100 XP

Summary

2 minutes

Here are some of the key takeaways from this module:

- Cloud solutions are often complex, consisting of tens or even hundreds of resources that must be provisioned and configured to work together.
- Automation helps ensure that solutions can be deployed (and updated) reliably and repeatedly.
- One way to automate the provisioning of cloud solutions is to use scripting languages such as Bash and PowerShell. However, these scripts can grow very complex, especially if they're to be used to *update* solutions as well as provision them.
- Another way to achieve automation is to use configuration-management (CM) systems such as Chef, Puppet, and Ansible.
- CM systems embody the concept of Infrastructure-as-Code (IaC), in which the state of a system is defined using a declarative or imperative language. With IaC, you can make a change to a system by changing the code that defines it.
- One example of IaC is Azure Resource Manager templates, which define Azure resources (and sets of resources) using JSON. Azure Resource Manager templates are Azure-specific and do not work with other cloud providers.
- Some configuration-management platforms, including Chef, Puppet, and Ansible, can work with multiple cloud service providers.
- Puppet and Ansible rely on declarative methodologies in which desired system states are specified, and the tools determine the best methods for achieving those states.
- Chef uses a simpler language that renders instructions in the form of symbolic recipes.
- Infrastructure orchestration provides an even higher level of automation and manages the entire lifecycle of a solution.
- An infrastructure-orchestration platform such as HashiCorp Terraform plans for the implementation of all the services and applications a solution may require, and renders that plan in a form that may be reviewed and even rejected before it is implemented.

Module complete:

Unlock achievement