Manage your Azure Cloud with ARM Templates

Basic Structure

Here is the basic structure of an Azure ARM template:

- Mandatory
- Layout of ARM template
- Defines ARM template design

Schema

- Mandatory
- Convenience and tacking

Content Version

- Not mandatory
- Provides flexibility
- Values provided with customized deployment

Parameters

- Not mandatory
- Values used to simplify language expressions

Variables

- Mandatory
- Key component of ARM templates
- Defines collection of resources

Resources

- Not mandatory
- Values returned after deployment

Output

Key Benefits

Some of the key benefits of using the Azure Resource Manager are:

- Template-driven
- Multi-service
- Idempotent
- Declarative
- Multi-region
- Access control

What You Need to Know: Introduction to ARM Templates

Manage your Azure Cloud with ARM Templates

Key Tools

Notepad

- Can edit JSON templates
- Limited functionality

Visual Studio

- Easy way to start
- Automatically generates JSON for Resource groups you create

Visual Studio Code

- One of the best IDEs
- Lightweight, loads fast
- Lacks in ARM authoring support

Sublime Text

- ARM package, quick authoring
- Not very stable