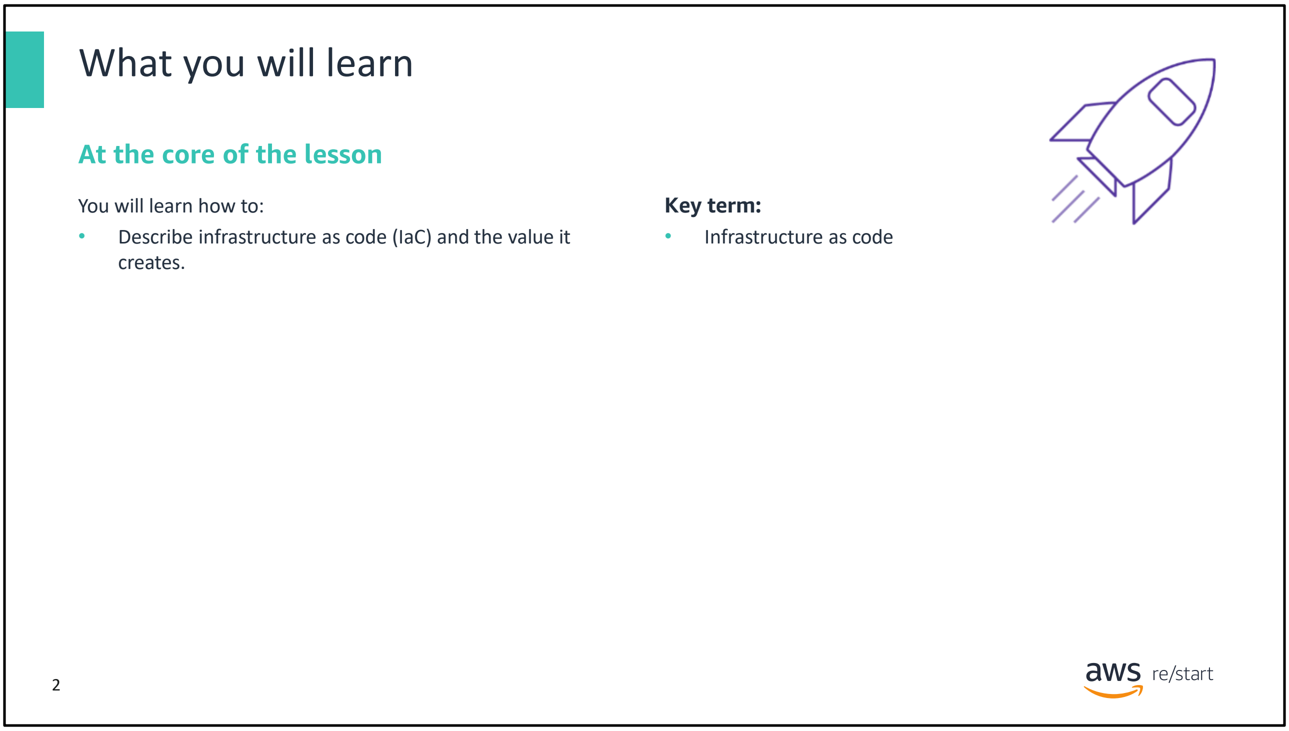
Infrastructure as Code

This lesson explores the concept of infrastructure as code (IaC).



What you will learn

At the core of the lesson

You will learn how to:

