Contents

[ARM 1](#_Toc53626731)

[ARM Syntax 2](#_Toc53626732)

[CLI 2](#_Toc53626733)

[Cli structure 2](#_Toc53626734)

[Finding commands 2](#_Toc53626735)

[Globally available arguments 2](#_Toc53626736)

[Interactive mode 3](#_Toc53626737)

[Power Shell 3](#_Toc53626738)

[PS Verbs: 3](#_Toc53626739)

[New 3](#_Toc53626740)

[Add 3](#_Toc53626741)

[Disable 3](#_Toc53626742)

[Get 4](#_Toc53626743)

[Set 4](#_Toc53626744)

[Invoke 4](#_Toc53626745)

[Remove 4](#_Toc53626746)

[Similar Verbs for Different Actions 4](#_Toc53626747)

[Common Commands 4](#_Toc53626748)

[Get-Help 4](#_Toc53626749)

[Get-Command 5](#_Toc53626750)

[Azure Blueprint 5](#_Toc53626751)

# ARM

## ARM Syntax

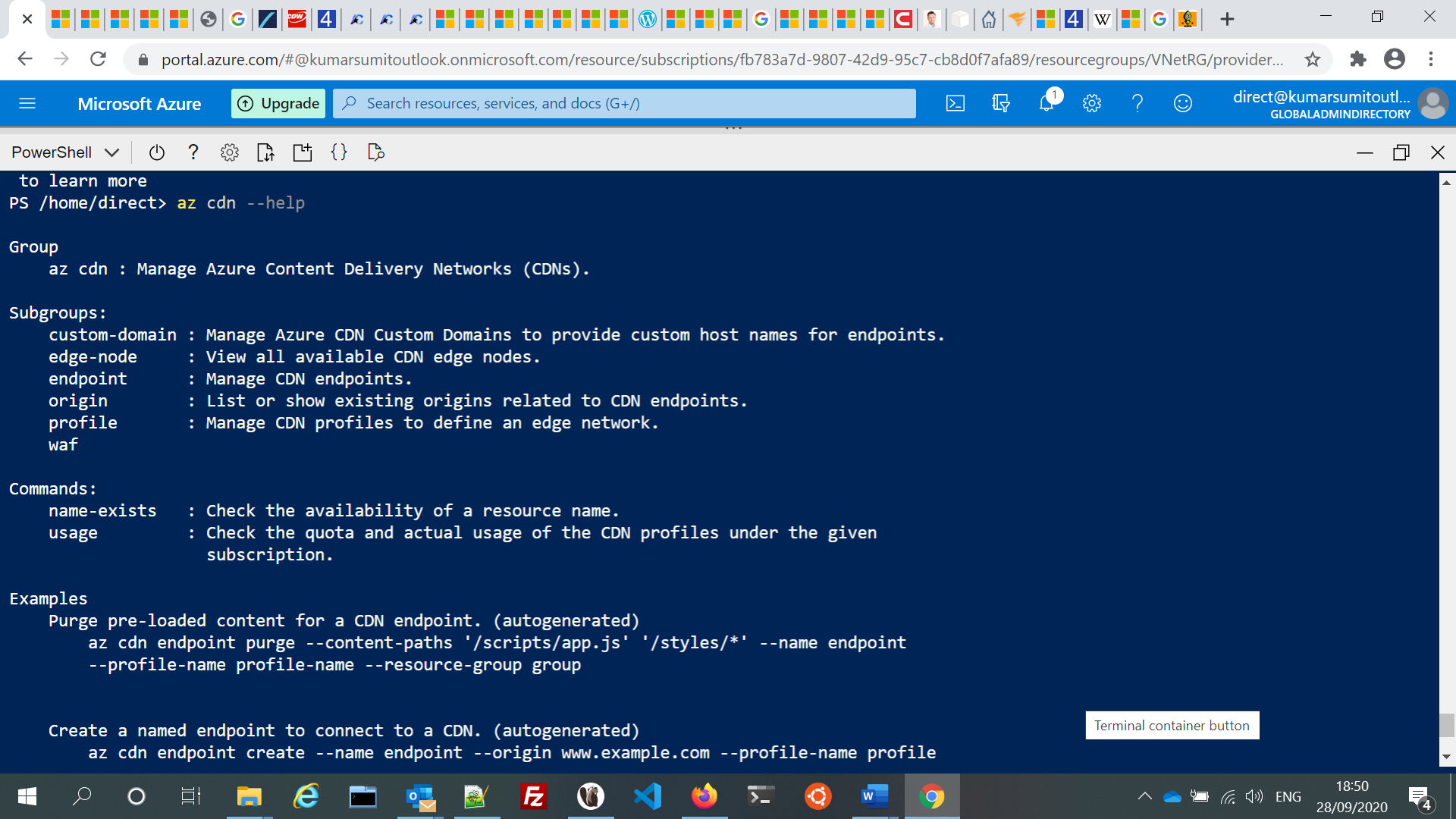
<https://docs.microsoft.com/en-us/azure/azure-resource-manager/templates/template-syntax>

