



Azure Master Class: **Introduction To Azure ARM Templates**



Expected Learning Outcomes

Azure Master Class: Introduction To Azure ARM Templates

By the end of this section, you should be able to:

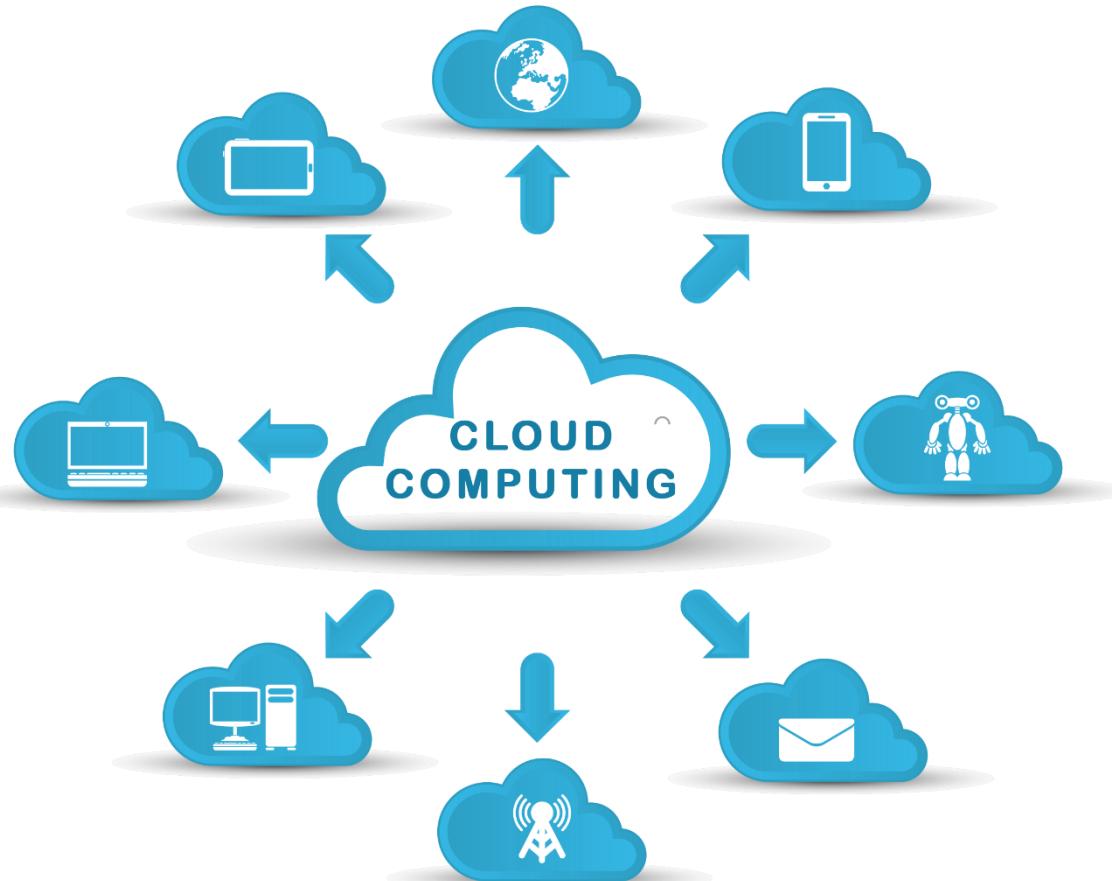


- Understand basic cloud computing concepts
- Describe the Azure Resource Manager
- Define resource groups and resource providers
- Outline the basic structure of an ARM template
- Create an ARM template
- Install Visual Studio Code



@tetranoode





"Cloud computing is a kind of Internet-based computing that provides shared processing resources and data to computers and other devices on demand."

