Module 2
Deploying Resources with Azure Resource Manager

https://chmurowisko.pl

Chmurowisko

Agenda

Chmurowisko

- ARM Templates
- Role-Based Access Control (RBAC)
- Resource Policies
- Security
- Building Blocks

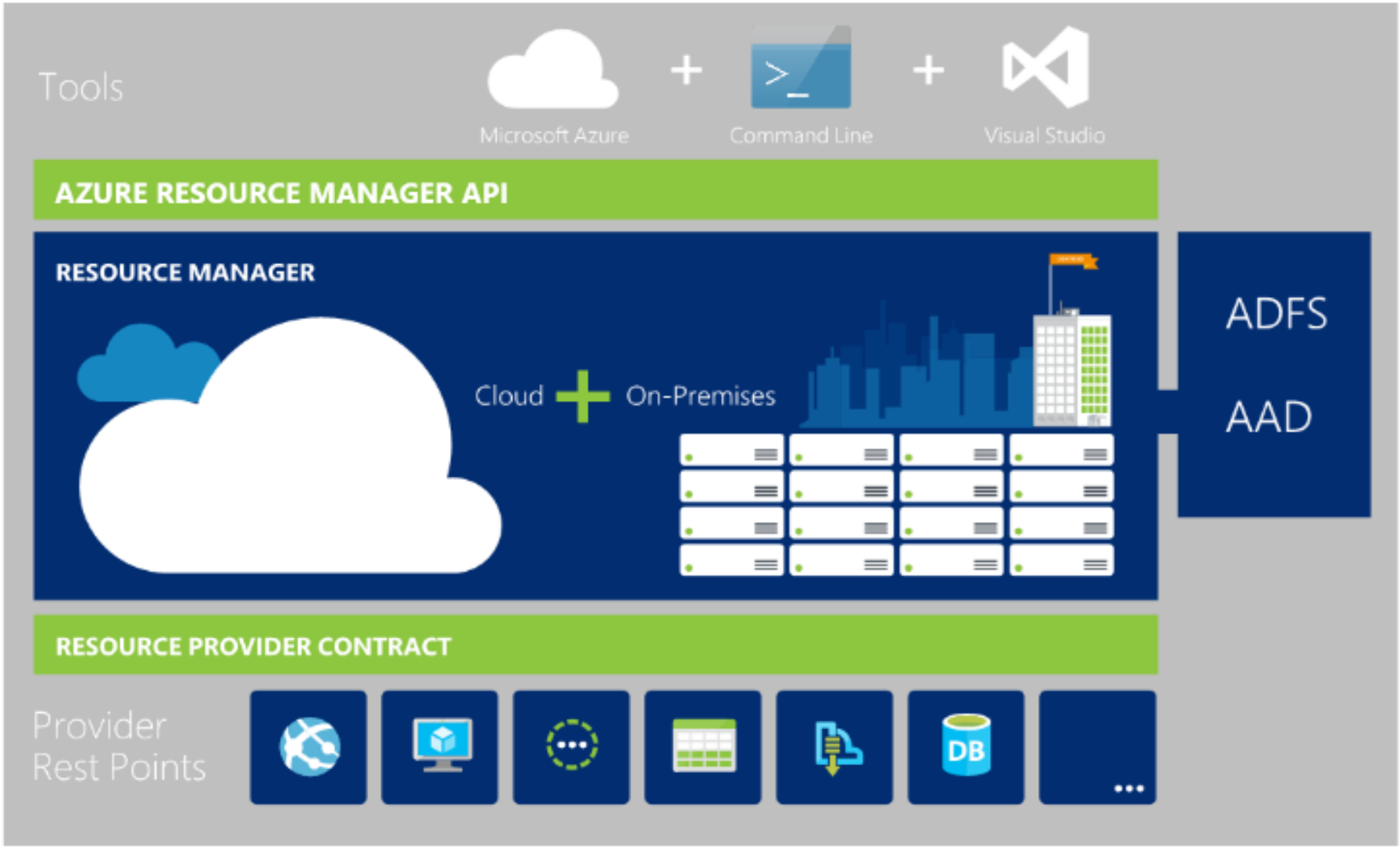
Lesson 1: ARM Templates

- Azure Resource Manager
- JSON Templates
- Debugging

Azure Resource Manager

Chmurowisko

Consistent Management Layer



Azure Resource Manager

Chmurowisko

- Provide a scalable, repeatable method for deploying Azure resources
- Helps in deploying whole solutions, not just one resource
- All resources in the ARM model are built using JSON templates
- Samples - https://github.com/Azure/azure-quickstart-templates

Deploying Resources

Chmurowisko

- PowerShell
- Cross Platform Command-Line Interface (all client platforms)
- Client Libraries for various languages
- Visual Studio, Visual Studio Code
- Portal template deployment

- All use the REST API: The REST API is available here: https://docs.microsoft.com/rest/api/resources

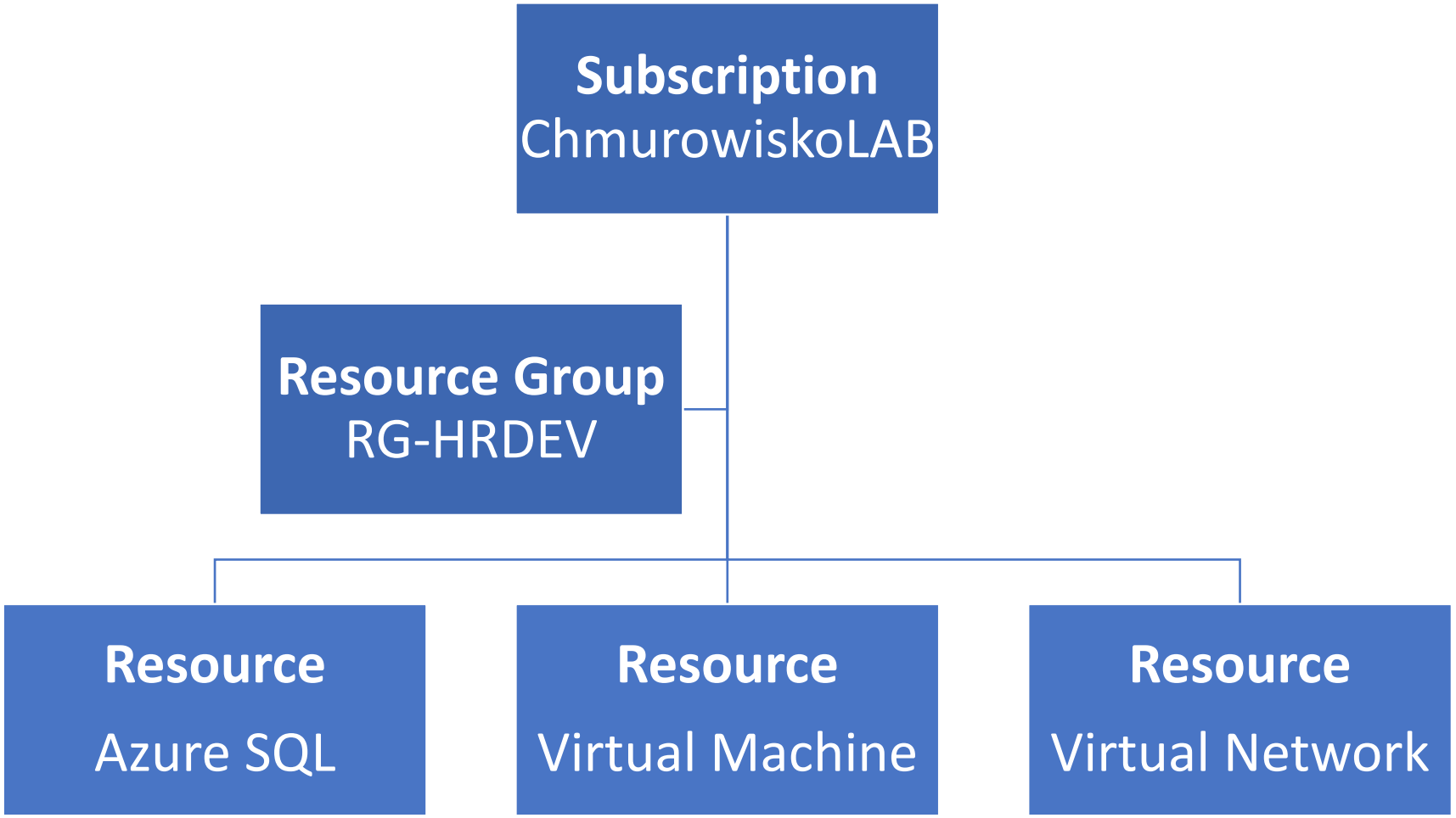
```
usage: az group deployment create [-h] [--verbose] [--debug]
                                  [--output {json,jsonc,table,tsv}]
                                  [--query JMESPATH] --resource-group
                                  RESOURCE_GROUP_NAME
                                  [--template-file TEMPLATE_FILE]
                                  [--template-uri TEMPLATE_URI]
                                  [--name DEPLOYMENT_NAME]
                                  [--parameters PARAMETERS [PARAMETERS ...]]
                                  [--mode {Incremental,Complete}] [--no-wait]
```

PowerShell | Copy

```
New-AzureRmResourceGroupDeployment
    [-Name <String>]
    -ResourceGroupName <String>
    [-Mode <DeploymentMode>]
    [-DeploymentDebugLogLevel <String>]
    [-RollbackToLastDeployment]
    [-RollBackDeploymentName <String>]
    [-Force]
    [-AsJob]
    -TemplateFile <String>
    [-ApiVersion <String>]
    [-Pre]
    [-DefaultProfile <IAzureContextContainer>]
    [-WhatIf]
    [-Confirm]
    [<CommonParameters>]
```

Resource Group Deployment

Chmurowisko



What is JSON?