# CLI

<https://docs.microsoft.com/en-us/cli/azure/get-started-with-azure-cli>

## Cli structure

**az <group> <subgroups> <commands> <parameters> <global Options>**



## Finding commands

To search for commands, use [az find](https://docs.microsoft.com/en-us/cli/azure/reference-index" \l "az-find).

az find secret

Use the --help argument to get a complete list of commands and subgroups of a group.

az network nsg –help

## Globally available arguments

* --help prints CLI reference information about commands and their arguments and lists available subgroups and commands.
* --output changes the output format. The available output formats are json, jsonc (colorized JSON), tsv (Tab-Separated Values), table (human-readable ASCII tables), and yaml.

## Interactive mode

The CLI offers an interactive mode that automatically displays help information and makes it easier to select subcommands.

az interactive

# Power Shell

<https://jussiroine.com/2019/03/mastering-azure-cli/>

<https://4sysops.com/archives/getting-started-with-azure-powershell/>

PowerShell has following types of commands:

* cmdlets
* aliases
* functions
* filters
* scripts
* applications

## PS Verbs:

<https://docs.microsoft.com/en-us/powershell/scripting/developer/cmdlet/approved-verbs-for-windows-powershell-commands>

### New

Creates a resource. (The Set verb can also be used when creating a resource that includes data, such as the Set-Variable cmdlet.)

### Add

Adds a resource to a container, or attaches an item to another item. For example, the Add-Content cmdlet adds content to a file. This verb is paired with Remove.

### Disable

Configures a resource to an unavailable or inactive state. For example, the Disable-PSBreakpoint cmdlet makes a breakpoint inactive. This verb is paired with Enable.

### Get

Specifies an action that retrieves a resource. This verb is paired with Set.

### Set

Replaces data on an existing resource or creates a resource that contains some data. For example, the Set-Date cmdlet changes the system time on the local computer. (The New verb can also be used to create a resource.) This verb is paired with Get

### Invoke

Performs an action, such as running a command or a method.

### Remove

Deletes a resource from a container. For example, the Remove-Variable cmdlet deletes a variable and its value. This verb is paired with Add.

## Similar Verbs for Different Actions

**New vs. Set**

The New verb is used to create a new resource. The Set verb is used to modify an existing resource, optionally creating the resource if it does not exist, such as the Set-Variable cmdlet.

**Find vs. Search**

The Find verb is used to look for an object. The Search verb is used to create a reference to a resource in a container.

**Get vs. Read**

The Get verb is used to retrieve a resource, such as a file. The Read verb is used to get information from a source, such as a file.

**Invoke vs. Start**

The Invoke verb is used to perform an operation that is generally a synchronous operation, such as running a command. The Start verb is used to begin an operation that is generally an asynchronous operation, such as starting a process.

**Ping vs. Test**

Use the Test verb.

## Common Commands

### Get-Help

<https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.core/get-help>

PS Alias : help, man

### Get-Command

<https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.core/get-command>

PS Alias

# Azure Blueprint

<https://docs.microsoft.com/en-us/azure/governance/blueprints/overview>

Blueprints are a declarative way to orchestrate the deployment of various resource templates and other artifacts such as:

* Role Assignments
* Policy Assignments
* Azure Resource Manager templates (ARM templates)
* Resource Groups

