IBM Systems

Cloud Pak for Network Automation 2.1 Administration and Configuration

TN777 (Classroom)
ZN777 (Self-paced)

Course description

This course is intended to teach the skills that are needed to successfully install, configure, and use Cloud Pak for Network Automation (CP4NA) 2.1. This course is the second part of instructor-led or self-paced training in Cloud Pak for Network Automation and focuses on hands-on lab exercises to describe, deploy, and manage VOIP and telco components. The first portion of the course is TN776 that focuses on the fundamental principles, architecture, and terminology of CP4NA.

For information about other related courses, see the IBM Training website:

ibm.com/training

General information

Delivery method

Classroom or self-paced virtual classroom (SPVC)

Course level

ERC 1.0

Product and version

Cloud Pak for Network Automation 2.1

Audience

This course is intended for systems engineers, technical support professionals, system administrators, administrators, and deployers of CP4NA 2.1.

Learning objectives

After completing this course, you should be able to:

- Understand the installation process for CP4NA
- Edit assembly and resource descriptors
- Use the Lifecycle Manager to push project descriptors to the CP4NA server
- Instruct the CP4NA server to deploy assemblies and resources in a virtual environment such as Red Hat Openstack™ or Red Hat OpenShift
- Remove assemblies and resources from deployment
- Automate scaling of resources to meet traffic loads
- Automate testing to validate deployed resources

Prerequisites

- Completion of basic training on IBM Cloud Paks
- Completion of course TN776 – IBM Cloud Pak for Network Automation Foundations
- Proficiency in deployment of Red Hat OpenShift™ containers
- Understanding of telecommunications concepts
- Proficiency in command-line tools such as the vi editor in Linux and Red Hat OpenShift commands
- Ability to use VPN, ssh, and GitHub repository
- Ability to edit Red Hat OpenShift configuration files

Duration

2 days

Skill level

Intermediate

Notes

The following unit and exercise durations are estimates, and might not reflect every class experience. If the course is customized or abbreviated, the duration of unchanged units will probably increase.

Course agenda

Unit 1. Installing Cloud Pak for Network Automation 2.1
Duration: 1 hour and 30 minutes

Overview	This unit describes the procedure to install Cloud Pak for Network Automation by providing an example video and a step-by-step explanation.
Learning objectives	After completing this unit, you should be able to: • Describe the steps necessary to install Cloud Pak for Network Automation • Do an operator-based installation of CP4NA • Understand that automatic updates are not available for CP4NA 2.1 currently.

Unit 2. Introduction to course lab exercises
Duration: 1 hour

Overview	This unit describes the lab exercises in this course. It first explains the resource Lifecycle state model and the lab architecture. It then explains all the lab exercises you do in the course to give you an overview of what students accomplish during this course.
Learning objectives	After completing this unit, you should be able to: • Understand the lab architecture • Understand that descriptors are created on your client desktop • Know the utility that is used to push descriptors to the CP4NA server • Explain the difference between the managed-from and managed-to environments • Understand the progress of labs throughout the course • Explain the basic concepts of the Resource Lifecycle State Model • List some components of a basic VOIP service

Unit 3. Verify desktop tools
Duration: 30 minutes

Overview	This module introduces Exercise 1 in which students configure connections and verify their client desktop tools function correctly.
Learning objectives	After completing this unit, you should be able to: • Use the Lifecycle manager control tool (lmctl) to verify connection to CP4NA server • Access the CP4NA server GUI • Access the Red Hat Openstack GUI • List other tools on your desktop workstation • Create an SSH key pair to access OpenStack • Create a default deploymentLocation to ensure that your work does not conflict with the work of other students

Exercise 1. Verify your desktop tools
Duration: 30 minutes

Overview	In this exercise, students verify the function of their desktop tools. They configure an SSH key pair to communicate with Red Hat OpenStack. They also create a default deployment location that keeps their work separate from the efforts of other students.
Learning objectives	After completing this exercise, you should be able to: • Use the Lifecycle manager control tool (lmctl) to verify connection to CP4NA server • Access the CP4NA server GUI • Access the Red Hat Openstack GUI • List other tools on your desktop workstation • Create an SSH key pair to access OpenStack • Create a default deploymentLocation to ensure that your work does not conflict with the work of other students

