	19	8.4.x
8.4.9.0	March 2024	Install and Configure AD2C with a note	19	8.4.x

kyndryl

Contents

kyndryl

Introduction

The Kyndryl Automated Discovery, Deployment, and Configuration Service (AD2C Service) tool is an automated and stand-alone tool that is used in tandem with Kyndryl Resiliency Orchestration (RO) for automated discovery simplification. The AD2C Service tool overcomes the problem of manual discovery of the components, which is time-consuming and error-prone. Using the AD2C Service tool, you can reduce the time taken to implement the Kyndryl RO and the entire discovery process by following these steps:

- **Data Gathering**—Includes the data gathering from different application servers and database servers.
- **Data Validation**—Includes the validation of the collected data using custom rules.
- **Ingestion**—Includes ingestion of the gathered data into the Kyndryl RO.

For more information on the Discovery Simplification process refer to "*Kyndryl Resiliency Orchestration AD2C Service User Guide*".

Document Scope

This document includes the steps to install the Kyndryl AD2C Service tool using Podman containerization platforms. The document also provides details on the software and hardware requirements for launching this tool.

Glossary of Terms and Abbreviations

Term	Meaning
AD2C	Automated Discovery, Deployment, and Configuration
AD2C Server container host machine **Or**	The container host machine on which the AD2C Service container is being hosted

Term	Meaning
AD2C Service container host machine	
RHEL	Red Hat Enterprise Linux
RO	Resiliency Orchestration
RO Server machine	The machine on which the RO application is installed and running.

AD2C Service Support

For further assistance and more information on the AD2C Service solution, reach out to the **Kyndryl RO Support** team.

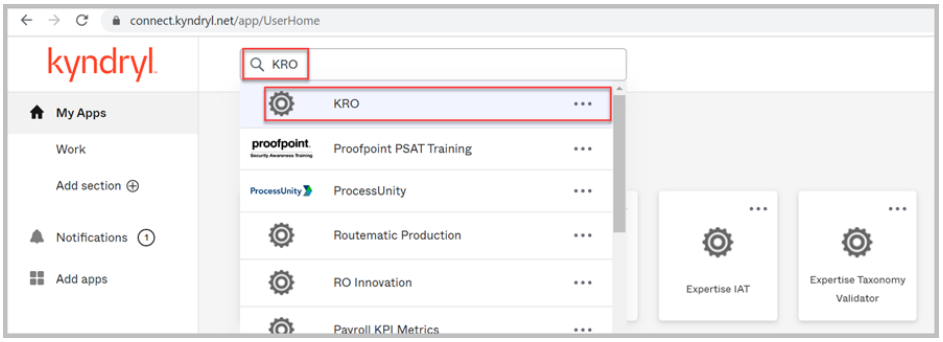
Prerequisites

- Download AD2C binaries from JFrog
- Other Prerequisites

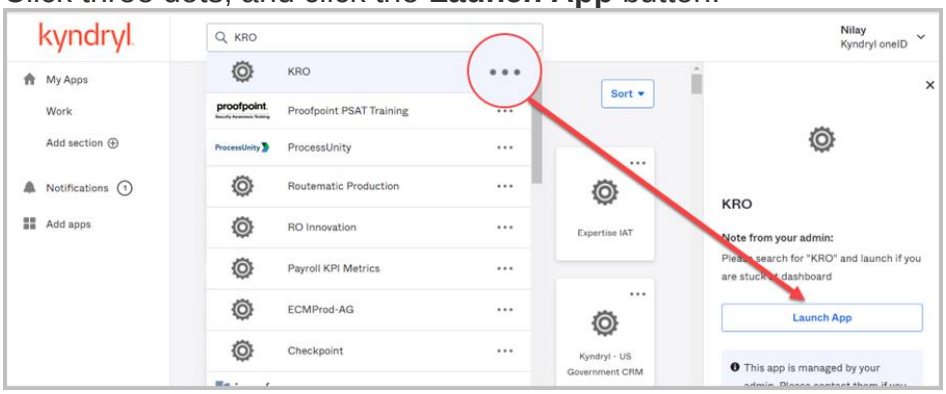
Download AD2C binaries from JFrog:

To log in to JFrog by follow these steps:

1. Open this URL: https://kyndrylresiliency.jfrog.io/ui/login/
2. Click **SAML SSO** button.
3. Click **Sign in with Okta FastPass**. Verify your identity. You need to go through the Two-Factor Authentication. After successful verification of your credentials, the **My Apps** portal appears.
4. In the **Search your apps** search bar, type **KRO**. KRO appears in the suggestion list:



5. Click three dots, and click the **Launch App** button:



6. Launch the URL https://kyndrylresiliency.jfrog.io/ui/repos/tree/General/ad2c
7. Expand the folder **ad2c** by clicking ">"
8. Select the folder based on the release month. For more information about AD2C version, refer to the table below.