- Wikipedia



Benefits Of Cloud Computing





Capex To Opex





Agility



@tetranoode





Flexibility



Green
Energy



@tetranoode





€
Savings



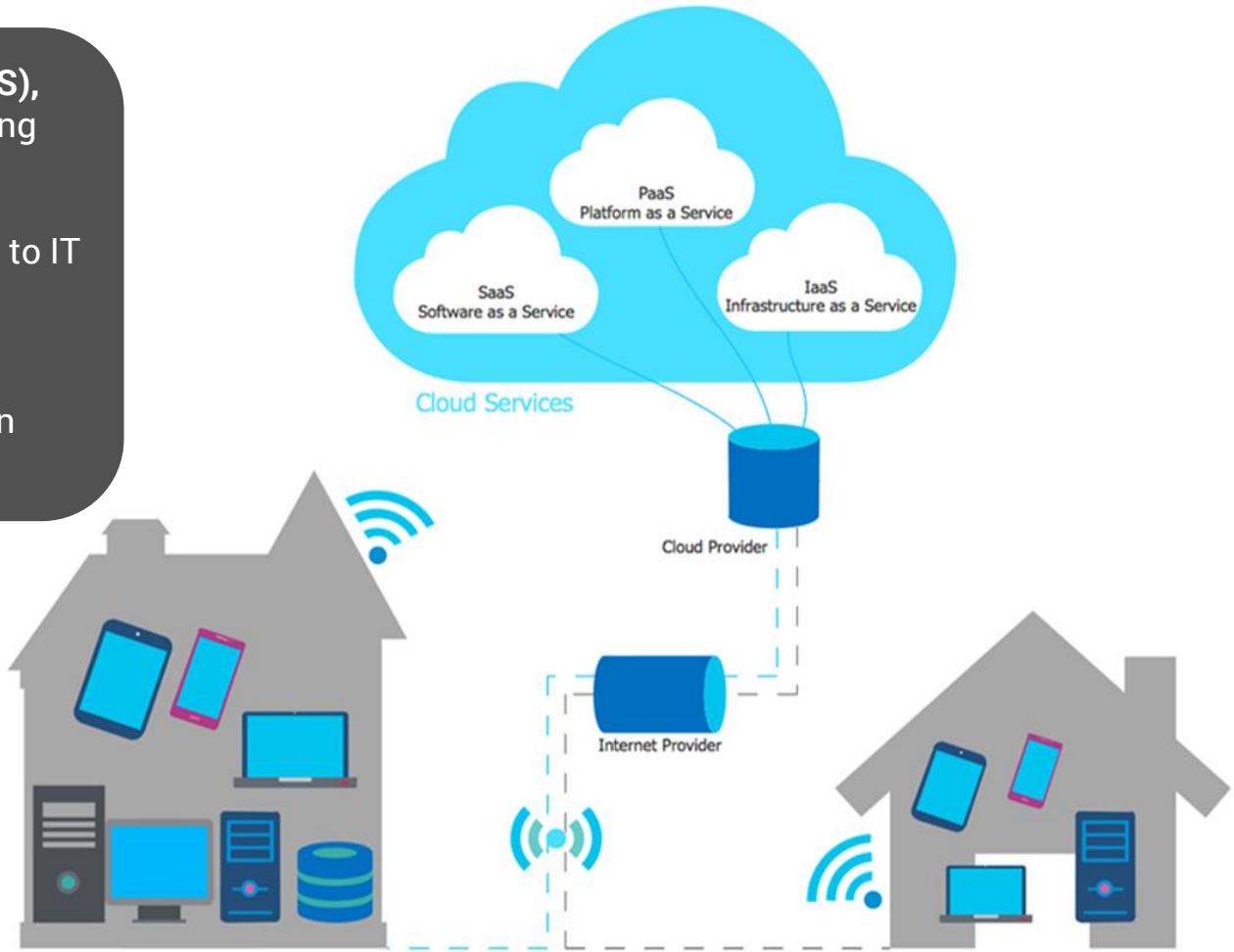
Growth



@tetranoode



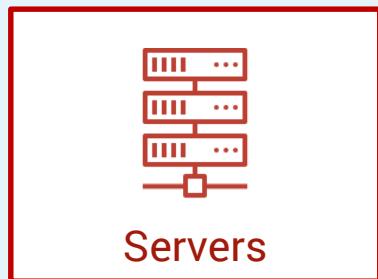
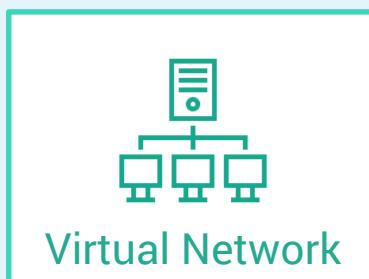
