**Lab Manual- Manage Azure Resources with ARM/BICEP Teamplate**

Table of Contents

[1 OBJECTIVE 3](#_Toc112511876)

[2 BICEP Playground 3](#_Toc112511877)

[3 PRE-REQUISISTE 4](#_Toc112511878)

[4 Provision Resource with ARM/Bicep with QuickStart 4](#_Toc112511879)

[4.1 **Build the Template with Github – Review the Quick Start option** 4](#_Toc112511880)

[4.2 **Visualize Template with Github – Review the Quick Start option** 9](#_Toc112511881)

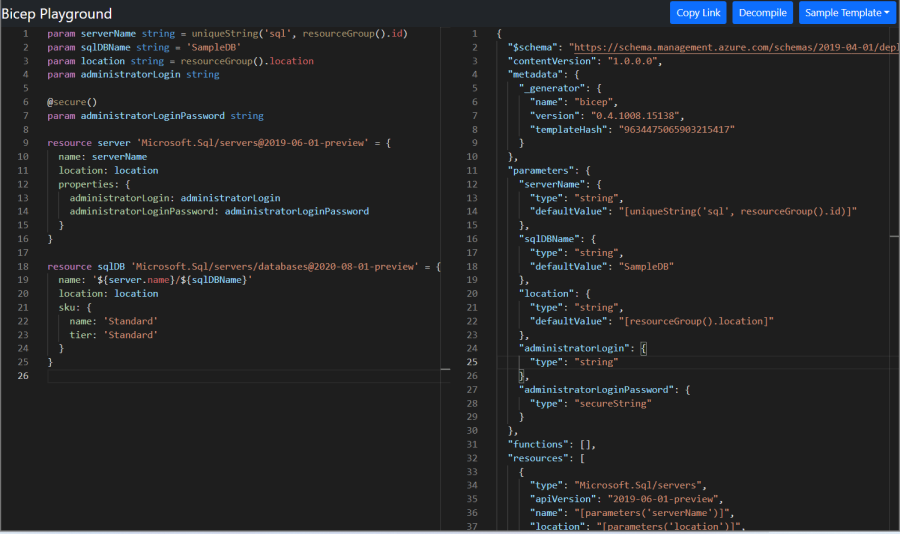
# OBJECTIVE

When deploying Azure resources with an Infrastructure as Code tool, you need to understand what resource types are available, and what values to use in your files

**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.

**Bicep** is a domain-specific language (DSL) that uses declarative syntax to deploy Azure resources similer to ARM . Syntax of Biceps is shorter than ARM. During deployment, the Bicep CLI converts a Bicep file into ARM template JSON.

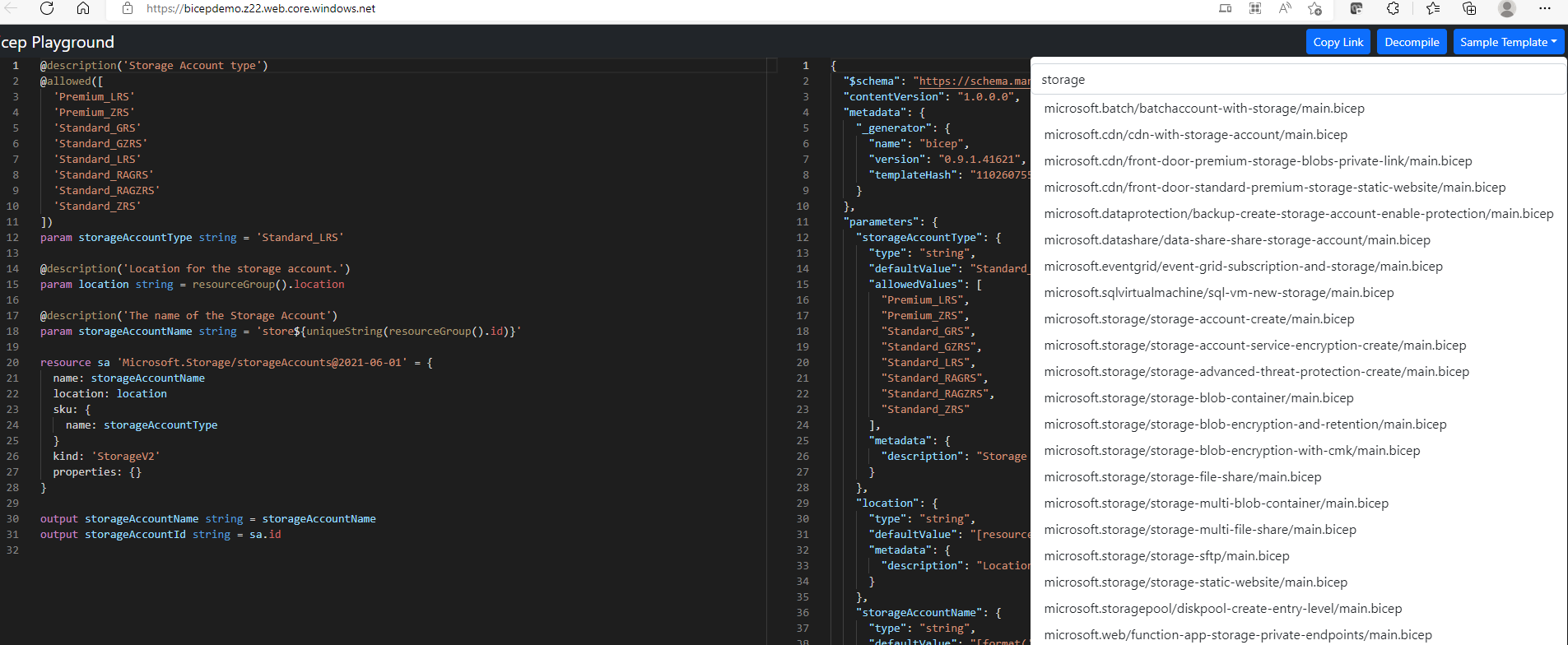
The following examples show the difference between a Bicep file and the equivalent JSON template. Both examples deploy a SQL Server .



