



# KodeKloud

© Copyright KodeKloud

Follow us on <https://kodekloud.com/> to learn more about us.

# Administer Azure Resources

© Copyright KodeKloud

These Learn modules are part of the AZ-104: Prerequisites for Azure administrators  
(<https://docs.microsoft.com/learn/paths/az-104-administrator-prerequisites/learning-path>).

# Learning Objectives



Configure Azure Resources with Tools



Configure Resources with ARM Templates

# Learning Objectives



Configure Azure Resources with Tools



Compare Administrator Tools

# Learning Objectives

**B** Configure Resources with ARM Templates

**01** Azure Resource Manager

**02** Azure Resource Manager Templates

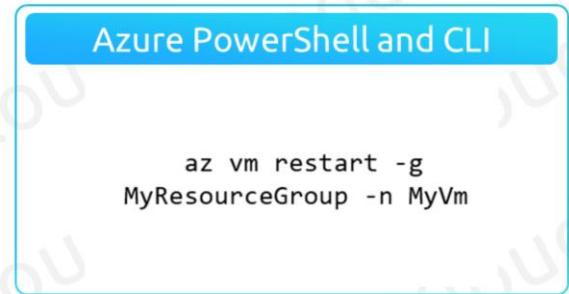
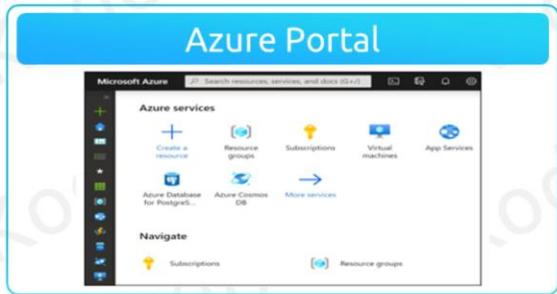
**03** ARM template structure

**04** Bicep



# Compare Administrator Tools

# Compare Administrator Tools



# Compare Administrator Tools

## Azure Portal



- Oversee and organize resources
- Graphical user interface
- Centralized platform for learning resources
- Customizable user experience
- Available as a mobile app
- Provides Cloud Shell access
- Suitable for single-instance creation tasks

## Azure Cloud Shell



Welcome to Azure Cloud Shell

Bash | PowerShell

- User-friendly, accessible via web browsers
- Choice of Bash or PowerShell
- Seamless sign-in with existing credentials
- Session-specific and individualized usage
- Sessions timeout after 20 minutes of inactivity

## Azure PowerShell and CLI

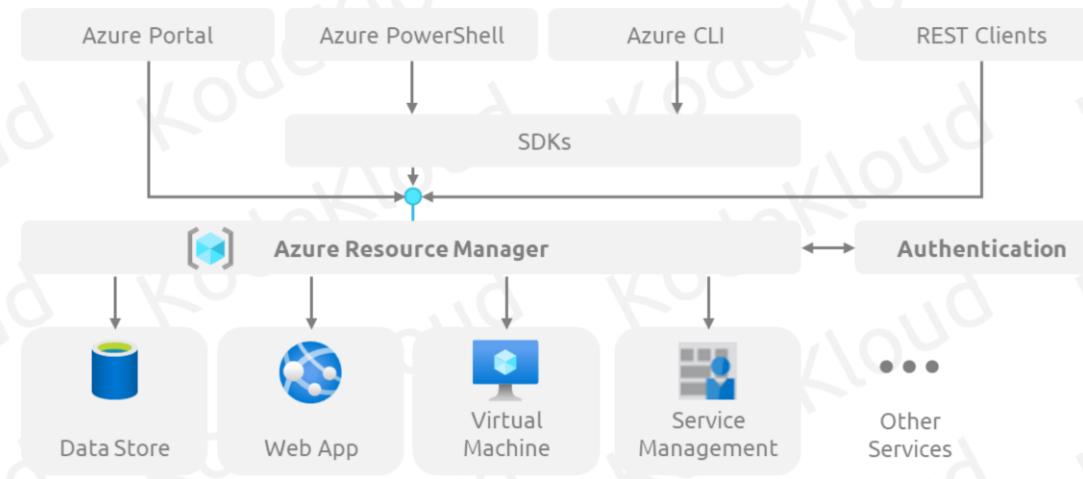
```
az vm restart -g  
MyResourceGroup -n MyVm
```

- Terminal-based utilities
- Supports command-line and script execution
- Compatible with multiple operating systems
- Ideal for consistent, automated deployments
- Traditional programming environment experience



# Azure Resource Manager

# Azure Resource Manager



© Copyright KodeKloud

## Management layer

Azure Resource Manager or ARM is the management layer responsible for creating, updating and managing resources.

