



Administration Tools

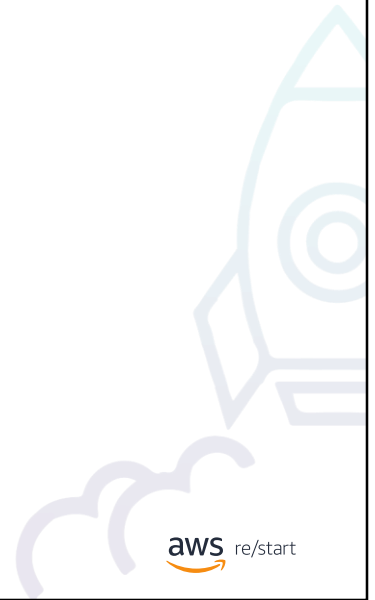
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Welcome to Administration Tools.

What you will learn

At the core of the lesson

You will identify various tools to automate administration.



In this lesson, you will learn about various automation tools.

Software development kits (SDKs)

You can use SDKs to access AWS services programmatically and write administrative scripts in different programming languages.

- .NET
- C++
- Go
- Java
- JavaScript
- Node.js
- PHP
- Python
- Ruby



AWS Tools
and SDKs

The concept of infrastructure as code (IaC) is fundamental to cloud computing and differentiates the cloud from more traditional IT environments. The Amazon Web Services (AWS) software development kits (SDKs) and the application programming interfaces (APIs) provide you with tools for managing resources that are built on AWS through an IaC approach.

AWS provides SDKs that are language-specific. These SDKs also contain APIs for the languages that are listed. A developer or administrator can use both the SDKs and the APIs to incorporate the connectivity and functionality of the wide range of AWS services into their code without writing functions from scratch. The AWS SDKs have extensive documentation about how to use them, including guides about how to get started, developer guides, API references, and community forums or developer blogs. In addition to the languages identified in this slide, AWS also provides SDKs for the Kotlin, Rust, and Swift programming languages.

For more information about the AWS SDKs, see the Tools to Build on AWS webpage at <https://aws.amazon.com/developer/tools/>.

AWS CloudFormation

You can use CloudFormation to create, update, and delete a set of AWS resources as a single unit.

- You define the resources in a **template**, which can be written in JSON or YAML.
- CloudFormation provisions the resources defined in a template as a single unit called a **stack**.
- Key features of CloudFormation include the ability to do the following:
 - Preview how proposed changes to a stack will impact the existing environment.
 - Detect drift.
 - Invoke an AWS Lambda function.



AWS CloudFormation



AWS CloudFormation is another useful tool that you can use to create, update, or delete entire AWS infrastructure deployments predictably and repeatedly.

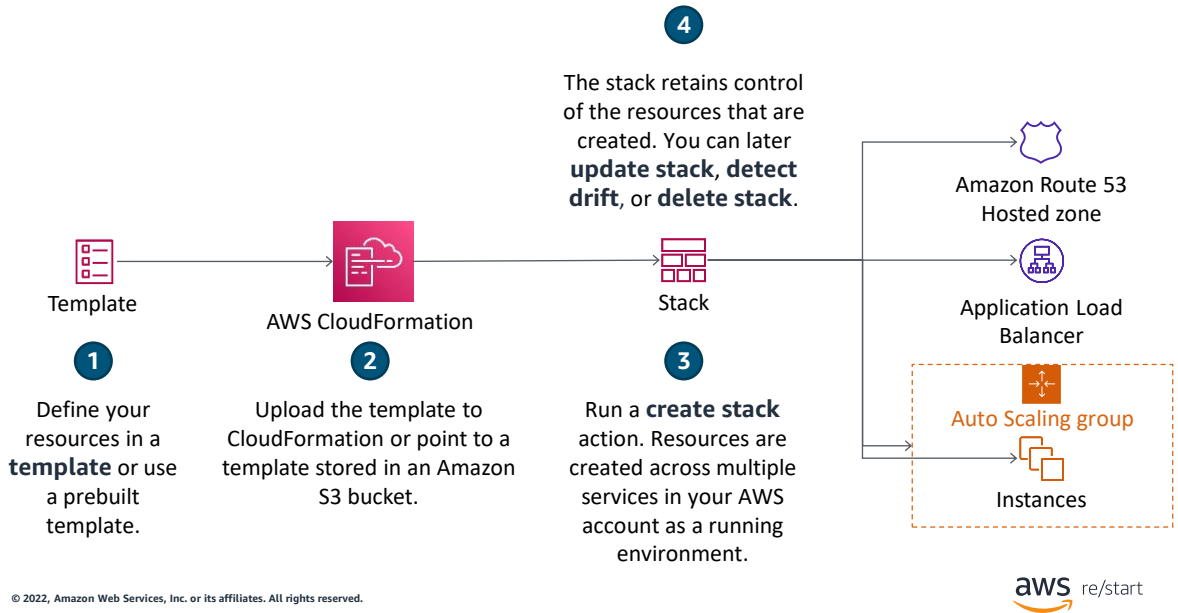
With CloudFormation, an entire infrastructure can be modeled in a single text file that is written in JSON or YAML. You can use the text file template to define all the required AWS resources for a stack. A CloudFormation stack is a collection of AWS resources that can be managed as a single unit. You can create, update, or delete a collection of resources by creating, updating, or deleting stacks. Resources in the stack can include Amazon Elastic Compute Cloud (Amazon EC2) instances, Amazon Relational Database Service (Amazon RDS) instances, and virtual private clouds (VPCs). They also can include many other resources that are created from AWS services.

