



Configuration Management in the Cloud

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

This lesson explores the benefits of configuration management in the cloud. It identifies AWS services that you can use to implement configuration management strategies.

What you will learn

At the core of the lesson

You will learn how to:

- Describe the benefits of configuration management
- Identify key AWS services for configuration management
- Create a configuration management strategy

Key terms

- Configuration management



In this lesson, you will learn how to:

- Describe the benefits of configuration management
- Identify key AWS services for configuration management
- Create a configuration management strategy

Benefits of configuration management

- Increase efficiency
- Validate every change before release
- Reduce cost by removing unwanted resources
- Enforce security at every layer
- Deploy configuration changes to running instances
- Make configuration automated and repeatable

Management tools



Amazon CloudWatch



AWS CloudFormation



AWS OpsWorks



Amazon Web Services (AWS) offers several methods to help configure and manage infrastructure that is deployed on the AWS Cloud. It is important to plan the configuration and orchestration of AWS resources proactively instead of reactively. Resources can span multiple AWS services. Typically, resources include the following and other resources that comprise your deployed infrastructure on AWS:

- Amazon Elastic Compute Cloud (Amazon EC2) instances
- Auto Scaling groups
- Security groups
- Elastic Load Balancing load balancers

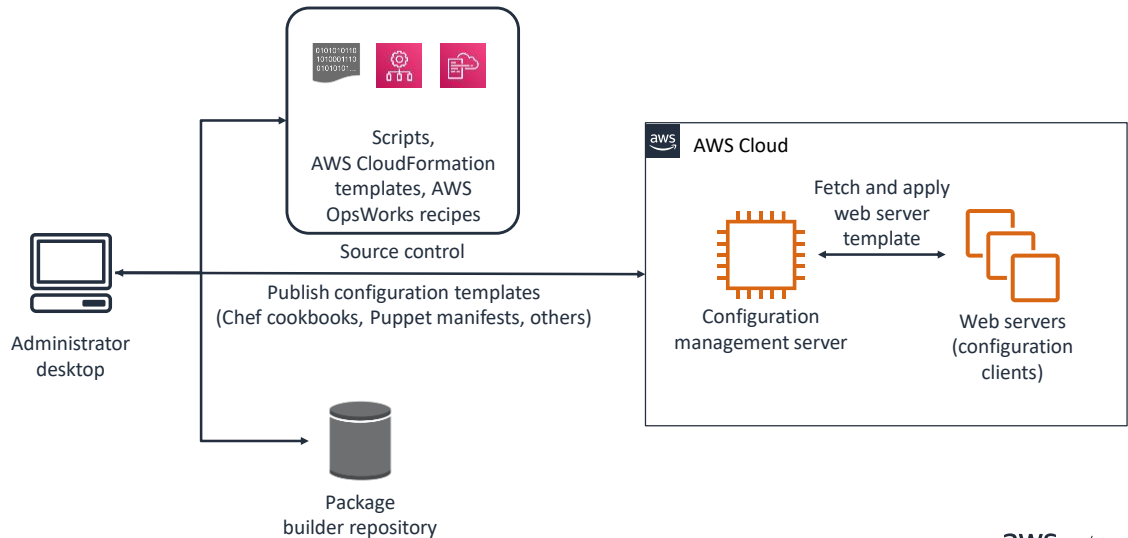
The manual configuration of distributed systems is time-consuming and error-prone, and it can lead to inconsistently configured systems. Therefore, the ability to automate, monitor, and track configuration changes is a key component of any configuration management solution.

Configuration management provides many benefits. One of the key benefits is that it enables you to launch new resources quickly and consistently in an automated or semiautomated way. It is commonly used for deploying configuration changes to running instances with increased consistency and limited server downtime.

Some of the managed services and tools from AWS that customers commonly use for

configuration management include AWS CloudFormation and AWS OpsWorks. The Amazon CloudWatch service complements these services by providing a monitoring service that can trigger actions when changes to your resources occur.

Configuration management tools



4

aws re/start

Many AWS customers use configuration tools to launch new EC2 instances and other resources in an automated, repeatable manner. The flexibility of the configuration management tools enables the implementation of a complete configuration management strategy.

The diagram illustrates one way in which configuration software can be integrated within an AWS architecture. Assume that you wanted to use configuration software to initialize a set of Amazon EC2 instances as web servers. The scenario details are as follows:

1. An administrator configures a configuration server.

This is usually a standalone Amazon EC2 instance. The instance will host a set of templates, which describe all the applications, files, and configurations necessary to launch AWS resources. For example, a web server template might specify how to automate the launch of an EC2 instance. The web server template might then specify the following for web server content:

- Installation of HTTP
- Configuration of the web server `httpd.conf` file
- Installation of all necessary programming language environments (such as PHP or Ruby)
- Creation of a directory and file structure on the EC2 instance

The configuration server would probably contain a number of recipes for configuring dozens of different types of servers. Examples of servers include *MySQL server*, *NAT server*, *Windows IIS server*, *Maven repository*, and so on.