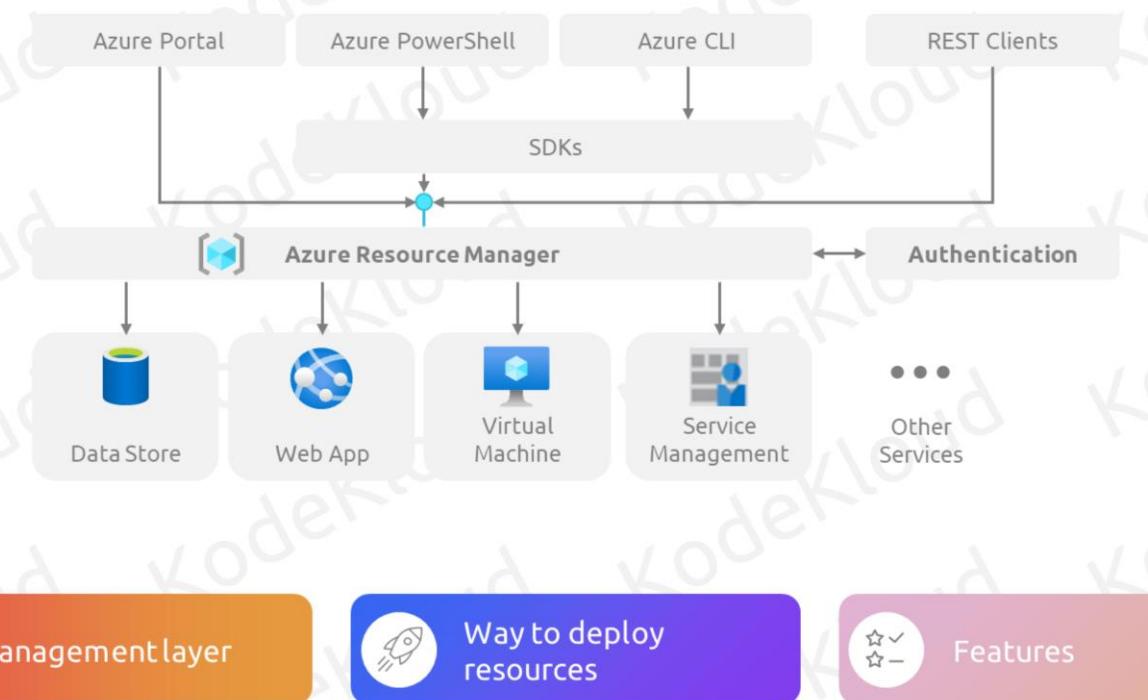
## Way to deploy resources

Regardless of whether you are using Azure Portal, Azure PowerShell, Azure CLI or REST API, Azure Resource Manager offers a way to deploy and manage the resources.

## **Features**

Access Control, Locks, Tags, Resource Groups, and Templates are some of the features offered by ARM, which was not available in the previous model – Azure Service Manager

# Azure Resource Manager



© Copyright KodeKloud

## Management layer

Azure Resource Manager or ARM is the management layer responsible for creating, updating and managing resources.

## Way to deploy resources

Regardless of whether you are using Azure Portal, Azure PowerShell, Azure CLI or REST API, Azure Resource Manager offers a way to deploy and manage the resources.

## **Features**

Access Control, Locks, Tags, Resource Groups, and Templates are some of the features offered by ARM, which was not available in the previous model – Azure Service Manager



# Azure Resource Manager Templates

# Azure Resource Manager (ARM) Templates

```
{  
  "$schema":  
    "https://schema.management.azure.com/schemas/2019-04-  
    01/deploymentTemplate.json#",  
  "contentVersion": "1.0.0.0",  
  "parameters": {},  
  "functions": [],  
  "variables": {},  
  "resources": [],  
  "outputs": {}  
}
```



Declarative automation



Consistent and  
reusable



Error prone tasks and  
simplify deployment



Linkable and  
help in complex  
deployment

© Copyright KodeKloud

## Declarative automation

ARM templates uses JSON file. In declarative automation, you need to declare the resources but not how to create them. Creating the resources is Resource Manager's responsibility.

