

Infrastructure as Code best practices in the Cloud.

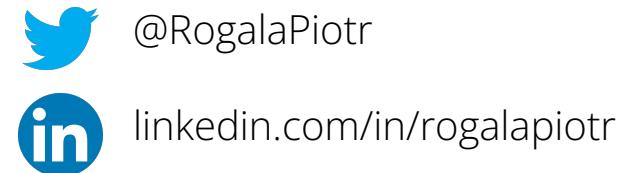
Piotr Rogala



Piotr Rogala

Working in  **Nordcloud**
an IBM Company

as Engineering Practice Lead



Questions?



JustCloud.pl

O mnie

Blog

Meetup's

Kontakt



Kontakt

Jeśli chcesz się ze mną skontaktować to możesz to zrobić za pomocą tego formularza lub poprzez [LinkedIn](#), do zobaczenia!

E-mail

Message

Send message

Agenda.

1. Intro to the cloud
2. IaC and GitOps
3. Perception of IaC
4. Demo
5. Summary



Intro to the cloud.

Cloud Vendors



Azure



Google Cloud

 Alibaba Cloud

The Alibaba Cloud logo, featuring a dark grey stylized bracket shape followed by the text "Alibaba Cloud".

Services comparison

<https://comparecloud.in/>

Public Cloud Services Comparison

About

Follow @iilyas-it83 230

Star 1,287

Fork 971

Issue 23

Discuss

Sponsor

Category	Service	AWS	Azure	Google Cloud	IBM Cloud	ORACLE CLOUD	Alibaba Cloud	HUAWEI CLOUD
Compute	Shared Web hosting	 AWS Amplify	 Web Apps	 Firebase	 Web hosting services		 Web Hosting  Simple Application Server	
Compute	Virtual Server	 Amazon EC2	 Azure Virtual Machine	 Compute Engine	  Classic Virtual Server  Virtual Server for VPC (x86 & s390x)  Power Systems Virtual Servers  Hyper Protect Virtual Server (LinuxONE)  Quantum Services	 Oracle Cloud Infrastructure Compute,	 Alibaba ECS	 Huawei Cloud Elastic Cloud Server
Compute	Bare Metal Server	 Amazon EC2 Bare Metal Instance	 Azure Bare Metal Servers (Large Instance Only for SAP Hana)	 Bare Metal Solution	 Bare Metal Servers	 Oracle Bare Metal Servers	 ECS Bare Metal Instance	 Huawei Cloud Bare Metal Server

Azure

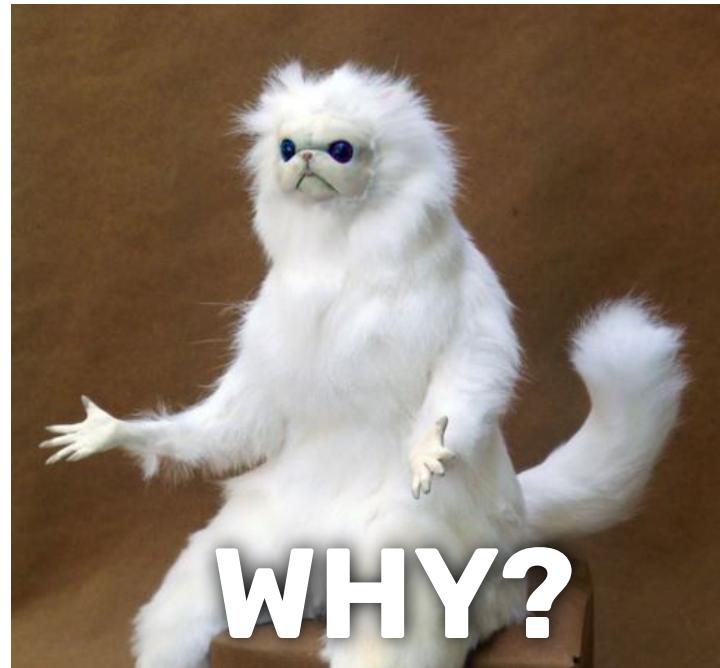
- Public Cloud (Azure)
 - **Poland region** in GA!
- Private Cloud
 - Azure Stack Hub & HCI
- Regions Worldwide: 60+
- Cost: Pay-as-you-go
- Services: Ready to use
 - IaaS / PaaS / SaaS
- Simple / Quick start