Chmurowisko

- JavaScript Object Notification (JSON) - method for passing data and objects in a formatted style
- It is based on a subset of the JavaScript Programming Language, Standard ECMA-262 3rd Edition - December 1999
- JSON is built on two structures:
  - A collection of name/value pairs.
  - An ordered list of values.

```
{"widget": {
    "debug": "on",
    "window": {
        "title": "Sample Konfabulator Widget",
        "name": "main_window",
        "width": 500,
        "height": 500
    },
    "image": {
        "src": "Images/Sun.png",
        "name": "sun1",
        "hOffset": 250,
        "vOffset": 250,
        "alignment": "center"
    },
    "text": {
        "data": "Click Here",
        "size": 36,
        "style": "bold",
        "name": "text1",
        "hOffset": 250,
        "vOffset": 100,
        "alignment": "center",
        "onMouseUp": "sun1.opacity = (sun1.opacity / 100) * 90;"
    }
}}
```

Chmurowisko

JSON

Copy

```
{
    "$schema": "http://schema.management.azure.com/schemas/2015-01
    "contentVersion": "",
    "parameters": {  },
    "variables": {  },
    "functions": [  ],
    "resources": [  ],
    "outputs": {  }
}
```

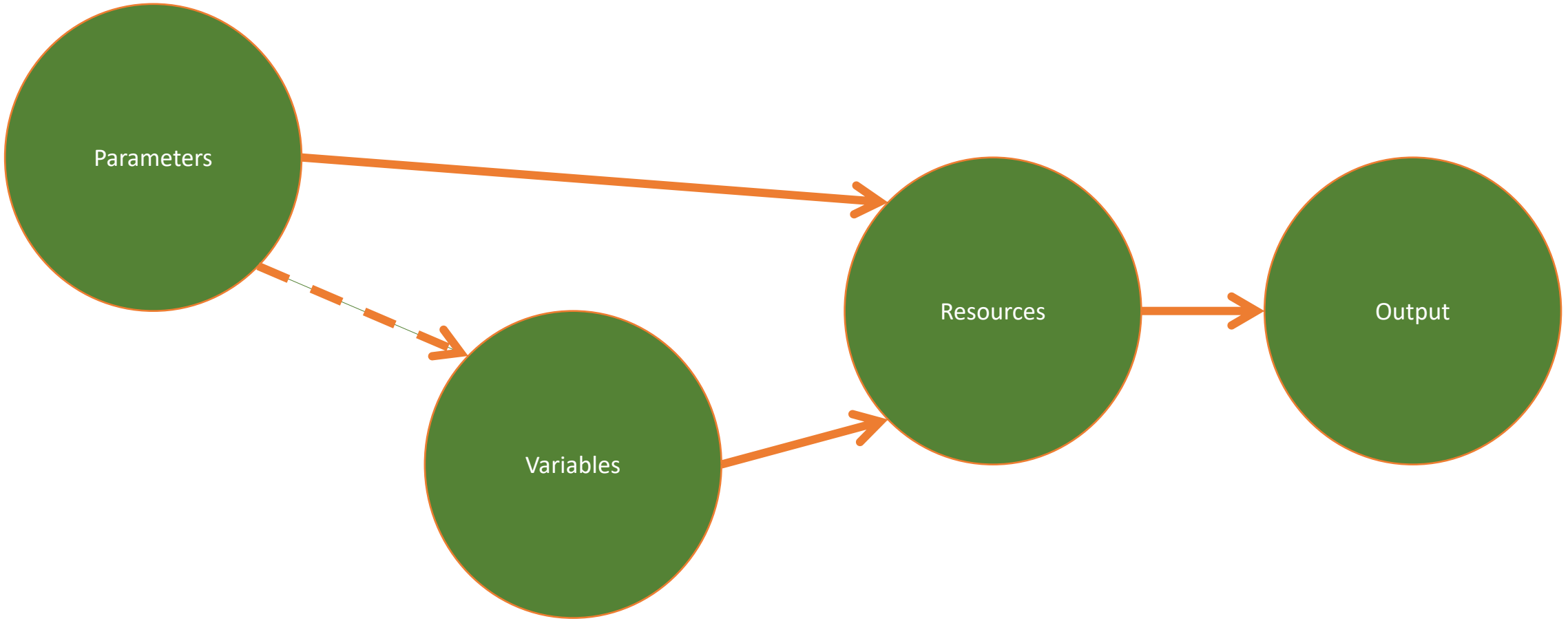
Sample template

Chmurowisko

edvm-template.json

```
1  {
2      "$schema": "http://schema.management.azure.com/schemas/2015-01-01/deploymer
3      "contentVersion": "1.0.0.0",
4      "parameters": {
5          "location": {
6              "type": "string"
7          },
8          "virtualMachineName": {
9              "type": "string"
10         }
11     },
12     "variables": {
13         "vnetId": "[resourceId('day2demorg','Microsoft.Network/virtualNetworks'
14         "subnetRef": "[concat(variables('vnetId'), '/subnets/', parameters('sub
15         "diagnosticsExtensionName": "IaaSDiagnostics"
16     },
17     "resources": [
18         {
19             "name": "[parameters('virtualMachineName')]",
20             "type": "Microsoft.Compute/virtualMachines",
21             "apiVersion": "2016-04-30-preview",
22             "location": "[parameters('location')]",
23             "dependsOn": [
24                 "[concat('Microsoft.Network/networkInterfaces/', parameters('ne
25                 "[concat('Microsoft.Compute/availabilitySets/', parameters('ava
26                 "[concat('Microsoft.Storage/storageAccounts/', parameters('diag
```

Chmurowisko



Chmurowisko

rg-securityworkshop - Automation script

Resource group - PREVIEW

Download | Add to library | Deploy

7 resource types cannot be exported yet and are not included in the template. See error details. →

Automate deploying resources with Azure Resource Manager templates in a single, coordinated operation. Define resources and configurable input parameters and deploy with script or code. Learn more about template deployment.

Template | Parameters | CLI | PowerShell | .NET | Ruby

- Parameters (272)
- Variables (0)
- Resources (240)
  - [parameters('automationAccounts...
  - [parameters('disks_mifurmvm01_O...
  - [parameters('virtualMachines_kali0...
  - [parameters('virtualMachines_seb_...
  - [parameters('virtualMachines_sec_...
  - [parameters('virtualMachines_sec_...
  - [parameters('virtualMachines_sec_...
  - [parameters('virtualMachines_sec_...
  - [parameters('virtualMachines_sec_...
  - [parameters('virtualMachines_sec_...
  - [parameters('virtualMachines_sec_...
  - [parameters('virtualMachines_sec_...
  - [parameters('virtualMachines_sec_...
  - [parameters('virtualMachines_sec...
  - [parameters('managedClusters_az...
  - [parameters('applicationGateways...
  - [parameters('loadBalancers_sec_n...
  - [parameters('networkInterfaces_ka...

```
{
    "$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#",
    "contentVersion": "1.0.0.0",
    "parameters": {
        "networkSecurityGroups_AADDS_chmurowiskolab.onmicrosoft.com_NSG_sourceAddressPrefix": {
            "defaultValue": null,
            "type": "SecureString"
        },
        "networkSecurityGroups_AADDS_chmurowiskolab.onmicrosoft.com_NSG_sourceAddressPrefix_1": {
            "defaultValue": null,
            "type": "SecureString"
        },
        "networkSecurityGroups_AADDS_chmurowiskolab.onmicrosoft.com_NSG_sourceAddressPrefix_2": {
            "defaultValue": null,
            "type": "SecureString"
        },
        "publicIPAddresses_waf_fe01_ip_domainNameLabel": {
            "defaultValue": null,
            "type": "SecureString"
        },
        "certificates_AzureClassicRunAsCertificate_base64Value": {
            "defaultValue": null,
            "type": "SecureString"
        },
        "certificates_AzureRunAsCertificate_base64Value": {
```

Why templates and how they are evaluated?

Chmurowisko

- Idempotency – same template to many RG's give you same results
- Simplified orchestration – one template -> many resources
- Build whole solution with dependencies using parameters, variables and functions
- If one template is not enough, linked templates will help a bit.

DEMO
Azure Resource Group Deployment History

Chmurowisko

ARM Functions

Chmurowisko

Array and object functions

Resource Manager provides several functions for working with arrays and objects.

- array
- coalesce
- concat
- contains
- createArray
- empty
- first
- intersection
- json
- last
- length
- min
- max
- range
- skip
- take
- union

Comparison functions

Resource Manager provides several functions for making c
templates.