- **Infrastructure as a Service (IaaS),**
Providing Infrastructure hosting
- **Platform as a Service (PaaS),**
Providing framework services to IT
directors
- **Software as a Service (SaaS),**
Providing software application
service to end-users

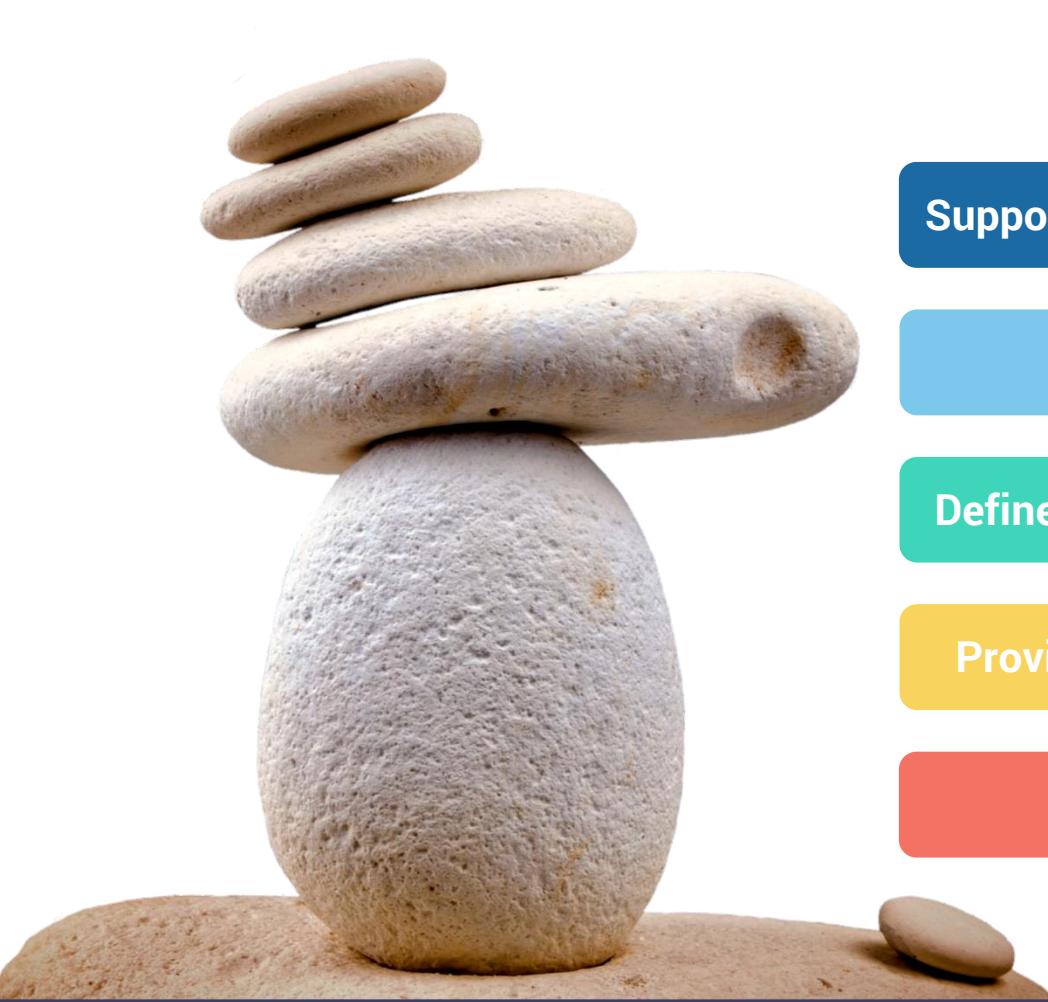


What is ARM?