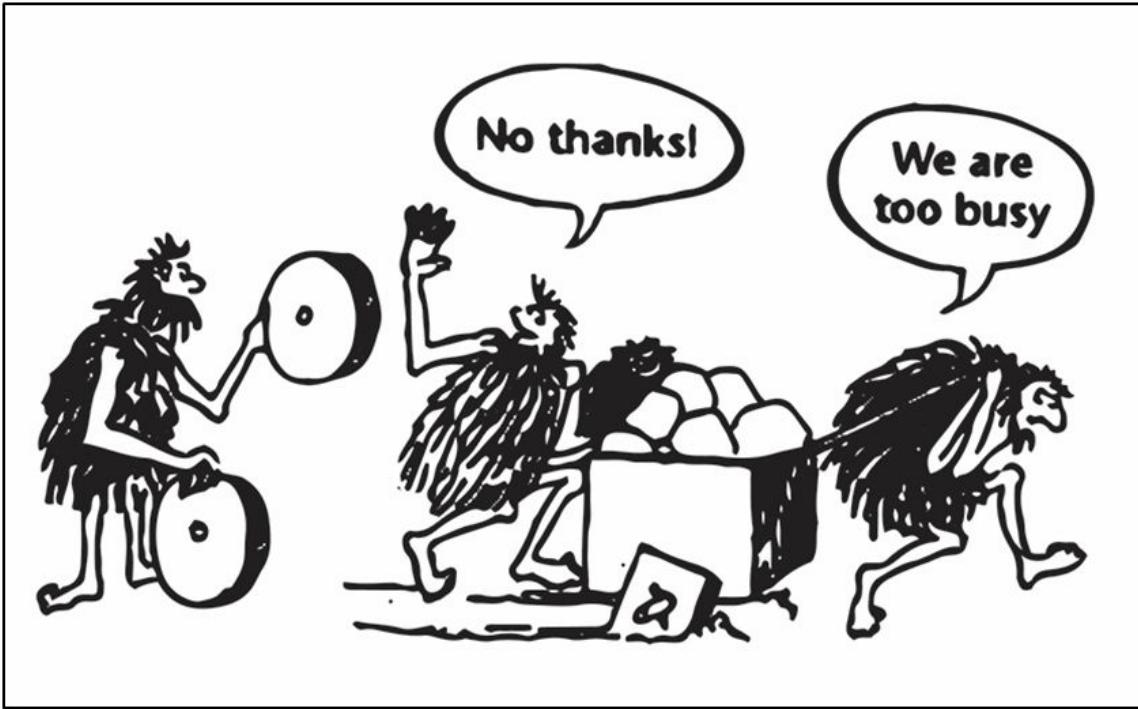


IaC and GitOps.

Why automation is important?

- Speedup delivery
- Scalability
- Consistency and Standardization
 - Simple to repeat same process
- Reduced Human Error
- Cost Optimization
- Agility and Flexibility





What is what?

Infrastructure as Code

- Script for deployment Infrastructure
- Manageable code
- Simple to recreate
- Reusable code

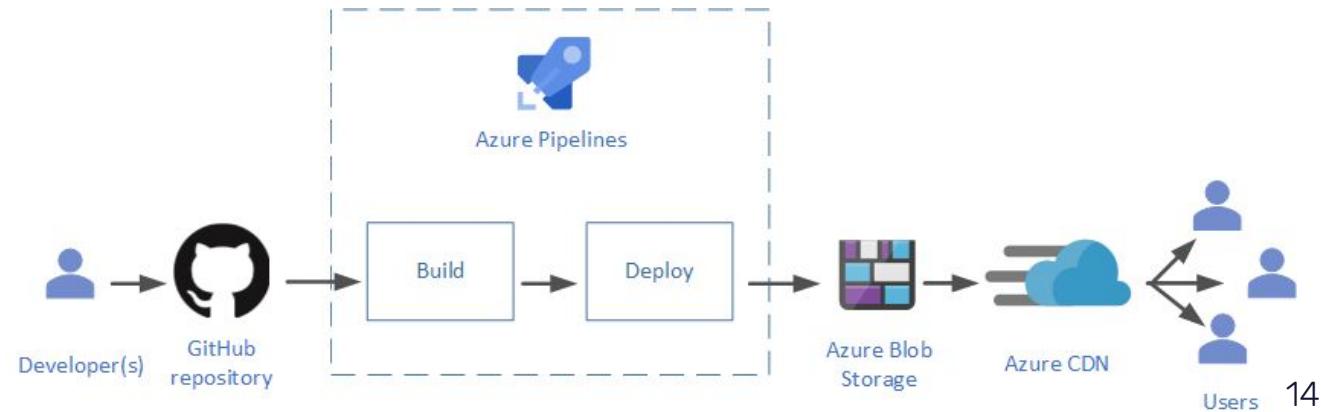
GitOps

- CI/CD process
- Code versioning
- Work on code with team
- Supporting pipelines

Perception of IaC.

Conclusions after implementation

- Everything was supposed to be automatic
- Complicated CI/CD processes
- Time-consuming code to write
- Don't touch if it works - Production env
- Code fixes mess up existing processes



Why automation failing?

1. Project launched from the portal
2. No documentation
3. Customer wants DevOps
4. Customer wants to manage from code

Why?

