



**#GlobalAzure**  
**#GlobalAzureMilano**

# Implementare strategie di DevOps con le funzionalità built-in di Azure App Service

Massimo Bonanni  
MICROSOFT

# What is Kudu?

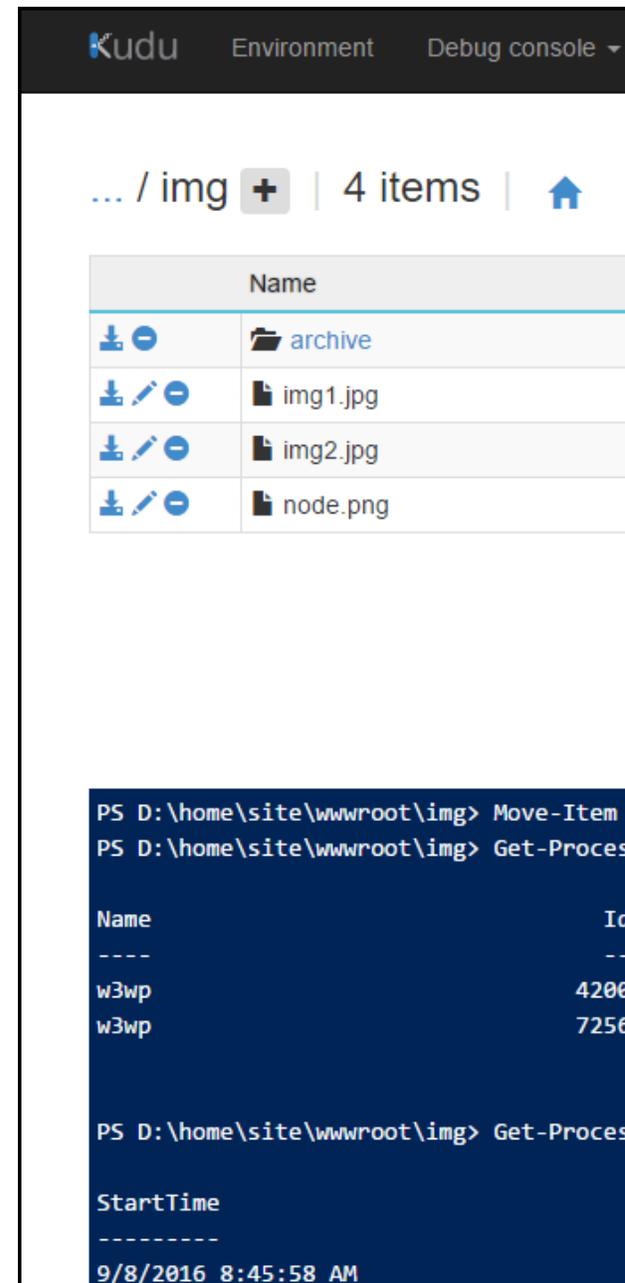
 Kudu is the engine behind some features in Azure App Service that are related to source-control-based deployment and other deployment methods.

 Every time you deploy an App Service, a Kudu instance is created with the `*.scm.azurewebsites.net` FQDN.

 Some Kudu's features:

- Publishing a website from Source Control
- Deployment hooks
- Web Hooks
- Azure Site Extensions
- Post Deployment Action Hooks
- WebJobs

 Kudu is Open-Source project.



The screenshot shows the Kudu web interface. At the top, there is a navigation bar with the Kudu logo, "Environment", and "Debug console". Below the navigation bar, there is a breadcrumb trail: "... / img + | 4 items | Home". A table lists the contents of the /img directory:

Name	
 archive	archive
 img1.jpg	img1.jpg
 img2.jpg	img2.jpg
 node.png	node.png

Below the file explorer, there is a terminal window showing PowerShell commands and output:

```
PS D:\home\site\wwwroot\img> Move-Item
PS D:\home\site\wwwroot\img> Get-Process

Name                               Id
----                               -
w3wp                                4200
w3wp                                7250

PS D:\home\site\wwwroot\img> Get-Process

StartTime
-----
9/8/2016 8:45:58 AM
```

# Kudu architecture

## Security

The Kudu site runs in the same sandbox as the real site

Kudu site cannot do anything that the site itself wouldn't be able to do.

Kudu can access anything the site itself can access to (all app settings, secrets, certificates).

## Auth\*

Kudu site is always authenticated which by default the real site is not authenticated.