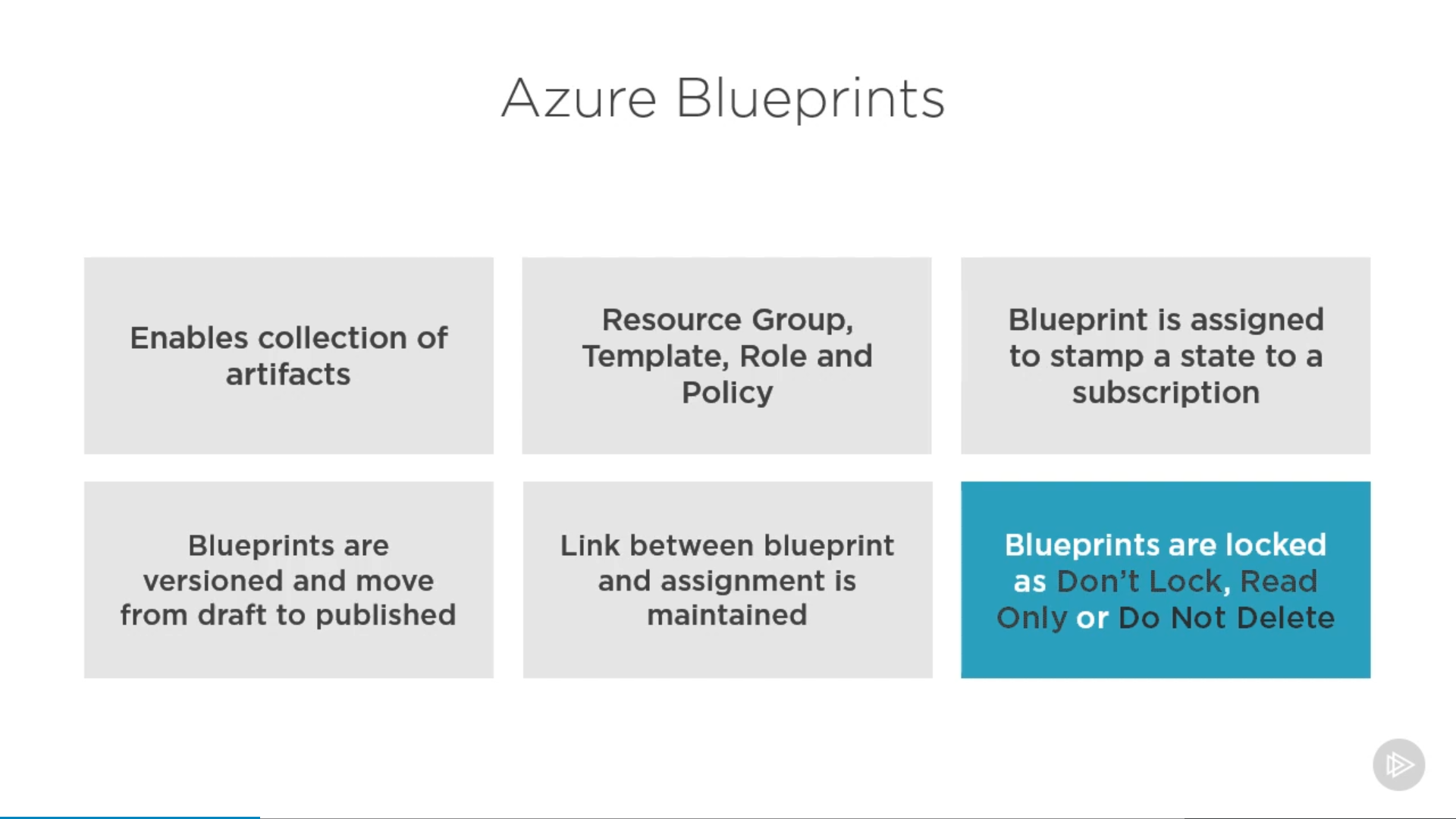
The Azure Blueprints service is backed by the globally distributed Azure Cosmos DB. Blueprint objects are replicated to multiple Azure regions.

Blueprint Stage:

* Blueprint definition (what should be deployed)
* Blueprint publishing: When a blueprint is first created, it's considered to be in **Draft** mode. When it's ready to be assigned, it needs to be **Published**. Publishing requires defining a **Version** string
* Blueprint assignment (what was deployed) : Each **Published** **Version** of a blueprint can be **assigned** to an existing **management group or subscription**.