# BICEP Playground

The [Bicep Playground](https://aka.ms/bicepdemo) lets you view Bicep and equivalent JSON side by side. You can compare the implementations of the same infrastructure.

<https://bicepdemo.z22.web.core.windows.net/>



# PRE-REQUISISTE

* Accounts in Azure
* A local Computer with 4 CPU, 16 GB RAM, 200 GB disk space

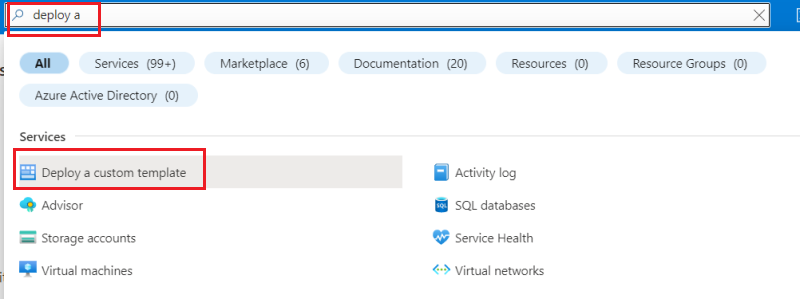
# Provision Resource with ARM/Bicep with QuickStart

In this quickstart, you learn how to create an Azure Resource Manager template (ARM template) in the Azure portal. You edit and deploy the template from the portal.

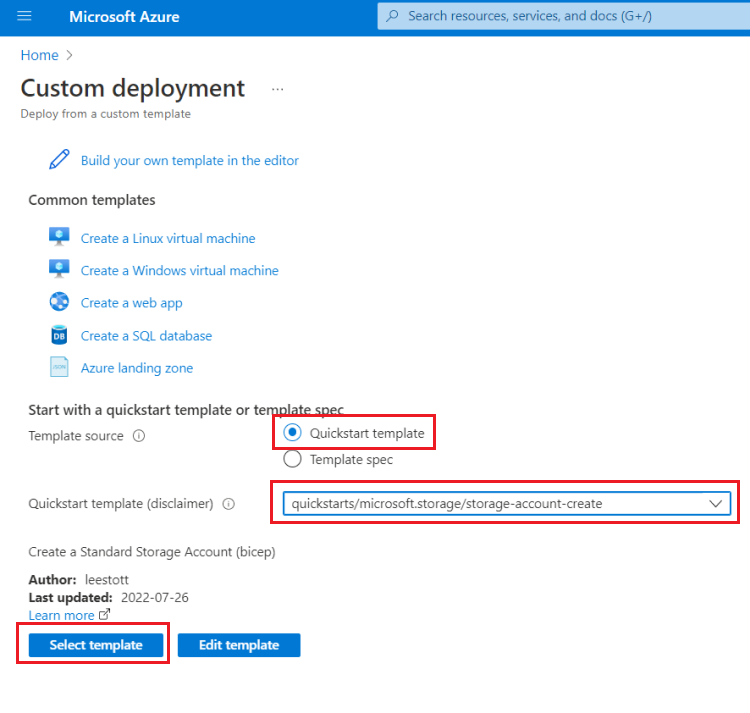
After completing the tutorial, you deploy an Azure Storage account. The same process can be used to deploy other Azure resources.

