#Coding

Dal caos al deploy : orchestrare il ciclo di vita delle app con una sola CLI

#4DEV

Massimo Bonanni





AGIC

KEY2



BizzyNow™

make yourself better

READY2USE

Max, il trainer sempre in viaggio

Ogni sessione/corso richiede una o piu' demo funzionanti.

Ogni demo richiede codice, infrastruttura, ambienti e deployment.

Non un lungo elenco di comandi.
Un workflow affidabile per preparare il palco/aula.

Quel workflow è azd.



Dal codice al cloud, con un flusso ripetibile

01 Preparare

Codice, infrastruttura e configurazione nello stesso progetto.

Onboarding più semplice.

02 Distribuire

Un workflow coerente per creare le risorse e pubblicare l'app.

Meno passaggi manuali.

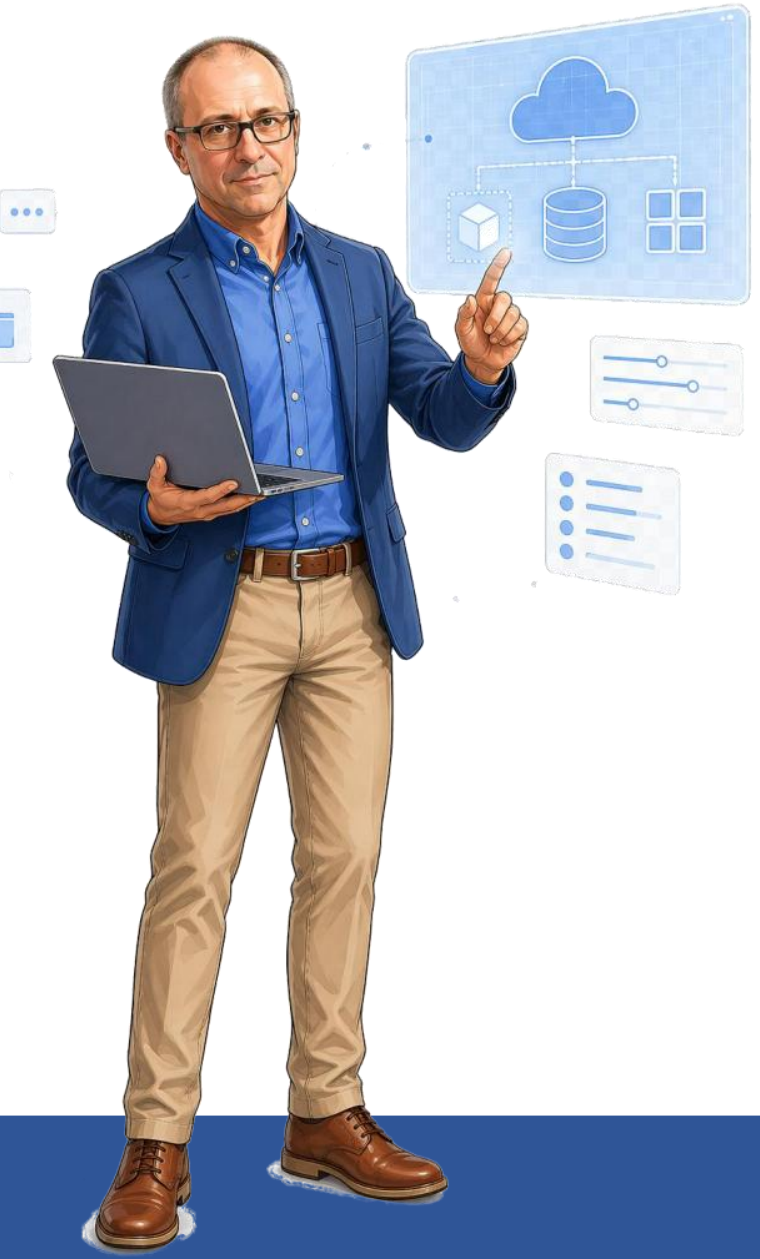
03 Ripetere

Ambienti distinti e gli stessi comandi nelle pipeline.

Standard utili al team.

Non elimina la complessità di Azure: rende ripetibile il modo in cui la gestisci.