Azure Masterclass: Introduction to ARM Templates

Azure
Resource
Manager





Support for pre-defined, structured templates

Based on JSON syntax

Define the resources to be deployed in Azure

Provides security and auditing capabilities

Supports multiple client SDKs



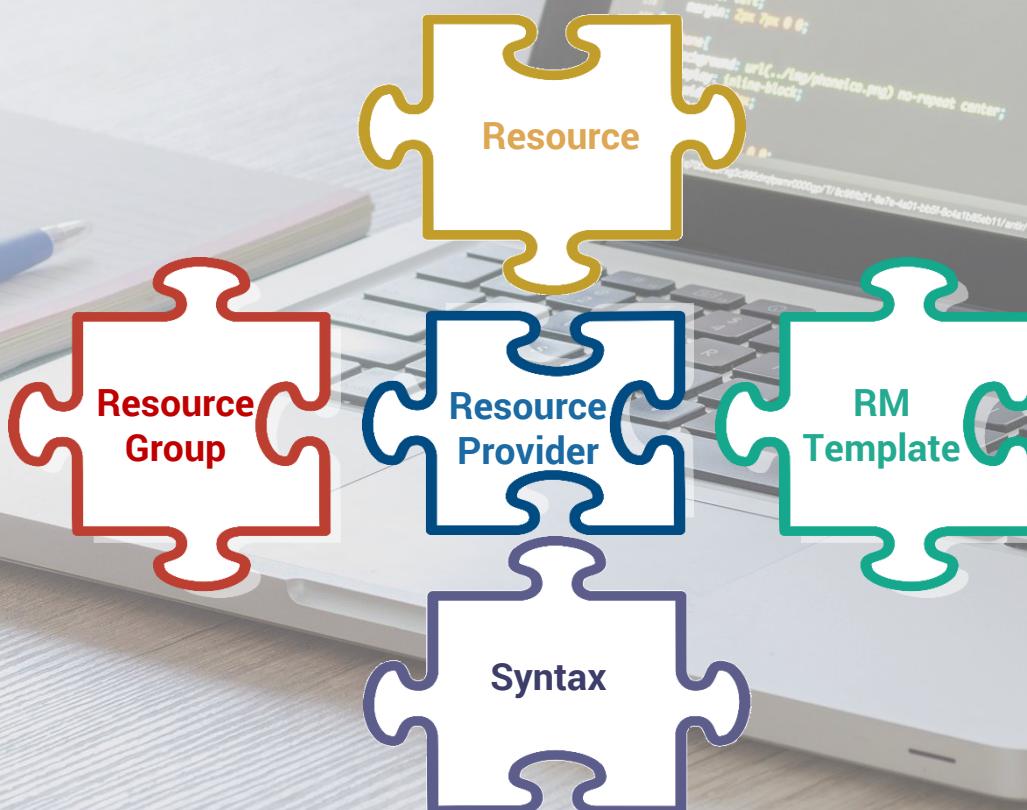
Deployment

Tip



Basic Terminology

Azure Masterclass: Introduction to ARM Templates



Benefits of using Azure Resource Manager

Azure Masterclass: Introduction to ARM Templates

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



Template-driven



Multi-service



Idempotent



Declarative



Multi-region



Access control



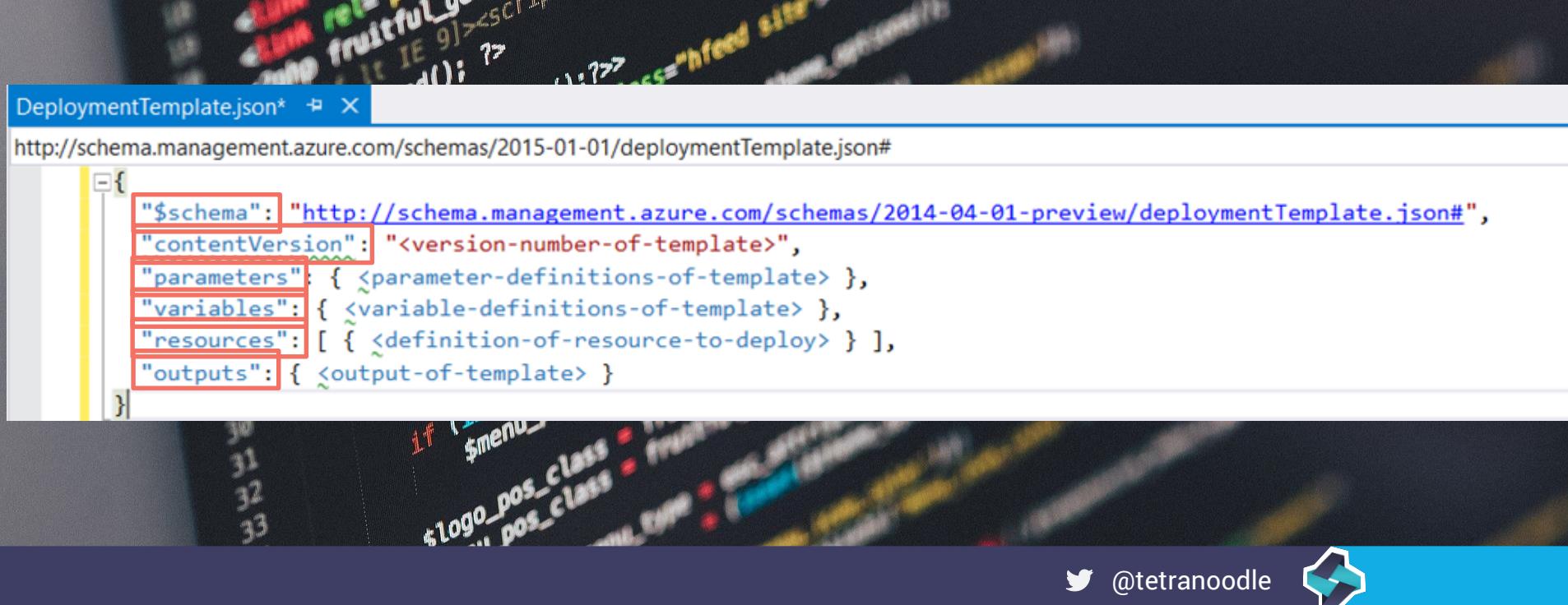
@tetranoode



Basic Structure of an ARM Template

Azure Masterclass: Introduction to ARM Templates

Here is the basic structure of an ARM template:



The screenshot shows a code editor window titled "DeploymentTemplate.json*" containing the schema definition for an ARM template. The URL "http://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#" is shown above the code. The JSON structure is as follows:

```
{ "$schema": "http://schema.management.azure.com/schemas/2014-04-01-preview/deploymentTemplate.json#", "contentVersion": "<version-number-of-template>", "parameters": { <parameter-definitions-of-template> }, "variables": { <variable-definitions-of-template> }, "resources": [ { <definition-of-resource-to-deploy> } ], "outputs": { <output-of-template> } }
```

The top-level properties "\$schema", "contentVersion", "parameters", "variables", "resources", and "outputs" are highlighted with red boxes. The code editor has a dark theme with syntax highlighting for JSON. In the background, there is a blurred view of another terminal or code editor showing some C# code.