AD2C Version Matrix

The following is the list of AD2C image referenced with RO version:

Release Month	RO Release	AD2C Image
Jun-23	8.4.0.0	ad2c_podman_image_8.4.0.0_1
May-23	8.3.11.0	ad2c_podman_image_release_8.3.11.0
Apr-23	8.3.10.0	ad2c-apr2023-release
Mar-23	8.3.9.0	ad2c-mar2023-kotakpatch ad2c-mar2023-release

kyndryl

Release Month	RO Release	AD2C Image
Feb-23	8.3.8.0	ad2c-feb2023-podman-release
Jan-23	8.3.7.0	ad2c-jan2023-podman-release:1.0 ad2c-jan2023-podman-release:2.0
Dec-22	8.3.6.0	ad2c-dec2022-podman-release:1.0

Note: For RO Versions 8.4.1.0 and later, the installer includes the AD2C image by default.

Other Prerequisites

The requirements and prerequisites to install AD2C service are as follows:

- The AD2C Service can be installed by running an installer that creates a podman container on a host machine.
- The host machine for the AD2C deployment needs to be an RHEL machine.
- The AD2C Service container can be hosted on the RO Server machine itself, or on a different machine.

Software Pre-requisites on the AD2C Server Machine

To use the AD2C Service, you need to install the following software:

Software	Supported Versions
RHEL	7.x, 8.x, 9.x
Podman	1.6.x is for RHEL 7.x 4.x.x or higher is for RHEL 8.x, 9.x **Note:** For RHEL 8.0 and higher versions, Podman is available by default.
conmon	2.0.x for RHEL 7.x 2.1.x or later for RHEL 8.x, 9.x **Note:** conmon should be available on the server before AD2C is installed.

kyndryl

Additional Requirements on the RO Server

The following additional requirements are applicable on the RO Server, to be able to use the AD2C Service:

1. The SSH Service must be enabled on the RO Server machine, to allow the AD2C Server machine to act as an SSH client.
2. On the RO Server machine, there needs to be a user whose UserID and Password shall be used for making an SSH connection from the AD2C Server to the RO Server. In this document, we refer to this as `<user>.`
3. For the `user>`:
   A. There must be a home directory.
   B. The home directory of `<user>` must contain a sub-directory called `javaclient`.
   C. The `<user>` must have read-write-execute permissions to the `javaclient` directory.
   D. The `<user>` must be a member of the following user groups in the RO Server machine:
      - `tomcatusergroup`.
      - `panacestomcatgroup`.

For example: If the user is a rouser, then the command to add the user to the above-mentioned groups is:

```
sudo usermod rouser -a -G
tomcatusergroup,panacestomcatgroup,panacesusergroup
```

Hardware Requirements

- 15 GB space must be available in the user home directory (`/home/<user>`) or the `path/location` where we are planning to install AD2C. The size of the AD2C installer can be around 1.2 GB.
- If AD2C Installation is performed on a different server other than RO then, Port 8444 and Port 22 should be opened from the RO server to the AD2C server.

- AD2C application uses 300x(x=1,2,3 etc.) Port. Ensure this port is opened on the AD2C server.

AD2C Custom Security Certificate Generation

AD2C container has already been configured with an SSL certificate. But if a user/ customer wants to add their own certificate, then follow these steps to generate the new certificate. There are following two methods to generate the certificate:

- Manual steps to create AD2C custom certificate
- Automated steps to create AD2C custom certificate

Manual steps to create AD2C custom certificate

Ref Link for generating cert/key - PEM files
https://dev.to/devland/how-to-generate-and-use-an-ssl-certificate-in-nodejs-2996

1. Generate a Private Key
openssl genrsa -out key.pem

2. Create a CSR (Certificate Signing Request)
openssl req -new -key key.pem -out csr.pem

The following Kyndryl details are provided by default in parameter. However, it is recommended to modify these details according to customer requirements for the self signed certificate.