Inizializzare un progetto con azd



azd init

Analizza il codice

Scansiona la cartella corrente.

Scegli un template

App e IaC pronte.

Parti dal minimo

Crea lo scheletro del progetto.

Con GitHub Copilot

Setup assistito · Preview

Anatomia di un template azd

`.azure`

Configurazioni e stato degli ambienti.

Normalmente esclusa dal controllo versione.

`infra`

Definizioni infrastructure-as-code.

Bicep o Terraform per creare le risorse Azure.

`src`

Codice sorgente dell'applicazione.

azd prepara e distribuisce i servizi configurati.

`azure.yaml`

Il contratto del progetto.

Collega servizi, percorsi del codice e destinazioni Azure.

Struttura tipica, non obbligatoria: percorsi e generazione IaC dipendono dal progetto.

Ambienti: stessa app, contesti diversi

Che cos'è?

Una configurazione con un nome, legata al deployment dell'app.

Esempi: dev, test, prod.

Che cosa contiene?

Subscription e region
Variabili e impostazioni
Nomi delle risorse

Valori specifici per ambiente.

Perché è utile?

Deployment ripetibili
Sandbox personali
Collaborazione più sicura

Stesso codice, contesti diversi.

Configurazione separata ≠ isolamento di sicurezza: servono IaC, naming, reti e RBAC.

Dove vive la configurazione degli ambienti

```
.azure
  dev
    .env
    config.json
  test
    .env
    config.json
  config.json
```

.env

Variabili specifiche di ogni ambiente.

config.json nell'ambiente

Impostazioni e stato gestiti da azd.

.azure\config.json

Ambiente selezionato per il progetto.

```
azd env new dev      azd env select dev
```

Variabili d'ambiente, senza segreti in chiaro

Una configurazione per ambiente

Coppie chiave-valore in .env.
Backend remoto opzionale, da configurare.

Input → IaC → output

Gli output Bicep diventano variabili azd.
Il mapping verso l'app va configurato.

.azure\sharp4dev\.env

```
AZURE_ENV_NAME="sharp4dev"
AZURE_LOCATION="westeurope"
```

UN VALORE ALLA VOLTA

```
azd env set API_TIMEOUT 5000
azd env get-value API_TIMEOUT
```

Niente segreti in .env o Git: usa Azure Key Vault e, quando possibile, managed identity.

azd up: tre fasi, un solo workflow

Prepara l'app, crea le risorse, distribuisce il codice.