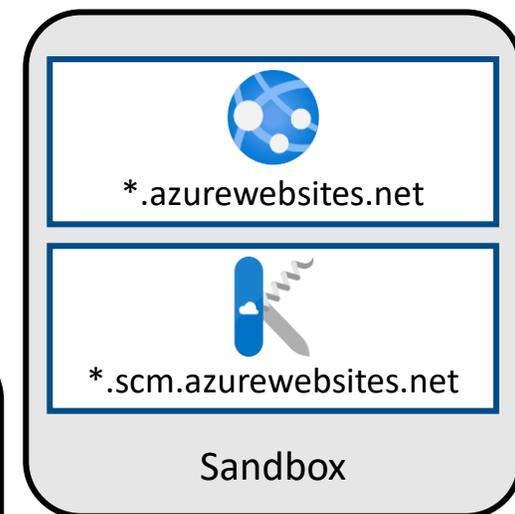
Only users with publishing/deployment permission can access Kudu.

## Process

Kudu service site runs in the same sandbox as the site but in a different process.

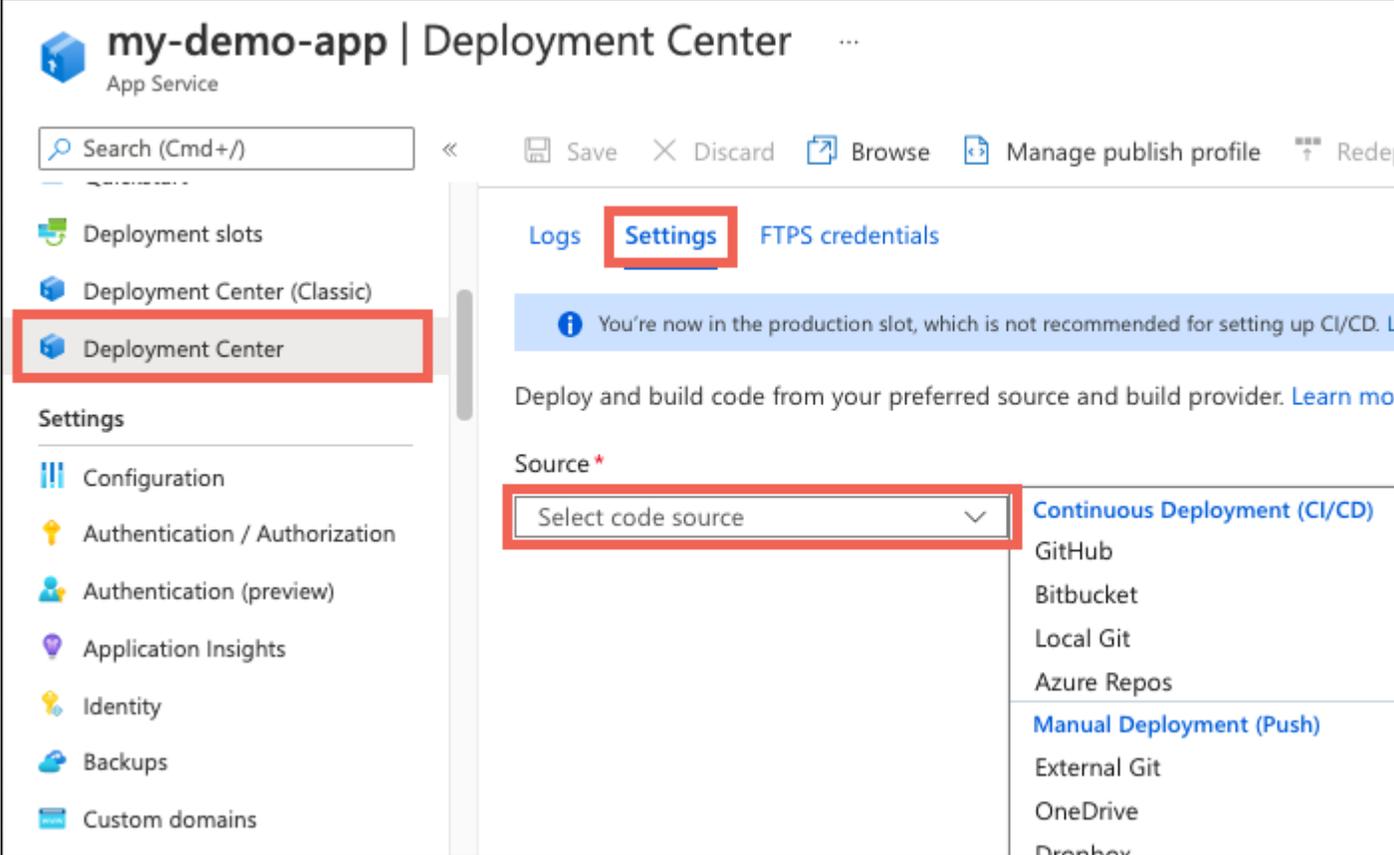
The only thing that really 'connects' the Kudu site to the real site is the file system.