Note:

* Assigning a blueprint definition to a management group means the assignment object exists at the management group. The deployment of artifacts still targets a subscription.
* **Blueprint are either associated with management group or subscription.**
* **Even an owner of subscription can’t remove blueprint resource locks.**



### How it's different from ARM templates

The service is designed to help with environment setup. This setup often consists of a set of resource groups, policies, role assignments, and ARM template deployments. A blueprint is a package to bring each of these artifact types together and allow you to compose and version that package, including through a CI/CD pipeline.

Nearly everything that you want to include for deployment in Azure Blueprints can be accomplished with an ARM template. However, an ARM template is a document that doesn't exist natively in Azure – each is stored either locally or in source control.

With Azure Blueprints, the relationship between the blueprint definition (what should be deployed) and the blueprint assignment (what was deployed) is preserved. This connection supports improved tracking and auditing of deployments.

Each blueprint can consist of zero or more ARM template artefacts.

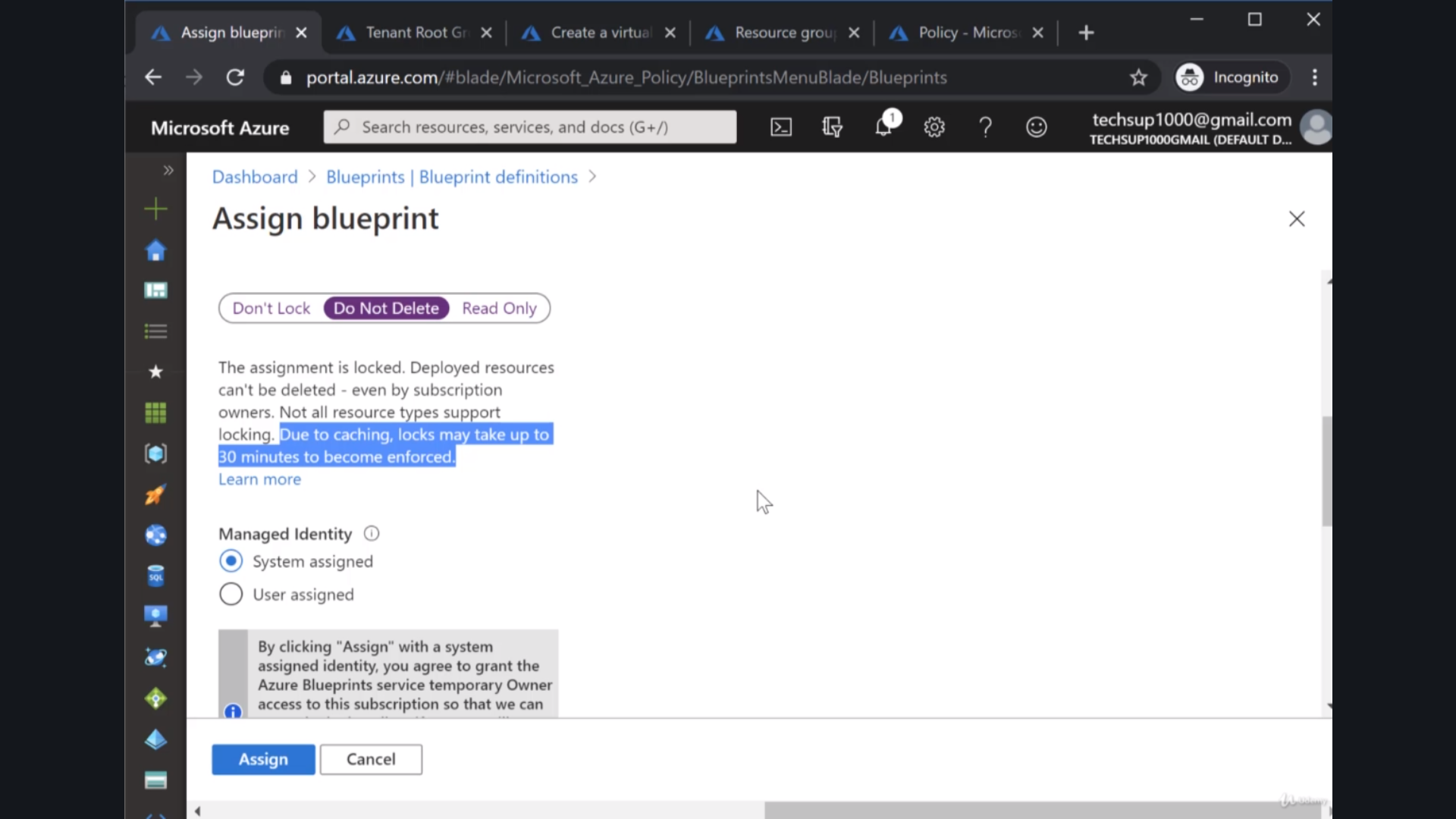
### How it's different from Azure Policy

A policy can be included as one of many artifacts in a blueprint definition.

