

Describe features and tools for managing and deploying Azure resources

1. Introduction

<https://learn.microsoft.com/en-us/training/modules/describe-features-tools-manage-deploy-azure-resources/1-introduction>

Introduction

Completed

This module introduces you to features and tools for managing and deploying Azure resources. You learn about the Azure portal (a graphic interface for managing Azure resources), the command line, and scripting tools that help deploy or configure resources. You also learn about Azure services that help you manage your on-premises and multicloud environments from within Azure.

Learning objectives

After completing this module, you'll be able to:

- Describe the Azure portal.
- Describe Azure Cloud Shell, including Azure CLI and Azure PowerShell.
- Describe the purpose of Azure Arc.
- Describe Azure Resource Manager (ARM), ARM templates, and Bicep.

2. Describe tools for interacting with Azure

<https://learn.microsoft.com/en-us/training/modules/describe-features-tools-manage-deploy-azure-resources/2-describe-interacting-azure>

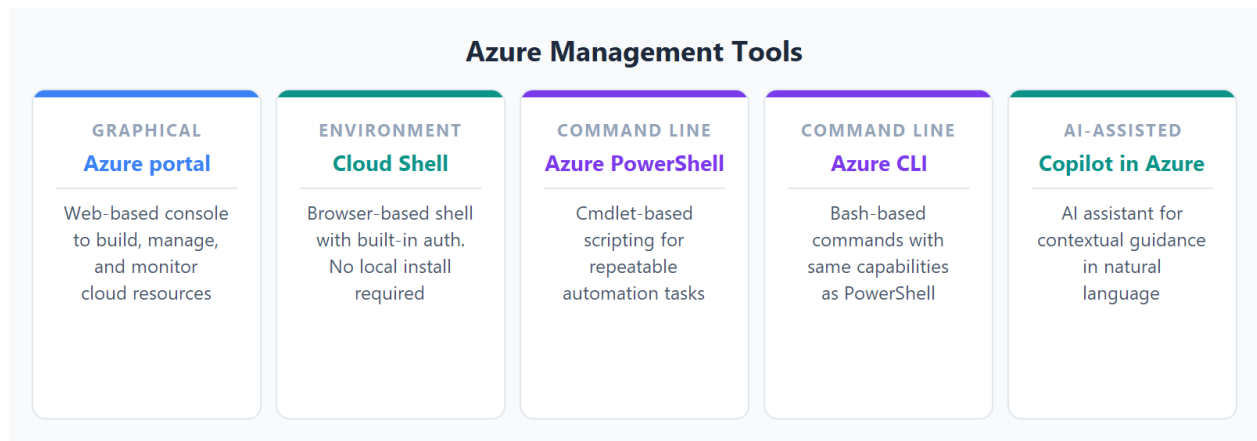
Describe tools for interacting with Azure

Completed

To get the most out of Azure, you need a way to interact with the Azure environment, the management groups, subscriptions, resource groups, resources, and so on. Azure provides multiple tools for managing your environment, including the:

- Azure portal
- Azure PowerShell
- Azure Command Line Interface (CLI)

Many teams also use Copilot in Azure to accelerate common operations tasks such as exploring services, understanding configuration options, and drafting commands or scripts.



AI-assisted operations with Copilot in Azure

Copilot in Azure is an AI assistant experience that can help administrators work faster by providing contextual guidance in natural language. Depending on your environment, some Copilot workflows can be agent-like, where the assistant helps coordinate multi-step tasks.

At a fundamentals level, treat Copilot as an operational assistant. You should still validate recommendations, confirm permissions, and review deployment changes before applying them in production.

What is the Azure portal?

The Azure portal is a web-based, unified console that provides an alternative to command-line tools. With the Azure portal, you can manage your Azure subscription by using a graphical user interface. You can:

- Build, manage, and monitor everything from simple web apps to complex cloud deployments
- Create custom dashboards for an organized view of resources
- Configure accessibility options for an optimal experience