- Deadlines
- Lack of knowledge
- Not my problem

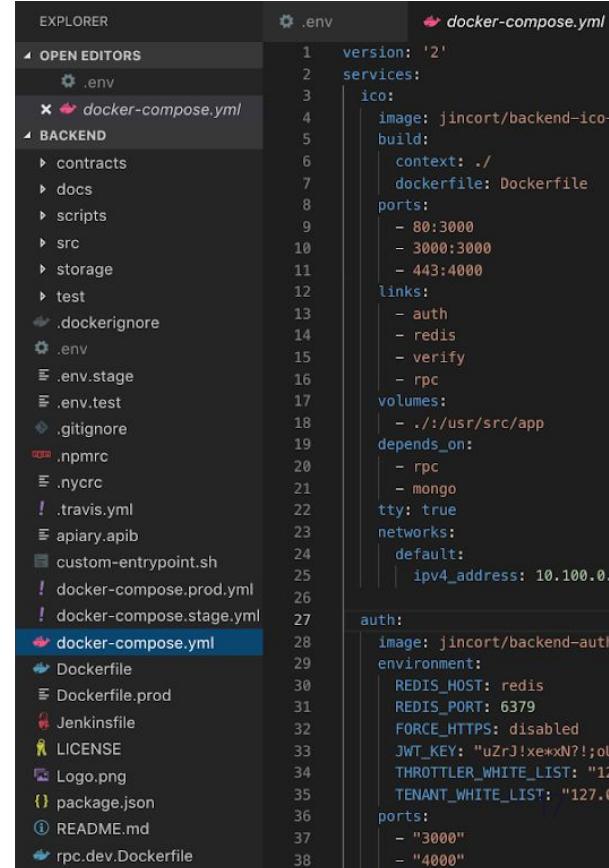


Why it is happening?

- It was supposed to be beautiful and automatic
- Changes to the platform messed something up in the process
- Adding exceptions in automation
- Restart CI/CD process fails
- Lack of standards
- No separation between Infrastructure and Applications
- Lack of code versioning

Infrastructure as Code (IaC) - example

- Infrastructure as Code
 - Templates
 - ARM templates / Bicep
 - Terraform
 - Scripts
 - PowerShell
 - CLI
- Code **must** allow the management of
- Implementation must work idempotently
- Use Git and CI/CD
- Disaster recovery capability
- Documentation = Code



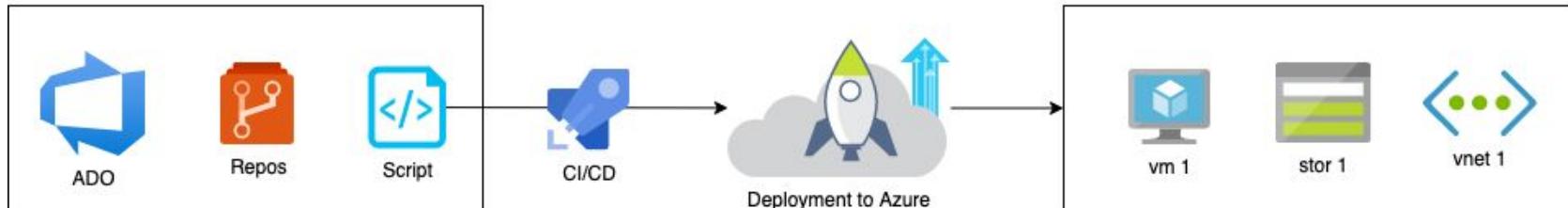
The screenshot shows a code editor interface with the following details:

- EXPLORER** sidebar:
 - OPEN EDITORS: .env, docker-compose.yml
 - BACKEND: contracts, docs, scripts, src, storage, test, .dockerignore, .env, .env.stage, .env.test, .gitignore, .npmrc, .nyrc, .travis.yml, apiary.apib, custom-entrypoint.sh, docker-compose.prod.yml, docker-compose.stage.yml, docker-compose.yml, Dockerfile, Dockerfile.prod, Jenkinsfile, LICENSE, Logo.png, package.json, README.md, rpc.dev.Dockerfile
- .env** file content (highlighted):

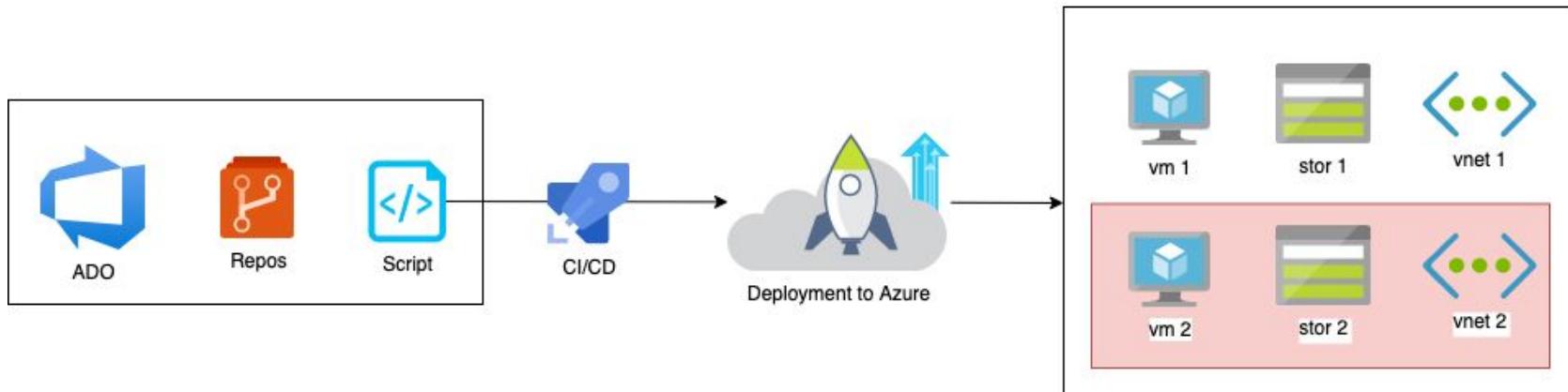
```
version: '2'
services:
  ico:
    image: jincort/backend-ico-build
    context: ./
    dockerfile: Dockerfile
    ports:
      - 80:3000
      - 3000:3000
      - 443:4000
    links:
      - auth
      - redis
      - verify
      - rpc
    volumes:
      - ./:/usr/src/app
    depends_on:
      - rpc
      - mongo
    tty: true
  networks:
    default:
      ipv4_address: 10.100.0.1
  auth:
    image: jincort/backend-auth
    environment:
      REDIS_HOST: redis
      REDIS_PORT: 6379
      FORCE_HTTPS: disabled
      JWT_KEY: "uZrJ!xe*xN?!.;o"
      THROTTLER_WHITE_LIST: "127.0.0.1"
      TENANT_WHITE_LIST: "127.0.0.1"
    ports:
      - "3000"
      - "4000"
```