- equals
- less
- lessOrEquals
- greater
- greaterOrEquals

Deployment value functions

Resource Manager provides the following functions for getting values from sections of the template and values related to the deployment:

- deployment
- parameters
- variables

Logical functions

Resource Manager provides the following functions for working with logical conditions:

- and
- bool
- if
- not
- or

Numeric functions

Resource Manager provides the following functions for working with integers:

- add
- copyIndex
- div
- float
- int
- min
- max
- mod
- mul
- sub

ARM Functions

Resource functions

Resource Manager provides the following functions for getting resource v

- listAccountSas
- listKeys
- listSecrets
- list*
- providers
- reference
- resourceGroup
- resourceId
- subscription

String functions

Resource Manager provides the following functions for working with strings:

- base64
- base64ToJson
- base64ToString
- concat
- contains
- dataUri
- dataUriToString
- empty
- endsWith
- first
- guid
- indexOf
- last
- lastIndexOf
- length
- padLeft
- replace
- skip
- split
- startsWith
- string
- substring
- take
- toLower
- toUpper
- trim
- uniqueString
- uri
- uriComponent
- uriComponentToString

Chmurowisko

Functions – How to use them?

Chmurowisko

```
{
    "$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#",
    "contentVersion": "1.0.0.0",
    "resources": [
        {
            "apiVersion": "2016-01-01",
            "type": "Microsoft.Storage/storageAccounts",
            "name": "[concat(copyIndex(),'storage', uniqueString(resourceGroup().id))]",
            "location": "[resourceGroup().location]",
            "sku": {
                "name": "Standard_LRS"
            },
            "kind": "Storage",
            "properties": {},
            "copy": {
                "name": "storagecopy",
                "count": 4,
                "mode": "serial",
                "batchSize": 2
            }
        }
    ],
    "outputs": {}
}
```

Functions – Create your own

```
"functions": [
    {
        "namespace": "nicNameCount",
        "members": {
            "getNicName": {
                "parameters": [
                    {
                        "name": "prefix",
                        "type": "string"
                    },
                    {
                        "name": "vmType",
                        "type": "string"
                    },
                    {
                        "name": "id",
                        "type": "string"
                    }
                ],
                "output": {
                    "type": "string",
                    "value": "[concat(toLower(parameters('prefix')),'-',parameters('vmType'),'-nic-'
                }
            }
        }
    }
],
```

Chmurowisko

Chmurowisko

- *az group deployment create --name chmurowiskodeployment -g mftest02123 --template-file storageAccount_template1.json –debug*

- *az group deployment show -g mf01rg -n mf01rgdep* ***--query properties.correlationId***

- *az group deployment* ***operation list*** *-g mf01rg -n mf01rgdep*

- *az monitor activity-log list* ***--correlation-id "028da53d-e51e-4f77-b0d1-cf5694a9db56"***

Template Limits

Chmurowisko

- Limit the size of your template to 1 MB, and each parameter file to 64 KB.
- The 1-MB limit applies to the final state of the template (after all copy functions)
- Other limits:
  - 256 parameters
  - 256 variables
  - 800 resources (including copy count)
  - 64 output values
  - 24,576 characters in a template expression

Lesson 1: ARM Templates - Summary

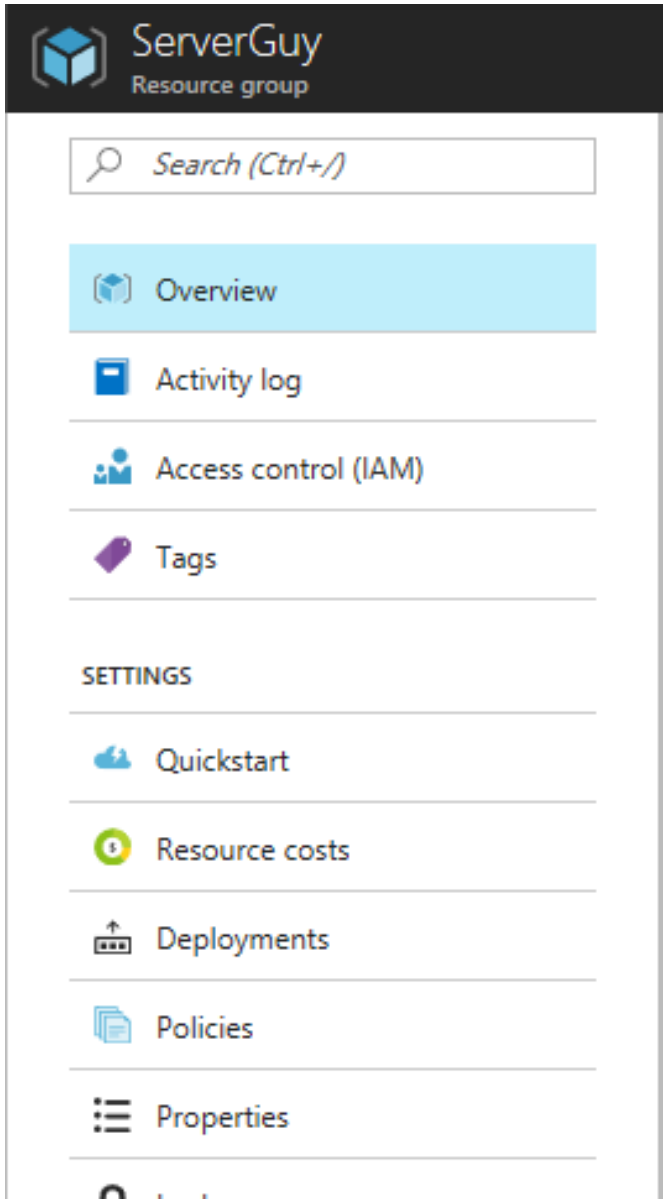
- Azure Resource Manager
- JSON Templates
- Debugging

Lesson 2: Role-Based Access Control (RBAC)

- Role-Based Access Control
- Role Assignment
- Resource Scope

- Azure role-based access control allows granular access by users, groups and applications to resources

- Available through portal.azure.com, each resource has an Access Control (IAM) blade



Chmurowisko

Chmurowisko

- Many roles available
- if not suitable for the purpose, custom roles can be created

ROLE NAME	DESCRIPTION
Contributor	Contributors can manage everything except access.
Owner	Owner can manage everything, including access.
Reader	Readers can view everything, but can't make changes.
User Access Administrator	Lets you manage user access to Azure resources.
Virtual Machine Contributor	Lets you manage virtual machines, but not access to them, and not the virtual network or storage account they're connected to.

