Azure Master Class:

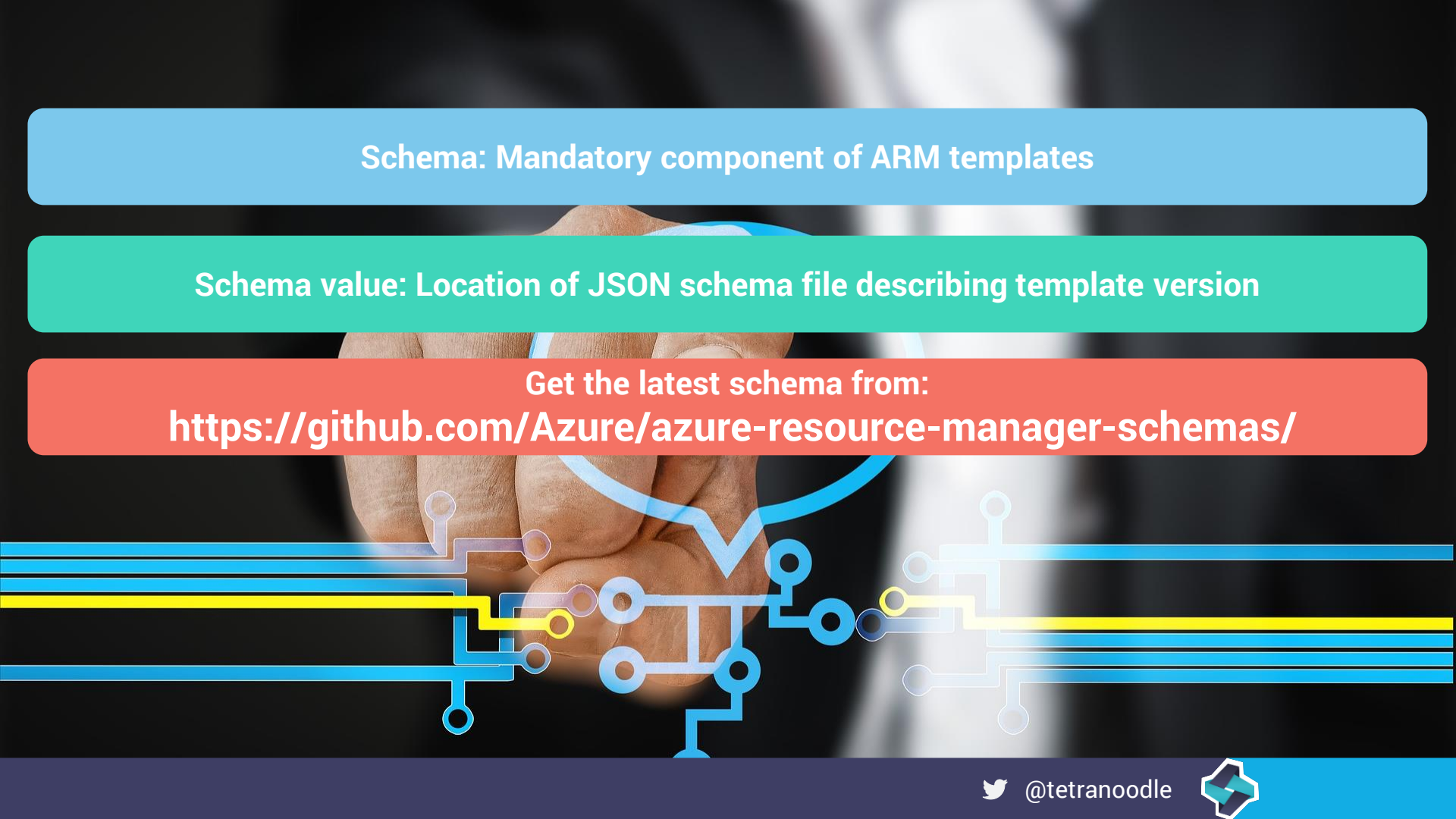
Deep-dive Into Azure ARM Templates

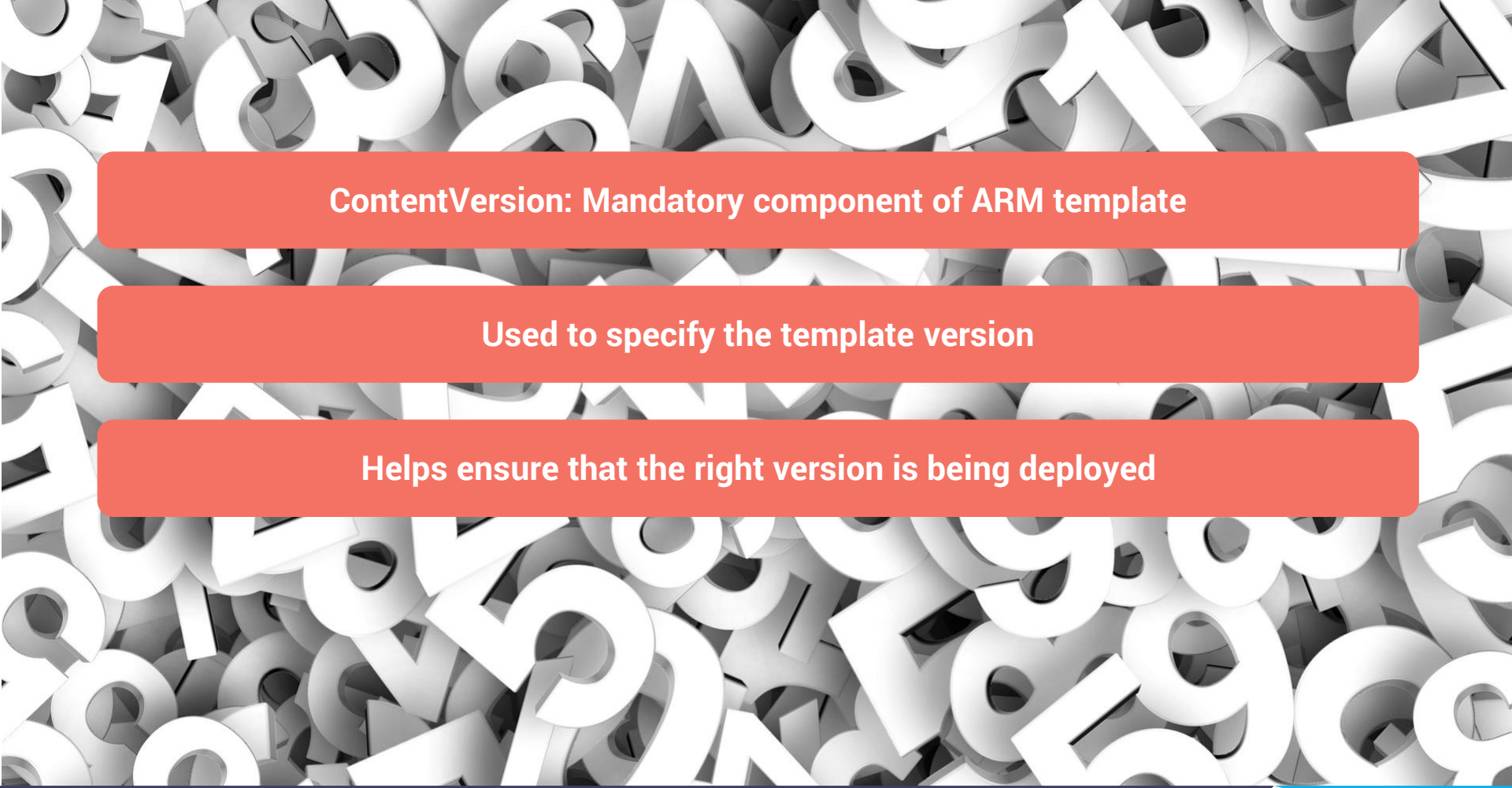
Expected Learning Outcomes

Azure Masterclass: Deep-dive into ARM Templates

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

- **Walkthrough the various sections of the ARM template**
- **Differentiate between single and nested templates**
- **Describe incremental and complete deployment modes**


Schema: Mandatory component of ARM templates
Schema value: Location of JSON schema file describing template version
Get the latest schema from:
https://github.com/Azure/azure-resource-manager-schemas/


ContentVersion: Mandatory component of ARM template
Used to specify the template version
Helps ensure that the right version is being deployed