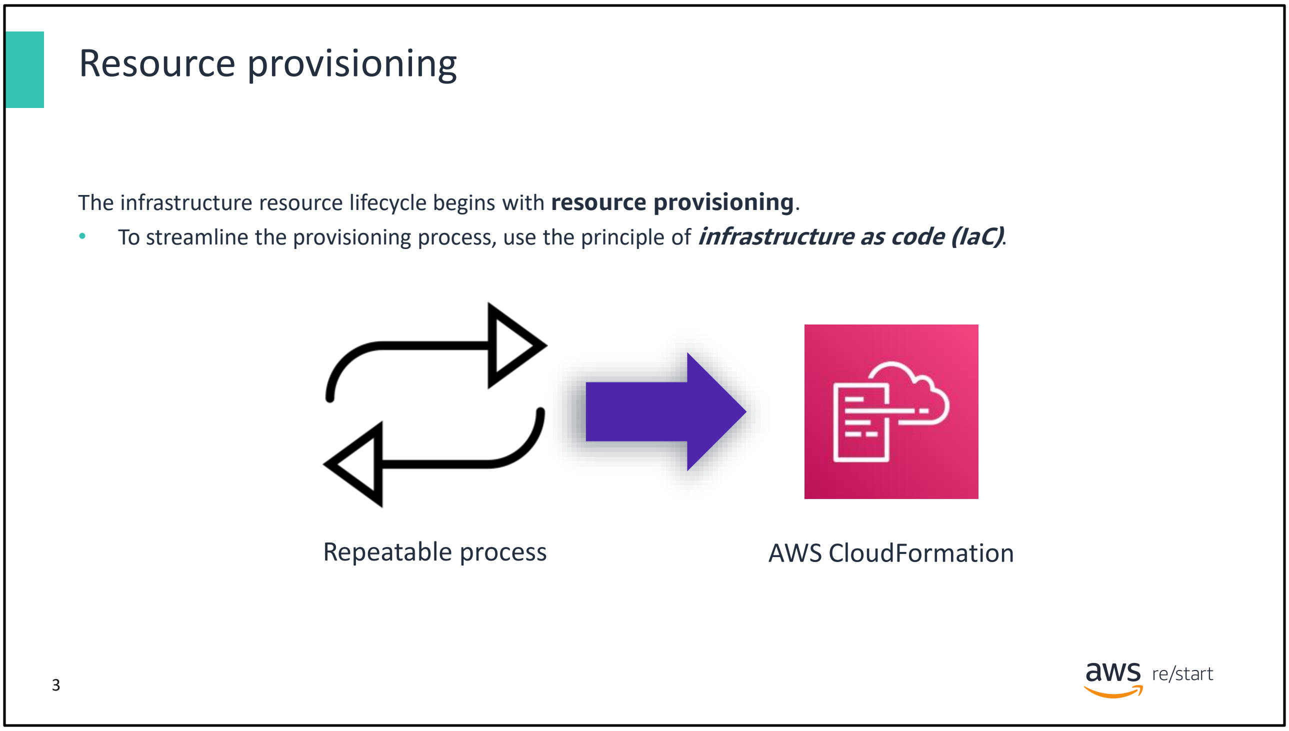
- Describe infrastructure as code (IaC) and the value it creates.

Key term:

- Infrastructure as code

In this lesson, you will learn how to:

- Describe infrastructure as code (IaC) and the value that it creates



Resource provisioning

The infrastructure resource lifecycle begins with **resource provisioning**.