Unit 4. Create an assembly project and descriptor
Duration: 15 minutes

Overview	This unit introduces Exercise 2 in which students create their first CP4NA project. They modify a project file and then create an assembly descriptor. You learn the two types of assemblies. You edit configuration and YAML files to ensure that your projects remain distinct from the projects that are created by other students. You customize your project properties.
Learning objectives	After completing this unit, you should be able to: • Customize your project properties on your client desktop environment • Create or modify files to instantiate a Voice over IP (VoIP) network with network rules

Exercise 2. Create an assembly project and descriptor
Duration: 45 minutes

Overview	This exercise covers the creation of your first assembly project. You learn the two types of assemblies. You edit configuration and YAML files to ensure that your projects remain distinct from the projects that are created by other students. You customize your project properties.
Learning objectives	After completing this exercise, you should be able to: • Customize your project properties on your client desktop environment • Create or modify files to instantiate a Voice over IP (VoIP) network with network rules

Unit 5. Deploy assemblies and resources
Duration: 20 minutes

Overview	This unit introduces Exercise 3 in which students deploy the assembly that they described in the previous exercise. Students learn how to push projects in to the CP4NA server. Students then instruct CP4NA to deploy these resources in the virtual environment. Red Hat OpenStack provides the virtual environment for these exercises.
Learning objectives	After completing this unit, you should be able to: • Push descriptors from your client desktop to the CP4NA server • Instruct CP4NA server to deploy assemblies and resources • Verify the existence of deployed resources in the OpenStack environment

Exercise 3. Deploy assemblies and resources
Duration: 1 hour

Overview	In this exercise, students push the defined project with its assemblies and resources that are created in the preceding exercise to the CP4NA server. Students then instruct CP4NA to deploy these resources in the virtual environment. Red Hat OpenStack provides the virtual environment for these exercises
Learning objectives	After completing this exercise, you should be able to: • Push descriptors from your client desktop to the CP4NA server • Instruct CP4NA server to deploy assemblies and resources • Verify the existence of deployed resources in the OpenStack environment

Unit 6. Add a jumphost server
Duration: 15 minutes

Overview	This unit introduces Exercise 4 in which students add a jumphost server by using the same process as in the previous exercise. They create a deployment descriptor, push it to the CP4NA server, and then deploy the resources. They then verify that the jumphost server is properly deployed. This same process is repeated multiple times throughout this course to reinforce the basic procedures.
Learning objectives	After completing this unit, you should be able to: • Edit and configure a descriptor file for a jumphost • Push the descriptor to the CP4NA server • Deploy the jumphost server • Verify the deployment of the jumphost server

Exercise 4. Add a jumphost server
Duration: 30 minutes

Overview	This exercise has students add a jumphost server by using the same process as in the previous exercise. They create a deployment descriptor, push it to the CP4NA server, and then deploy the resources. They then verify that the jumphost server is properly deployed. This same process is repeated multiple times throughout this course to reinforce the basic procedures
Learning objectives	After completing this exercise, you should be able to: • Edit and configure a descriptor file for a jumphost • Push the descriptor to the CP4NA server • Deploy the jumphost server • Verify the deployment of the jumphost server

Unit 7. Configure server behaviors
Duration: 30 minutes

Overview	This unit introduces Exercise 5 in which students configure server behaviors like autoscaling.
Learning objectives	After completing this unit, you should be able to: • Install a VoIP server and gateway • Configure scaling for the VoIP server • Explain scaling in, scaling out, and heal operations in CP4NA • Explain what triggers a healing process • Explain the difference between a load metric and an integrity metric

Exercise 5. Configure server behaviors
Duration: 1 hour and 30 minutes

Overview	This exercise is fairly lengthy and some students might require extra time. In this exercise, you create a VoIP server and gateway. After you deploy these devices, you configure some testing and behaviors for these instantiated entities. Pay close attention to the instructions for this exercise.
Learning objectives	After completing this exercise, you should be able to: • Install a VoIP server and gateway • Configure scaling for the VoIP server • Explain scaling in, scaling out, and heal operations in CP4NA • Explain what triggers a healing process • Explain the difference between a load metric and an integrity metric

Unit 8. Simulate traffic to test scaling
Duration: 20 minutes

Overview	This unit introduces Exercise 6 in which students deploy a traffic simulator that enables them to test the autoscaling that they configured in the previous exercise. This exercise depends upon the successful completion of the previous exercise. Students create and deploy a traffic simulator project and deploy it. They then use this traffic simulator to increase load on the VoIP gateway to trigger the scaling of VoIP servers in the environment. Pay close attention to the instructions for this exercise.
Learning objectives	After completing this unit, you should be able to: • Install a SIPP traffic simulator and deploy it • Start the traffic simulator to increase the load on the VoIP gateway • Observe scaling of the VoIP server instances