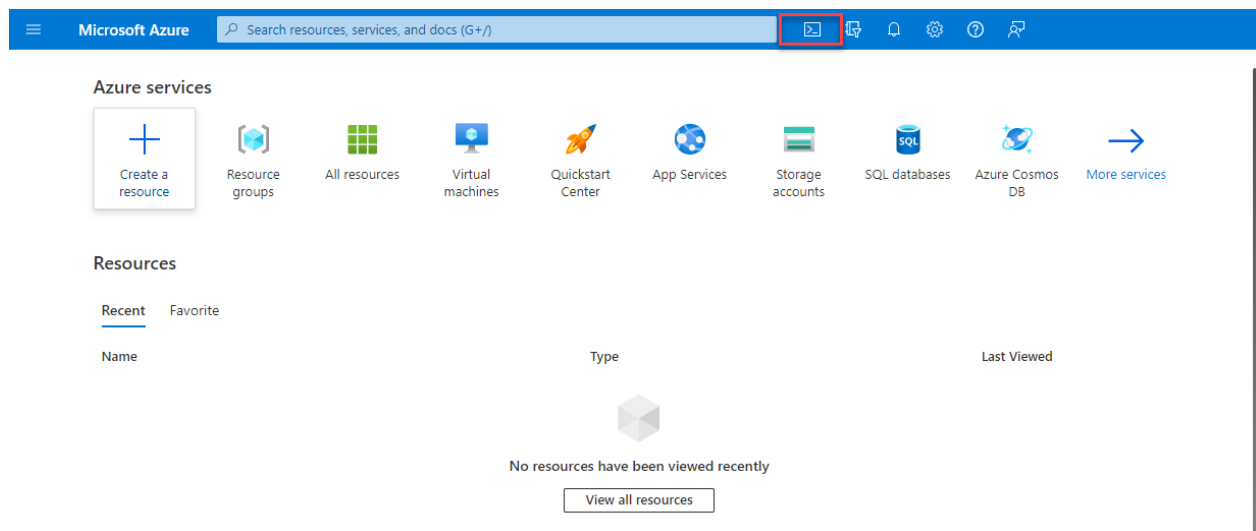
The following video introduces you to the Azure portal:

The Azure portal is designed for resiliency and continuous availability. It maintains a presence in every Azure datacenter. This configuration makes the Azure portal resilient to individual datacenter failures and avoids network slowdowns by being close to users. The Azure portal updates continuously and requires no downtime for maintenance activities.

Azure Cloud Shell

Azure Cloud Shell is a browser-based shell tool that allows you to create, configure, and manage Azure resources using a shell. Azure Cloud Shell support both Azure PowerShell and the Azure Command Line Interface (CLI), which is a Bash shell.

You can access Azure Cloud Shell via the Azure portal by selecting the Cloud Shell icon:



Azure Cloud Shell has several features that make it a unique offering to support you in managing Azure. Some of those features are:

- It is a browser-based shell experience, with no local installation or configuration required.
- It is authenticated to your Azure credentials, so when you log in it inherently knows who you are and what permissions you have.
- You choose the shell you're most familiar with; Azure Cloud Shell supports both Azure PowerShell and the Azure CLI (which uses Bash).

What is Azure PowerShell?

Azure PowerShell is a shell with which developers, DevOps, and IT professionals can run commands called command-lets (cmdlets). These commands call the Azure REST API to perform management tasks in Azure. Cmdlets can be run independently to handle one-off changes, or they may be combined to help orchestrate complex actions such as:

- The routine setup, teardown, and maintenance of a single resource or multiple connected resources.
- The deployment of an entire infrastructure, which might contain dozens or hundreds of resources, from imperative code.

Capturing the commands in a script makes the process repeatable and automatable.

In addition to being available via Azure Cloud Shell, you can install and configure Azure PowerShell on Windows, Linux, and Mac platforms.

What is the Azure CLI?

The Azure CLI is functionally equivalent to Azure PowerShell, with the primary difference being the syntax of commands. While Azure PowerShell uses PowerShell commands, the Azure CLI uses Bash commands.

The Azure CLI provides the same benefits of handling discrete tasks or orchestrating complex operations through code. It's also installable on Windows, Linux, and Mac platforms, as well as through Azure Cloud Shell.

Due to the similarities in capabilities and access between Azure PowerShell and the Bash based Azure CLI, it mainly comes down to which language you're most familiar with.

