Module 13:
ARM Template

Infrastructure as code

- *Infrastructure as code* enables you to describe, through code, the infrastructure that you need for your application.

- Infrastructure as code enables you to maintain both your application code and everything you need to deploy your application in a central code repository. In that repository, you can build, test, and deploy as a unit.

"The process of managing and provisioning computing infrastructure and their configuration through machine processable definition files" - Wikipedia

Tools for IaC – Infrastructure Provisioning

- ARM Template by Microsoft
- CloudFormation by AWS
- Heat by OpenStack
- Terraform by Hashicorp

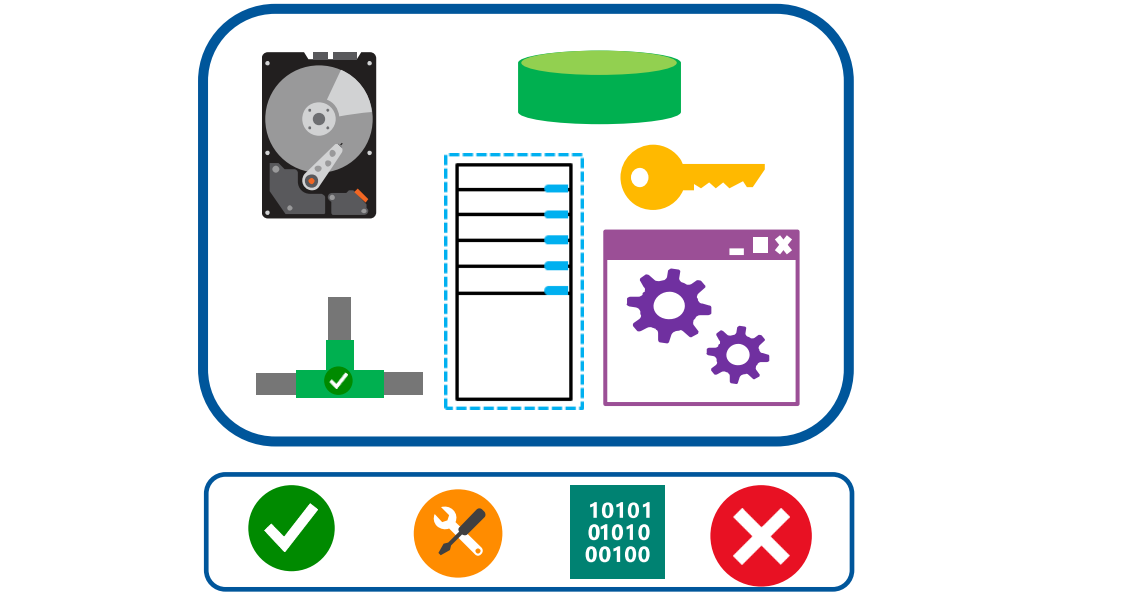
Benefits of IaC

Why Use ARM templates?

- Make deployments faster and more repeatable
- Improve consistency by providing a common language
- Enable you to deploy multiple resources in the correct order by mapping out resource dependencies
- Reduce manual, error-prone tasks
- Templates can be linked together to provide a modular solution

What is Azure Resource Manager?

- *Azure Resource Manager* is a management layer in which a resource group and all the resources within it are created, configured, managed, and deleted



JavaScript Object Notation (JSON)

- Lightweight data-interexchange format based on a subset of JavaScript

- JSON is based on two structure types
  - A collection of name/value pairs.
  - An ordered list of values.

- Supports schemas that enable intellisense/autocomplete in JSON supported editors

JSON Basics

Each JSON file has a start and end bracket

```
{

}
```

JSON objects are defined within the bracket. Syntax for a property is name : value

String values use double quotes "

```
{
  "name": "value",
  "inchesInFoot": 12
}
```

Defining Arrays

Arrays are a single property with multiple values
Values are defined within [ ] brackets

```
{
"availableColors": [
    "blue",
    "red",
    "white"
  ]
}
```

Template components

JSON stores data stored as an object in text. A JSON document is a collection of key-value pairs

JSON files can contain the following sections:

- Parameters
- Variables
- Functions
- Resources
- Outputs

Anatomy of ARM Template

- $schema
  - The URL to the JSON schema that defines the version of the template language
- contentVersion
  - Version of your template. This is useful to ensure you are deploying the correct version of the template
- parameters
  - Define inputs for the template
- variables
  - Custom values usually created from parameters or output from other templates
- resources
  - What resources in Azure (VMs, Databases, etc) the template actually defines
- outputs
  - Return values (if any) that the template produces

```
{
    "$schema":
"https://schema.management.azure.com/schem
as/2015-01-01/deploymentTemplate.json#",
    "contentVersion": "1.0.0.0",
    "parameters": {
    },
    "variables": {
    },
    "resources": [
    ],
    "outputs": {
    }
}
```