Too complex - processes

One time deployment - IaC

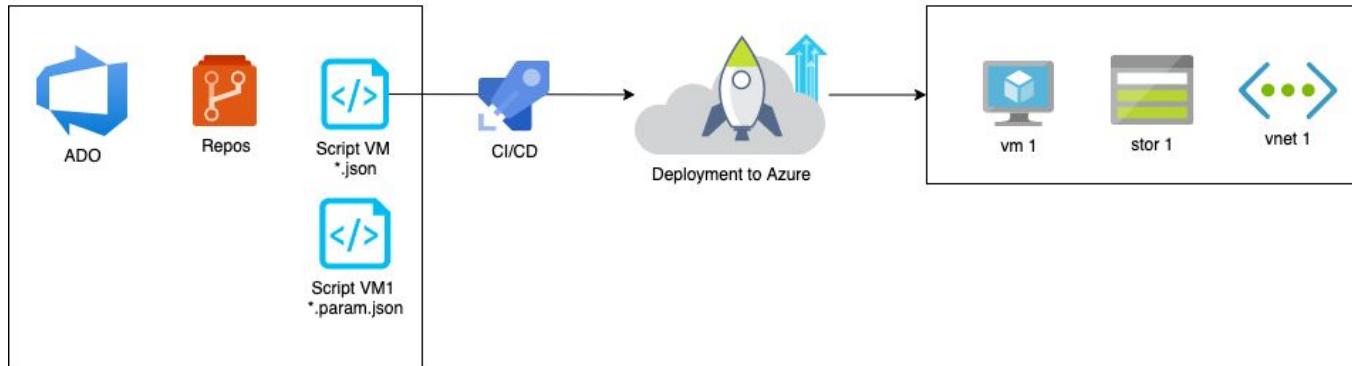


Second shot of "One time deployment - IaC"