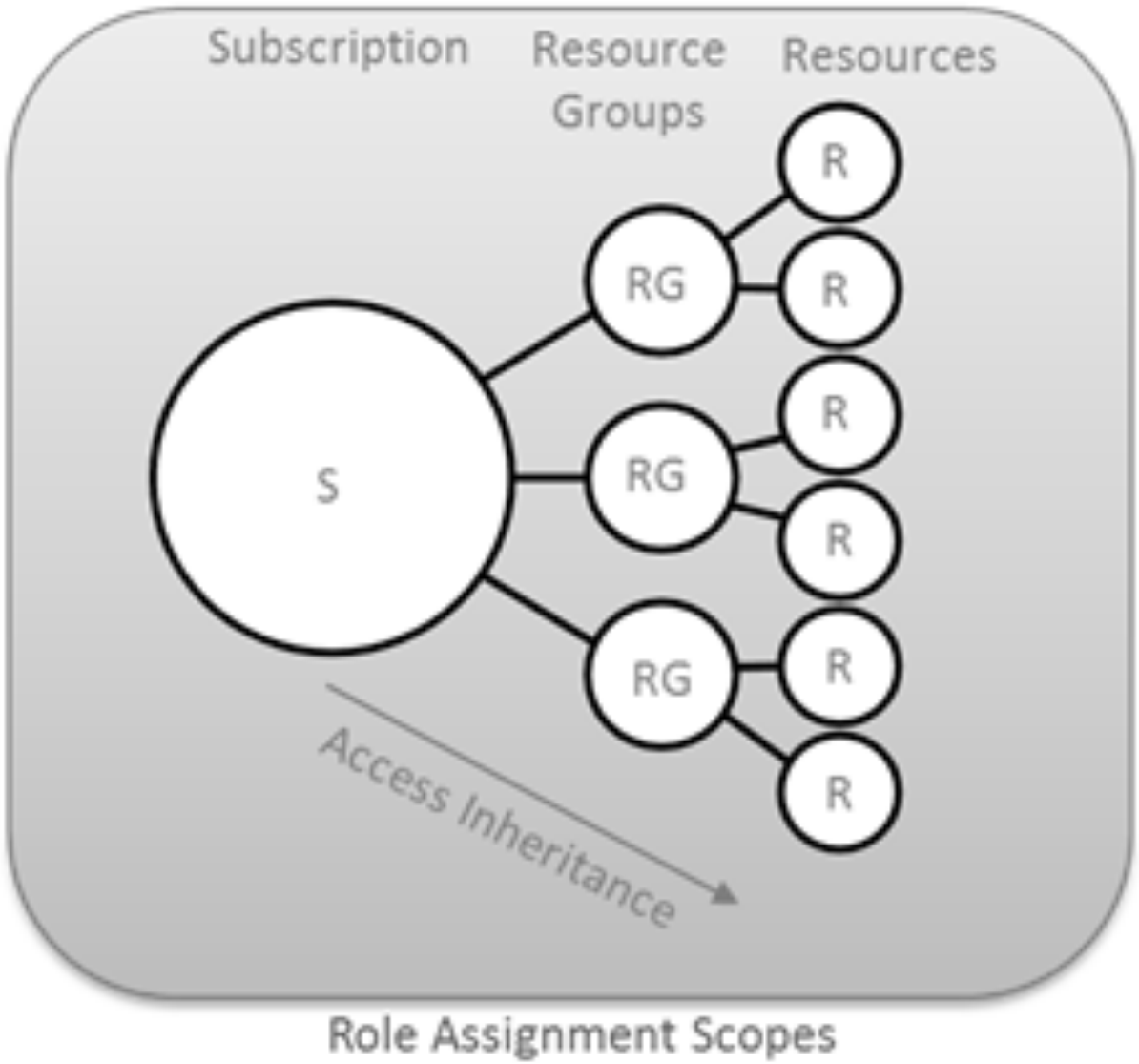
Chmurowisko

- **Users**: From the same Azure AD and same subscription

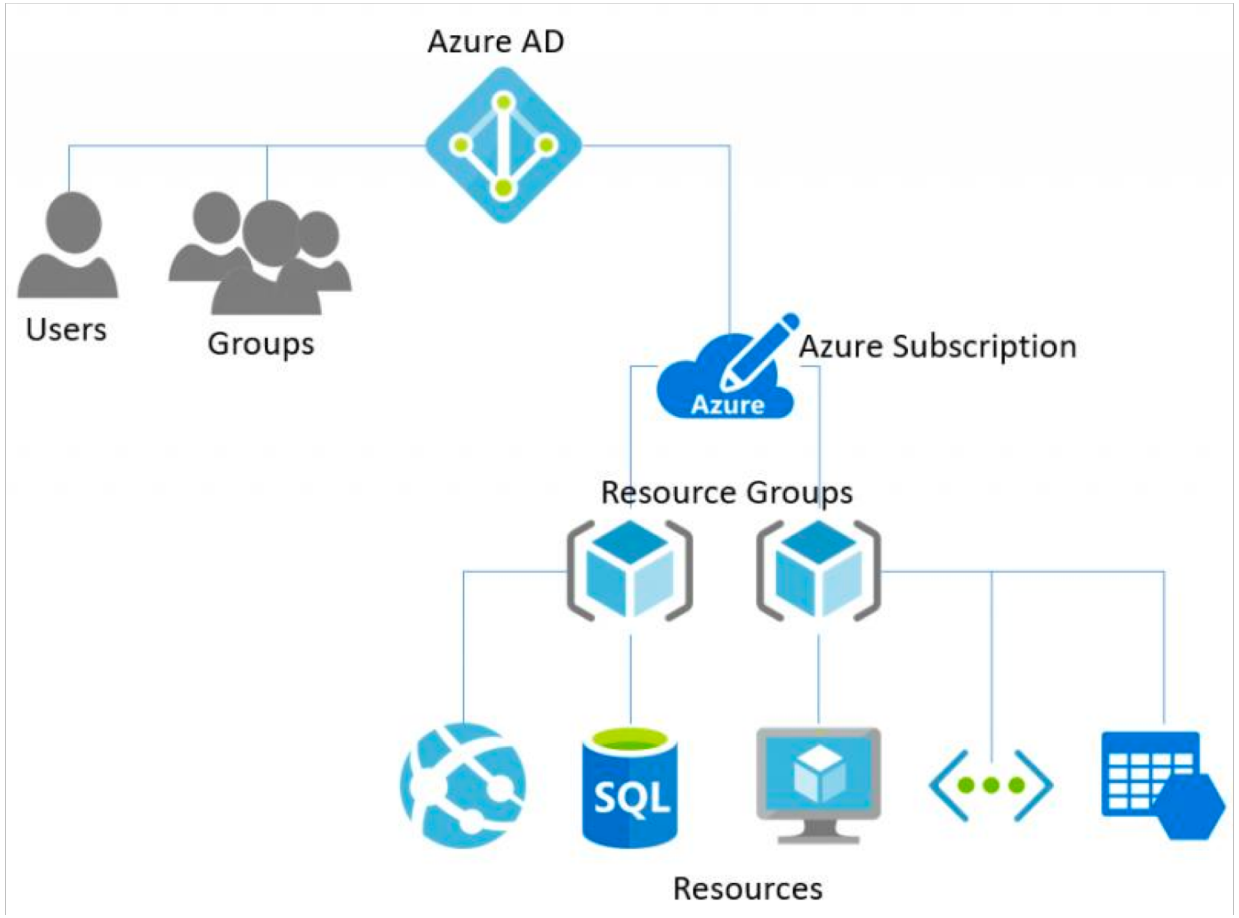
- **Groups**: If a role is assigned to a group, a user receives the rights of the role when added to the group. The user also automatically loses access to the resource after getting removed from the group

- **Service principals**: Services can be granted access to Azure resources by assigning roles via the Azure module for PowerShell to the Azure AD service principal representing that service

Chmurowisko



Role Assignment Scopes



Custom Role

JSON Kopiuj

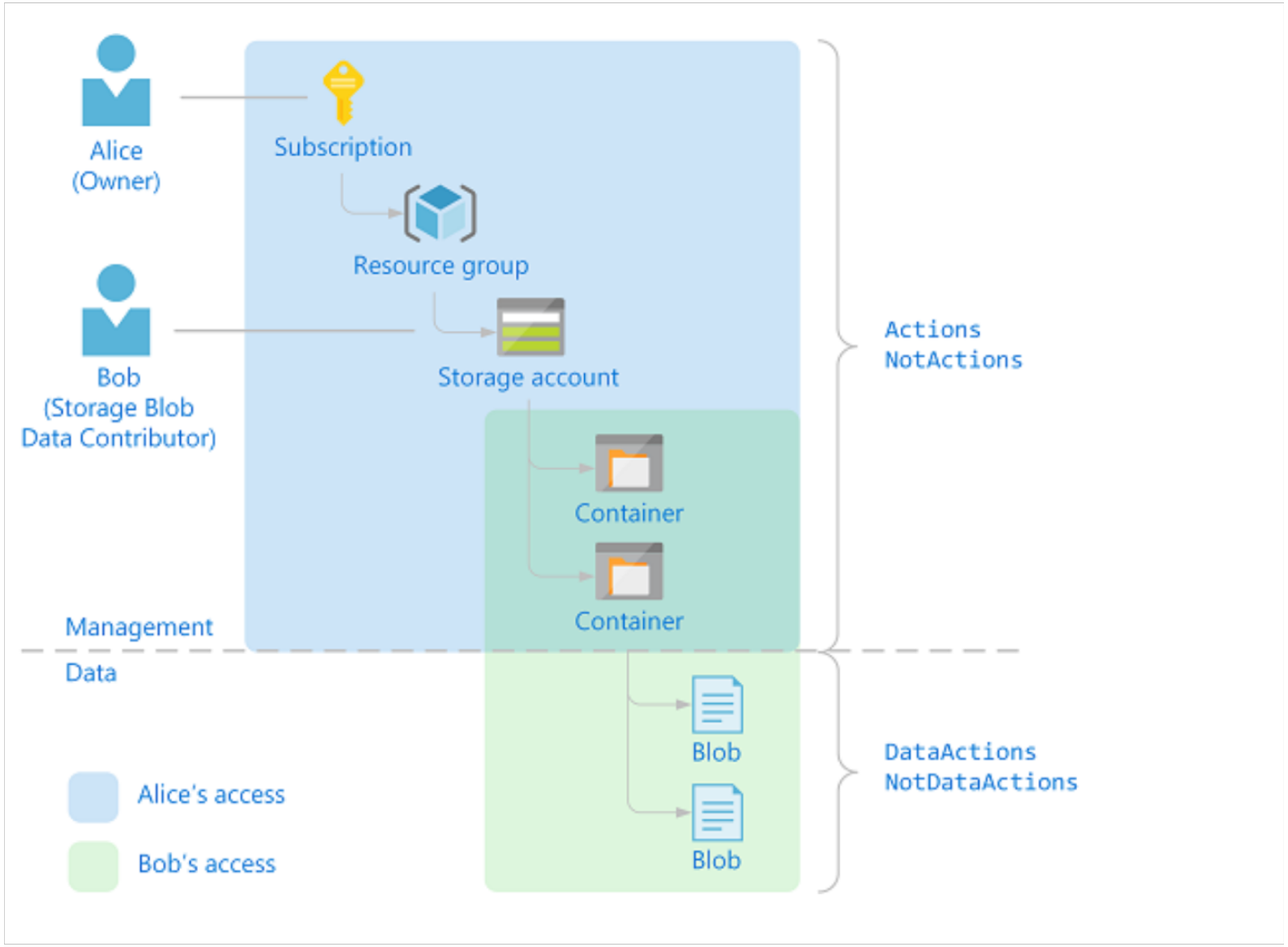
```
{
  "Name":  "Virtual Machine Operator",
  "Id":  "88888888-8888-8888-8888-888888888888",
  "IsCustom":  true,
  "Description":  "Can monitor and restart virtual machines.",
  "Actions":  [
                "Microsoft.Storage/*/read",
                "Microsoft.Network/*/read"
                ],
  "NotDataActions":  [

                    ],
  "AssignableScopes":  [
                         "/subscriptions/{subscriptionId1}",
                         "/subscriptions/{subscriptionId2}",
                         "/subscriptions/{subscriptionId3}"
                       ]
}
```

- Actions

If a user is assigned a role that excludes an operation in `NotActions`, and is assigned a second role that grants access to the same operation, the user is allowed to perform that operation. `NotActions` is not a deny rule – it is simply a convenient way to create a set of allowed operations when specific operations need to be excluded.