3. Describe the purpose of Azure Arc

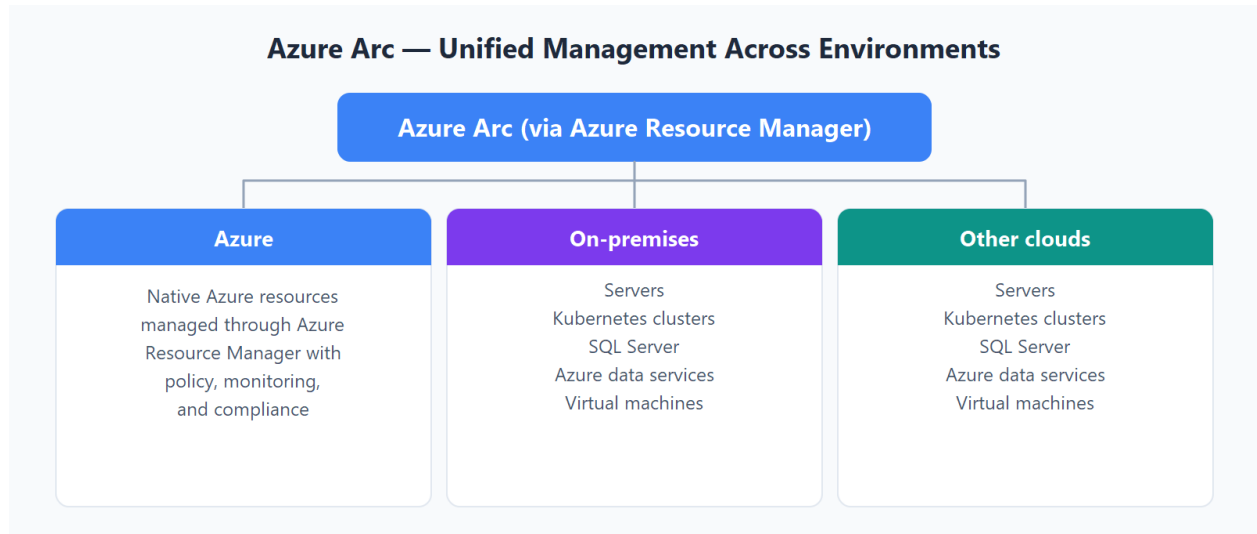
<https://learn.microsoft.com/en-us/training/modules/describe-features-tools-manage-deploy-azure-resources/3-describe-purpose-of-azure-arc>

Describe the purpose of Azure Arc

Completed

Managing hybrid and multicloud environments can rapidly get complicated. Azure provides a host of tools to provision, configure, and monitor Azure resources. What about the on-premises resources in a hybrid configuration or the cloud resources in a multicloud configuration?

Azure Arc works with Azure Resource Manager to extend your Azure compliance and monitoring to hybrid and multicloud configurations. Azure Arc simplifies governance and management by delivering a consistent multicloud and on-premises management platform.



Azure Arc provides a centralized, unified way to:

- Manage your entire environment together by projecting your existing non-Azure resources into Azure Resource Manager.
- Manage multicloud and hybrid virtual machines, Kubernetes clusters, and databases as if they are running in Azure.
- Use familiar Azure services and management capabilities, regardless of where they live.
- Continue using traditional ITOps while introducing DevOps practices to support new cloud and native patterns in your environment.
- Configure custom locations as an abstraction layer on top of Azure Arc-enabled Kubernetes clusters and cluster extensions.

Practical scenario

Imagine your organization runs workloads in Azure, an on-premises datacenter, and another public cloud. With Azure Arc, operations teams can apply consistent governance, policy, and inventory tracking across these environments from Azure, instead of managing each environment with disconnected tools.

What can Azure Arc do outside of Azure?

Currently, Azure Arc allows you to manage the following resource types hosted outside of Azure:

- Servers
- Kubernetes clusters
- Azure data services
- SQL Server

- Virtual machines (preview)

4. Describe Azure Resource Manager and Azure ARM templates

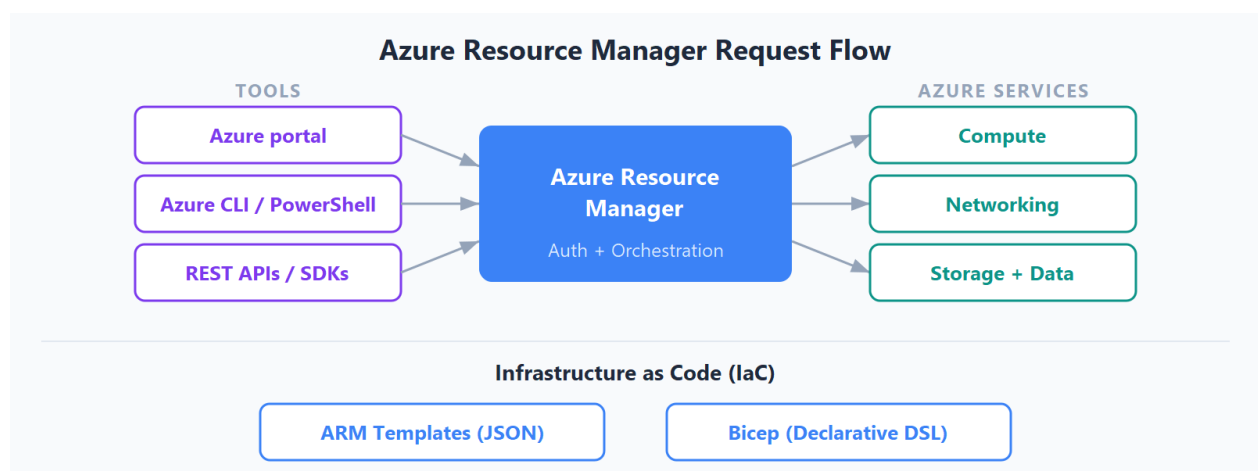
<https://learn.microsoft.com/en-us/training/modules/describe-features-tools-manage-deploy-azure-resources/4-describe-azure-resource-manager-azure-arm-templates>

Describe Azure Resource Manager and Azure ARM templates

Completed

Azure Resource Manager is the deployment and management service for Azure. It provides a management layer that enables you to create, update, and delete resources in your Azure account. Anytime you do anything with your Azure resources, Azure Resource Manager is involved.