The CPU/RAM/Disk used by the Kudu service is counted toward the site's quota.



# App Service Deployment Center

Azure App Service enables continuous deployment from **GitHub**, **Bitbucket**, and **Azure Repos** repositories by pulling in the latest updates.



The screenshot displays the Azure App Service Deployment Center interface for an application named "my-demo-app". The interface is divided into a left-hand navigation pane and a main content area. In the navigation pane, the "Deployment Center" option is highlighted with a red box. Below it, a "Settings" section is visible, containing various configuration options such as "Configuration", "Authentication / Authorization", "Authentication (preview)", "Application Insights", "Identity", "Backups", and "Custom domains". The main content area shows the "Settings" tab selected, with a red box around the "Settings" label. A warning message states: "You're now in the production slot, which is not recommended for setting up CI/CD." Below this, there is a "Source" dropdown menu, also highlighted with a red box, which is currently set to "Select code source". The dropdown menu is open, showing a list of options: "Continuous Deployment (CI/CD)" (with sub-options: GitHub, Bitbucket, Local Git, Azure Repos) and "Manual Deployment (Push)" (with sub-options: External Git, OneDrive, Dropbox).

# What are the build providers?

## GitHub Actions

It is available only for GitHub deployment.

It creates workflow in the GitHub workflow folder.

## App Service Build Service

It is the deployment and build engine native to App Service (AKA Kudu).

When this option is selected, App Service adds a webhook into the repository you authorized.

Any code push to the repository triggers the webhook, and App Service pulls the changes and performs deployment tasks.

## Azure Pipelines

You can configure a pipeline to build, test, and deploy your app to App Service.

Don't configure it in App Service, but go to Azure DevOps.

# Deployment Credentials

## Publish Profile

- It is a **configuration file** that contains the necessary information and settings for deploying application
- You can download it from the Portal (the option is disabled when Basic Authentication is off)