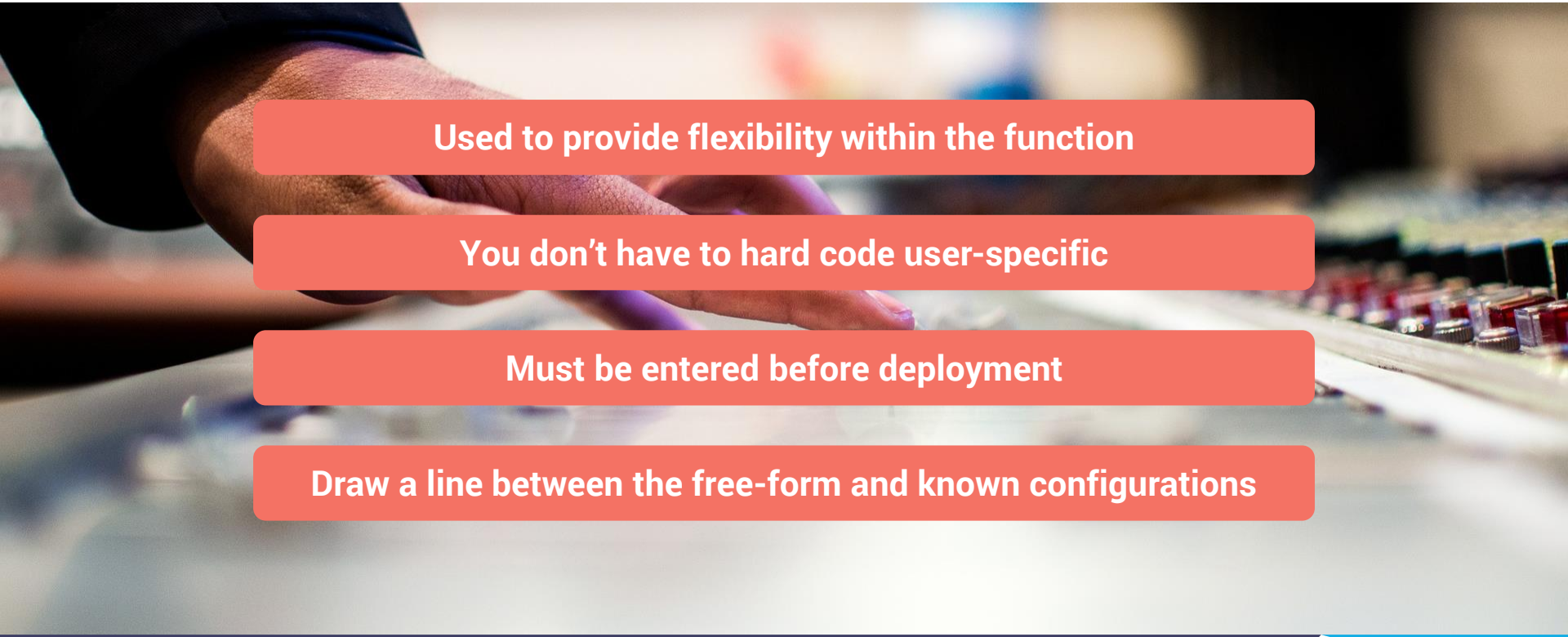
Schema And ContentVersion

DEMO


Play Video

Parameters

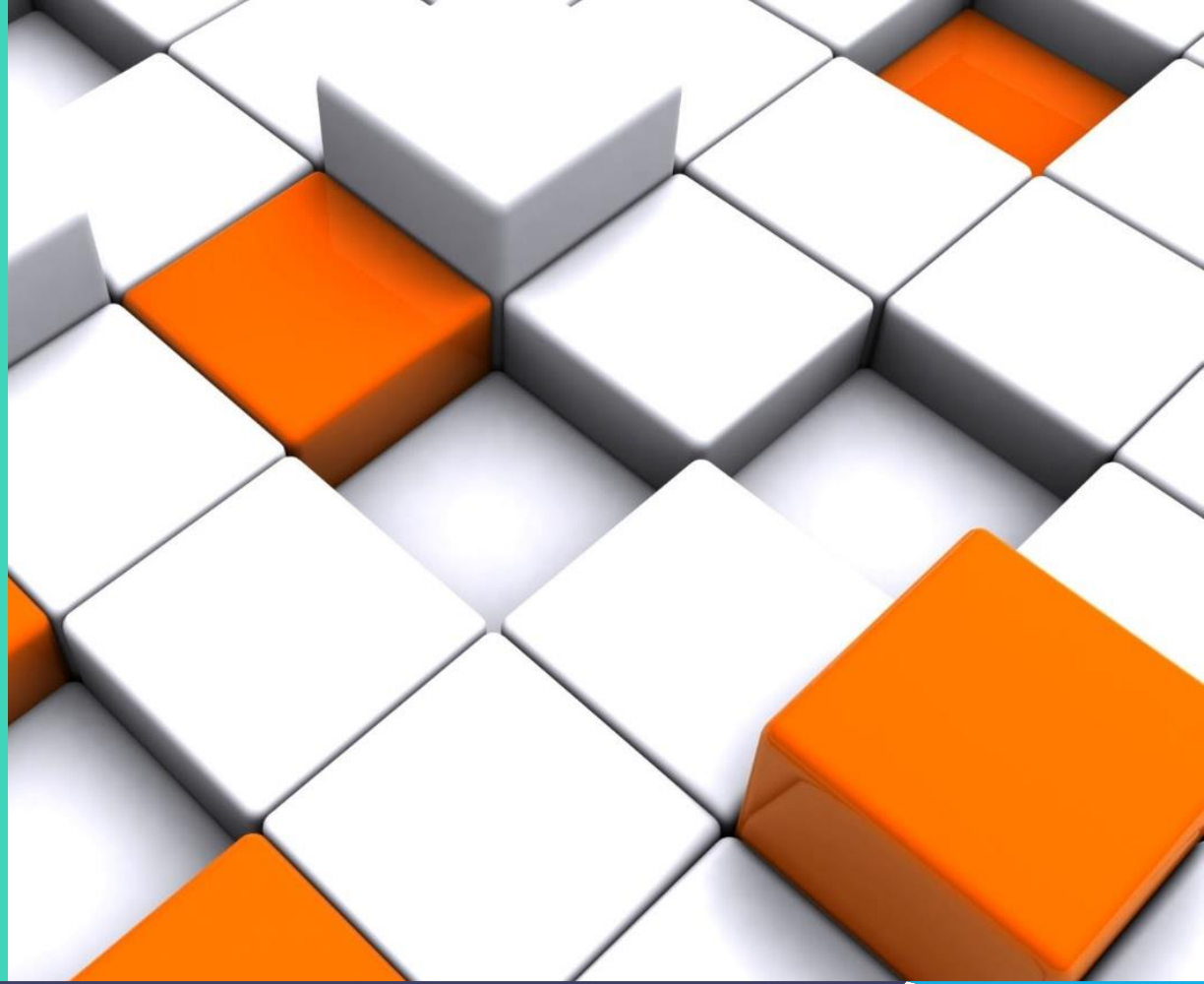
Azure Masterclass: Deep-dive into ARM Templates



- Used to provide flexibility within the function
- You don’t have to hard code user-specific
- Must be entered before deployment
- Draw a line between the free-form and known configurations

Parameter properties used to set values:

- parameterName
- type
- defaultValue
- allowedValues
- minValue
- maxValue
- minLength
- maxLength
- description



Parameters

DEMO


Play Video


Variables: Store and reuse values for resource properties
Don’t need to specify at deployment
Create variables for use across template
Types: standard & complex

Resources

Azure Masterclass: Deep-dive into ARM Templates

Resources:

- Collection of resources
- Array of resources, child resources
- Define properties
- Describe resource-specific properties

Elements:

- apiVersion
- type
- name
- location
- tags
- dependsOn
- properties
- resources
- condition
- comments