Custom Role

Chmurowisko



Custom Role

Chmurowisko

Actions

```
Microsoft.Storage/storageAccounts/blobServices/containers/delete
```

```
Microsoft.Storage/storageAccounts/blobServices/containers/read
```

```
Microsoft.Storage/storageAccounts/blobServices/containers/write
```

DataActions

```
Microsoft.Storage/storageAccounts/blobServices/containers/blobs/delete
```

```
Microsoft.Storage/storageAccounts/blobServices/containers/blobs/read
```

```
Microsoft.Storage/storageAccounts/blobServices/containers/blobs/write
```

Assignable Scopes

Chmurowisko

Scenario	Example
Role is available for assignment in a single subscription	`"/subscriptions/c276fc76-9cd4-44c9-99a7-4fd71546436e"`
Role is available for assignment in two subscriptions	`"/subscriptions/c276fc76-9cd4-44c9-99a7-4fd71546436e", "/subscriptions/e91d47c4-76f3-4271-a796-21b4ecfe3624"`
Role is available for assignment only in the Network resource group	`"/subscriptions/c276fc76-9cd4-44c9-99a7-4fd71546436e/resourceGroups/Network"`
Role is available for assignment in all scopes	`"/"`

Lesson 2: Role-Based Access Control - Summary

- Role-Based Access Control
- Role Assignment
- Resource Scope

Lesson 3: Resource Policies

- Azure Resource Policies
- Policy vs RBAC
- Built-In Policies
- Policy Definition
- Policy Assignment
- Policies for Naming Conventions
- Policy validation

Azure Resource Policies

Chmurowisko

- Provides resource conventions in an organization and consists of:
  - **policy definition** - describe when and what action to take
  - **policy assignment** - apply the policy definition to a scope

Policy vs RBAC

Chmurowisko

RBAC

- RBAC controls user access (need RBAC to create resources)
- Policies control resources (need RBAC to use policies)

The Contributor role cannot create or apply policies

Permissions

- To define requires: Microsoft.Authorization/policydefinitions/write
- To apply requires: Microsoft.Authorization/policyassignments/write

Built-In Policies

Chmurowisko

Azure provides built-in policy definition limiting the number users need to define;

Some examples are:

- Allowed locations
- Allowed resource types
- Allowed storage account SKUs
- Allowed virtual machine SKUs
- Not allowed resource types

Definitions are stored in JSON

Policy Definition

Chmurowisko

How to define:

- Use All Mode
- Use Parameters
- Policy Rule contains simple if and then blocks

```
{
    "if": {
        <condition> | <logical operator>
    },
    "then": {
        "effect": "deny | audit | append | auditIfNotExists | deployIfNotExists"
    }
}
```

Policy Definition

Chmurowisko

Effect

Policy supports the following types of effect:

- **Deny**: generates an event in the audit log and fails the request
- **Audit**: generates a warning event in audit log but does not fail the request
- **Append**: adds the defined set of fields to the request
- **AuditIfNotExists**: enables auditing if a resource does not exist
- **DeployIfNotExists**: deploys a resource if it does not already exist.

Policy Assignment

- Using PowerShell:

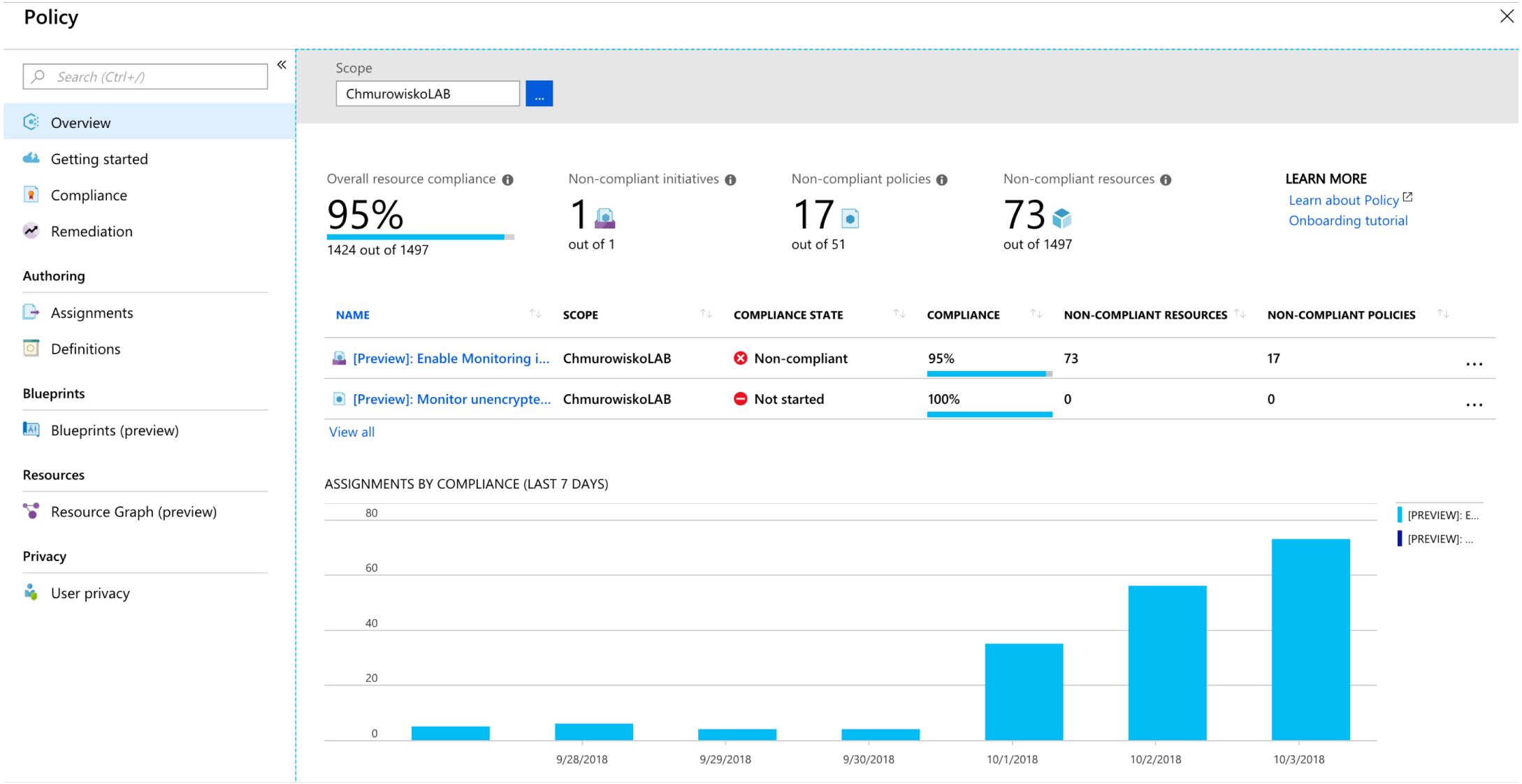
```
$rg = Get-AzureRmResourceGroup -Name “RG-SZKOLACHMURY"

$definition = Get-AzureRmPolicyDefinition -Id
/providers/Microsoft.Authorization/policyDefinitions/a57364a-
7474-ed43-c564-bf8b9038c4c

New-AzureRMPolicyAssignment -Name ”VM Sizes Assignment” -Scope
$rg.ResourceId -PolicyDefinition $definition
```

```
$rg = Get-AzureRmResourceGroup -Name “RG-SZKOLACHMURY"
$definition = Get-AzureRmPolicyDefinition -Id /providers/Microsoft.Authorization/policyDefinitions/a57364a-7474-ed43-c564-bf8b9038c4c
New-AzureRMPolicyAssignment -Name VM Sizes Assignment -Scope $rg.ResourceId -PolicyDefinition $definition
```

Policy Assignment

- Using the Portal

Chmurowisko



Policies for Naming Conventions

Prescribe how organization resources are named:

- Wildcard
- Pattern
- Tags
- Multiple patterns

```
{
    "properties": {
        "displayName": "Name pattern with like condition.",
        "description": "Enforce a naming pattern on resources with the like condition.",
        "mode": "Indexed",
        "parameters": {
            "namePattern": {
                "type": "String",
                "metadata": {
                    "description": "Pattern to use for names. Can include wildcard (*)."
                }
            }
        },
        "policyRule": {
            "if": {
                "not": {
                    "field": "name",
                    "like": "[parameters('namePattern')]"
                }
            },
            "then": {
                "effect": "deny"
            }
        }
    }
}
```

Policies for Naming Conventions

Chmurowisko

- Pattern:

```
{
  "if": {
    "not": {
      "field": "name",
      "match": "contoso??????"
    }
  },
  "then": {
    "effect": "deny"
  }
}
```