You can also use CloudFormation to preview how proposed changes to a stack might impact their running resources. For example, you might want to find out whether your changes will delete or replace any critical resources. CloudFormation makes only the changes to the stack after the user decides to run them.

Another feature of CloudFormation is drift detection. Performing a drift detection operation on a stack determines whether the stack's actual configuration differs, or has drifted, from its expected template configuration. A resource is considered to have drifted if any of its actual property values differ from the expected property values or if the property or resource has been deleted. A stack is considered to have drifted if one or more of its resources have drifted. If the resource in the stack supports drift detection, the drift detection operation returns detailed information about the drift status of that resource.

CloudFormation also supports custom extensions to your stack template that can be built by using AWS Lambda. You can write custom provisioning in a Lambda function and configure a CloudFormation stack to invoke the Lambda function when it creates the stack. For example, you can write custom provisioning logic for tasks. This logic could look up the most recent Amazon Machine Image (AMI) IDs so that you can use the IDs in your stacks.

How CloudFormation works



This diagram illustrates how CloudFormation works:

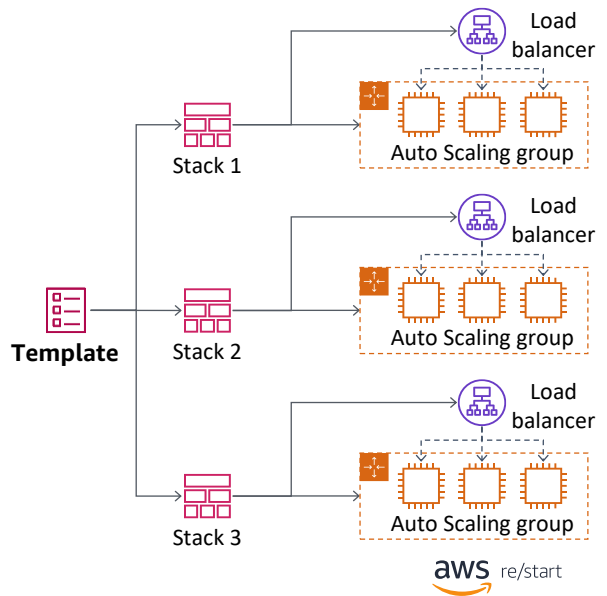
1. First, you define the AWS resources that you want to create. In this example, you create EC2 instances, an Application Load Balancer, an Auto Scaling group, and an Amazon Route 53 hosted zone. You define the resources in a CloudFormation template. You can create the template from scratch, or you can use a prebuilt or sample template.
2. Next, you upload the template to CloudFormation. Alternatively, you can store the template on Amazon Simple Storage Service (Amazon S3) and point CloudFormation to the location where it is stored.
3. Third, you run the create stack action. When you do so, the CloudFormation service reads through what is specified in the template and creates the desired resources in your AWS account. A single stack can create and configure resources for multiple AWS services in a single Region.
4. Finally, you can observe the progress of the stack-creation process. After the stack has successfully completed, the AWS resources that it created exist in your account. The stack object remains, and it acts like a handle to all the resources that it created. This technique is helpful when you want to take actions later. For example, you might want to update the stack to create additional AWS resources or modify existing resources. Alternatively, you might want to delete the stack, which will clean up and delete the resources that the stack created.