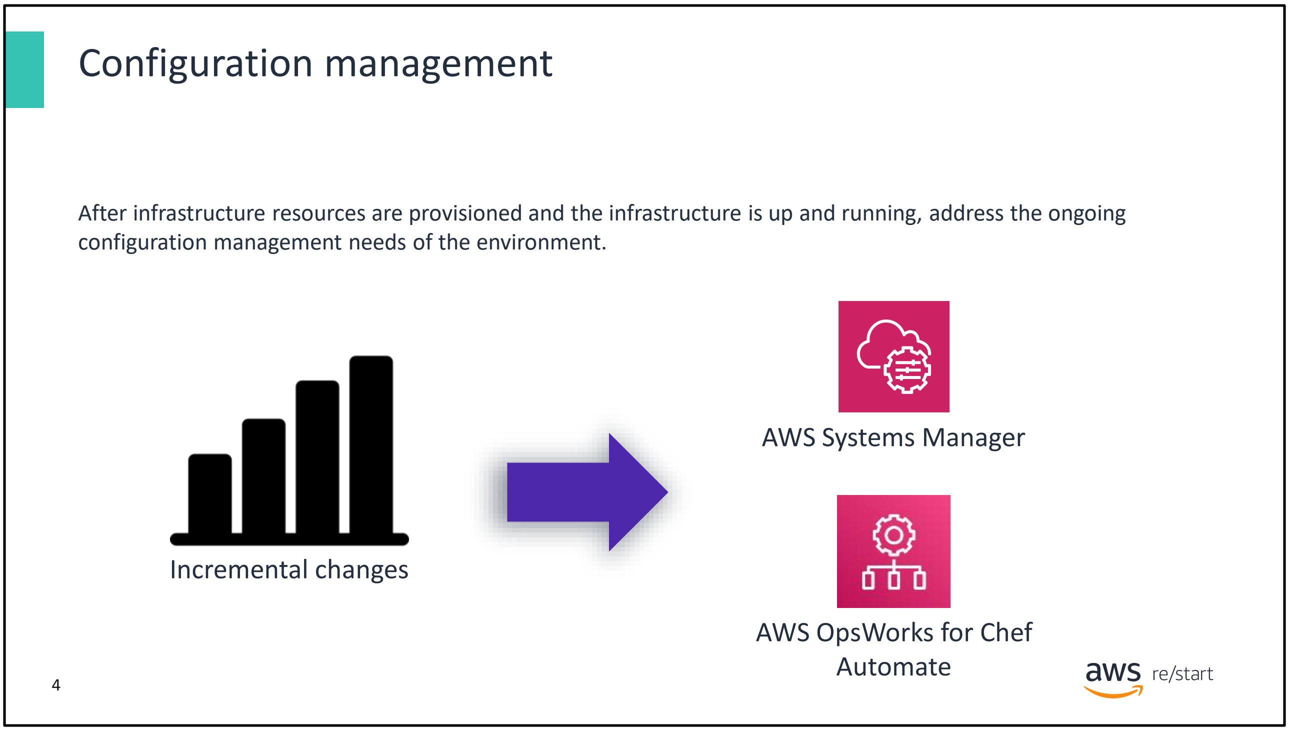
- To streamline the provisioning process, use the principle of ***infrastructure as code (IaC)***.

Repeatable process
AWS CloudFormation

The creation process for infrastructure resources needs to be consistent and reliable. Infrastructure as code (IaC) is a framework for implementing such disciplines. AWS CloudFormation is a service for creating infrastructure using code.

Here are some examples when repeatable, consistent deployments are needed:

- The creation of replica environments for experimentation of new services, testing in a mirror of the production infrastructure, or the quick deployment of recovery environments after a disaster.
- The creation of temporary environments for demonstrations. Typically when a service is being demonstrated, it is done using infrastructure that is not a mirror of a real environment. With IaC, a copy of production can be quickly deployed and used for the demonstration.
- To lower costs, development environments for programmers can be stood up when needed. Then, when it is not needed, the environment can be deleted. This flexibility not only saves money but also ensures the developers are creating software that will work with the current production systems.



Configuration management

After infrastructure resources are provisioned and the infrastructure is up and running, address the ongoing configuration management needs of the environment.

Incremental changes
AWS Systems Manager
AWS OpsWorks for Chef Automate
aws re/start

After you provision your infrastructure resources and the infrastructure is up and running, you must address the ongoing configuration management needs of the environment.

Consider the following situations:

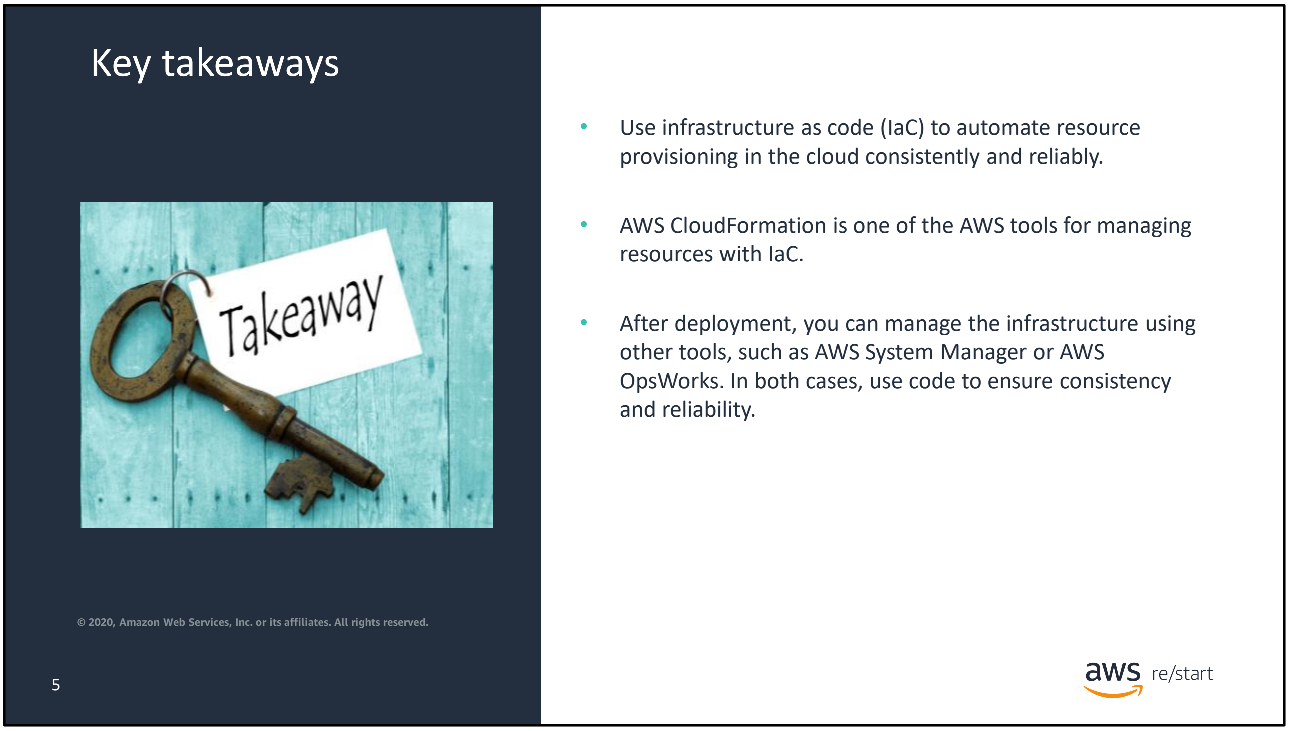
- A release manager wants to deploy a version of an application across a group of servers. If problems are encountered, the release managers wants to be able to roll back to a working version.
- A system administrator receives a request to install a new operating system package in developer environments, but to keep the other environments untouched.
- An application administrator needs to periodically update a configuration file across all servers that host an application.

One way to address these situations is to return to the provisioning stage, provision fresh resources with the required changes, and remove the earlier resources. This approach, which is also known as *infrastructure immutability*, ensures that the provisioned resources are built anew— according to the code baseline—each time a change is made. This process reduces configuration drift. A configuration drift means that the actual configuration of the resources differs, or has *drifted*, from the expected configuration.

There are times, however, when you might want to take a different approach. Some environments have high levels of durability in which transactions or data are stored permanently. Therefore, committed changes to a database can be recovered or reconstructed entirely. For such environments, it might be preferable to establish ways to make incremental changes to the current resources instead of reprovisioning them. You can manage these changes by using AWS Systems Manager and AWS OpsWorks for Chef Automate.

AWS Systems Manager gives engineers insight into the existing AWS infrastructure. This includes automating many repetitive tasks and shortening the time to identify operational issues. System Manager also has automation capability to help improve the accuracy and efficiency of maintenance tasks.

Like System Manager, AWS OpsWoks for Chef Automate can help automate and improve the efficiency of many tasks by providing a managed Chef server and tools to help with deployments, testing, security, as well as visibility and status of resources in cloud environment.



Key takeaways

- Use infrastructure as code (IaC) to automate resource provisioning in the cloud consistently and reliably.
- AWS CloudFormation is one of the AWS tools for managing resources with IaC.
- After deployment, you can manage the infrastructure using other tools, such as AWS System Manager or AWS OpsWorks. In both cases, use code to ensure consistency and reliability.

- Use infrastructure as code (IaC) to automate resource provisioning in the cloud consistently and reliably.
- AWS CloudFormation is one of the AWS tools for managing resources with IaC.
- After deployment, you can manage the infrastructure using other tools, such as AWS System Manager or AWS OpsWorks. In both cases, use code to ensure consistency and reliability.