## **Build the Template with Github – Review the Quick Start option**

1. In a web browser, go to the [Azure portal](https://portal.azure.com/) and sign in.
2. From the Azure portal search bar, search for **deploy a custom template** and then select it from the available options.

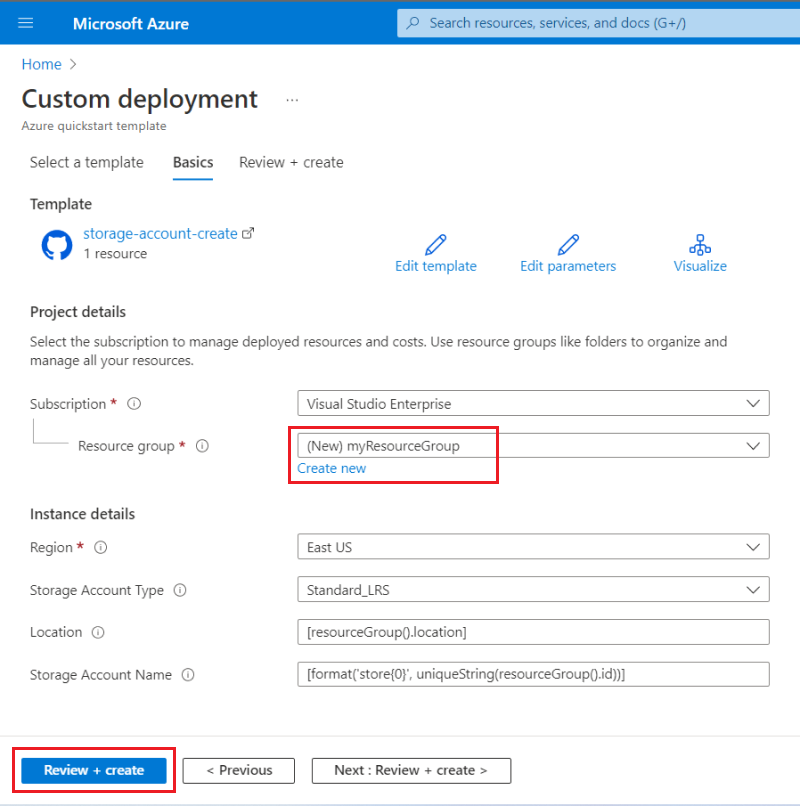


1. For **Template** source, notice that **Quickstart template** is selected by default. You can keep this selection. In the drop-down, search for quickstarts/microsoft.storage/storage-account-create and select it. After finding the quickstart template, select **Select template.**