Field Name	Description	Values
Country Name	(2-letter code)	US
State or Province Name	(full name)	New York
Locality Name	(eg, city) [Default City]	New York City
Organization Name	(eg, company) [Default Company Ltd]	Security and Resiliency Services
Organizational Unit Name	(eg, section)	Security and Resiliency Services
Common Name	(eg, your name or your server's hostname)	Kyndryl Inc
Email Address	your mail	Email address

kyndryl

3. Generate the SSL Certificate

openssl x509 -req -days 365 -in csr.pem -signkey key.pem -out cert.pem

The above command will prompt for a password, or the below command can be used where the password is passed in the command itself.

openssl x509 -req -days 365 -passin pass:<password> -in csr.pem -signkey key.pem -out cert.pem

4. Copy the key.pem, cert.pem to container path /app/src/middleware/server/sslcert

podman cp key.pem <ContainerID>:/home/ad2cuser/app/src/middleware/server/sslcert
podman cp cert.pem <ContainerID>:/home/ad2cuser/app/src/middleware/server/sslcert

After copying, change the names as key.pem and cert.pem in the *server/bin/www file*

To make the new certificates effective, restart the containers. If a new container needs

to be created using an image, follow these steps:

5. Create an image from the container

podman commit <container_id_or_name> <new_image_name>

6. Deploy the container from the image created

Replace the user path, and the image names as per the creation.

podman run --rm -i --security-opt=no-new-privileges -p 3000:3000 -d -v /home/rouser/ad2c/output:/home/ad2cuser/app/src/discovery-simplification/output:Z -v /home/rouser/ad2c/source:/home/ad2cuser/app/src/discovery-simplification/source:Z --user $uid:$gid --uidmap $uid:0:1 --uidmap 0:1:$uid --uidmap $(($uid+1)):$(($uid+1)):$(($subuidSize-$uid)) --gidmap $gid:0:1 --gidmap 0:1:$gid --gidmap $(($gid+1)):$(($gid+1)):$(($subgidSize-$gid)) localhost/new_image_name

kyndryl

Automated steps to create AD2C custom certificate

1. The script and the property files are placed under **source/scripts** folder.
2. Before running the script, ensure to fill the property file (ad2c_cert_params.txt) in this folder.
3. Country name, locality name , Org details , email etc has to be provided in the property file.
4. User has to set the Flagvalue as 1 , 2 or 3 in the property file based on the requirement as detailed in the table below:

Setting Flagvalue	Description
1	Generates the certificates only
2	Generates certificates, updates these certs to the container mentioned by the user and restarts the container.
3	Generates certs, updates these certs to the container mentioned by user, creates new image and then deploys a new container from the image so that the new container is also created with newly generated certs.

5. Enter the running container id in the property file, which is updated with new certs
6. Enter the sourcepath of this running container in the property file.
7. You can enter any name that will be used by the script as an image name during the creation process.
8. Provide the port number in the property file where new container will be deployed by the script.
9. After filling these details, ensure to run the script *Generate_ad2c_certificate.sh* as os user (say for example - rouser) and not as root user.

10. When the script execution starts, you will be asked to enter few more details like validity of cert, password for cert etc.
11. The logs of this script execution will be stored in the following path: */home/rouser/genAD2CCertLog.txt.*
12. When the script execution is completed, you can view the new certs and the container updated with the new cert.

Install and Configure AD2C

- **On the Kyndryl RO Server**, log in as a sudo-enabled user. Run the following commands to the `<user>` and provide the necessary privileges:

  ```
  sudo usermod <user> -a -G
  tomcatusergroup,panacestomcatgroup,panacesusergroup
  ```

- **On the RO Server, it is recommended to create javaclient subdirectory:**

  a. On the Kyndryl RO Server, login as <user>, and within the directory `/home/<user>,` create a sub-directory named `javaclient`.

  b. Provide 744 permissions to <*user*>, in this directory:

  ```
  chmod -R 744 /home/<rouser>/javaclient
  ```

  Note: If javaclient folder already exists, ensure you delete and create one with `<user>` (not with sudo user).