## Consistent and reusable

Environments deployed via ARM template will be consistent. With the help of parameters, we can share and reuse the

template to create environment from scratch.

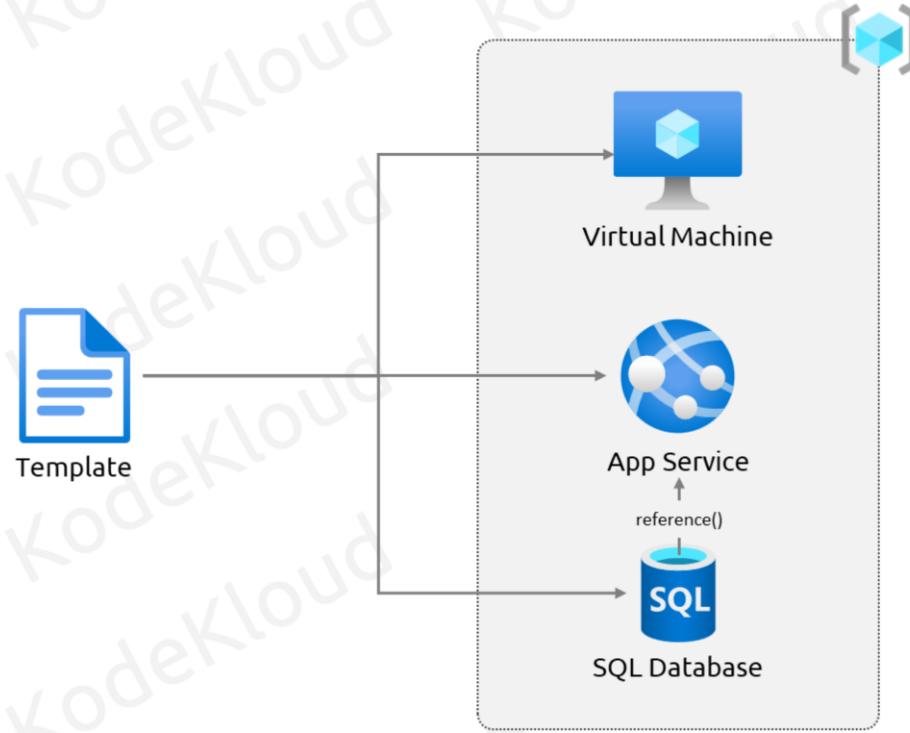
### **Error prone tasks and simplify deployment**

If we are creating environment manually chances of human error will be there and with ARM templates, we can deploy all the resources we define in a single operation.

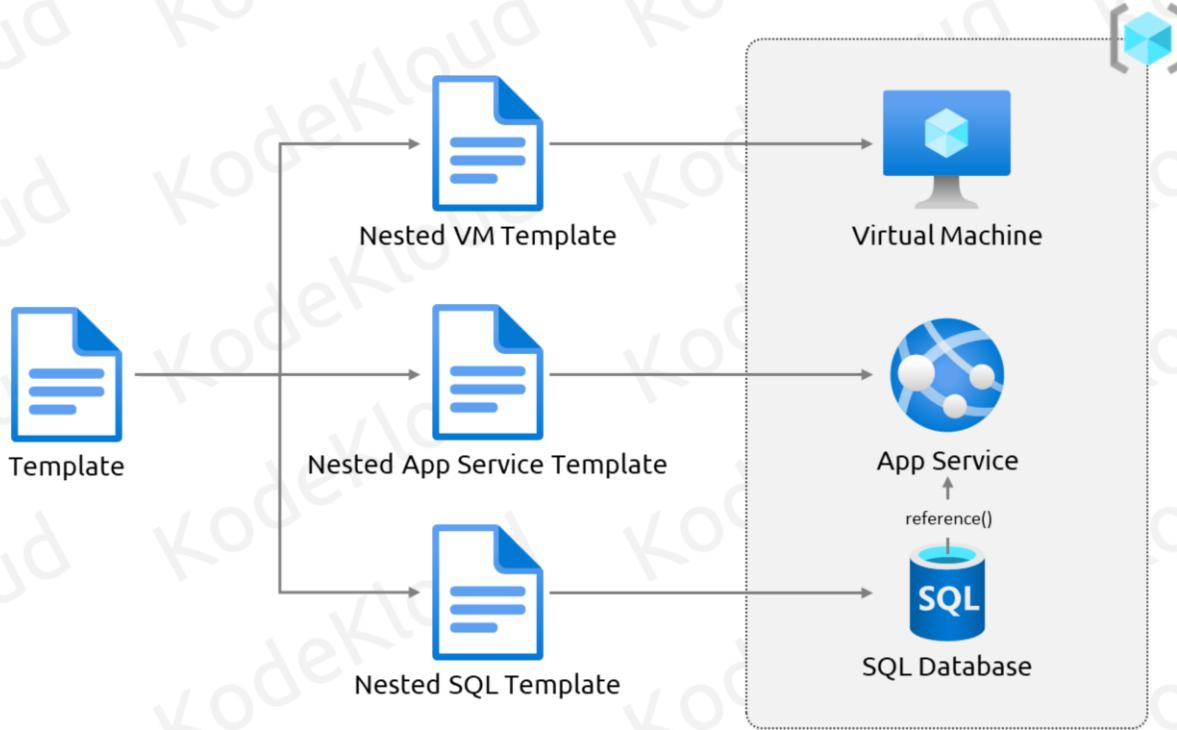
### **Linkable and helps complex deployment**