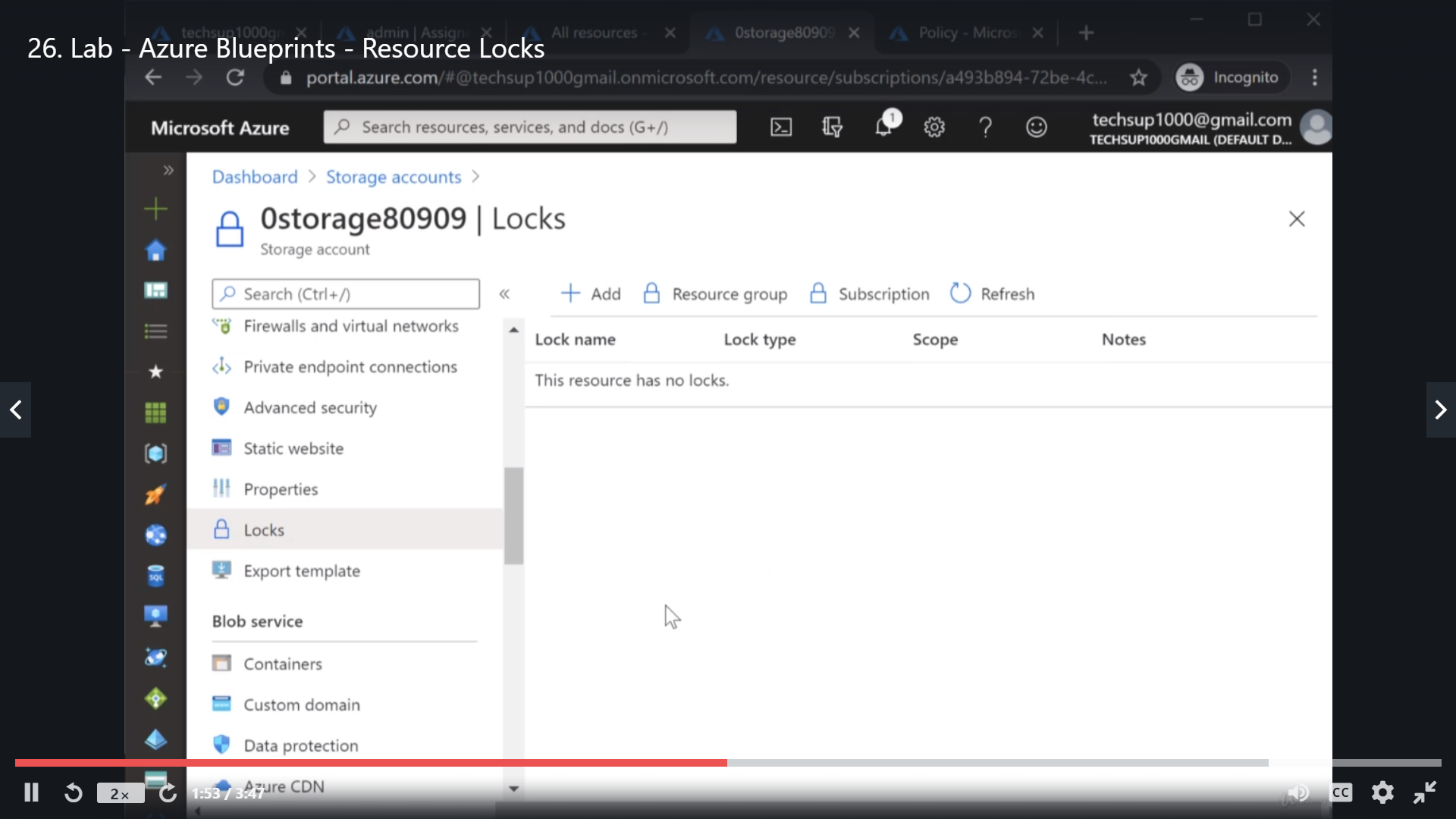
Including a policy in a blueprint enables the creation of the right pattern or design during assignment of the blueprint.