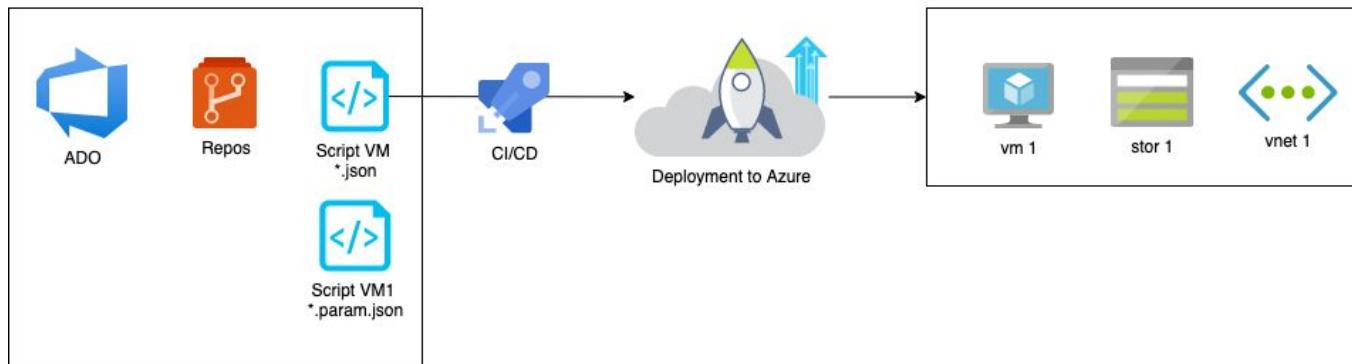


Deployment is non-idempotent

IaC deployment

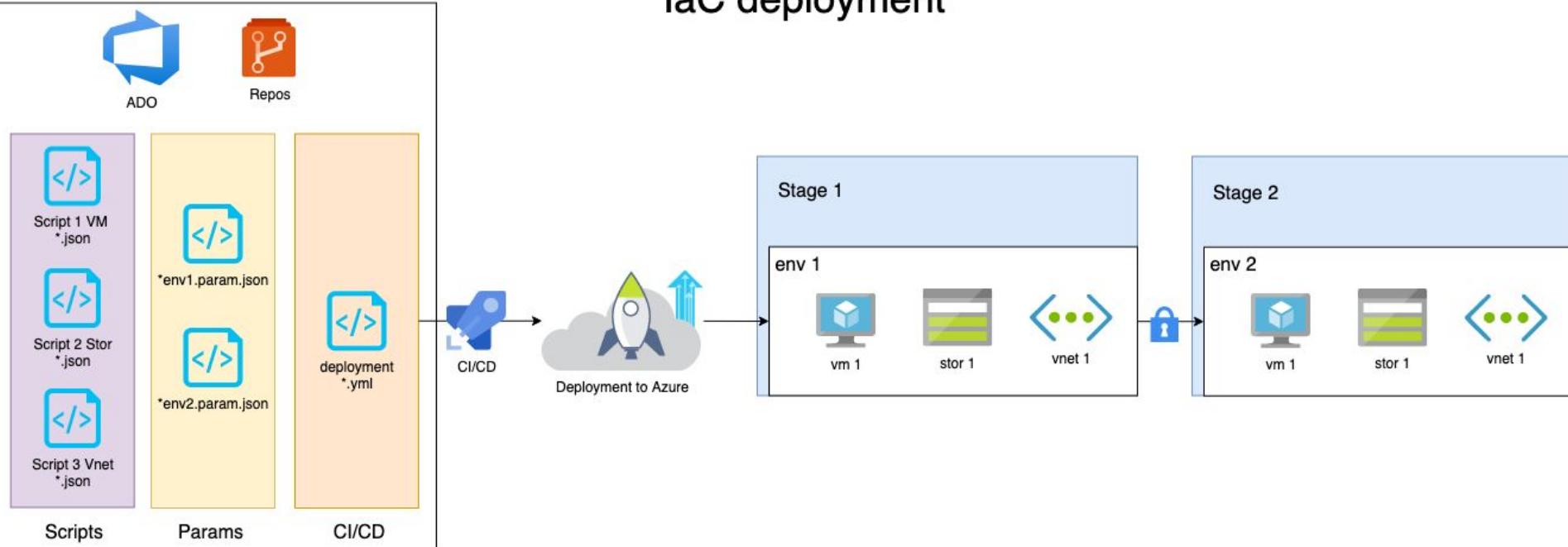


Second shot of "IaC deployment"