Note:

- AD2C installer auto detect the RHEL OS version of AD2C deployment VM and extract the OS compatible AD2C image as well as execute the container deployment steps specific to that OS version.
- AD2C deployment on RHEL 7.x VM, installer executes 7.x compatible steps and for VM having RHEL 8.x or later installer executes those OS-specific steps.

kyndryl

Steps to be followed as a non-root user

Follow all the below installation steps using non-root user (ex. hostuser). Ensure the installation directory(where AD2C needs to be installed) must have space greater than 15 GB. Also, ensure that no other AD2C container is running in the environment with the same port that you are planning to use.

1. Download the tar file to the AD2C server at a specific location.
2. Untar the tar file to get the `Ad2c_Installer.bin` and

   `Ad2c_Installer.properties` files.

```
[rouser@rheldevopsro installer_files]$ ll
total 844128
-rw-------. 1 rouser rouser 864378720 Jul 11 11:53 Ad2c_Installer.bin
-rw-------. 1 rouser rouser        79 Jul 11 11:53 Ad2c_Installer.properties
```

3. Edit the file AD2C_Installer.properties -

   INSTALLER_UI=silent

   (Currently supporting silent mode)

   `USER_INSTALL_DIR=/tmp/IA_Ad2c`

   (Specify the path where host user should have the permission to create the directory, ex /home/rouser/IA_Ad2c)

   `USER_INPUT_HOST_PORT=3000`

   (If 3000 port is used by other application in your env choose 300x port, x=1,2,3).

```
INSTALLER_UI=silent

USER_INSTALL_DIR=/home/rouser/IA_Ad2c

USER_INPUT_HOST_PORT=3000
```

4. Give 777 permissions to the binary files and execute below command -

kyndryl

```
./Ad2c_Installer.bin -f Ad2c_Installer.properties
```

```
[rouser@rheldevopsro installer_files]$ chmod 777 *
[rouser@rheldevopsro installer_files]$ ll
total 844128
-rwxrwxrwx. 1 rouser rouser 864378720 Jul 11 11:53 Ad2c_Installer.bin
-rwxrwxrwx. 1 rouser rouser        87 Jul 11 14:56 Ad2c_Installer.properties
```

```
[rouser@rheldevopsro installer_files]$ ./Ad2c_Installer.bin -f Ad2c_Installer.properties
Preparing to install
Extracting the JRE from the installer archive...
Unpacking the JRE...
Extracting the installation resources from the installer archive...
Configuring the installer for this system's environment...

Launching installer...

8. final log file name=/home/rouser/IA_Ad2c/Ad2c_Install_07_11_2023_15_39_23.log
installUnixJRE:  the source VM tar: /tmp/install.dir.3243627/Linux/resource/vm.tar
    exists = true
installUnixJRE:  the source VMRoot: /tmp/install.dir.3243627/Linux/resource/jre
    exists = true
installUnixJRE:  the dest VMRoot: /home/rouser/IA_Ad2c/jre
    exists = true
#
# INSTALLING VM: /home/rouser/IA_Ad2c/jre
#

installUnixJRE: Using new TAR technique...
Destination path for tar extraction (sans 'jre') =
    /home/rouser/IA_Ad2c

installUnixJRE: install shell script:
#!/bin/sh
echo "InstallUnixJRE Script begun..."
```

```
#!/bin/sh
echo "InstallUnixJRE Script begun..."
cd '/home/rouser/IA_Ad2c/jre'
rm -rf  *''
cd '/home/rouser/IA_Ad2c'
tar xf '/tmp/install.dir.3243627/Linux/resource/vm.tar'
chmod -R '775' '/home/rouser/IA_Ad2c/jre'
echo "...InstallUnixJRE Script complete."
##### SCRIPT END ############

XMLScriptWriter: No Installation Objects were skipped
Retrying Installables deferred in pass 0
Deferral retries done because:
There were no deferrals in the last pass.
Podman image load
Loaded image: localhost/ad2c_podman_image_8.4.0.0_152:latest
 --
Getting image source signatures
Copying blob sha256:7560feb73e7c7a4e077bfdeaf40ecc834b25968c5dec812da28d57b3631868ca
Copying blob sha256:e85c36266c834f75b181621c2af753854f92c534eec0a6931c5c115d584fc411
Copying blob sha256:fc74cadda37159fef0cad4876f43c1486cb4f6f8163248b022b568779185a14c
Copying blob sha256:d91b2949283ab74ad98179b5e52dca6c5ce9952f23038246cc0015dee43e2f23
Copying blob sha256:b273590db0483215e22c48aa8c0cc9d76952e4332f0ab5f6b28022e10ad0e6b3
Copying blob sha256:97baacbaa7da3cfb0282ab16a60656f40c38ccca72f5e6fa850bb59e2acdde69
Copying blob sha256:e715f2fc003a7afe486966020fc08835c260c115ecddaa332b4ed3ac5ca6daf7
Copying blob sha256:5bb6849f8f84e4b3c2826fb61e1610711ddb1736420132222a0314ee985dcdd4
Copying blob sha256:0cdc90331669c0d65b79c635f18541abf8fc9bf59ccd2db6a00835904dfd3bb5
Copying config sha256:823c45bc77c4974b6a45c39873d59c4f2d5e6a254e26f1b9bc10a1d35e49eb68
Writing manifest to image destination
Storing signatures
 ---
```

```
Copying config sha256:823c45bc77c4974b6a45c39873d59c4f2d5e6a254e26f1b9bc10a1d35e49eb68
Writing manifest to image destination
Storing signatures
 ---
imgId = localhost/ad2c_podman_image_8.4.0.0_152:latest



----------


-----
podman run --rm -i  --security-opt=no-new-privileges -p 3000:3000 -d -v /home/rouser/IA_Ad2c/output:/home/ad2cu
ser/app/src/discovery-simplification/output:Z -v /home/rouser/IA_Ad2c/source:/home/ad2cuser/app/src/discovery-s
implification/source:Z --user 3001:3001 --uidmap 3001:0:1 --uidmap 0:1:3001 --uidmap 3002:3002:62535 --gidmap 3
001:0:1 --gidmap 0:1:3001 --gidmap 3002:3002:62535 localhost/ad2c_podman_image_8.4.0.0_152:latest


==========podman run command==========
Podman contianer run 1e5eb59968c8ac2bd3b65526bd5483610c564f4857a559102314a94385973a1f

8. final log file name=/home/rouser/IA_Ad2c/Ad2c_Install_07_11_2023_15_39_23.log
```