@tetranoode



Basic Structure of an ARM Template

Azure Masterclass: Introduction to ARM Templates

Schema

Content Version

Parameters

Variables

Resources

Output

Not mandatory

Values returned after deployment



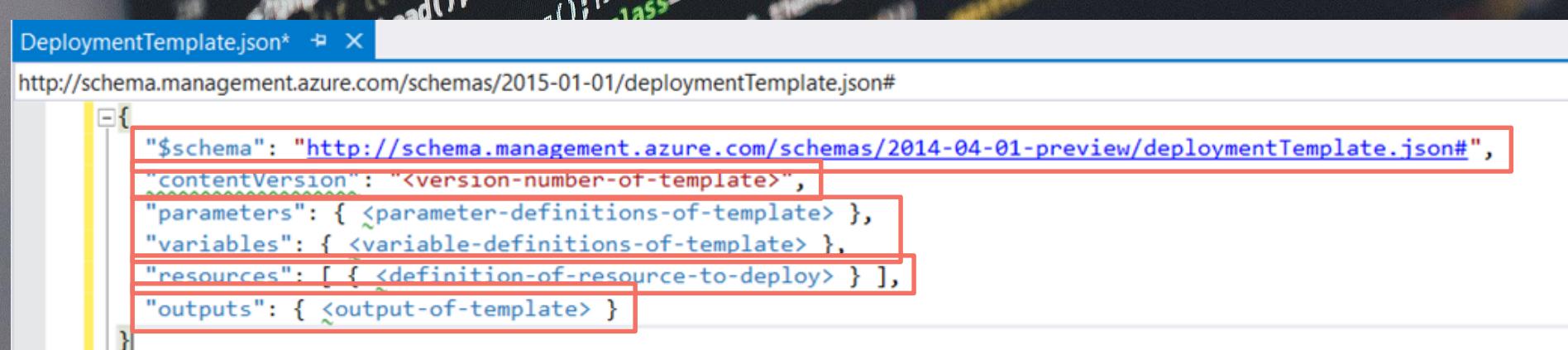
@tetranoode



Sample ARM Template

Azure Masterclass: Introduction to ARM Templates

If you take an empty resource template, it looks like this:



The screenshot shows a code editor window titled "DeploymentTemplate.json*" with a red "X" button. Below the title bar is a URL: "http://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#". The main content area contains the following JSON schema:

```
{ "$schema": "http://schema.management.azure.com/schemas/2014-04-01-preview/deploymentTemplate.json#", "contentVersion": "<version-number-of-template>", "parameters": { <parameter-definitions-of-template> }, "variables": { <variable-definitions-of-template> }, "resources": [ { <definition-of-resource-to-deploy> } ], "outputs": { <output-of-template> } }
```

The entire JSON structure is highlighted with a red border.



@tetranoode



ARM Templates In Azure

▶ DEMO



Tools for ARM Templates

Azure Masterclass: Introduction to ARM Templates

You can use Azure to create ARM resources

Azure is useful, but may not be scalable

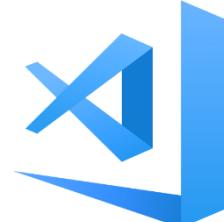
Use one of many available IDEs or tools to edit JSON templates



Notepad



Visual Studio



Visual Studio Code



Sublime Text

Tools for ARM Templates

Azure Masterclass: Introduction to ARM Templates



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

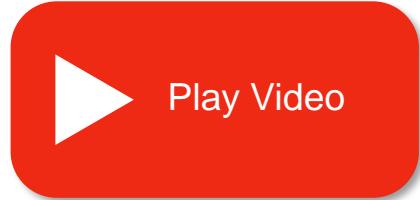


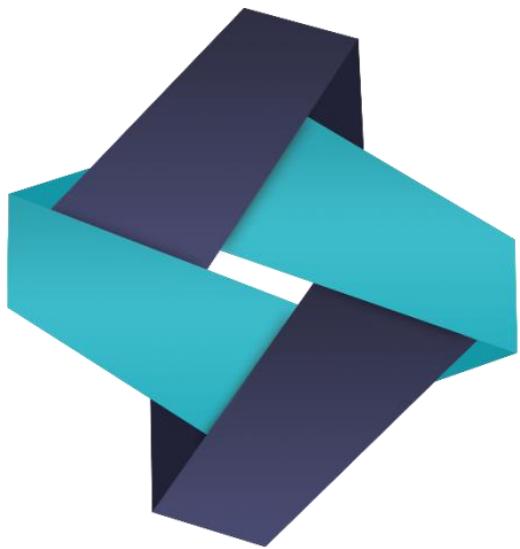
@tetranoode



Install Visual Studio Code

▶ DEMO





@tetrano