01

Packaging

Prepara gli artefatti

`azd package`

02

Provisioning

Crea o aggiorna le risorse

`azd provision`

03

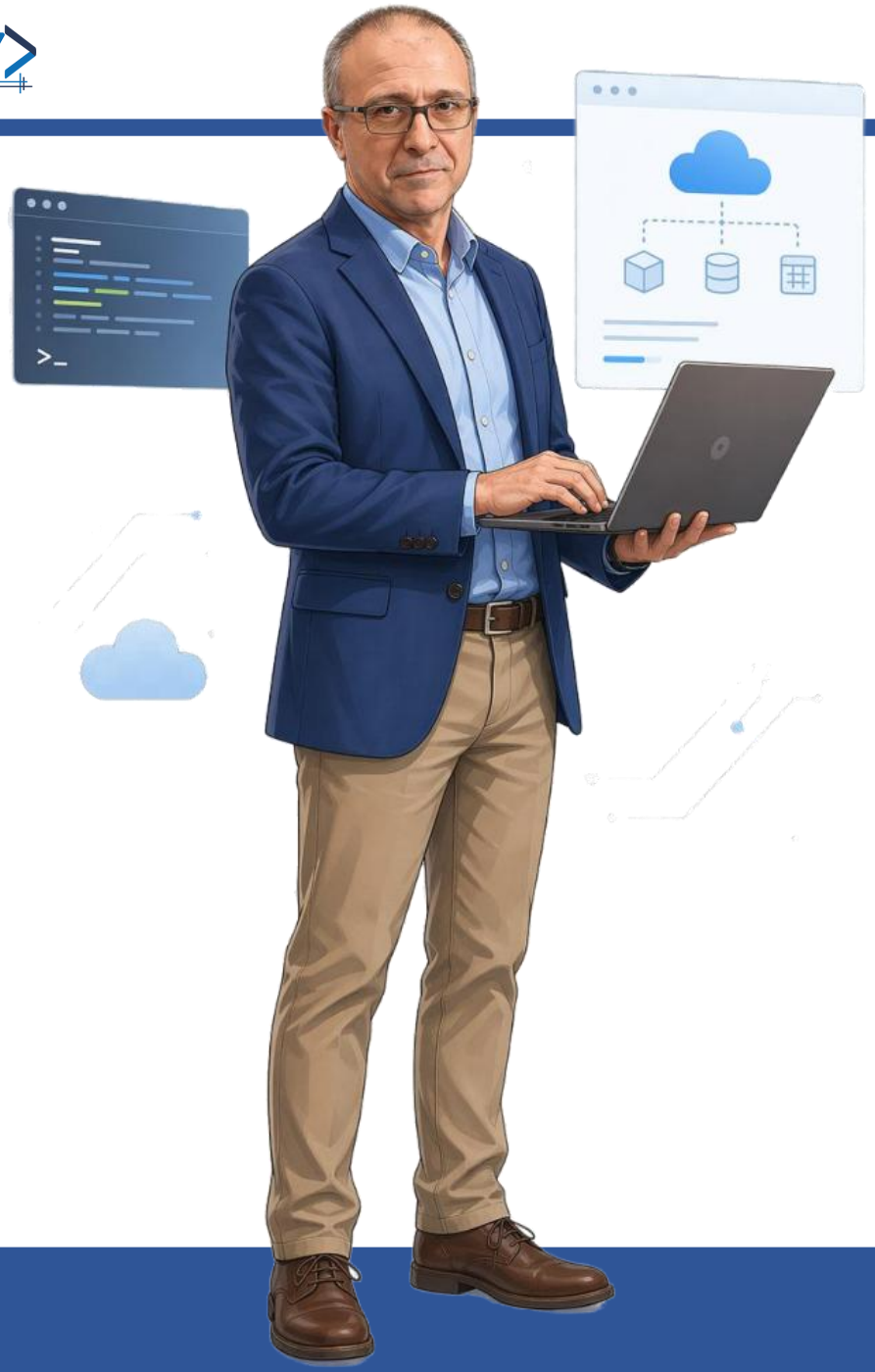
Deployment

Pubblica l'applicazione

`azd deploy`

`azd up`

Il workflow del commando up e' personalizzabile.



Demo

Hook · Personalizzare il ciclo di vita

Script prima o dopo i comandi e gli eventi dei servizi, in `azure.yaml`.

`preup`

`prepackage`	`preprovision`	`predeploy`
Packaging	**Provisioning**	**Deployment**
`postpackage`	`postprovision`	`postdeploy`

`postup`

Bash / PowerShell: sh / pwsh
Anche Python, JavaScript, TypeScript e .NET

Oltre il primo deployment: maturità e limiti

Estensioni · framework GA

Registro ufficiale e fonti personalizzate.
Installabili anche nei Dev Containers.

Aspire · scelta consapevole

Workflow azd esistenti supportati.
Per nuovi deploy: valutare aspire deploy.

AI · template ed estensioni

RAG, agenti e MCP.
azd add: Beta, progetti supportati.

CI/CD · pipeline config (Beta)

GitHub Actions e Azure Pipelines.
OIDC con prerequisiti e limiti.

azd e az: complementari, non concorrenti

Azure Developer CLI · azd

Ciclo di vita dell'applicazione
Template, codice e infrastruttura
Provisioning e deployment ripetibili
Configurazione delle pipeline

«Porta questa applicazione su Azure.»

Azure CLI · az

Controllo granulare delle risorse
Scripting e attività amministrative
Ampia copertura dei servizi Azure
Gestione di tenant e subscription

«Gestisci questa risorsa Azure.»

azd orchestra il workflow. az e l'infrastructure-as-code restano strumenti fondamentali.





Massimo Bonanni

Sr Technical Trainer @ Microsoft

massimo.bonanni@microsoft.com
https://info.massimobonanni.it