Resources

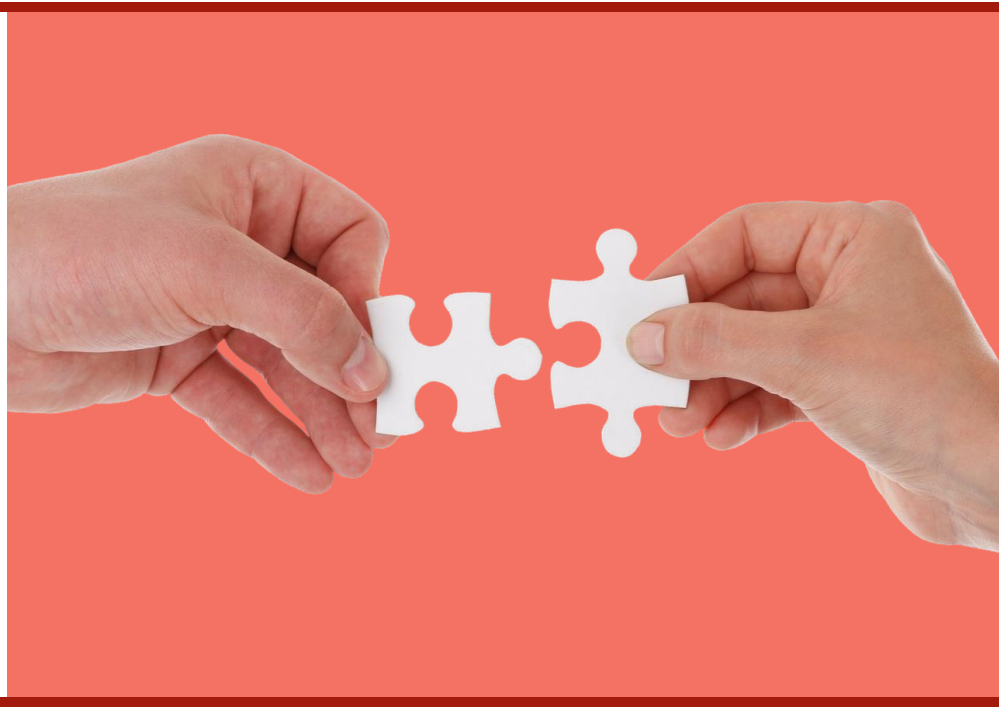
DEMO


Play Video

Output

Azure Masterclass: Deep-dive into ARM Templates

- Used to specify values returned from deployment
- Values are passed into another deployment as connection string for deployment
- Output elements: OutputName, type, value



Single Templates vs. Nested Templates

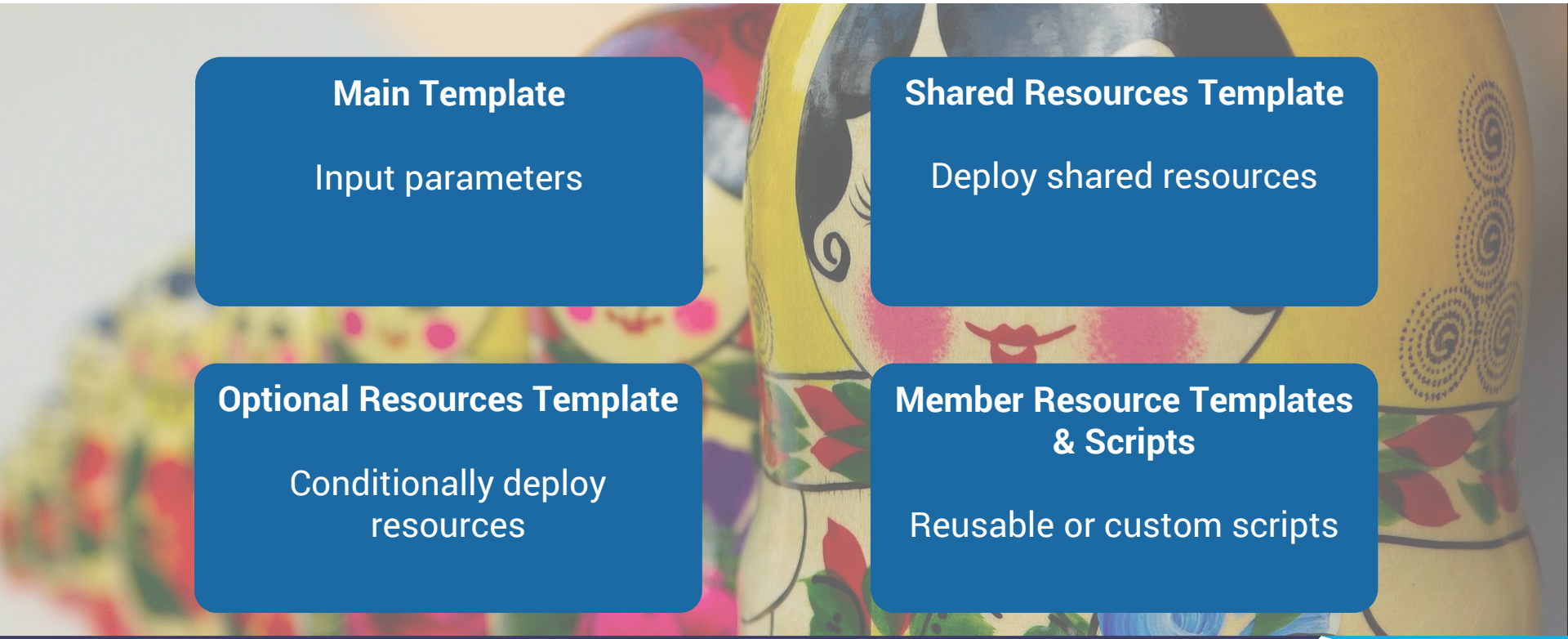
Azure Masterclass: Deep-dive into ARM Templates