Lesson 3: Resource Policies - Summary

- Azure Resource Policies
- Policy vs RBAC
- Built-In Policies
- Policy Definition
- Policy Assignment
- Policies for Naming Conventions
- Policy validation

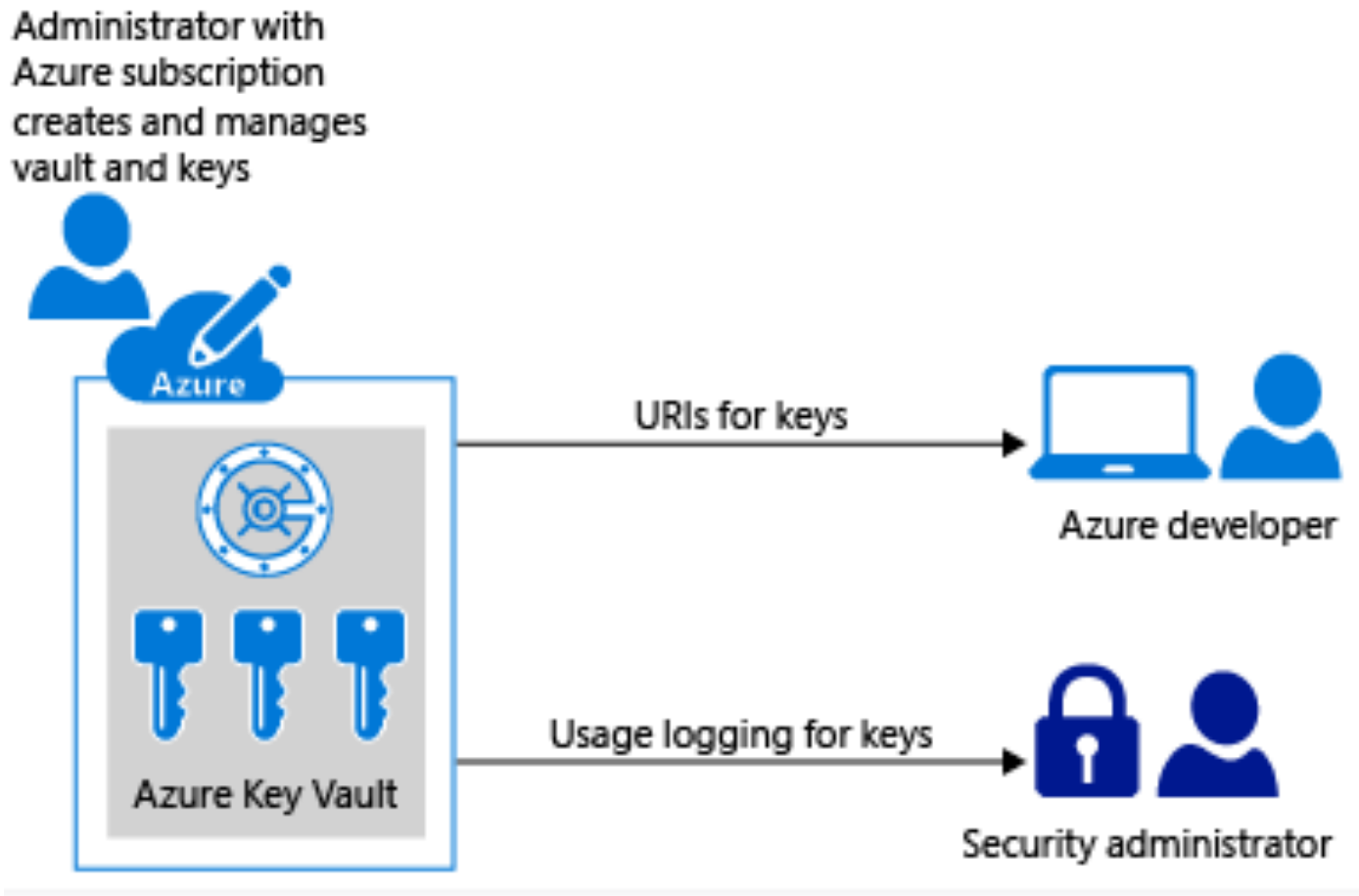
Lesson 4: Security

- Azure Key Vault
- Key Vault Use in ARM Templates

Azure Key Vault

Chmurowisko

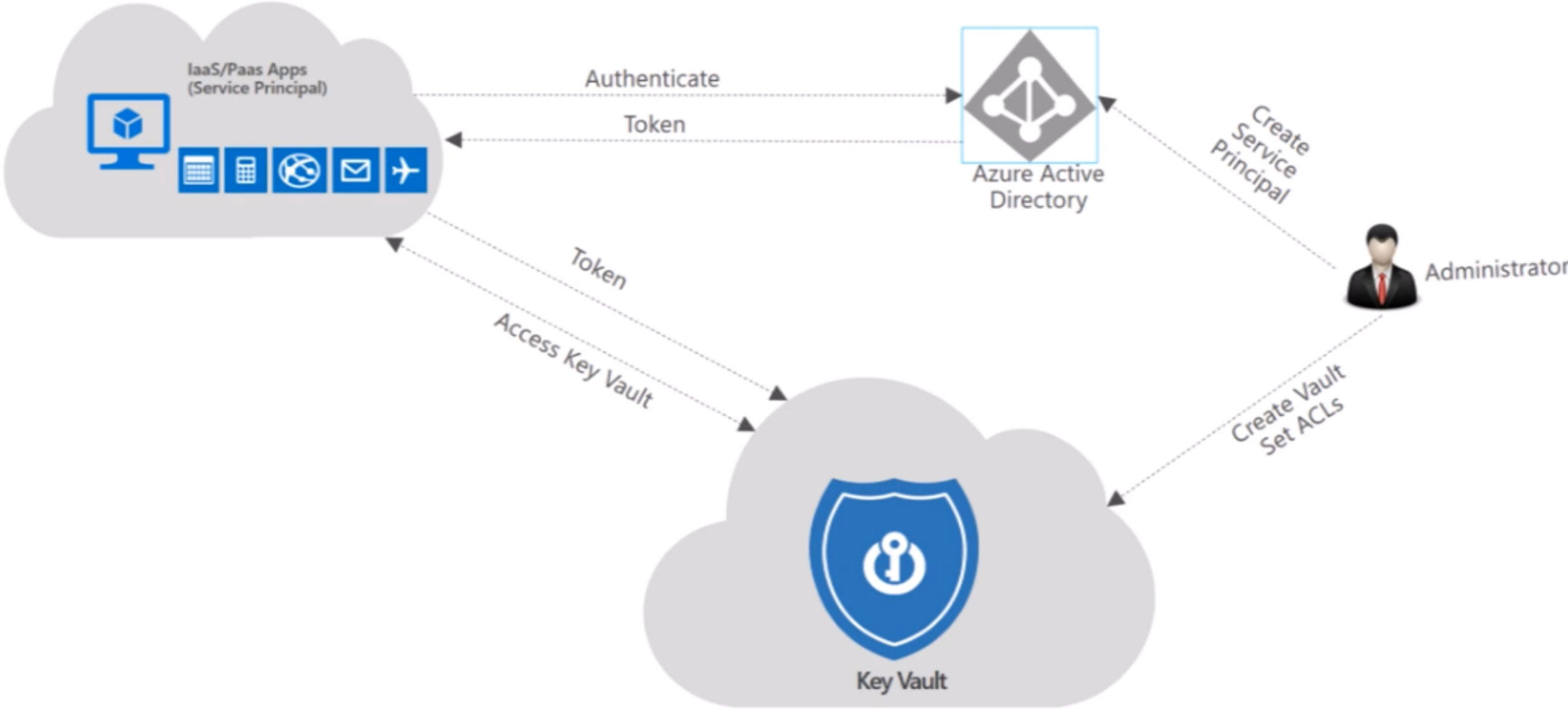
- While deploying resources, secrets are required ☺
- These should not be passed but stored in the Azure Key Vault.



Key Vault Use in Azure

- Application access without passing credentials

Chmurowisko



KeyVault is based on Thales HSM’s

Chmurowisko

More information about Thales HSMs and Microsoft services

Thales e-Security is a leading global provider of data encryption and cyber security solutions to the financial services, high technology, manufacturing, government, and technology sectors. With a 40-year track record of protecting corporate and government information, Thales solutions are used by four of the five largest energy and aerospace companies. Their solutions are also used by 22 NATO countries, and secure more than 80 per cent of worldwide payment transactions.

Microsoft has collaborated with Thales to enhance the state of art for HSMs. These enhancements enable you to get the typical benefits of hosted services without relinquishing control over your keys. Specifically, these enhancements let Microsoft manage the HSMs so that you do not have to. As a cloud service, Azure Key Vault scales up at short notice to meet your organization’s usage spikes. At the same time, your key is protected inside Microsoft’s HSMs: You retain control over the key lifecycle because you generate the key and transfer it to Microsoft’s HSMs.

KeyVault in ARM Templates

Chmurowisko

Several steps to allow Key Vault use in template deployment:

- Deploy a Key vault and Secret in it, Set the Secret
- Enable access to the secret
- Either:
  - Reference the secret with a static ID
  - Reference the secret with a dynamic ID

Set Key Vault enabledForTemplateDeployment property to true at creation.