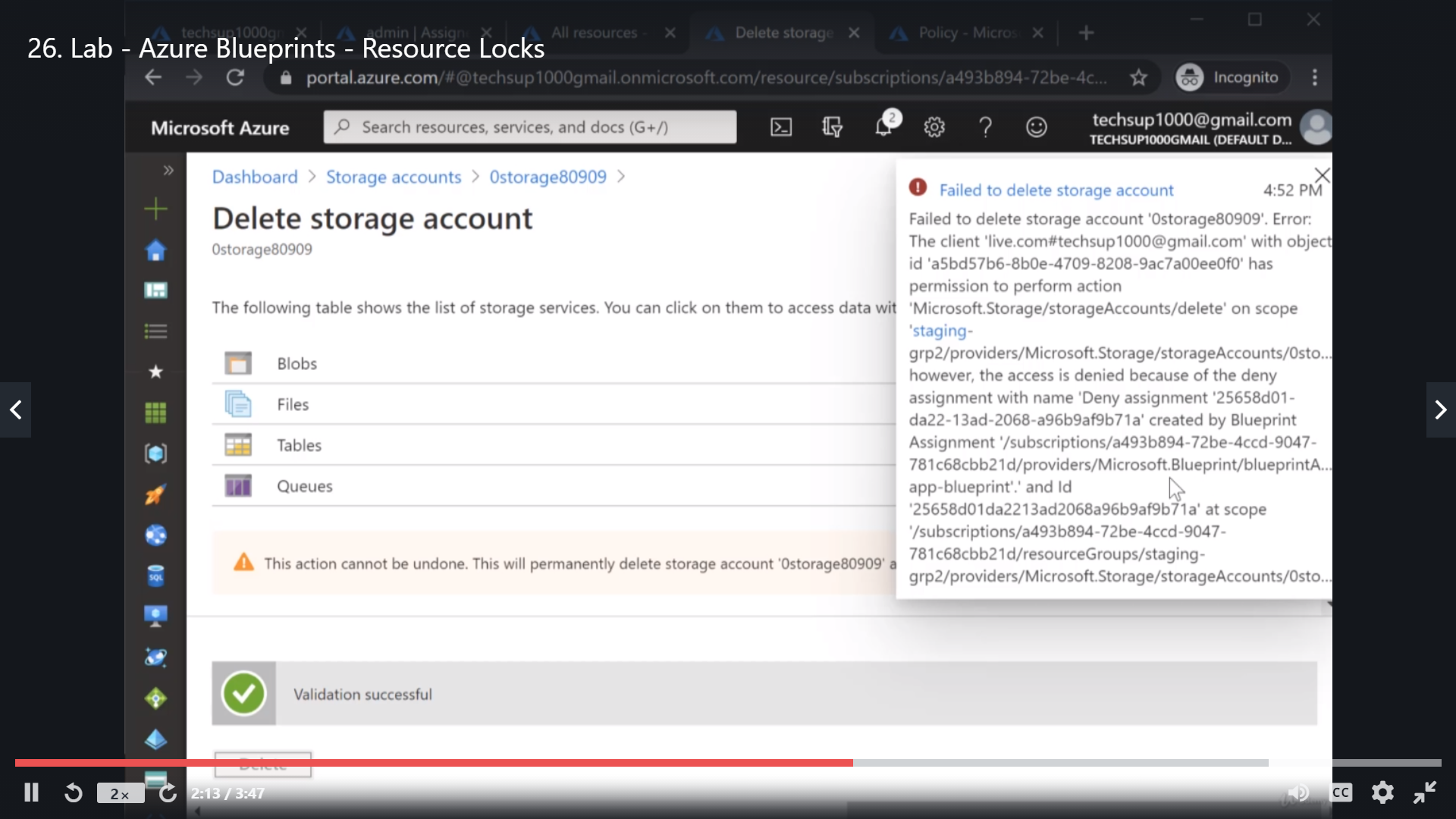
### Resource Locking in blueprint

<https://docs.microsoft.com/en-us/azure/governance/blueprints/concepts/resource-locking>



Despite Locks being applied at blueprint level, no lock gets visible at the resource level. However, if you delete the resource it will throw an error.





Now if you go to Assigned Blueprint and choose unassign, After the blueprint getting unassigned one can delete the resources.