Exercise 6. Simulate traffic to test scaling
Duration: 1 hour

Overview	This exercise builds upon the successful completion of the previous exercise. Students create and deploy a traffic simulator project and deploy it. They then use this traffic simulator to increase load on the VoIP gateway to trigger the scaling of VoIP servers in the environment. Pay close attention to the instructions for this exercise.
Learning objectives	After completing this exercise, you should be able to: • Install a SIPP traffic simulator and deploy it • Start the traffic simulator to increase the load on the VoIP gateway • Observe scaling of the VoIP server instances

Unit 9. Automate behavior tests
Duration: 30 minutes

Overview	This unit introduces Exercise 7 in which students create an automated testing scenario. This exercise builds upon the successful completion of the previous exercise. Students create a behavior testing scenario. Students drag blocks from test templates onto the screen and then completing the names of metrics to be collected, assemblies, and cluster names. Students then declare the expected behavior that must be seen by the testing scenario to validate a service. This testing is then automated. Then, a newly provisioned service has these tests run against it automatically. After tests complete successfully, the test assembly is deleted to leave the other components in a production-ready state. Pay close attention to the lab instructions for this exercise.Students must ensure that the testing blocks appear in the correct order when doing this exercise.
Learning objectives	After completing this unit, you should be able to: • Create a Behavior Testing Scenario using tnco-load Topic (transaction_load and managed_servers) • Learn how to run and automate end-to-end complex testing scenarios • Automate the manual testing steps from the preceding exercise

Exercise 7. Automate behavior tests
Duration: 1 hour and 30 minutes

Overview	This exercise builds upon the successful completion of the previous exercise. Students create a behavior testing scenario. Students drag blocks from test templates onto the screen and then completing the names of metrics to be collected, assemblies, and cluster names. Students then declare the expected behavior that must be seen by the testing scenario to validate a service. This testing is then automated. Then, a newly provisioned service has these tests run against it automatically. After tests complete successfully, the test assembly is deleted to leave the other components in a production-ready state. Pay close attention to the lab instructions for this exercise.Students must ensure that the testing blocks appear in the correct order when doing this exercise.
Learning objectives	After completing this exercise, you should be able to: • Create a Behavior Testing Scenario using tnco-load Topic (transaction_load and managed_servers) • Learn how to run and automate end-to-end complex testing scenarios • Automate the manual testing steps from the preceding exercise

Unit 10. Demonstration – Use CP4NA with CI/CD tool **Duration: 30 minutes**	
Overview	This unit is a recorded video demo of integrating a CI/CD processes with CP4NA. Customers can order services through a service dashboard. A Jenkins CI/CD pipeline uses API calls to create a project in CP4NA. The project is then deployed and tested. When Jenkins sees that the service is active, it notifies the service dashboard to inform the customer that the service is ready. There is no exercise for this unit.
Learning objectives	After completing this unit, you should be able to: • Understand how CP4NA can use CI/CD processes. • Know that the CP4NA development team can assist with integration projects like the one seen in the demo.

Unit 11. Demonstration – Use the CP4NA API **Duration: 30 minutes**	
Overview	This unit is a recorded video demo that shows API features of CP4NA. On the IBM documentation site for CP4NA, there is a link to the CP4NA API with sample code. This demo shows a simple use of Podman software where the API code has been pasted into Podman. The demo shows that basic features can be setup in Podman to automate assembly creation, deletion, and configuration changes in CP4NA. The ability to use the API to automate CP4NA processes provides many possibilities for front-end, user-friendly methods to make self-service telco offerings possible.
Learning objectives	After completing this unit, you should be able to: • Understand what the API of CP4NA can do • Understand that you can paste API code snippets from CP4NA documentation into tools capable of making API calls to automate CP4NA functions

For more information

To learn more about this course and other related offerings, and to schedule training, see **ibm.com**/training

To learn more about validating your technical skills with IBM certification, see **ibm.com**/certify

To stay informed about IBM training, see the following sites:

IBM Training News: http://bit.ly/IBMTrainEN

YouTube: youtube.com/IBMTraining

Facebook: facebook.com/ibmtraining

Twitter: twitter.com/websphere_edu