Benefits of CloudFormation

CloudFormation provides the following benefits: reusability, repeatability, and maintainability.

With CloudFormation, you can do the following:

- Deploy complex environments rapidly.
- Duplicate the same environment.
- Ensure configuration consistency.
- Delete resources in a single action (delete the stack).
- Propagate the same change to all stacks (update the stacks).



CloudFormation provides the benefits of IaC. These benefits include reusability, repeatability, and maintainability.

If you build infrastructure with code, you gain benefits like the ability to rapidly deploy complex environments. With one template (or a combination of templates), you can build the same complex environments repeatedly.

In the example here, a single template is used to create three different stacks. Each stack can be created rapidly, usually in a matter of minutes. Each stack replicates complex configuration details consistently.

Suppose that Stack 2 is your test environment and Stack 3 is your production environment. Then, you can have greater confidence that if your test jobs performed well in the test environment, they will also perform well in the production environment. The template minimizes the risk that the test environment is configured differently from the production environment.

Also, if you want to make a configuration update in the test environment, you update the template with the change and update all the stacks. This process helps ensure that the modifications to a single environment will be reliably propagated to all environments that should receive the update.

Another benefit is that CloudFormation facilitates the clean up all the resources that were created in your account to support a test environment. After you no longer need these resources, you can clean them up in a single action by deleting the stack. This cleanup helps eliminate the cost associated with resources that you no longer need.

AWS OpsWorks

You can use OpsWorks to automate how servers are configured, deployed, and managed.

The following are features of OpsWorks:

- It automates configuration management.
- It is based on the **Chef** and **Puppet** popular open-source automation platforms.
- It is available in three versions:
 - AWS OpsWorks for Chef Automate
 - AWS OpsWorks for Puppet Enterprise
 - AWS OpsWorks Stacks

AWS OpsWorks is another useful tool. It is a configuration management service that provides managed instances of Chef and Puppet. Because Chef and Puppet are automation platforms, you can use code to automate the configurations of your servers. With OpsWorks, you can use Chef and Puppet to automate how servers are configured, deployed, and managed across your EC2 instances or on-premises compute environments.

OpsWorks has three offerings:

- AWS OpsWorks for Chef Automate provides a fully managed Chef Automate server. This server provides workflow automation for administrative and operational tasks, such as software and operating system configurations, continuous compliance, package installations, database setups, and more.
- AWS OpsWorks for Puppet Enterprise provides a managed Puppet Enterprise server and a suite of automation tools that provide workflow automation for orchestration, automated provisioning, and visualization for traceability. With Puppet Enterprise, you can define configurations for your servers in a format that you can maintain and version like your application source code.
- AWS OpsWorks Stacks is a configuration management service that helps you configure and operate applications of all kinds and sizes by using Chef.

For more information, see the AWS OpsWorks webpage at <https://aws.amazon.com/opsworks/>.

Checkpoint questions

1. Which AWS tool can a developer use to write an automated script in Python to access an AWS service?
2. In which CloudFormation component would an administrator define the AWS infrastructure resources that they want to provision?
3. Which AWS service provides managed instances of Chef and Puppet?



1. Which AWS tool can a developer use to write an automated script in Python to access an AWS service?

AWS SDK for Python

2. In which CloudFormation component would an administrator define the AWS infrastructure resources that they want to provision?

Template

3. Which AWS service provides managed instances of Chef and Puppet?

OpsWorks

Key ideas



- **AWS SDKs** provide **access to AWS services** by using **APIs** in a variety of programming languages.
- **CloudFormation** helps you create, update, and delete **AWS infrastructure deployments** predictably and repeatedly.
- **OpsWorks** helps you automate **configuration management** tasks.



Thank you



© 2022 Amazon Web Services, Inc. or its affiliates. All rights reserved. This work may not be reproduced or redistributed, in whole or in part, without prior written permission from Amazon Web Services, Inc. Commercial copying, lending, or selling is prohibited. Corrections, feedback, or other questions? Contact us at <https://support.aws.amazon.com/#/contacts/aws-training>. All trademarks are the property of their owners.