Template Parameters

- Input options that can be specified at template execution time.
- Can be overridden with separate parameters files

```
"parameters": {
   "<parameterName>" : {
     "type" : "<type-of-parameter-value>",
     "defaultValue": "<optional-default-value-of-parameter>",
     "allowedValues": [ "<optional-array-of-allowed-values>" ],
     "minValue": <optional-minimum-value-for-int-parameters>,
     "maxValue": <optional-maximum-value-for-int-parameters>,
     "minLength": <optional-minimum-length-for-string-secureString-array-parameters>,
     "maxLength": <optional-maximum-length-for-string-secureString-array-parameters>
   }
}
```

Parameter Types

- Allowed Types
  - string or secureString - any valid JSON string
  - int - any valid JSON integer
  - bool - any valid JSON boolean
  - object or secureObject - any valid JSON object
  - array - any valid JSON array

Parameters Examples

```
"StorageAccountUniqueName": {
   "type": "string",
   "metadata": {
     "description": "Unique name of storage account"
   }
},
```

Parameter description

```
"storageAccountType": {
          "type": "string",
          "defaultValue": "Standard_LRS",
          "allowedValues": [
              "Standard_LRS",
              "Standard_GRS",
              "Standard_RAGRS",
              "Premium_LRS"
          ]
      }
```

Default value and allowedValues

```
"instanceCount": {
     "type": "int",
     "minValue": 2,
     "maxValue": 100,
     "metadata": {
       "description": "Number of VM instances"
     }
   },
```

Minimum and Maximum values

Defining Variables

Named values that can store manipulated values from parameters or other resources

```
"parameters": {
    "username": {
      "type": "string"
    },
    "password": {
      "type": "secureString"
    }
 },
 "variables": {
    "connectionString": "[concat('Name=', parameters('username'),
';Password=', parameters('password'))]"
}
```

In this example **connectionString** is the variable and it is created by concatenating text and the user name and password parameters. It can be referenced in other resources

Helper Functions

Used to manipulate or return data from resources, parameter input or data from other resources

- Arithmetic, Array, Azure specific, Conversion, String, Template helpers

```
 "variables": {
      "usernameAndPassword": "[concat('parameters('username'),parameters('password'))]",
      "authorizationHeader": "[concat('Basic ', base64(variables('usernameAndPassword')))]"
    }
```

```
"websiteUri": {
      "type": "string",
      "value": "[concat('http://',reference(resourceId('Microsoft.Web/sites',
parameters('siteName'))).hostNames[0])]"
    }
```

```
"VMStorageName": "[concat('VMStorage', uniqueString(resourceGroup().id))]",
```

```
"location": "[resourceGroup().location]",
```

Resources

- The actual resource(s) the template will instantiate.
- Resources are defined out of resource providers
- Resources have properties that can be read and set
- Resources can have dependencies on other resources

Microsoft.Compute/
virtualMachines

Microsoft.Storage/
storageAccounts

Microsoft.Network/
networkInterfaces

Microsoft.Network/
publicIPAddresses

Microsoft.Network/
virtualNetworks

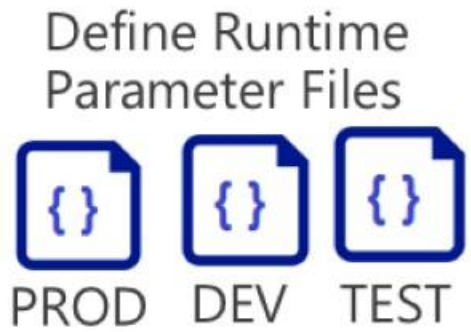
Creating multiple instances of a resource

- copy
  - Defines the number of iterations to make
- copyindex()
  - Returns the current index of the iteration. Used to make unique resource names.

```
"name": "[variables('uniqueStringArray')[copyIndex()]]",
"apiVersion": "2015-05-01-preview",
"copy": {
   "name": "storageLoop",
   "count": 5
},
```