When a user sends a request from any of the Azure tools, APIs, or SDKs, Azure Resource Manager receives the request. Azure Resource Manager authenticates and authorizes the request. Then, Azure Resource Manager sends the request to the Azure service, which takes the requested action. You see consistent results and capabilities in all the different tools because all requests are handled through the same API.



Azure Resource Manager benefits

With Azure Resource Manager, you can:

- Manage your infrastructure through declarative templates rather than scripts. A Resource Manager template is a JSON file that defines what you want to deploy to Azure.

- Deploy, manage, and monitor all the resources for your solution as a group, rather than handling these resources individually.
- Re-deploy your solution throughout the development life-cycle and have confidence your resources are deployed in a consistent state.
- Define the dependencies between resources, so they're deployed in the correct order.
- Apply access control to all services because RBAC is natively integrated into the management platform.
- Apply tags to resources to organize your subscription and support cost reporting.

The following video provides an overview of Azure Resource Manager.

Infrastructure as code

Infrastructure as code (IaC) means managing infrastructure through code and templates instead of manual configuration. At a fundamentals level, this can start with Azure CLI or Azure PowerShell and grow into repeatable environment deployments by using Azure Resource Manager templates and Bicep.

Azure Resource Manager templates

Azure Resource Manager templates describe desired Azure resources in declarative JSON. Azure validates the template before deployment, then orchestrates resource creation in the right order and in parallel when possible. Teams define the desired end state, and Azure Resource Manager handles deployment execution.

Templates can also call PowerShell or Bash deployment scripts when setup steps are needed before or after resource creation.

Benefits of using Azure Resource Manager templates

Azure Resource Manager templates provide several key benefits:

- **Declarative syntax:** Define what to deploy instead of writing step-by-step deployment commands.
- **Repeatable results:** Reuse the same template across environments for consistent outcomes.
- **Orchestration:** Azure Resource Manager handles dependency order and parallel deployment automatically.
- **Modularity:** Split templates into reusable components and nested templates.

- **Extensibility:** Add deployment scripts when additional setup actions are required.

Bicep

Bicep is a declarative language for deploying Azure resources through ARM. Compared to JSON ARM templates, Bicep is generally simpler and more concise.

Benefits of Bicep include:

- **Support for current Azure resources:** Bicep tracks Azure resource types and API versions.
- **Simple syntax:** Bicep is easier to read and write than equivalent JSON templates.
- **Repeatable deployments:** Bicep files are idempotent for consistent lifecycle deployments.
- **Built-in orchestration:** Azure Resource Manager handles dependencies and parallel deployment execution.
- **Modularity:** Reuse logic by organizing deployments into Bicep modules.

5. Module assessment

<https://learn.microsoft.com/en-us/training/modules/describe-features-tools-manage-deploy-azure-resources/5-knowledge-check>

Module assessment

Completed

6. Summary

<https://learn.microsoft.com/en-us/training/modules/describe-features-tools-manage-deploy-azure-resources/6-summary>

Summary

Completed

In this module, you were introduced to features and tools for managing and deploying Azure resources. You learned about the Azure portal (a graphic interface for managing Azure resources), command line, and scripting tools that help deploy or configure resources. You also learned about Azure services that help you manage your on-premises and multicloud environment from Azure.

You also learned where Copilot in Azure can assist with discovery, guidance, and multistep operational workflows while keeping administrators in control of final decisions.

Learning objectives

You should now be able to:

- Describe Azure portal.
- Describe Azure Cloud Shell, including Azure CLI and Azure PowerShell.
- Describe the purpose of Azure Arc.
- Describe Azure Resource Manager (ARM), ARM templates, and Bicep.
- Describe how Copilot in Azure can assist with operations tasks while humans remain responsible for validation and approval.

Additional resources

The following resources provide more information on topics in this module or related to this module.

- [Implement resource management security](#) in Azure is a Microsoft Learn learning path that provides more information on managing Azure resources.

Explore with Copilot

Tip

Try one of these prompts in Copilot Chat:

- "Compare Azure portal, Cloud Shell, Azure CLI, and Azure PowerShell across common admin tasks and explain when each is the best choice."
- "Design an agent-assisted deployment workflow with ARM templates or Bicep that includes human approval checkpoints for validation, staged rollout, and rollback decisions."
- "Explain how Azure Arc extends Azure management to on-premises and multicloud resources with one practical operations scenario."