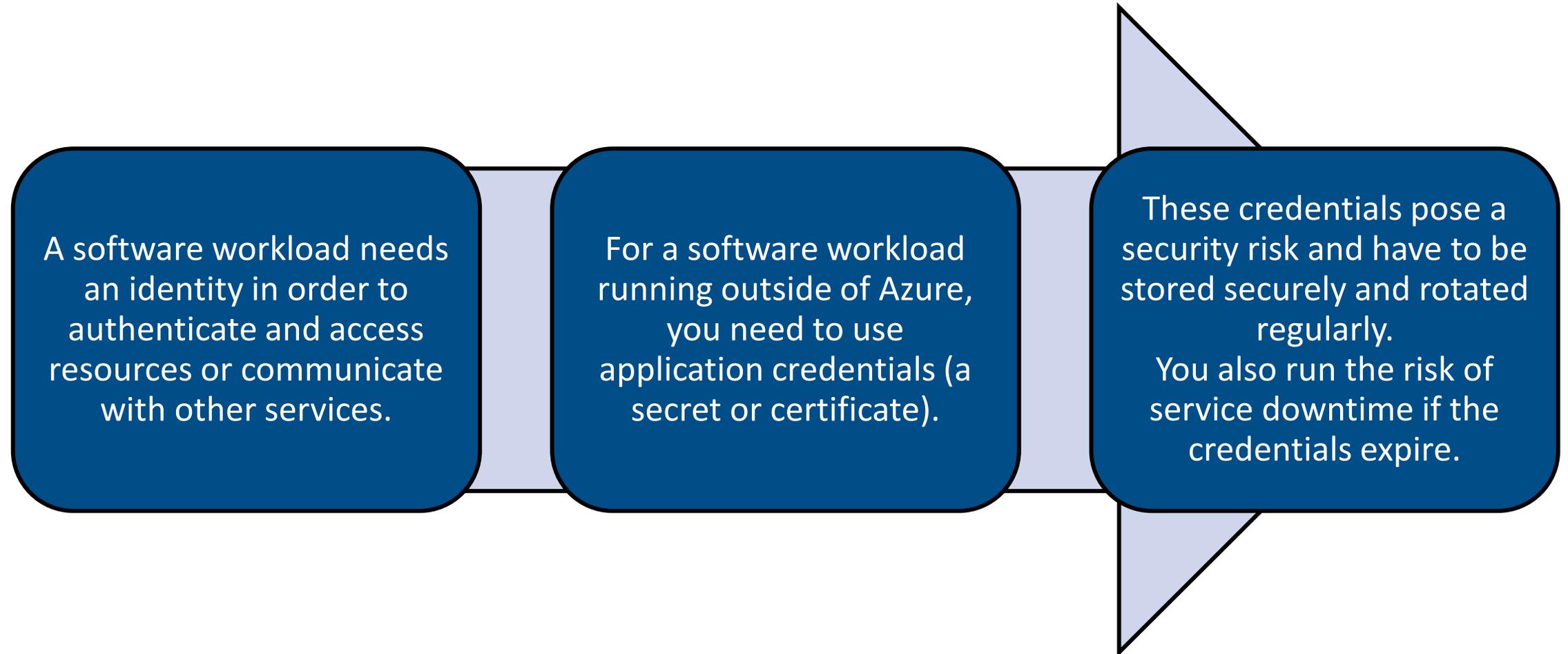
## Basic Authentication

- App Service provides basic authentication for FTP and WebDeploy clients
- It is a simple security method where a **username** and **password** are required to deploy applications
- From Security perspective, it is like a single, static key

## Service Principal

- RBAC for granular control over permissions
- **AppId**, **TenantID** and **Secret** setted as secrets in your pipeline
- Use Workload Identity Federation (GitHub) or Managed Identity (Azure DevOps)

# Before Workload Identity Federation



# ~~Before~~ Workload Identity Federation

You use workload identity federation to configure a user-assigned managed identity or app registration to trust tokens from an external Identity Provider, such as GitHub or Google.

Once that trust relationship is created, your workload exchanges trusted tokens from the external IdP for access tokens from Microsoft identity platform.

You eliminate the maintenance burden of manually managing credentials and eliminates the risk of leaking secrets or having certificates expire.

**It's not magic....it's OpenID Connect!!!**



**#GlobalAzure**  
**#GlobalAzureMilano**

# Demo

From 0 to Hero with Deployment center

# What happens to my app during deployment?



All the officially supported deployment methods make changes to the files in the `/home/site/wwwroot` folder of your app.

These files are used to run your app, so the deployment can fail because of locked files.

The app might also behave unpredictably during deployment because the files aren't all updated at the same time.

# What can I do?



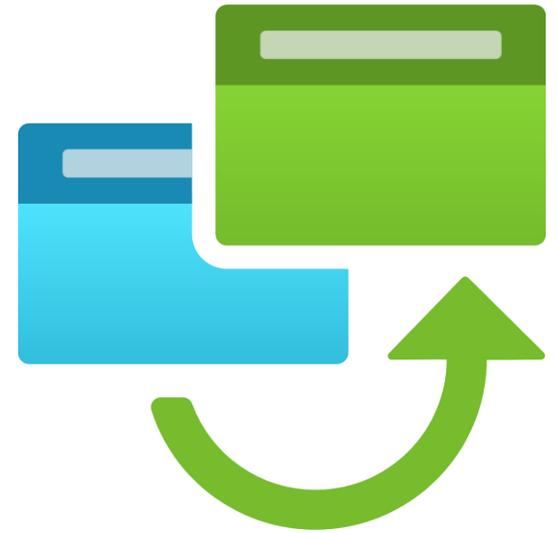
- ☹️ Run your app directly from the ZIP package, without unpacking it.
- ☹️ Stop your app or enable offline mode for it during deployment.
- ✓ Use Slots.

# What is a deployment slots?

An **App Service Slot** is a separate environment within **Azure App Service** that provides a staging area for deploying and testing your web app, mobile back end, or API app before swapping it into production.

Different App Service plan tiers support varying numbers of deployment slots.

There's no extra charge for using deployment slots.



# Slot benefits



You can validate app changes in a staging deployment slot before swapping it with the production slot.



All instances of the slot are warmed up before being swapped into production. This eliminates downtime when you deploy your app. The traffic redirection is seamless, and no requests are dropped because of swap operations.



You can automate this entire workflow by configuring auto swap when pre-swap validation isn't needed.



After a swap, the slot with previously staged app now has the previous production app. If the changes swapped into the production slot aren't as you expect, you can perform the same swap immediately to get your "last known good site" back.

# Slots and Deployment patterns

Create two slots (Blue and Green), deploy app to the Green then swap with the Blue

Blue-green deployments

Canary releases

Use the `x-ms-routing-name` header to specify which slot should respond to the http request.