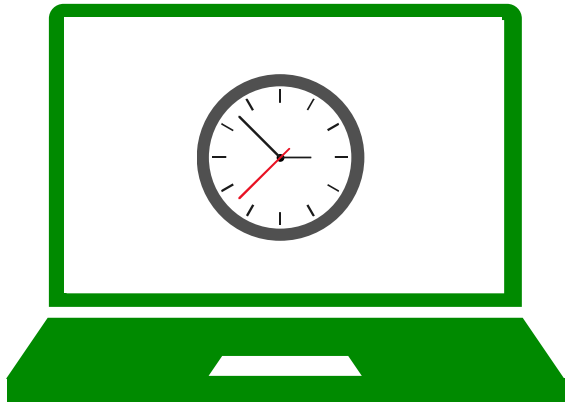
Parameter Files

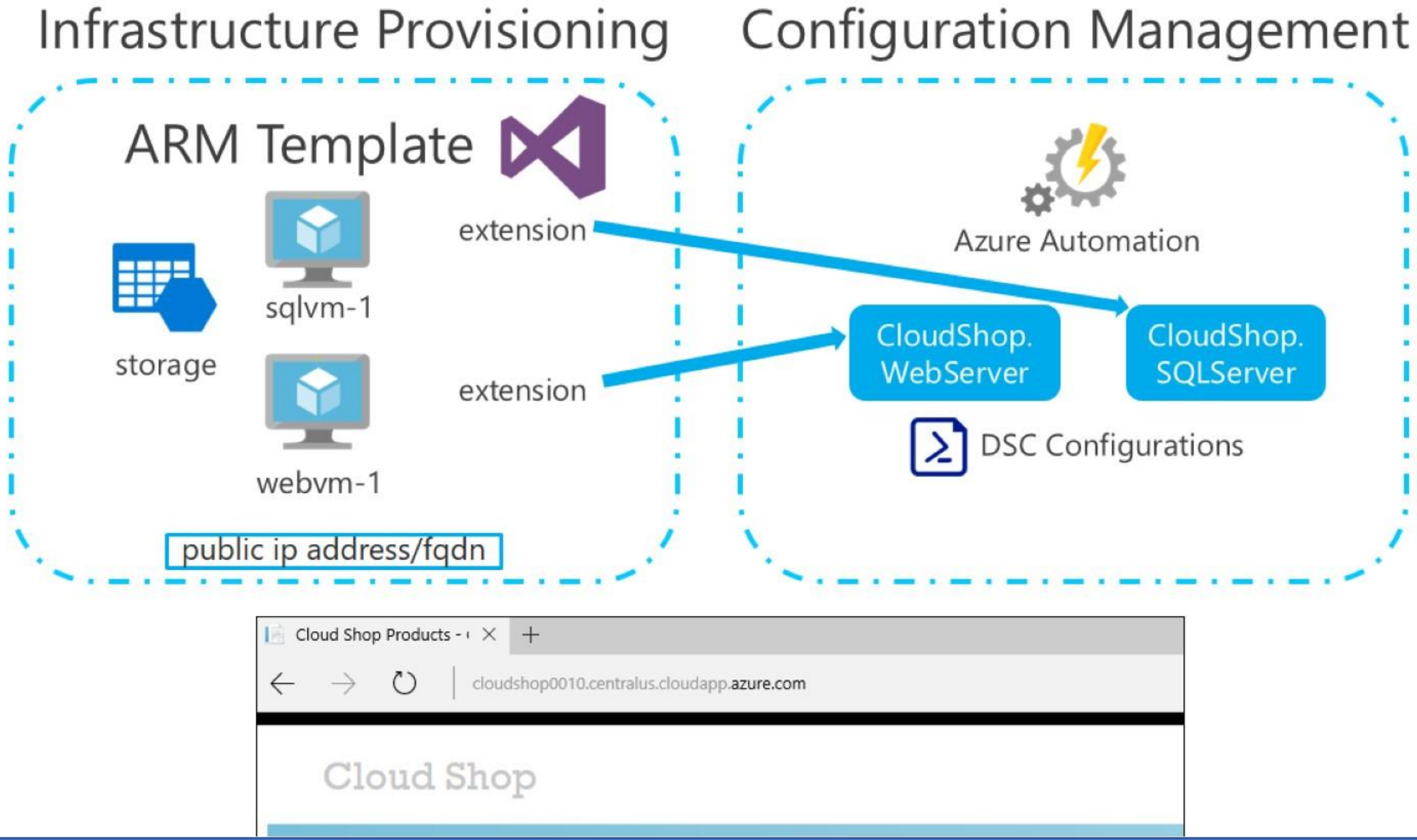
- Used to store settings specific to an environment



Demo:

Create & Deploy ARM Template using Visual Studio




Infrastructure Provisioning
Configuration Management
ARM Template
extension
Azure Automation
sqlvm-1
storage
extension
CloudShop.
WebServer
CloudShop.
SQLServer
DSC Configurations
webvm-1
public ip address/fqdn

Cloud Shop Products -
cloudshop0010.centralus.cloudapp.azure.com
Cloud Shop

Lab:

Authoring ARM Template using Visual Studio