You can write small ARM templates and link them to a parent template. This helps in managing different parts of the template efficiently. With ARM templates, we can deploy complex environments in the correct dependency order.

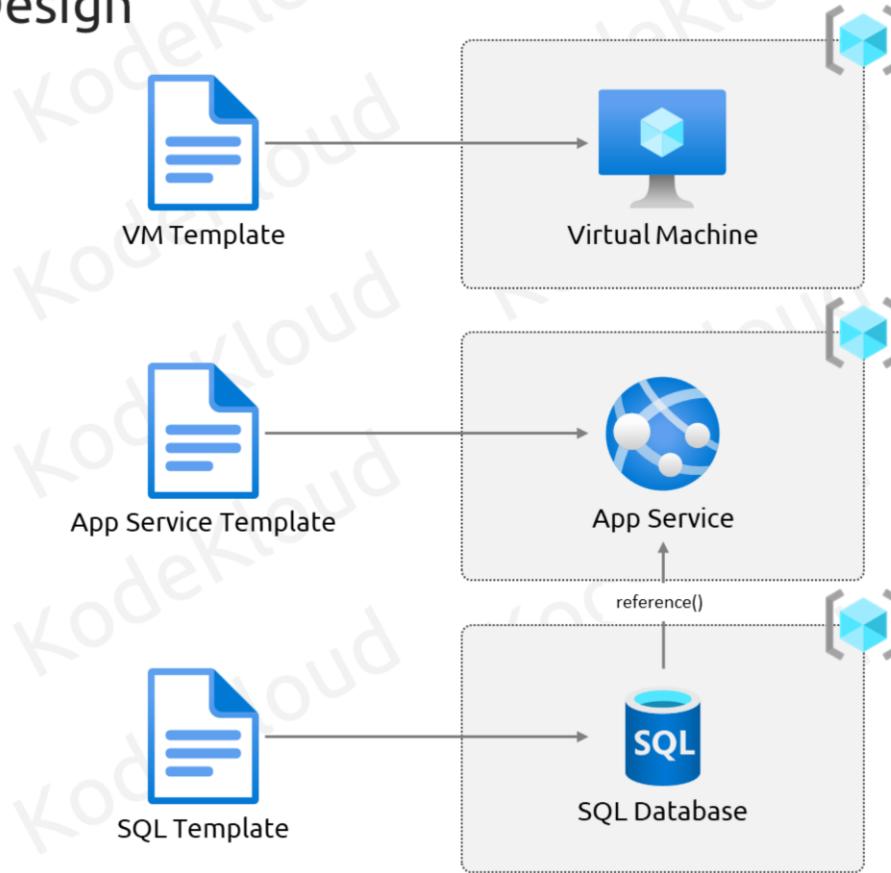
# ARM Template Design



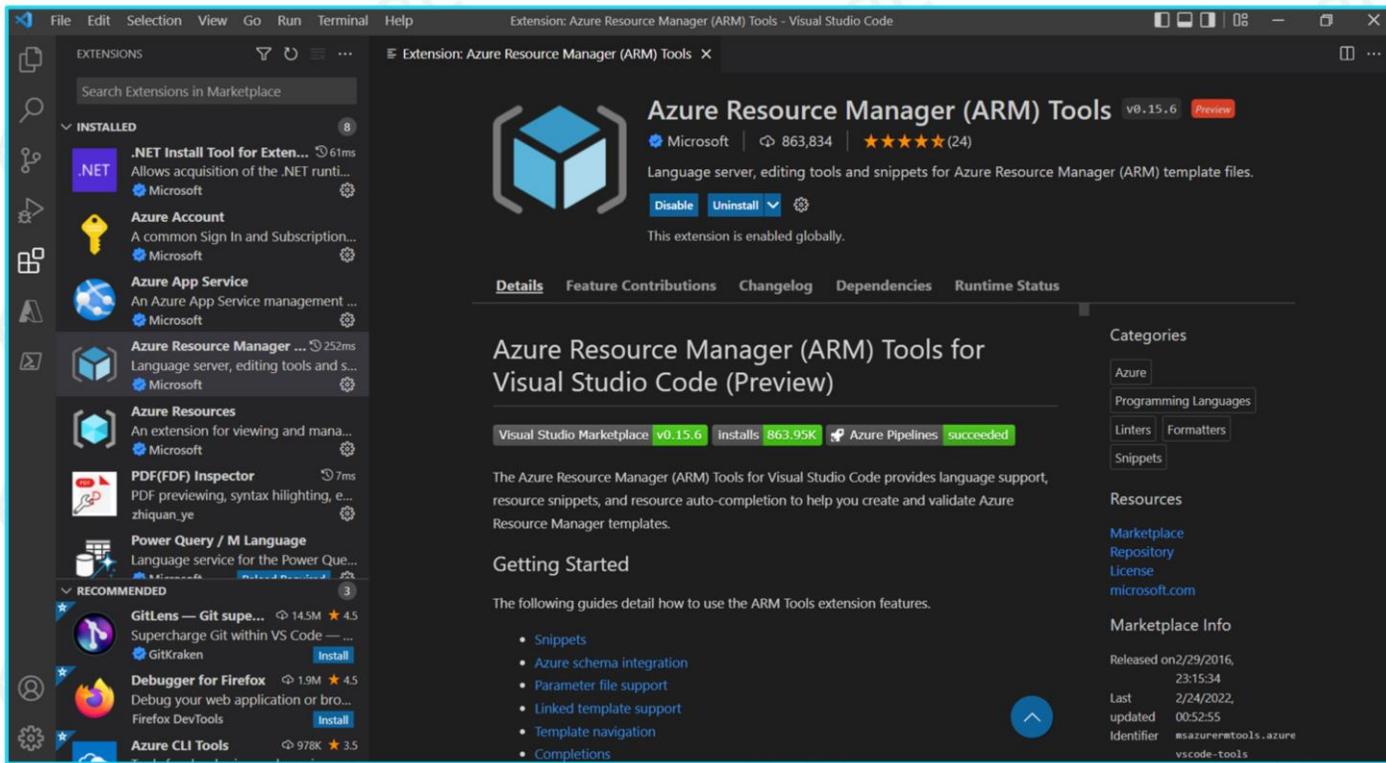
