

**Pulumi Fundamentals**

**Course Number:** DVOP-180WA
**Duration:** 1 day

**Overview**

Pulumi is an infrastructure-as-code (IaC) tool that enables developers and DevOps teams to define cloud resources using familiar programming languages (TypeScript, JavaScript, Python, Go, .NET, Java, and YAML). The Pulumi course teaches learners to leverage Pulumi’s architecture, use cases, and practical applications for deploying infrastructure across various cloud providers. This class uses AWS, Python, and GitHub.

**Prerequisites**

All learners must have a fundamental knowledge of AWS Cloud and Python programming.

**Materials**

All Pulumi training attendees receive comprehensive courseware.

**Software Needed on Each Student PC**

Attendees will not need to install any software on their computers for this class. The class will be conducted in a remote environment that Accelebrate will provide; students will only need a local computer with a web browser and a stable Internet connection. Any recent version of Microsoft Edge, Mozilla Firefox, or Google Chrome will work well.

**Objectives**

* Understand the basics of Pulumi
* Identify supported languages, cloud providers, and deployment targets
* Install and configure the Pulumi CLI
* Set up a Pulumi project
* Understand the structure of Pulumi projects and stack management
* Use Pulumi in automated pipelines

**Outline**

* Introduction to Pulumi
	+ What is Pulumi?
	+ Overview of IaC and Pulumi’s place in the ecosystem
* Pulumi Architecture and Key Concepts
	+ Stacks, resources and inputs/outputs
	+ Supported languages and cloud providers
* Getting Started
	+ Installation and setup
	+ Creating your first Pulumi project
* Pulumi Program Structure
	+ Defining resources with programming languages
	+ Using the Pulumi SDK and configuration options
* State Management
	+ Understanding Pulumi’s state and backends
* Pulumi Advanced Features
	+ Managing secrets and configurations
	+ Writing reusable components
	+ Leveraging dynamic resource providers
* Integration with CI/CD
	+ Setting up Pulumi in automated pipelines
* Pulumi Best Practices
	+ Modularization, versioning, and debugging tips
* Conclusion