5. When the installation is completed, check the status of the container using the command –

```
podman ps -a
```

```
[rouser@rheldevopsro installer_files]$ podman ps -a
CONTAINER ID  IMAGE                                          COMMAND     CREATED            STATUS
              PORTS                   NAMES
1e5eb59968c8  localhost/ad2c_podman_image_8.4.0.0_152:latest             About a minute ago  Up About a minute
 ago (healthy)  0.0.0.0:3000->3000/tcp  gallant_mahavira
```

6. Ensure the container should be up and running in healthy state. Now go to the installation directory path specified in the property file. Ensure that the source and output folders are available. The contents of the source/data directory are to be treated as sample data. Replace the contents with the actual data applicable to your environment and your requirements.

```
[padma@stln78ad2cro IA_Ad2c]$
[padma@stln78ad2cro IA_Ad2c]$
[padma@stln78ad2cro IA_Ad2c]$ ls
Ad2c_Install_01_05_2024_11_08_46.log  _Ad2c_installation  output  source
[padma@stln78ad2cro IA_Ad2c]$
[padma@stln78ad2cro IA_Ad2c]$
[padma@stln78ad2cro IA_Ad2c]$
[padma@stln78ad2cro IA_Ad2c]$
```

Source – The **source** directory is the location from which the AD2C Service accepts input data. It is also the place where the configuration settings are stored.

> **Note:** If you want to add a new file or modify an existing file inside the source folder, ensure that the file permission is 744 and the owner of that file is the same as the source folder.

Output – The **output** directory is the location where the output or results files are produced by the AD2C Service.

Note:

- After AD2C installation is successful, you must manually update $EAMSROOT, TOMCAT_HOME, RO IP, Panaces path, etc. in *runner.properties* file.
- Review the java_classpath.
- Verify the JSON library in RO and update the same in *runner.properties* file.