# ARM Template Design



# ARM Template Design



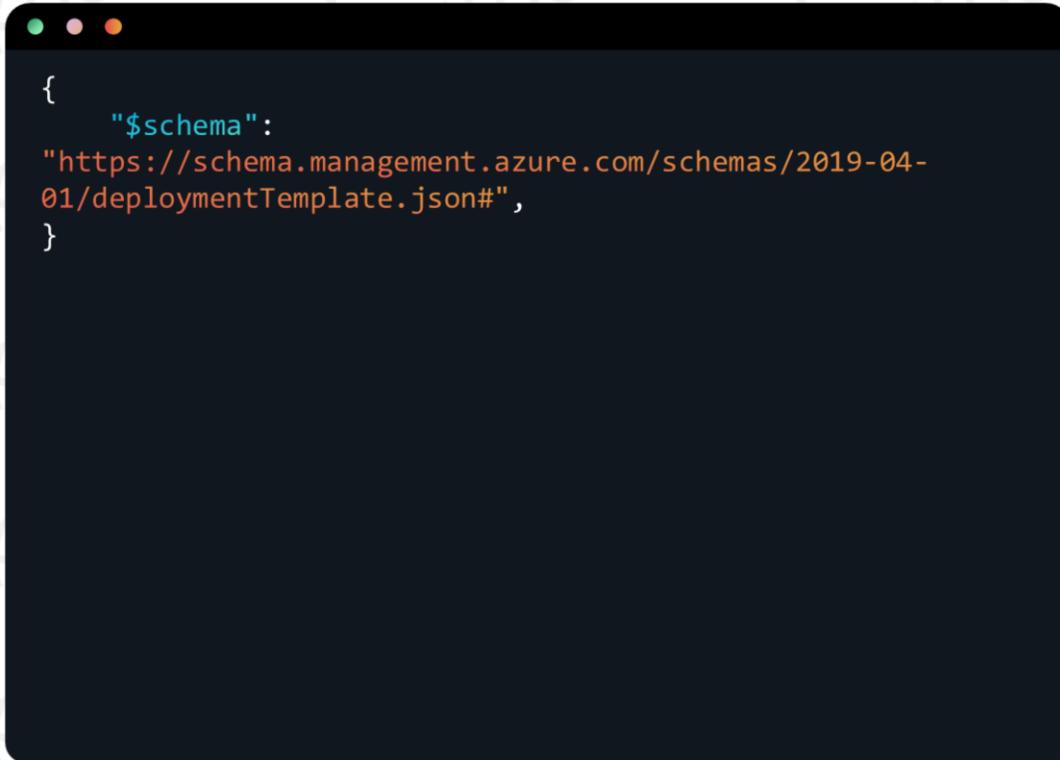
# ARM Extensions for VS Code (optional)





# ARM template structure

# ARM Template Structure



```
{  
  "$schema":  
  "https://schema.management.azure.com/schemas/2019-04-  
  01/deploymentTemplate.json#",  
}
```

