

Describe provisioning relational data services

4 minutes

In the sample scenario, Contoso has decided that the organization will require several different relational stores. As the data engineer, you've been asked to set up data stores using Azure SQL Database, PostgreSQL, and MySQL.

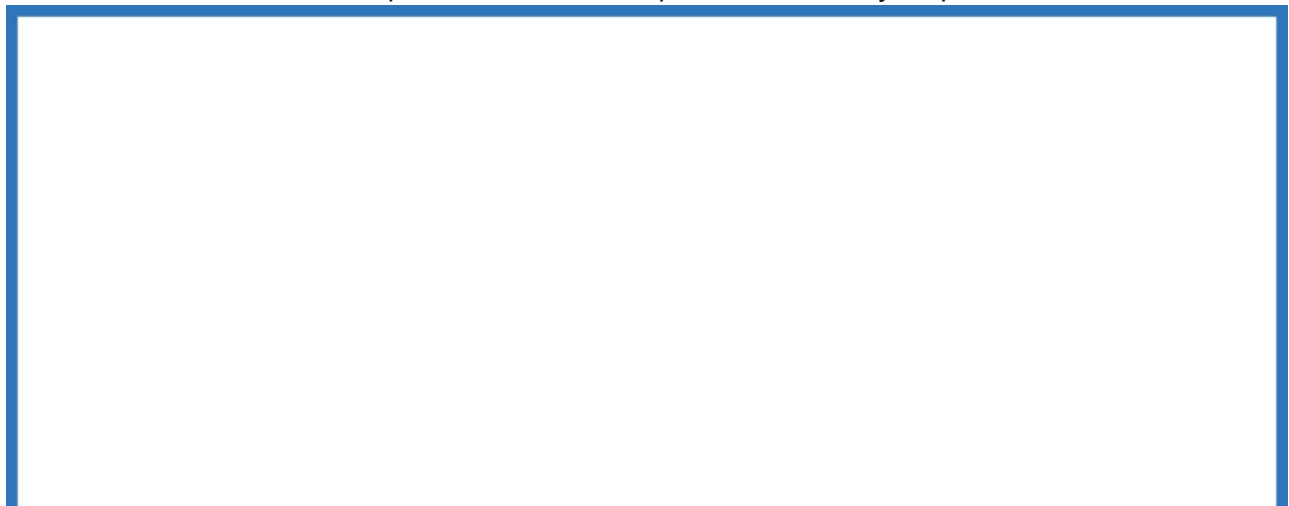
In this module, you'll learn how to provision these services.

What is provisioning?

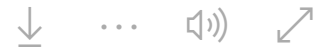
Provisioning is the act of running series of tasks that a service provider, such as Azure SQL Database, performs to create and configure a service. Behind the scenes, the service provider will set up the various resources (disks, memory, CPUs, networks, and so on) required to run the service. You'll be assigned these resources, and they remain allocated to you (and charged to you), until you delete the service.

How the service provider provisions resources is opaque, and you don't need to be concerned with how this process works. All you do is specify parameters that determine the size of the resources required (how much disk space, memory, computing power, and network bandwidth). These parameters are determined by estimating the size of the workload that you intend to run using the service. In many cases, you can modify these parameters after the service has been created, perhaps increasing the amount of storage space or memory if the workload is greater than you initially anticipated. The act of increasing (or decreasing) the resources used by a service is called *scaling*.

This video summarizes the process that Azure performs when you provision a service:



00:00 / 00:25



Azure provides several tools you can use to provision services:

- *The Azure portal.* This is the most convenient way to provision a service for most users. The Azure portal displays a series of service-specific pages that prompt you for the settings required, and validates these settings, before actually provisioning the service.
- *The Azure command-line interface (CLI).* The CLI provides a set of commands that you can run from the operating system command prompt or the Cloud Shell in the Azure portal. You can use these commands to create and manage Azure resources. The CLI is suitable if you need to automate service creation; you can store CLI commands in scripts, and you can run these scripts programmatically. The CLI can run on Windows, macOS, and Linux computers. For detailed information about the Azure CLI, read [What is Azure CLI](#).
- *Azure PowerShell.* Many administrators are familiar with using PowerShell commands to script and automate administrative tasks. Azure provides a series of commandlets (Azure-specific commands) that you can use in PowerShell to create and manage Azure resources. You can find further information about Azure PowerShell online, at [Azure PowerShell documentation](#). Like the CLI, PowerShell is available for Windows, macOS, and Linux.
- *Azure Resource Manager templates.* An Azure Resource Manager template describes the service (or services) that you want to deploy in a text file, in a format known as JSON (JavaScript Object Notation). The example below shows a template that you can use to provision an instance of Azure SQL Database.

JSON

Copy

```
"resources": [  
  {  
    "name": "sql-server-dev",  
    "type": "Microsoft.Sql/servers",  
    "apiVersion": "2014-04-01-preview",  
    "location": "[parameters('location')]",  
    "tags": {  
      "displayName": "SqlServer"  
    },  
    "properties": {}  
  }  
]
```

You send the template to Azure using the `az deployment group create` command in the Azure CLI, or `New-AzResourceGroupDeployment` command in Azure PowerShell. For more information about creating and using Azure Resource Manager templates to provision Azure resources, see [What are Azure Resource Manager templates?](#)

Next unit: Describe provisioning Azure SQL Database

Continue >
