**What are ARM templates?**

To implement infrastructure as code for your Azure solutions, use Azure Resource Manager (ARM) templates. The template is a JavaScript Object Notation (JSON) file that defines the infrastructure and configuration for your project. The template uses declarative syntax, which lets you state what you intend to deploy without having to write the sequence of programming commands to create it. In the template, you specify the resources to deploy and the properties for those resources.

## Why choose ARM templates?

If you're trying to decide between using ARM templates and one of the other infrastructure as code services, consider the following advantages of using templates:

* **Declarative syntax**: ARM templates allow you to create and deploy an entire Azure infrastructure declaratively. For example, you can deploy not only virtual machines, but also the network infrastructure, storage systems and any other resources you may need.
* **Repeatable results**: Repeatedly deploy your infrastructure throughout the development lifecycle and have confidence your resources are deployed in a consistent manner. Templates are idempotent, which means you can deploy the same template many times and get the same resource types in the same state. You can develop one template that represents the desired state, rather than developing lots of separate templates to represent updates.
* **Modular files**: You can break your templates into smaller, reusable components and link them together at deployment time. You can also nest one template inside another templates.
* **Create any Azure resource**: You can immediately use new Azure services and features in templates. As soon as a resource provider introduces new resources, you can deploy those resources through templates. You don't have to wait for tools or modules to be updated before using the new services.
* **Extensibility**: With [deployment scripts](https://docs.microsoft.com/en-us/azure/azure-resource-manager/templates/deployment-script-template), you can add PowerShell or Bash scripts to your templates. The deployment scripts extend your ability to set up resources during deployment. A script can be included in the template, or stored in an external source and referenced in the template. Deployment scripts give you the ability to complete your end-to-end environment setup in a single ARM template.

## Template file

Within your template, you can write [template expressions](https://docs.microsoft.com/en-us/azure/azure-resource-manager/templates/template-expressions) that extend the capabilities of JSON. These expressions make use of the [functions](https://docs.microsoft.com/en-us/azure/azure-resource-manager/templates/template-functions) provided by Resource Manager.

The template has the following sections:

* [Parameters](https://docs.microsoft.com/en-us/azure/azure-resource-manager/templates/template-parameters) - Provide values during deployment that allow the same template to be used with different environments.
* [Variables](https://docs.microsoft.com/en-us/azure/azure-resource-manager/templates/template-variables) - Define values that are reused in your templates. They can be constructed from parameter values.
* [User-defined functions](https://docs.microsoft.com/en-us/azure/azure-resource-manager/templates/template-user-defined-functions) - Create customized functions that simplify your template.
* [Resources](https://docs.microsoft.com/en-us/azure/azure-resource-manager/templates/template-syntax#resources) - Specify the resources to deploy.
* [Outputs](https://docs.microsoft.com/en-us/azure/azure-resource-manager/templates/template-outputs) - Return values from the deployed resources.

Templates are JSON files. To create templates, you need a good JSON editor. We recommend Visual Studio Code with the Resource Manager Tools extension

## Create an ARM template

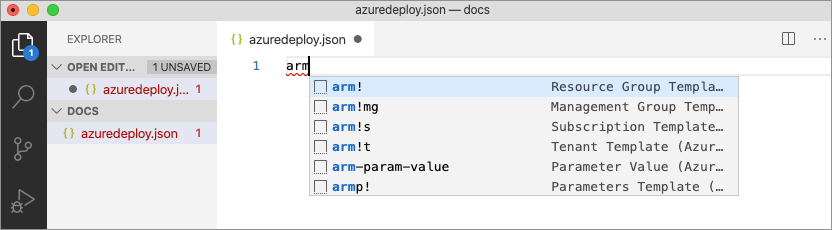
Create and open with Visual Studio Code a new file named azuredeploy.json. Enter arm into the code editor, which initiates Azure Resource Manager snippets for scaffolding out an ARM template.

Select arm! to create a template scoped for an Azure resource group deployment.

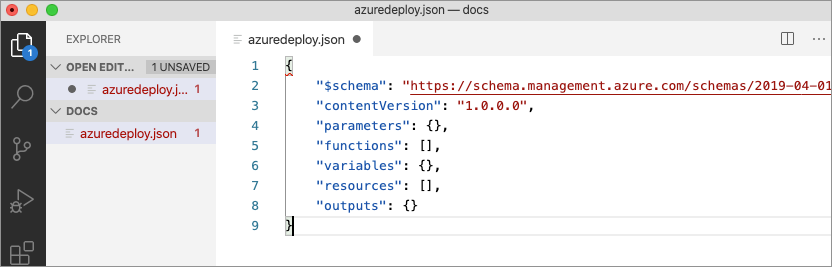
## Create an ARM template

Create and open with Visual Studio Code a new file named azuredeploy.json. Enter arm into the code editor, which initiates Azure Resource Manager snippets for scaffolding out an ARM template.

Select arm! to create a template scoped for an Azure resource group deployment.



This snippet creates the basic building blocks for an ARM template.



Notice that the Visual Studio Code language mode has changed from *JSON* to *Azure Resource Manager Template*. The extension includes a language server specific to ARM templates which provides ARM template-specific validation, completion, and other language services.