\$schema\*

contentVersion\*

parameters

variables

functions

resources\*

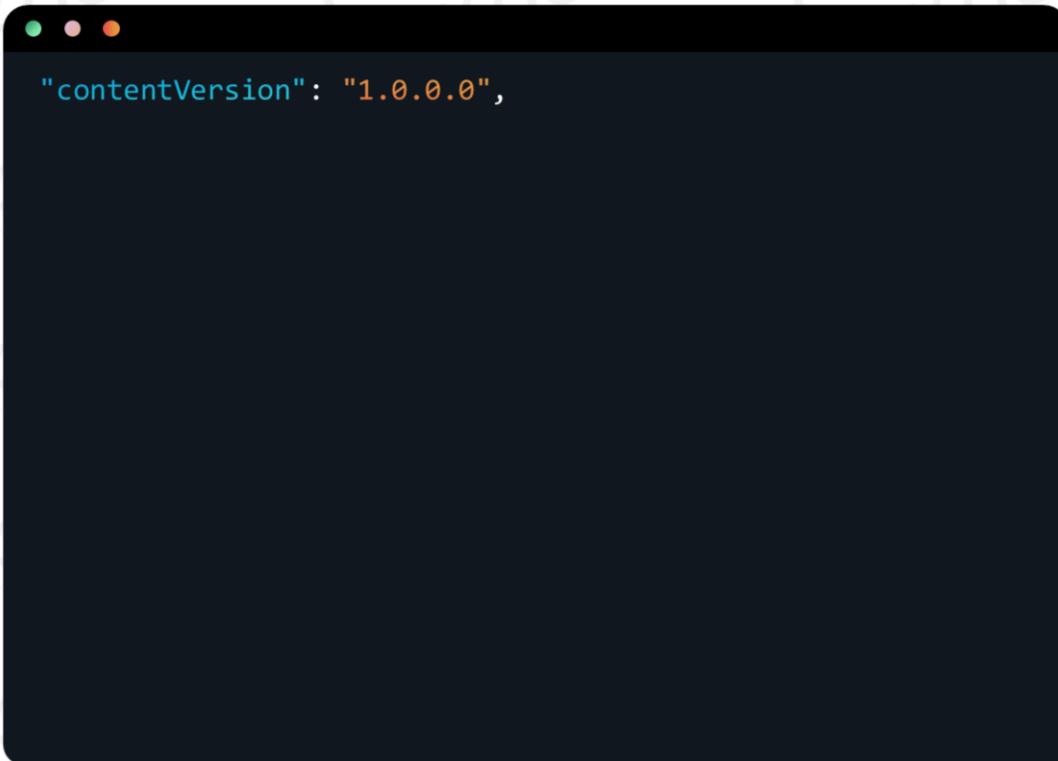
outputs

© Copyright KodeKloud

## Schema\*

References the location of the JSON file schema that describes the version of the template language. We can deploy ARM templates to different scopes like tenant, management groups, subscriptions; based on the scope that we are selecting the schema will change.

# ARM Template Structure



© Copyright KodeKloud

\$schema\*

contentVersion\*

parameters

variables

functions

resources\*

outputs

## Content Version\*

Used to version the template, the default value is 1.0.0.0. Any value can be given to this element. Content version is useful if you are storing your templates in a source control and would like to keep the changes tracked in different versions. Proper versioning will help users to pick the latest version of your template.

# ARM Template Structure



```
"parameters": {  
    "location": {  
        "type": "string",  
        "allowedValues" : [  
            "East US",  
            "West US"  
        ],  
        "defaultValue": "East US",  
        "metadata": {  
            "description": "Location of the resource"  
        }  
    },  
},
```

\$schema\*

contentVersion\*

parameters

variables

functions

resources\*

outputs

© Copyright KodeKloud

## Parameters

During resource deployment, the parameter value can be provided as an input to the template. Parameters help in making the templates reusable, where users can supply different values during execution without the need to modify the template.

# ARM Template Structure



```
"variables": {  
    "publicIPAddressName": "app-gw-pip"  
},
```

\$schema\*

contentVersion\*

parameters

variables

functions

resources\*

outputs

© Copyright KodeKloud

## Variables

Variables can be used to hardcode value to the templates. If you are referencing a value with the help of variables and if that value needs to be modified; instead of updating all occurrences, you just need to update the value of the variable.

# ARM Template Structure



```
"functions": [
    {
        "namespace": "userspace",
        "members": {
            "VMNameGenerator": {
                "parameters": [
                    {
                        "name": "userstring",
                        "type": "string"
                    }
                ],
                "output": {
                    "value": "function-return-value",
                    "type": "string"
                }
            }
        }
    }
],
```