This will permit access from Resource Manager templates during deployment.

```
keyvaultname=szklchm-keyvault01
rgname=rg-keyvault
az keyvault update -n $keyvaultname -g $rgname --set properties.enabledForDeployment=true
az keyvault update -n $keyvaultname -g $rgname --set properties.enabledForTemplateDeployment=true
```

ARM Template & Azure Key Vault

Chmurowisko

```
{
  "$schema": "http://schema.management.azure.com/schemas/2015-01-01/deploymentParameters.json#",
  "contentVersion": "1.0.0.0",
  "parameters": {
    "sqlAdministratorLogin": {
      "reference": {
          "keyVault": {
            "id": "/subscriptions/[redacted]/resourceGroups/[redacted]
t/providers/Microsoft.KeyVault/vaults/[redacted]"
          },
          "secretName": "sqlusername"
        }
    },
    "sqlAdministratorLoginPassword": {
      "reference": {
          "keyVault": {
            "id": "/subscriptions/[redacted]/resourceGroups/[redacted]
t/providers/Microsoft.KeyVault/vaults/[redacted]"
          },
          "secretName": "sqlpassword"
        }
    }
  }
}
```

Lesson 4: Security - Summary

- Azure Key Vault
- Key Vault Use in ARM Templates

Lesson 5: Building Blocks

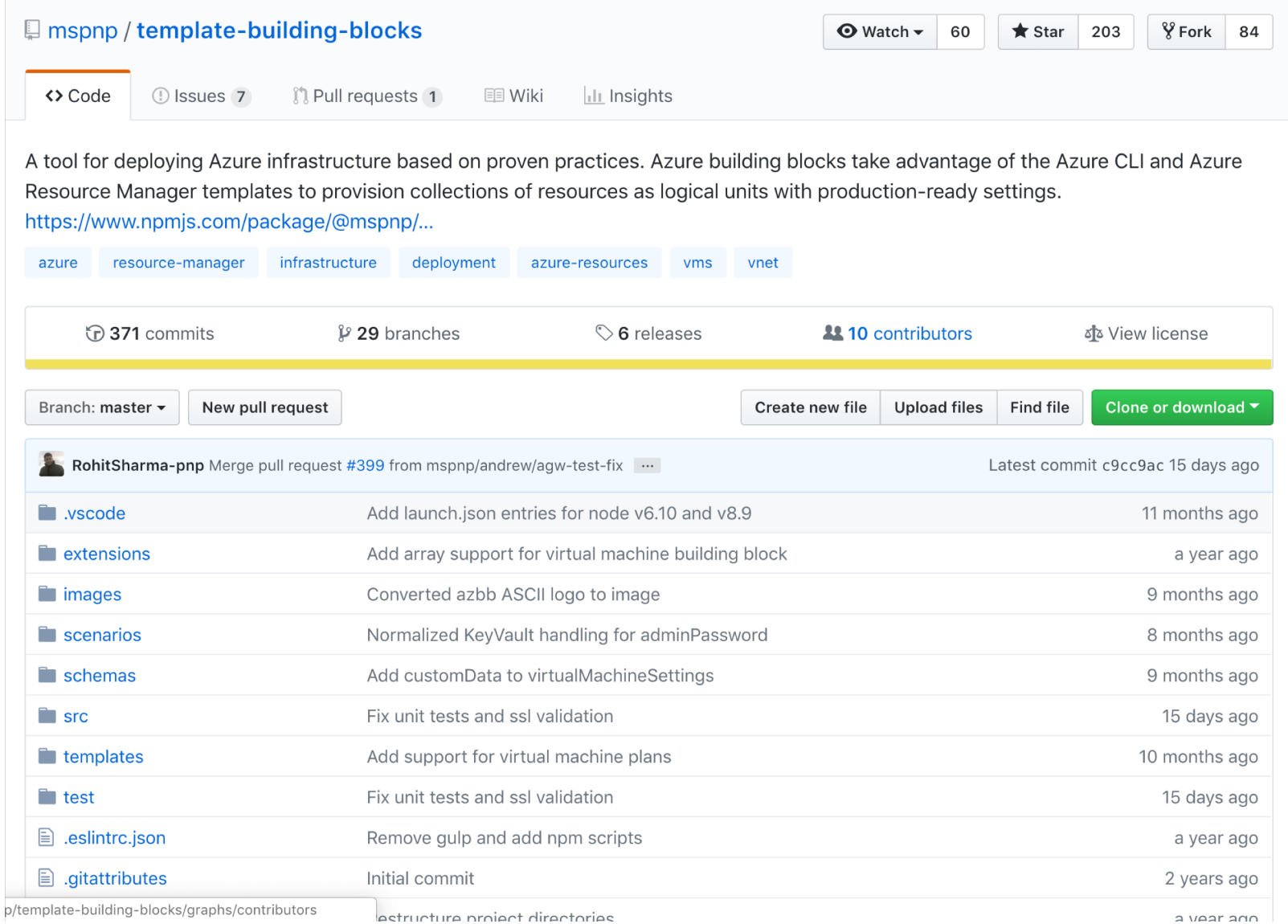
- Azure Building Blocks
- Deploying Resources using Building Blocks

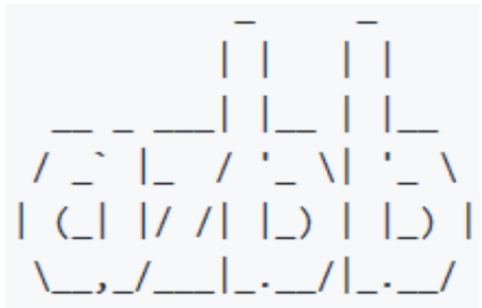
Azure Building Blocks

Chmurowisko

- Designed to simplify deployment of Azure resources
- Provides a command line tool and set of Azure Resource Manager templates

https://github.com/mspnp/template-building-blocks/





Azure Building Blocks: Simplifying Resource Deployment

> **Important Note**: Version 2.1.1 of Azure Building Blocks introduced breaking changes to versions earlier than 2.0.4. Versions earlier than 2.0.4 will no longer function. Please upgrade to version 2.0.4 or greater to continue using Azure Building Blocks.

The Azure Building Blocks project is a command line tool and set of Azure Resource Manager templates designed to simplify deployment of Azure resources. Users author a set of simplified parameters to specify settings for Azure resources, and the command line tool merges these parameters with best practice defaults to produce a set of final parameter files that can be deployed with the Azure Resource Manager templates.

Supported Resources

Chmurowisko

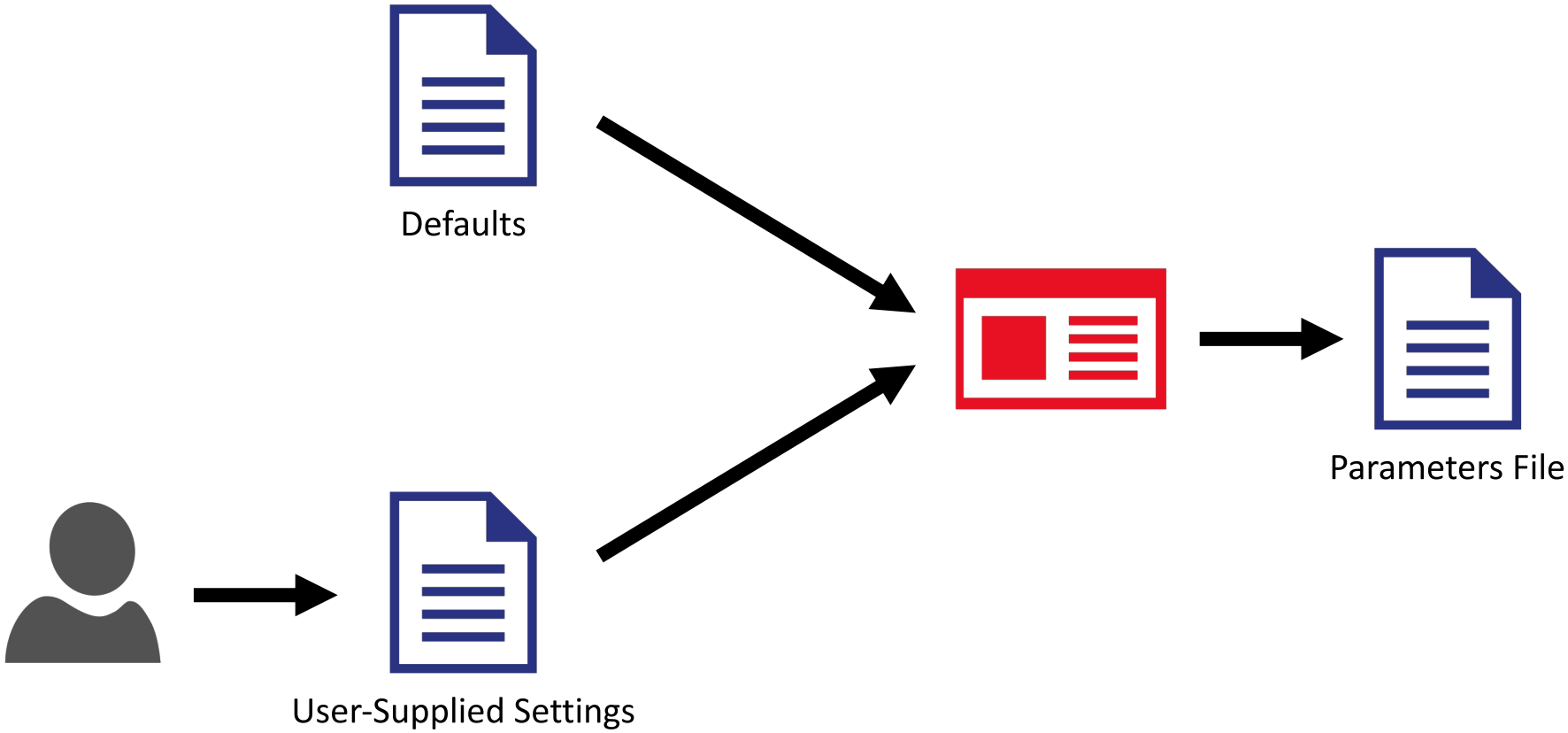
Building Blocks support the following resource types:

- Virtual Networks
- Virtual Machines
- Virtual Machine Extensions
- Load Balancers
- Route Tables
- Network Security Groups
- Virtual Network Gateways
- Virtual Network Connection

Deploying Resources Using Building Blocks

Chmurowisko

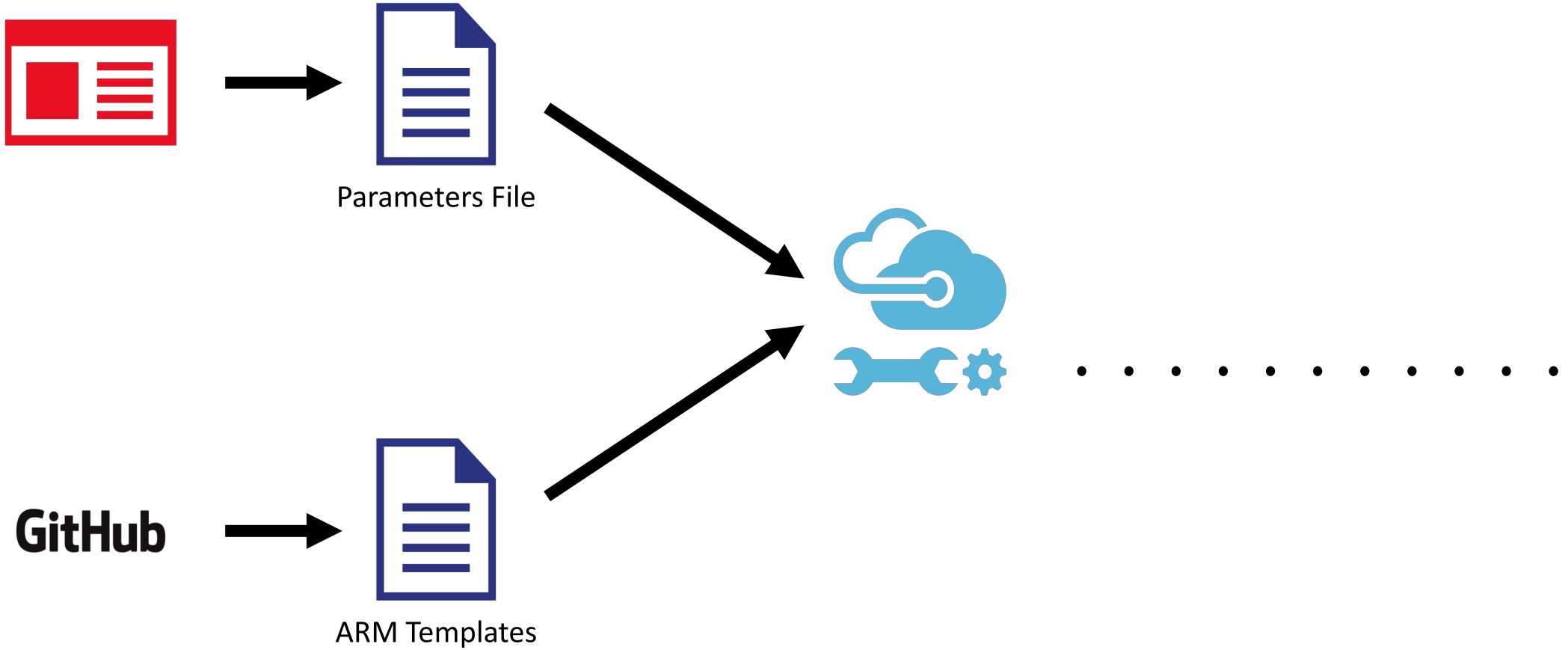
- Creating a Parameters File



Deploying Resources Using Building Blocks

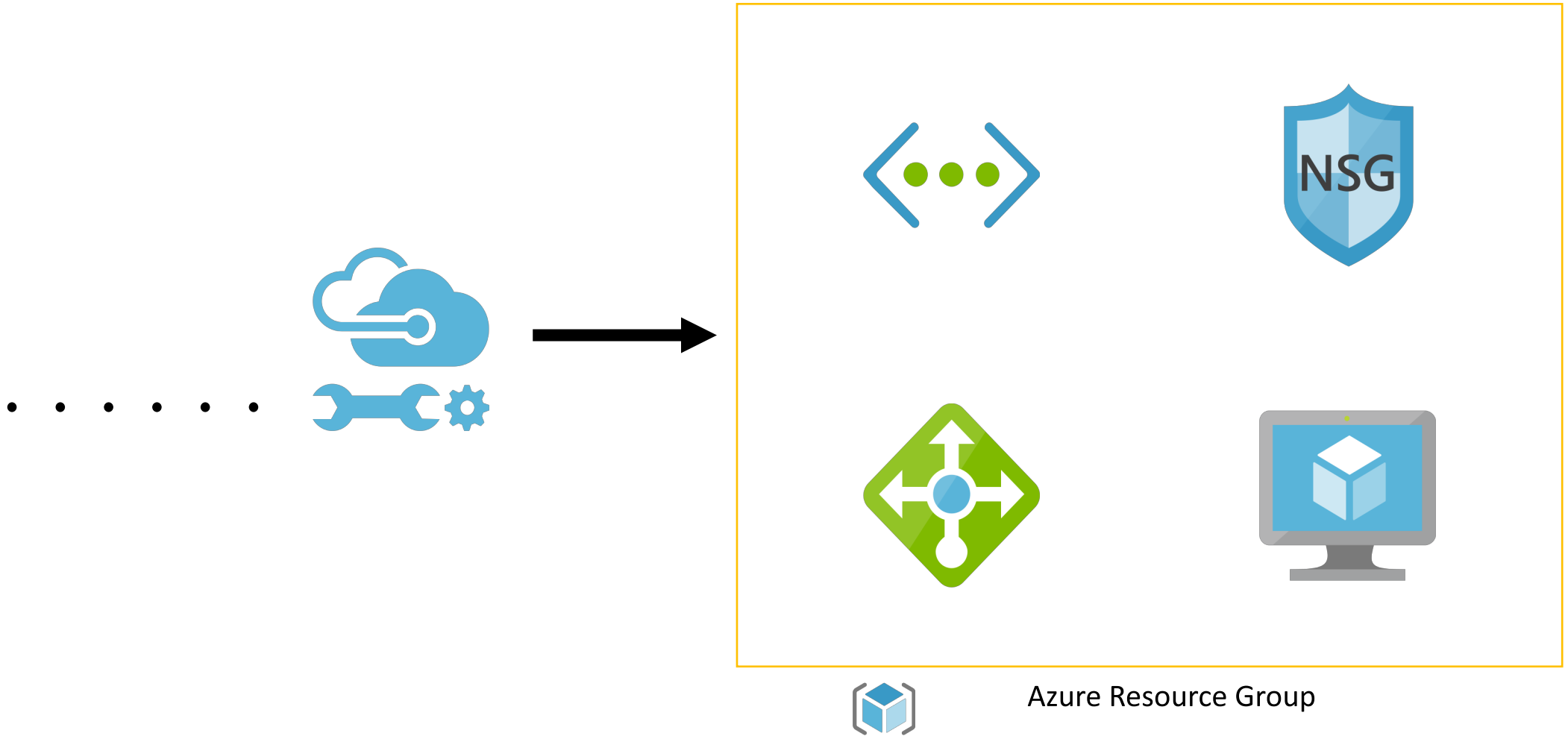
Chmurowisko

- Running a Parameters File



Deploying Resources Using Building Blocks

- Template Output



azbb

Chmurowisko

Examples

Azure Building Blocks parameters to deploy three identical VMs:

```
"type": "VirtualMachine",
"settings": {
    "vmCount": 3,
    "osType": "windows",
    "namePrefix": "test",
    "adminPassword": "testPassw0rd!23",
    "nics": [{"subnetName": "web"}],
    "virtualNetwork": {"name": "ra-vnet"}
}
```

Lesson 5: Building Blocks - Summary

- Azure Building Blocks
- Deploying Resources using Building Blocks

Summary

Chmurowisko

- ARM Templates
- Role-Based Access Control (RBAC)
- Resource Policies
- Security
- Linked Templates & Building Blocks