EDU-VRAOE: VMware vRealize Automation: Orchestration and Extensibility

Course Code: EDU-VRAOE
Duration: 5 days
Instructor-led Training (ILT) | Virtual Instructor-led Training (VILT)

OVERVIEW

During this five-day course, you focus on using VMware vRealize® Orchestrator™ to extend the functionality of VMware vRealize® Automation™. You learn how to provide XaaS (Anything as a Service) and implement Machine Lifecycle Extensibility using the VMware vRealize® Automation™ Event Broker. You also learn how to create vRealize Orchestrator workflows and vRealize Automation ABX actions. You learn about various features, including basic scripting implementation along with logic processing to implement a variety of functions to use in your environment. This course teaches implementing debugging, loops, conditions, and user interactions in vRealize Orchestrator. The course introduces the new vRealize Orchestrator HTML 5 interface, along with API calls and REST functions, to give you the groundwork to implement a variety of plugins and scripts. This course is designed to give you the tools to craft custom solutions in the product.

SKILLS COVERED

By the end of the course, you should be able to meet the following objectives:

- Describe the features and benefits of integrating vRealize Orchestrator and vRealize Automation
- Describe the role of vRealize Orchestrator workflows and content elements in automation
- Use the vRealize Orchestrator client to access and navigate the vRealize Orchestrator platform
- Use the vRealize Orchestrator client to import and run vRealize Orchestrator library workflows
- Design, develop, and run custom reusable vRealize Orchestrator workflows
- Integrate vRealize Automation with vRealize Orchestrator to deliver custom IT services
- Use the vRealize Automation event broker service to trigger specific vRealize Orchestrator workflows or ABX Actions
- Leverage the event broker to extend IaaS (Infrastructure-as-a-Service) machine lifecycle processes
- Use XaaS to extend vRealize Automation into other enterprise systems
- Use VMware APIs to run vRealize Orchestrator workflows
- Use the vSphere Client Code Capture feature

WHO SHOULD ATTEND?

- Experienced VMware administrators
- Automation and orchestration specialists
- System integrators
- Private cloud and public cloud administrators

PREREQUISITES

This course requires the following prerequisites:

- Knowledge of VMware vSphere®
- VMware vRealize Automation: Install, Configure, Manage [V8.3] course or equivalent knowledge
- Working knowledge of scripting or programming using JavaScript, Windows

PowerShell, Perl, Java, Python, or similar languages. All code is provided during class.

MODULES

Module 1: Course Introduction

- Introductions and course logistics
- Course objectives

Module 2: Overview of vRealize Automation and vRealize Orchestrator

- Define the purpose of vRealize Automation
- Outline the purpose of vRealize Orchestrator
- Describe the main components of vRealize Automation
- Describe the main components of vRealize Orchestrator

Module 3: Creating Schema Elements

- Invoking JavaScript from a vRealize Orchestrator workflow
- Invoking a vRealize Orchestrator Workflow from a vRealize Orchestrator workflow
- Invoking an action from a vRealize Orchestrator workflow
- Using vRealize Orchestrator workflows both synchronously and asynchronously

Module 4: Working with Variables

- Defining inputs, outputs, and variables in vRealize Orchestrator workflows
- Binding variables in vRealize Orchestrator workflows
- Wrapping vRealize Orchestrator workflows
- Using APIs and the API Explorer
- Creating actions in vRealize Orchestrator
- Using vRealize Orchestrator input forms
- Handling user interactions in vRealize Orchestrator

Module 5: Handling Exceptions, Logging, and Debugging

- Handling exceptions in vRealize Orchestrator workflows
- Using logs in vRealize Orchestrator workflows
- Debugging vRealize Orchestrator workflows

Module 6: Branching and Looping

- Using branching in vRealize Orchestrator workflows
- Using loops in vRealize Orchestrator workflows

Module 7: Working with Assets

- Using configuration elements in vRealize Orchestrator
- Using resources in vRealize Orchestrator
- Using packages in vRealize Orchestrator

Module 8: Working with Plug-Ins

- Downloading and installing Plug-Ins
- Using the SSH plug-In in vRealize Orchestrator
- Using the REST plug-in in vRealize Orchestrator
- Using the vRealize Automation plug-in in vRealize Orchestrator
- Using the PowerShell plug-in in vRealize Orchestrator

Module 9: Working with Versioning and Git

- Using versioning in vRealize Orchestrator

- Using Git in vRealize Orchestrator

Module 10: Scheduling, Sleeping, and Waiting

- Using scheduling in vRealize Orchestrator
- Using sleeping in vRealize Orchestrator
- Using waiting in vRealize Orchestrator

Module 11: Introduction to vRealize Automation Extensibility

- Introduction to extensibility
- Using ABX actions
- Using Python
- Using Nodejs
- Using PowerShell
- Using vRealize Automation Lifecycle

Module 12: Extending vRealize Automation with Event Broker

- Overview of vRealize Automation Event Broker
- Creating vRealize Automation subscriptions
- Data exchange between vRealize Automation and vRealize Orchestrator

Module 13: Using ABX Actions

- Overview of Action Based Extensibility (ABX)
- Comparison of vRealize Orchestrator and ABX
- Creating ABX Actions scripts, REST, and flows
- Using Day-2 Actions in vRealize Automation
- Describe the visualization capabilities of NSX Network Detection and Response

Module 14: Working with Services, Custom Resources, and Resource Actions

- Using vRealize Orchestrator as a content source in vRealize Automation
- Creating custom resources in vRealize Automation
- Creating resource actions in vRealize Automation
- Using Day-2 Actions in vRealize Automation

Module 15: Using vSphere Client Code Capture

- Enabling vSphere Client code capture
- Using vSphere Client code capture to capture code in vRO, Javascript, PowerCLI or other languages.
- Using the captured code in vRealize Orchestrator workflows or actions.

END OF PAGE