© Copyright KodeKloud

\$schema\*

contentVersion\*

parameters

variables

functions

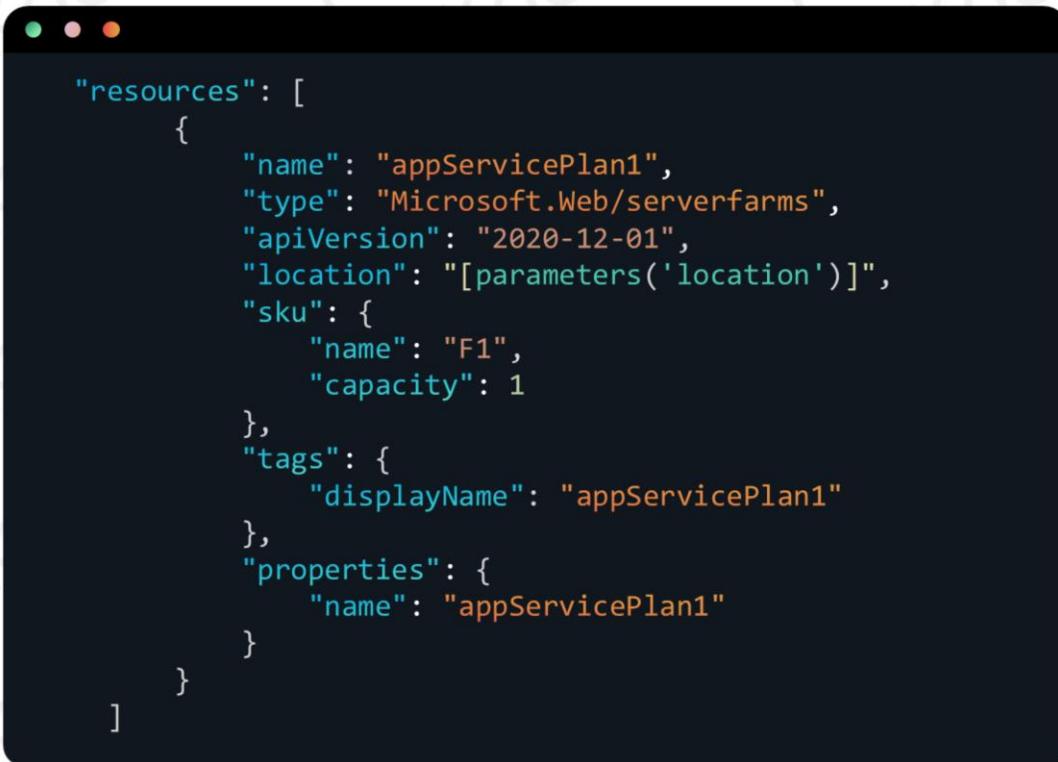
resources\*

outputs

## Functions

We can create user defined functions in ARM templates that can be used to replace repeated code blocks

# ARM Template Structure



```
"resources": [
    {
        "name": "appServicePlan1",
        "type": "Microsoft.Web/serverfarms",
        "apiVersion": "2020-12-01",
        "location": "[parameters('location')]",
        "sku": {
            "name": "F1",
            "capacity": 1
        },
        "tags": {
            "displayName": "appServicePlan1"
        },
        "properties": {
            "name": "appServicePlan1"
        }
    }
]
```

\$schema\*

contentVersion\*

parameters

variables

functions

resources\*

outputs

© Copyright KodeKloud

## Resources\*

Resources we intend to create, or update will be declared inside this element. Here, we can reference the parameters, variables, and functions we created earlier.

# ARM Template Structure

```
"outputs": {  
    "hostname": {  
        "type": "string",  
        "value": "[reference(resourceId('Microsoft.Network/publicIPAddresses',variables('publicIPAddressName'))).dnsSettings.fqdn]"  
    },  
}
```

\$schema\*

contentVersion\*

parameters

variables

functions

resources\*

outputs

© Copyright KodeKloud

## Outputs

Display values that are returned after deployment.



# Azure Bicep

# Consider Azure Bicep Files

 Simpler syntax compared to ARM templates

 Smaller module files

 Auto-detect dependencies

 VS Code extension available to make authoring smooth and seamless

Bicep File

```
resource storageAccount  
'Microsoft.Storage/storageAccounts  
@2021-01-01' = {  
    name: storageAccountName  
    location: location  
    tags: {  
        displayName:  
        storageAccountName  
    }  