Dark launching

A/B testing

Use traffic splitting between slots to expose new UI or new features to your users



**#GlobalAzure**  
**#GlobalAzureMilano**

# Demo

SuperHero with slots!



# So what?

## Deployment Center

Easy to use

Start with it and then  
personalize the  
workflow

## Deployment Slots

Zero-Downtime  
Deployments

Free to Use\*

Different DevOps  
patterns

# Any Questions?



Massimo Bonanni  
MICROSOFT



[aka.ms/maxlinkedin](https://aka.ms/maxlinkedin)

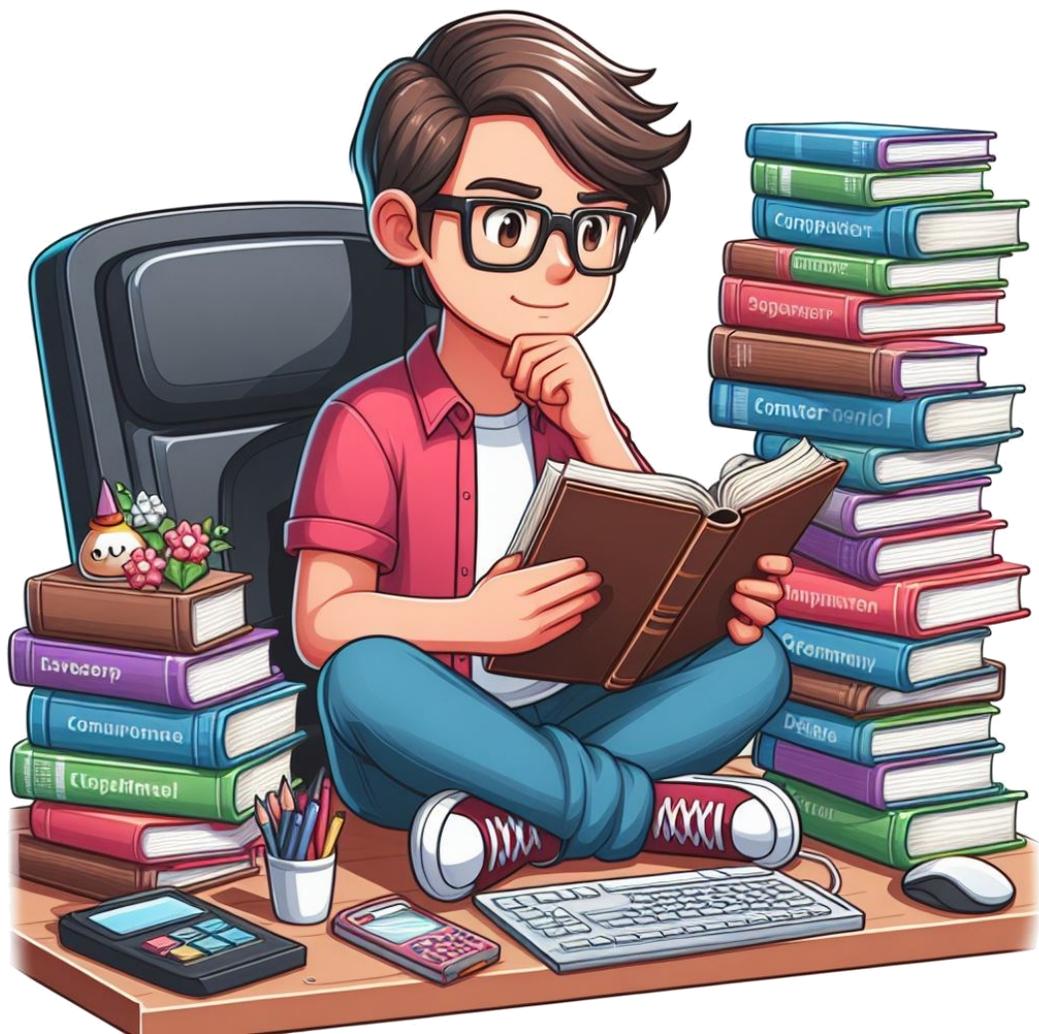


**#GlobalAzure**  
**#GlobalAzureMilano**

# THANK YOU!!!

Slides will be available on Global Azure 2024 page  
on Azure Meetup Milano website

# References



[Kudu service overview](#)



[projectkudu/kudu](#)



[Configure continuous deployment - Azure App Service](#)



[Set up staging environments - Azure App Service](#)