7. Access the AD2C Service from a browser

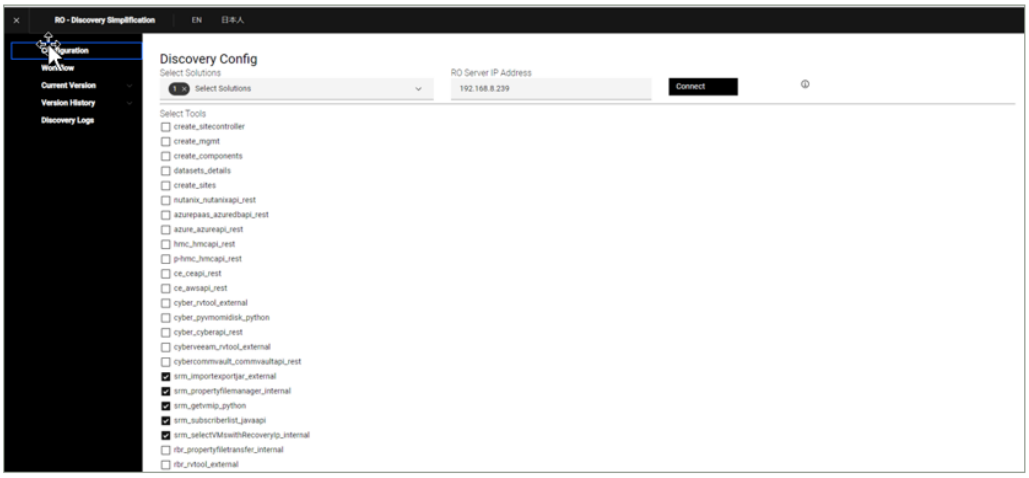
Once the container is up and running, you can access the AD2C Service UI on the browser by entering the AD2C host machine IP address and Port mentioned in the property file:

https://<HostMachineIPaddress:Port>/config

For example: https://101.4.6.7:3000/config

After the AD2C Service UI is loaded on the browser, you can navigate to different solutions for discovery.

Note: The AD2C Service will execute based on the data from the file runner.properties and Solution_Configuration.csv. These configuration files are located at the path <source>/config, where <source> refers to the source directory mentioned earlier in the Installation Steps**.**



Note:

-

Uninstallation of AD2C

To uninstall AD2C from the host server, follow the steps given below using the same user you used for installation:

1. Remove the AD2C container by running below commands-
    - Get the container id and image name by executing command –

podman ps -a

```
[rouser@rheldevopsro ~]$ podman ps -a
CONTAINER ID  IMAGE                                      COMMAND     CREATED      STATUS                  PORTS                   NAMES
7429c69d83af  localhost/ad2c_podman_image_8.4.2.0_1:latest           3 hours ago  Up 3 hours ago (healthy)  0.0.0.0:3000->3000/tcp  AD2C
[rouser@rheldevopsro ~]$
```

- Remove the container by executing –
  podman rm -f <container id>

```
[rouser@rheldevopsro ~]$ podman rm -f 7429c69d83af
WARN[0010] StopSignal SIGTERM failed to stop container AD2C in 10 seconds, resorting to SIGKILL
7429c69d83af39639b492183e27aaf5940bc7e4df7c7a0cea5fa672bf761e6d1
```

- Container should be deleted, validate again using – *podman ps -a*

2. Remove the AD2C image by running below commands –
   Note: Ensure to take backup of source folder files that we edited/modified, for future reference.
   - Get the image id corresponding to the image name that we get in the above command-
     podman images

```
[rouser@rheldevopsro ~]$ podman images
REPOSITORY                                TAG         IMAGE ID      CREATED       SIZE
localhost/ad2c_podman_image_8.4.2.0_1     latest      a93a7bf65bc6  3 days ago    885 MB
localhost/ad2c_podman_image_8.4.1.0_2     latest      0e6a78bfc9d4  3 weeks ago   884 MB
localhost/ad2c_podman_image_8.4.1.0_15    latest      8339edc57bc8  4 weeks ago   884 MB
```

- Remove the image using command –
  podman rmi <image id>

```
[rouser@rheldevopsro ~]$ podman rmi a93a7bf65bc6
Untagged: localhost/ad2c_podman_image_8.4.2.0_1:latest
Deleted: a93a7bf65bc61b8c3450bb00e3e1bfd4892560e11bb0d9d92ab784d16b734755
[rouser@rheldevopsro ~]$
```

3. Remove the source and output folders.