    kind: 'StorageV2'  
    sku: {  
        name: 'Standard_LRS'  
    } }
```



# Azure Quickstart Templates

## Azure Quickstart Templates

Deploy Azure resources through the Azure Resource Manager with community-contributed templates to get more done. Deploy, learn, fork and contribute back.



**What is Azure Resource Manager**

Azure Resource Manager allows you to provision your applications using a declarative template. In a single template, you can deploy multiple services along with their dependencies. You use the same template to repeatedly deploy your application during every stage of the application life cycle.

[Learn more >](#)

Search  x 🔍

1,057 Quickstart templates are currently in the gallery. [See all](#)

**Most popular**

[Migrate to Azure SQL database using Azure DMS](#)  
The Azure Database Migration Service (DMS) is designed to streamline the process of migrating on-premises databases to Azure. DMS will simplify the migration of e...  
 by Ashish Shinde, Last updated: 26/04/2021

[Secure VM password with Key Vault](#)  
This template allows you to deploy a simple Windows VM by retrieving the password that is stored in a Key Vault. Therefore the password is never put in plain text in the t...  
 by Brian Moore, Last updated: 11/05/2021

[Install Configuration Manager Current Branch in Azure](#)  
This template creates new Azure VMs based on which configuration you choose. It configures a new AD domain controller, a new hierarchy/standalone bench with SQL ...  
 by Yuanheng Yang, Last updated: 01/09/2021

[Configure WAF rate liming rule for Azure Front Door endpoint](#)  
This template configures a WAF rule for Azure Front Door to rate limit incoming traffic for a given frontend host.  
 by victorar, Last updated: 29/04/2021



# KodeKloud

© Copyright KodeKloud

Follow us on <https://kodekloud.com/> to learn more about us.