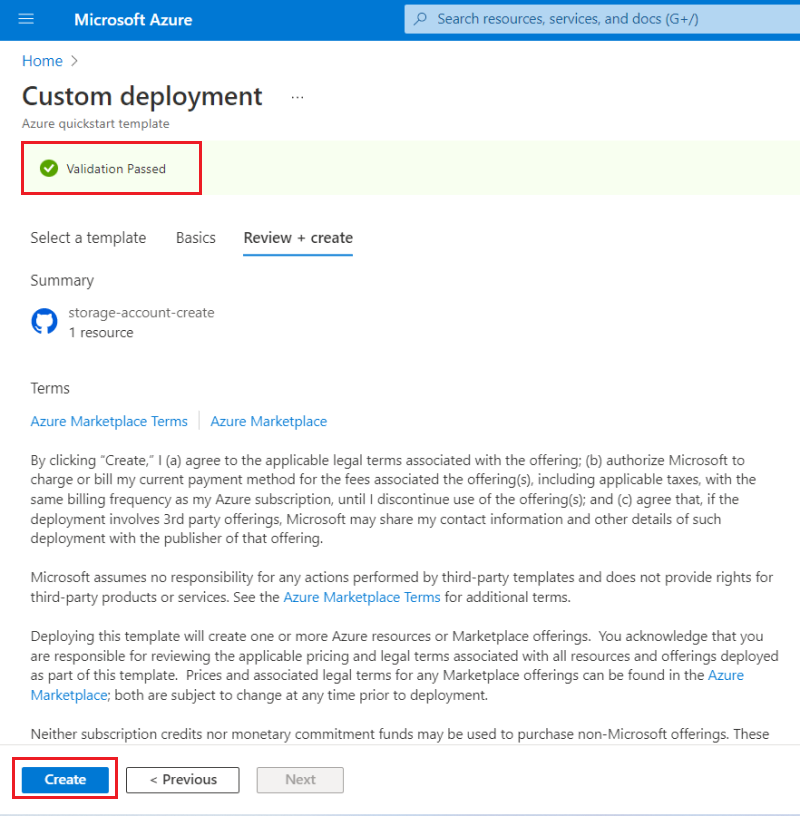


1. In the next blade, you provide custom values to use for the deployment.

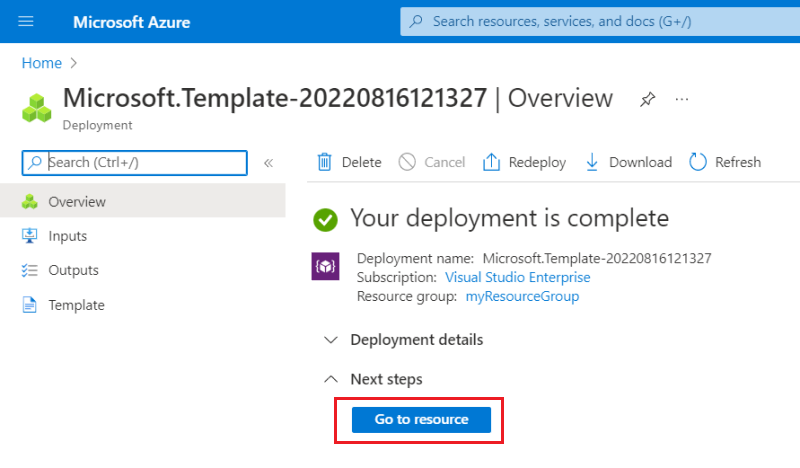
For **Resource group**, select **Create new** and provide myResourceGroup for the name. You can use the default values for the other fields. When you've finished providing values, select **Review + create**.



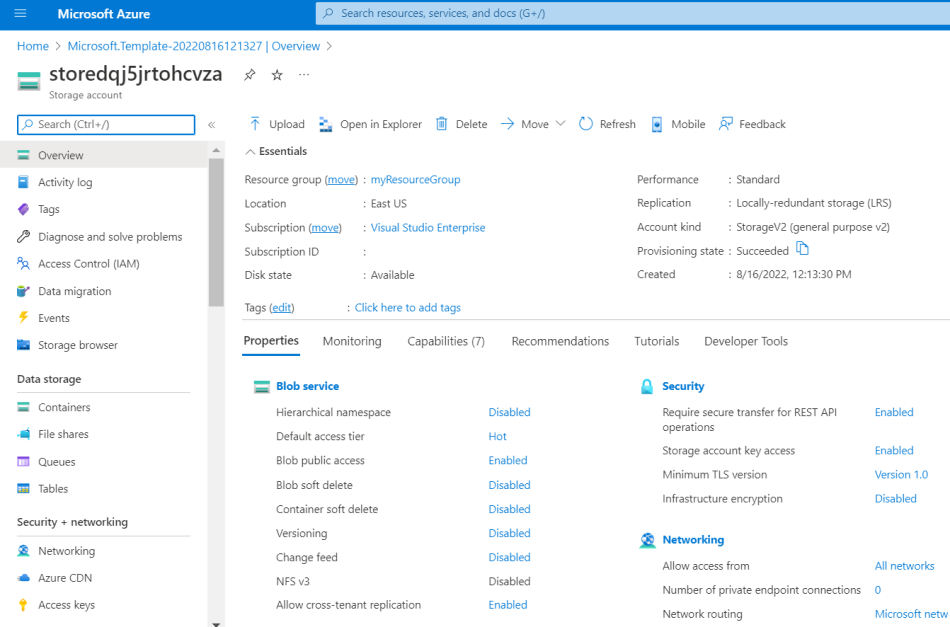
1. The portal validates your template and the values you provided. After validation succeeds, select **Create** to start the deployment.



1. Once your validation has passed, you'll see the status of the deployment. When it completes successfully, select **Go to resource** to see the storage account.



1. From this screen, you can view the new storage account and its properties.



## **Visualize Template with Github – Review the Quick Start option**

1. Select **Create-AD-Forest-with-Subdomain** template



1. Now Click **Edit Template**

Graphical user interface, text, application, email

Description automatically generated

1. Review the Code

Graphical user interface, text, application, email

Description automatically generated

1. Come Back to deployment screen

Graphical user interface, application

Description automatically generated

1. Click Visualize to Visualize the Resources it is going to created



Graphical user interface, diagram

Description automatically generated with medium confidence