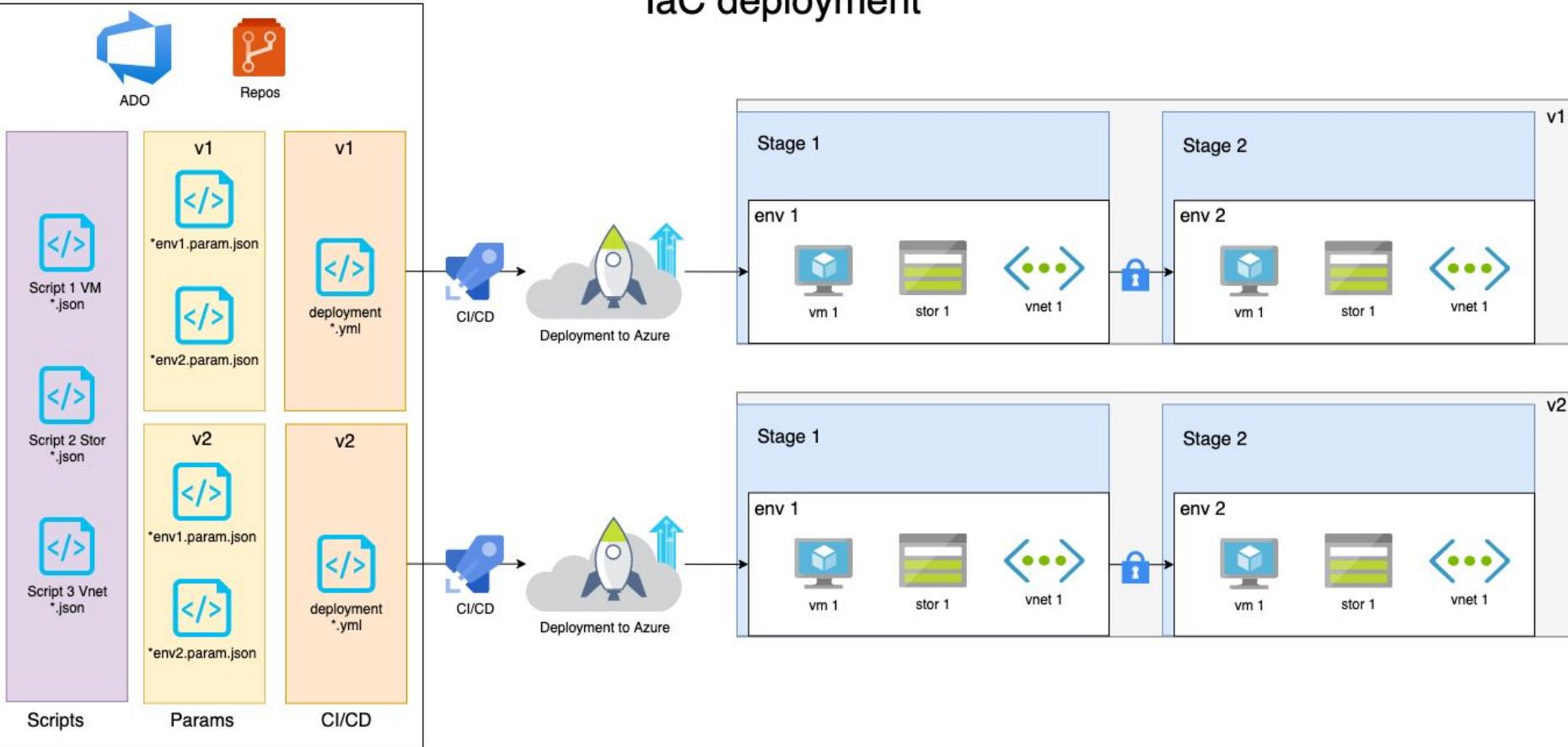


Deployment is idempotent

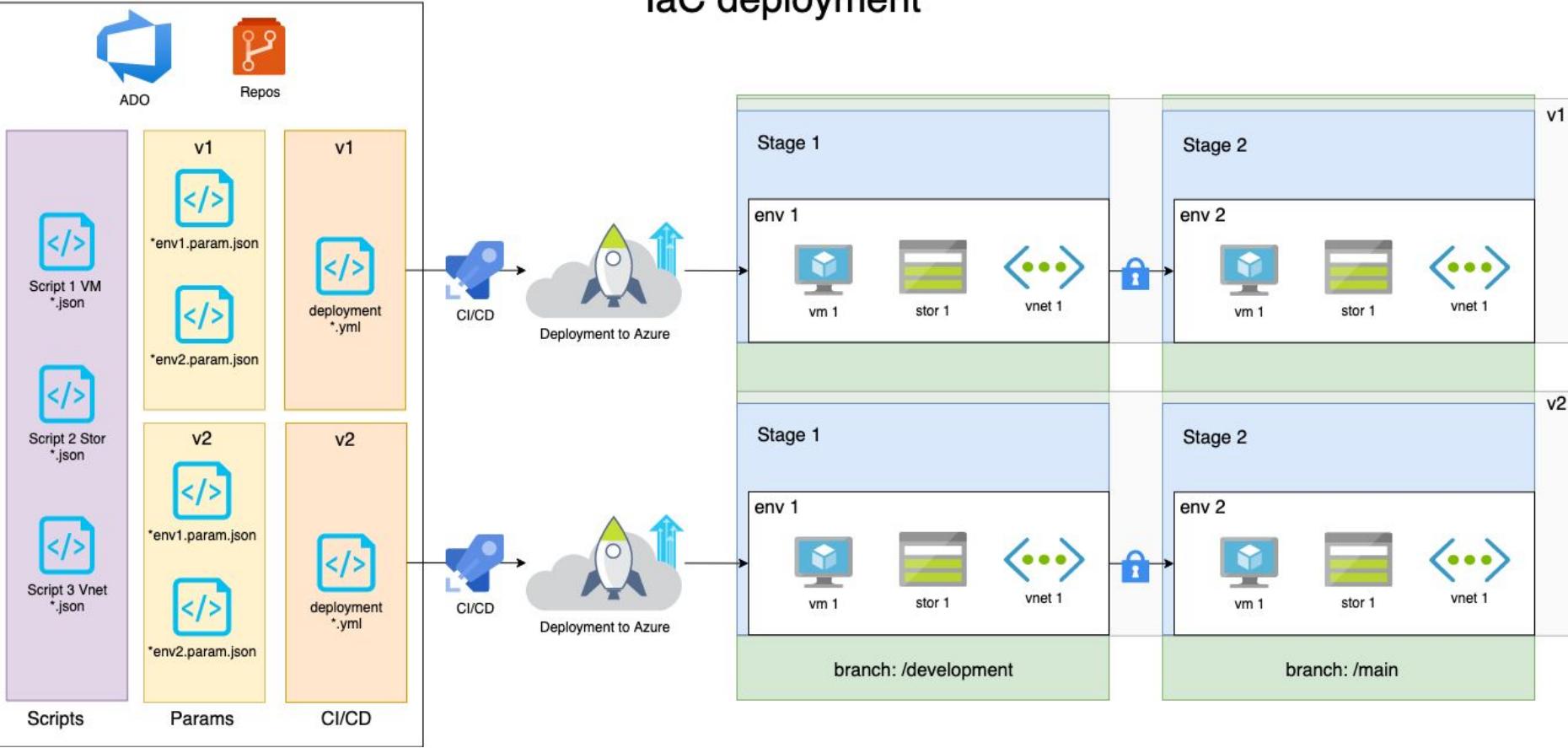
IaC deployment

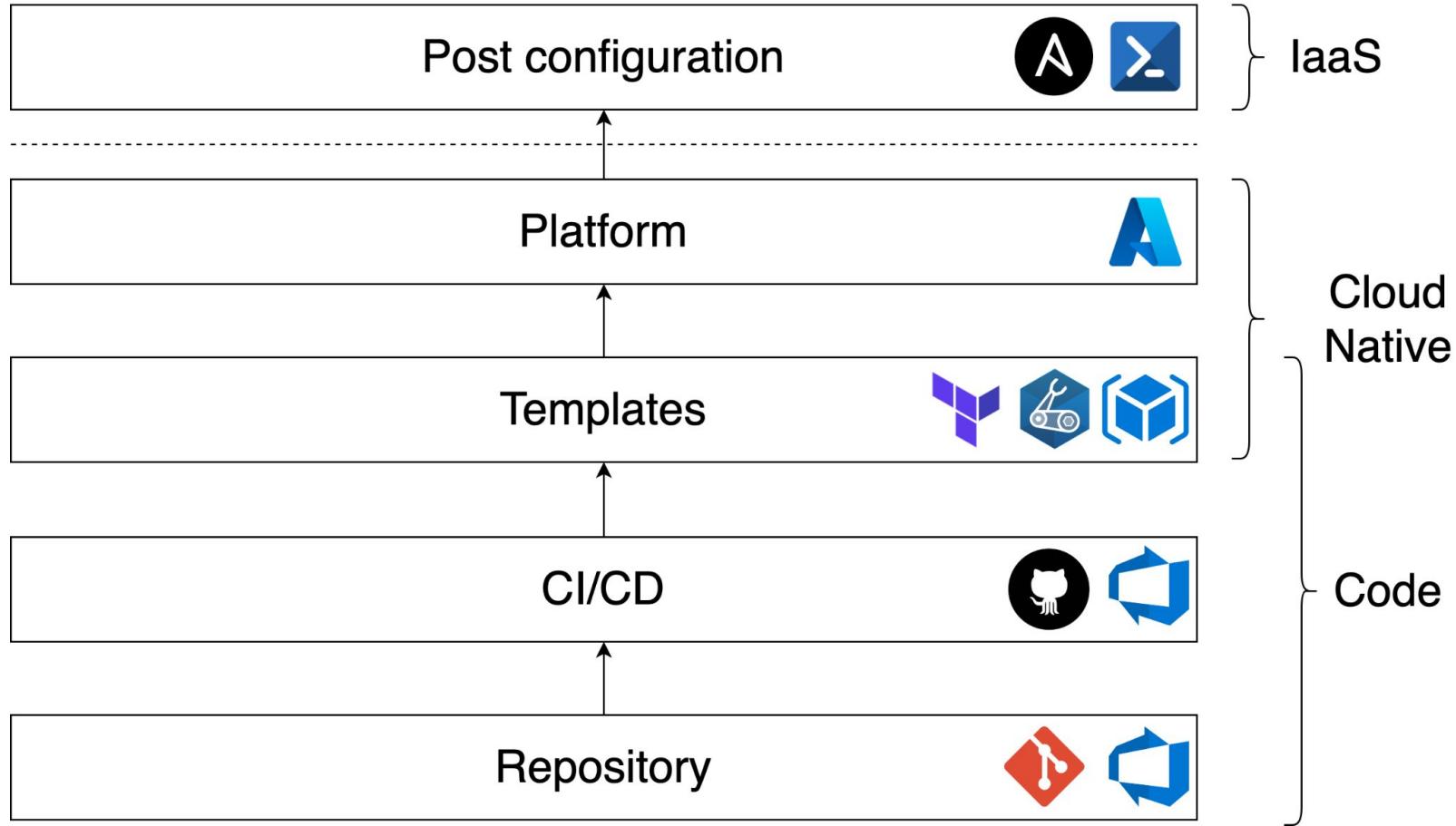


IaC deployment



IaC deployment







Demo

Example deployment in ADO

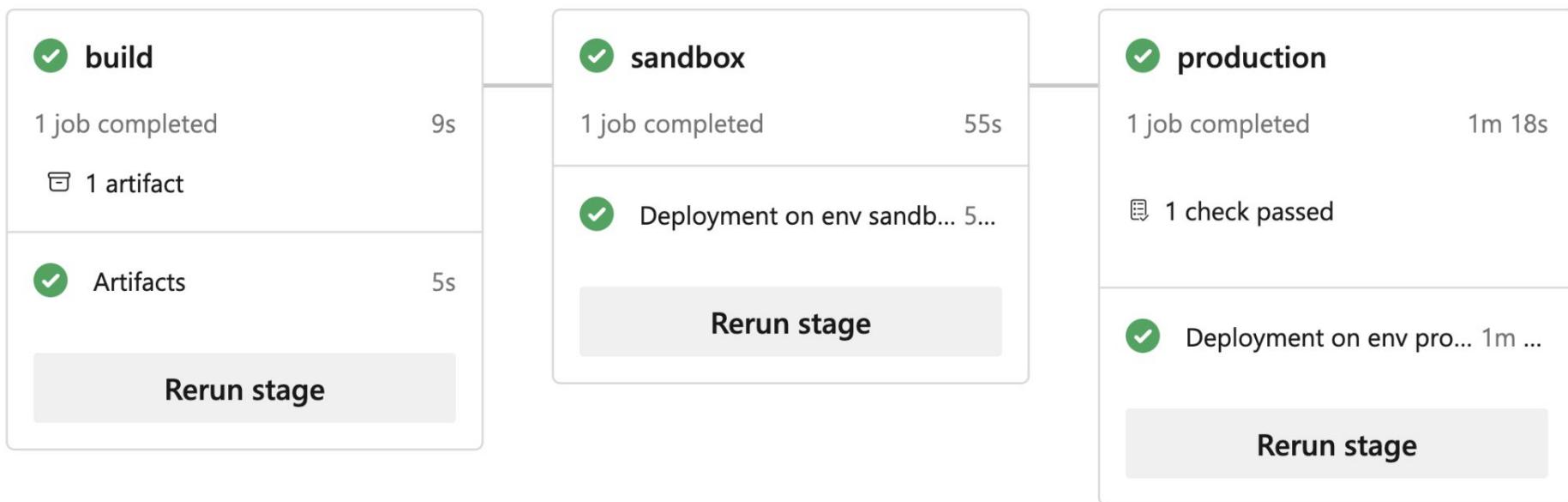
```
project-1
└── ci
    └── build.yml
    └── deployment.yml
    └── project-1.yml
└── scripts
    ├── createresourcegroup-project1.production.param.json
    ├── createresourcegroup-project1.sandbox.param.json
    └── createresourcegroup.json
    ├── storage-diag001.production.param.json
    └── storage-diag001.sandbox.param.json
    └── storage.json
└── README.md
```

```
project-1 > ci > ! project-1.yml > [ ] stages > {} 0 > displayName
You, 7 hours ago | 1 author (You)
  1 name: $(BuildDefinitionName)_${SourceBranchName}_(date:yyyyMMdd)$(rev:.r)
  2
  3 variables:
  4 - name: 'SandboxSubscriptionID'
  5   value: [REDACTED]
  6 - name: 'ProductionSubscriptionID'
  7   value: [REDACTED]
  8 - name: 'serviceConnectionSandbox'
  9   value: 'Sandbox'
 10 - name: 'serviceConnectionProd'
 11   value: 'Sandbox'
 12 - name: location
 13   value: 'westeurope'
 14 - name: agentPool
 15   value: 'Azure Pipelines'
 16 - name: agentImage
 17   value: 'windows-latest'
 18 - name: artifacts
 19   value: 'project-1'
 20