- Deploy either using a single template or a main template with multiple nested templates
- Nested templates are common for more advanced scenarios

- You can break down a solution into targeted components
- You can reuse nested templates with different main templates

Nested Templates – Best Practices

Azure Masterclass: Deep-dive into ARM Templates



Main Template

Input parameters

Shared Resources Template

Deploy shared resources

Optional Resources Template

Conditionally deploy resources

Member Resource Templates & Scripts

Reusable or custom scripts

Hello World Templates

DEMO


Play Video

Executing Templates With Azure PowerShell

DEMO


Play Video

Deployment Modes

Azure Masterclass: Deep-dive into ARM Templates

Incremental

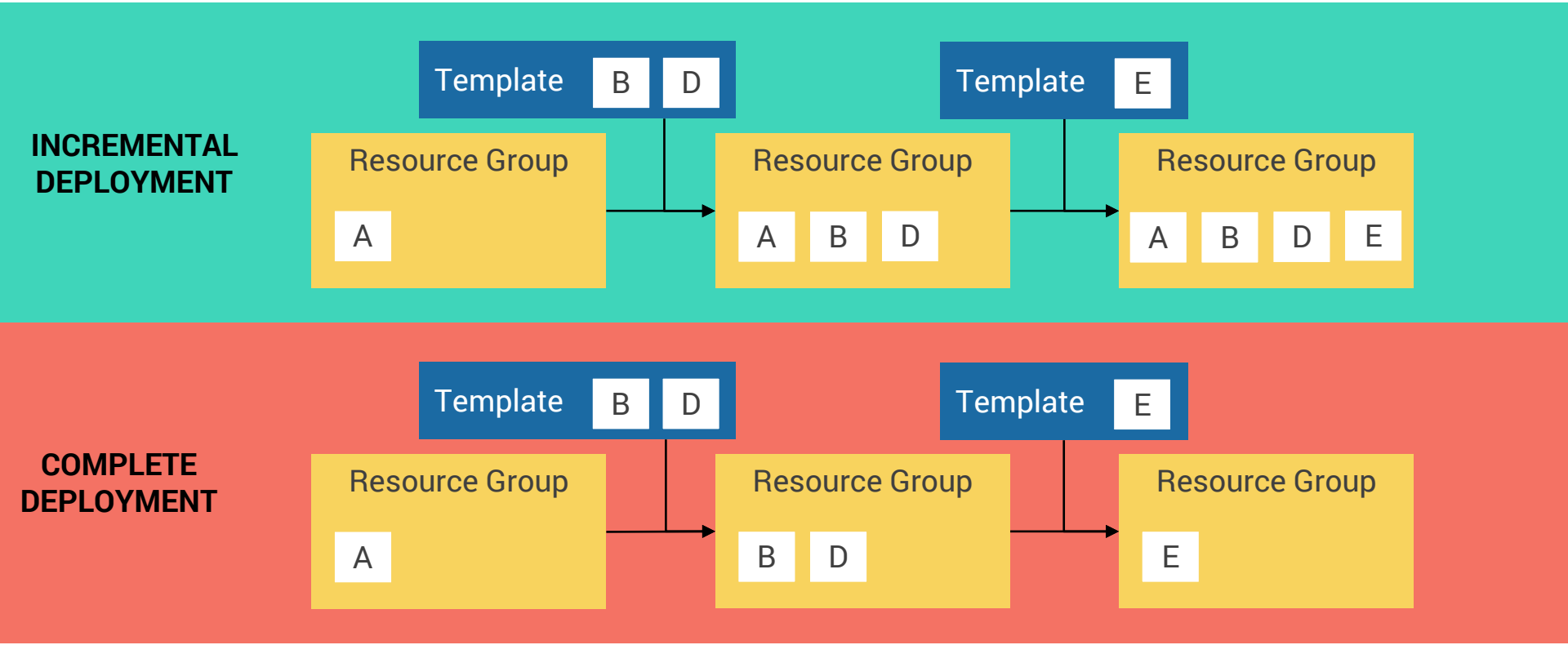
- Default mode
- Doesn't remove or modify resources
- Provides support for versioning and rolling back deployments

Complete

- Deletes resources not mentioned in the template
- Best for "Starting Fresh" or refactoring a solution

Deployment Modes

Azure Masterclass: Deep-dive into ARM Templates



Deployment Modes

DEMO


Play Video