2. Administrator creates the configuration templates. Then, the administrator publishes them to the configuration server and checks them into a source control system for version management and change tracking.
3. Administrator might use any combination of custom scripting, AWS CloudFormation, AWS OpsWorks, or other similar products to describe and create AWS Cloud deployments.

These deployments might consist of Amazon EC2 instances, virtual private cloud (VPC) configurations, databases, and other required AWS resources.

Technologies for configuring EC2 instances



User data



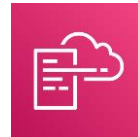
Amazon Machine
Images (AMIs)



Configuration and
deployment frameworks



AWS
OpsWorks



AWS
CloudFormation

5



AWS provides a number of technologies for configuring and deploying your Amazon EC2 instances and other AWS infrastructure. This module will explore some of these technologies. It will also discuss approaches for combining these technologies into a comprehensive configuration and deployment strategy.

Available features and resources include:

- **User data** – Enables you to author scripts that are run on instance launch. The use of scripts is one of the simplest approaches to configuration management. However, at the enterprise level, it becomes more difficult to manage and version infrastructure by only using scripts.
- **Amazon Machine Images (AMIs)** – By creating base images that are customized to the needs of your organization, you can pre-deploy installations and configurations into the EC2 instances that are launched from the AMI. By using this configuration method, you can potentially reduce deployment times.
- **Configuration and deployment frameworks** – Technologies such as Chef, Puppet, and Ansible enable you to configure new instances by using templates. You can update configurations dynamically in response to change.
- **AWS OpsWorks** – A configuration management service that provides managed instances of Chef and Puppet.

- **AWS CloudFormation** – An AWS service that enables you to configure architectures for repeatable deployments.

Using configuration software

Benefits of using a configuration server

- Can greatly simplify common administrative tasks.
- Offers a configuration that is idempotent.
 - Resources are allocated only once.
 - Manual changes are detected and rolled back.
- Supply user data to instance to kick off client configuration.
 - Install client and any required configuration and security credentials.
 - Specify the templates to be run.

Using a configuration server can simplify many common administrative tasks. The most common example is provisioning and de-provisioning user access to EC2 instances.

Simplified common tasks

Suppose that Jane, an engineer, was granted access to multiple EC2 instances across the fleet. That is, Jane has a private key that enables her to log in with her unique user ID and perform administrative tasks on these instances. What happens when Jane leaves the company? Without a configuration server, de-provisioning Jane's access to these servers can be a manual nightmare. However, by using a configuration server, a systems administrator can easily change Jane's status in the system. The administrator can then apply this change to all instances in the fleet with a few simple commands.

Idempotent configuration

One of the advantages of using a configuration server (such as Chef, Puppet, or Ansible) is that configuration is *idempotent*. Repeating commands and functions do not alter the surrounding system or environment. Resources that are created or configured by a configuration server are configured or created *only* once. If manual modifications are made to the configuration of an instance, the configuration server detects them and rolls them back. This ensures that an accidental change doesn't

compromise the integrity of the instance.

Creating a configuration strategy

1. Use Amazon Machine Images (AMIs).
 - Foundational components
 - Company-wide tools and standards
2. Use configuration software.
3. Define a source control and change policy.
4. Explore and experiment.

AWS customers use different combinations of the technologies that are discussed in this lesson to manage their AWS resources. The solution that each organization adopts depends on several factors, including the organization's production workflow and the organizational standards. None of these techniques is mutually exclusive. They can all be used in tandem.

One approach to creating a configuration strategy is to first consider which (if any) custom AMIs you should create. In general, configuring software into custom AMIs is best reserved for foundational, slow-changing components that are required and are standard throughout the organization.

After establishing what to include in any custom AMIs, the next step is to decide which configuration software to use. For example, you can use AWS CloudFormation or AWS OpsWorks to manage the configurable, rapidly changing components that you want to deploy.

A best practice is to have a version control system, such as Git or AWS CodeCommit, to maintain your infrastructure as code (IaC).

Finally, explore and experiment. Developing an effective configuration management solution is an iterative process whereby you try an approach and continually adjust it. You can iteratively improve the approach, the templates, and the IaC that you

develop. This way, your solution becomes more effective and appropriate for your business needs over time.

Key takeaways



© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

8

- Configuration management provides multiple benefits, including validation, automation, documentation, cost reduction, and security.
- AMIs can be pre-created to include required applications and files.
- AWS CloudFormation helps create, update, and delete AWS resources.
- AWS recommends managing a collection of AWS resources as a single unit through AWS CloudFormation.

aws re/start

Key takeaways from this module include:

- Configuration management provides multiple benefits, including validation, automation, documentation, cost reduction, and security.
- AMIs can be pre-created to include required applications and files.
- AWS CloudFormation helps create, update, and delete AWS resources.
- AWS recommends managing a collection of AWS resources as a single unit through AWS CloudFormation.