 21 trigger: none
 22 pr: none
 23
 24 stages:
 25
 26 ## Stage: Build
 27 - stage: build
 28   displayName: build You, 9 hours ago • upload changes
 29   jobs:
 30     - template: build.yml
 31       parameters:
 32         agentPool: '$(agentPool)'
 33         agentImage: '$(agentImage)'
 34         artifacts: '$(artifacts)'
 35
 36 ## Stage: Sandbox
 37 - stage: release_sandbox
 38   dependsOn: build
 39   displayName: sandbox
 40   jobs:
 41     - template: deployment.yml
 42       parameters:
```

```
36    ## Stage: Sandbox
37    - stage: release_sandbox
38      dependsOn: build
39      displayName: sandbox
40      jobs:
41        - template: deployment.yml
42        parameters:
43          deploymentStage: 'sandbox' # change to if based on
44          serviceConnection: '$(serviceConnectionSandbox)'
45          subscriptionID: '$(SandboxSubscriptionID)'
46          agentPool: '$(agentPool)'
47          agentImage: '$(agentImage)'
48          location: '$(location)'
49          artifacts: '$(artifacts)'
```

Stages

Jobs



Resource groups

justcloud.pl

[+ Create](#) [Manage view](#) ...

rg-pro

Name ↑↓

rg-project1-prod-001

...

rg-project1-sbx-001

...

rg-project1-sbx-001

Resource group


 Search

[+ Create](#)
[Manage view](#)

[Delete resource group](#)

[Export to CSV](#)
[Open query](#)

Overview

Activity log

Access control (IAM)

Tags

Resource visualizer

Events

Settings

Deployments

Security

Policies

Properties

Locks

Cost Management

^ Essentials

Subscription (move)

[Visual Studio Enterprise Subscription \(MVP Sandbox\)](#)

Deployments

[1 Succeeded](#)

Subscription ID

Location

West Europe

Tags ([edit](#))

Demo : Code Europe 2023

Resources

Recommendations

Filter for any field...

Type equals all [X](#)Location equals all [X](#)
[+ Add filter](#)
Showing 1 to 1 of 1 records. Show hidden types [?](#)
 No grouping

 Name ↑↓

Type ↑↓

Location ↑↓

 stdiagsbx001westeu2001

Storage account

West Europe

Summary.

Education vs. communication

- Sharing solutions within the team
- Using best practice
- Developing projects through small improvements
- Creating solutions that can be reusable in new implementations
- Building internal know-how



Tricky points...

- Create exceptions to the rule
- Inability to exclude an element
- Hard coded values
- No ability to deploy solutions in different places
- Scripts / Templates / Solutions as all-in-one
- Parameter files should define environment
- Rewriting solutions
- Separation of Infrastructure from Applications
(it depends)
- IaC and GitOps





Q&A

Thank you!

Made by Piotr Rogala.
piotr.rogala@nordcloud.com