DIRECTIONS 4 PARTNERS

Creating and maintaining Azure infrastructure professionally

Tobias Fenster, 4PS Germany

www.directions4partners.com

SPEAKER INTRO

TOBIAS FENSTER

- Managing Partner at 4PS Germany
  - Part of 4PS
  - Nordics partner: Fellowmind
- Microsoft Regional Director and MVP for Azure and Business Central
- tobiasfenster on Twitter and LinkedIn
- Blog URL tobiasfenster.io

Agenda

PART 01

- Introduction to Infrastructure as Code (IaC)

PART 02

- ARM templates

PART 03

- Bicep

PART 04

- Terraform

Agenda

PART 01
- Introduction to Infrastructure as Code (IaC)

PART 02
- ARM templates

PART 03
- Bicep

PART 04
- Terraform

What is IaC and why should you care?

- Infrastructure is growing more complex → Handling it manually takes too long and is too error prone but at the same time, larger teams are working on infrastructure
- Development and operations (the teams creating infrastructure for software created by development) need to work together more closely → DevOps! But how to do that?
- How the infrastructure changes over time must be tracked and you might have to go back
  "What did I do five weeks ago to make this work?"
- New infrastructure needs to be created quickly; maintenance must be done without interruption while at the same time needing less human interaction

Sounds like a familiar problem…

- Basically: Growing complexity, working in teams, tracking changes → Adopt best practices of development for infrastructure as well

- Conclusion: Treat your infrastructure as code! → IaC

-
  - Source control management including version history
  - Changes are done through pull request with reviews
  - Automated builds and automated tests

- My preferred flavor: Declarative IaC
  - Describe what you want, not how you get there
  - PowerShell (or other) describe how you set it up, not the result

What does it look like?


```
I need a VM with
    4 CPU cores and 8 GB of memory
    2 disks with 500 GB storage each
    Windows Server 2022 Core installed
    Containers enabled and Docker installed

I need a firewall setup with
    open ports 443 and 22

The VM must be
    secured by the firewall
    reachable from the internet
```

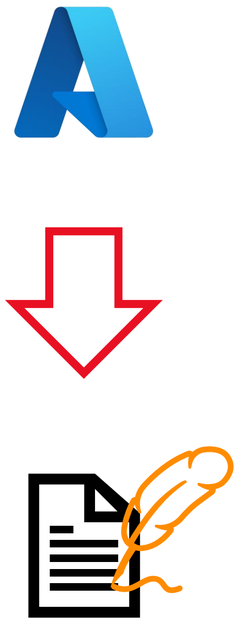
General approach

Create from scratch

- Start with an “empty sheet”
- Add your resources manually
- Deploy and validate whether you have modelled the right things in the right way

Import existing

- Create your infrastructure manually through the Azure Portal, validate that it works
- Import the result (full or components) into the IaC tool
- Possibly extend, abstract, modify…



Agenda

PART 01
- Introduction to Infrastructure as Code (IaC)

PART 02
- ARM templates

PART 03
- Bicep

PART 04
- Terraform

ARM templates

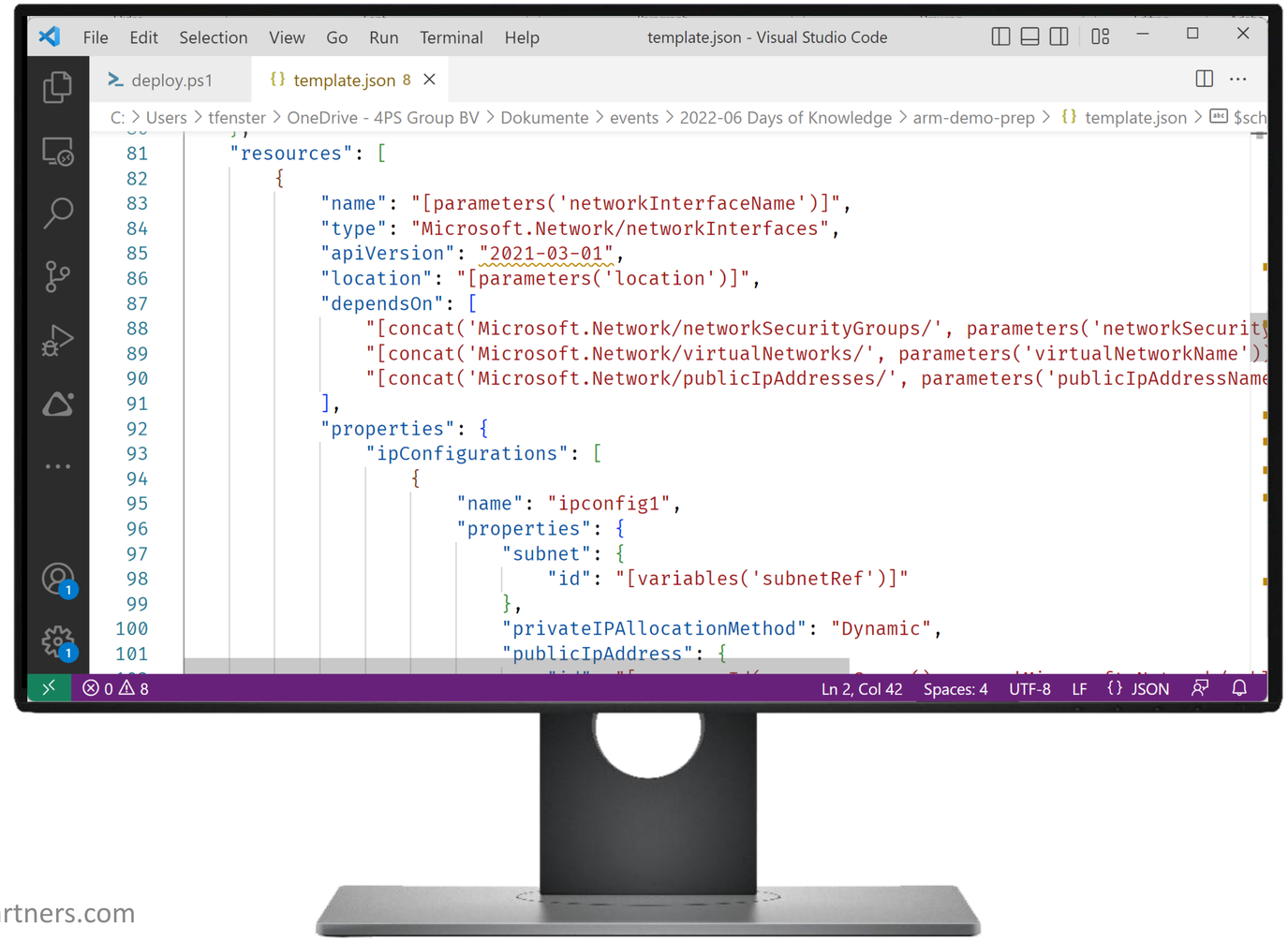
- IaC tool native to Azure: Might be a limitation, but also ensures quick and comprehensive support for Azure services
- Very verbose JSON syntax, which makes creating and maintaining a challenge
- Good support in VS Code through an extension
- Repeatable deployments with auto-managed orchestration
- Support for modules and “what if” deployments
- Vast example library (“Azure quickstart templates”)
- Good documentation: https://docs.microsoft.com/en-us/azure/azure-resource-manager/templates/overview

What does it look like?

ARM templates

Demo flow

- Create VM in Portal
- Download template
- Deploy template



Agenda

PART 01

- Introduction to Infrastructure as Code (IaC)

PART 02

- ARM templates

PART 03

- Bicep

PART 04

- Terraform

Bicep

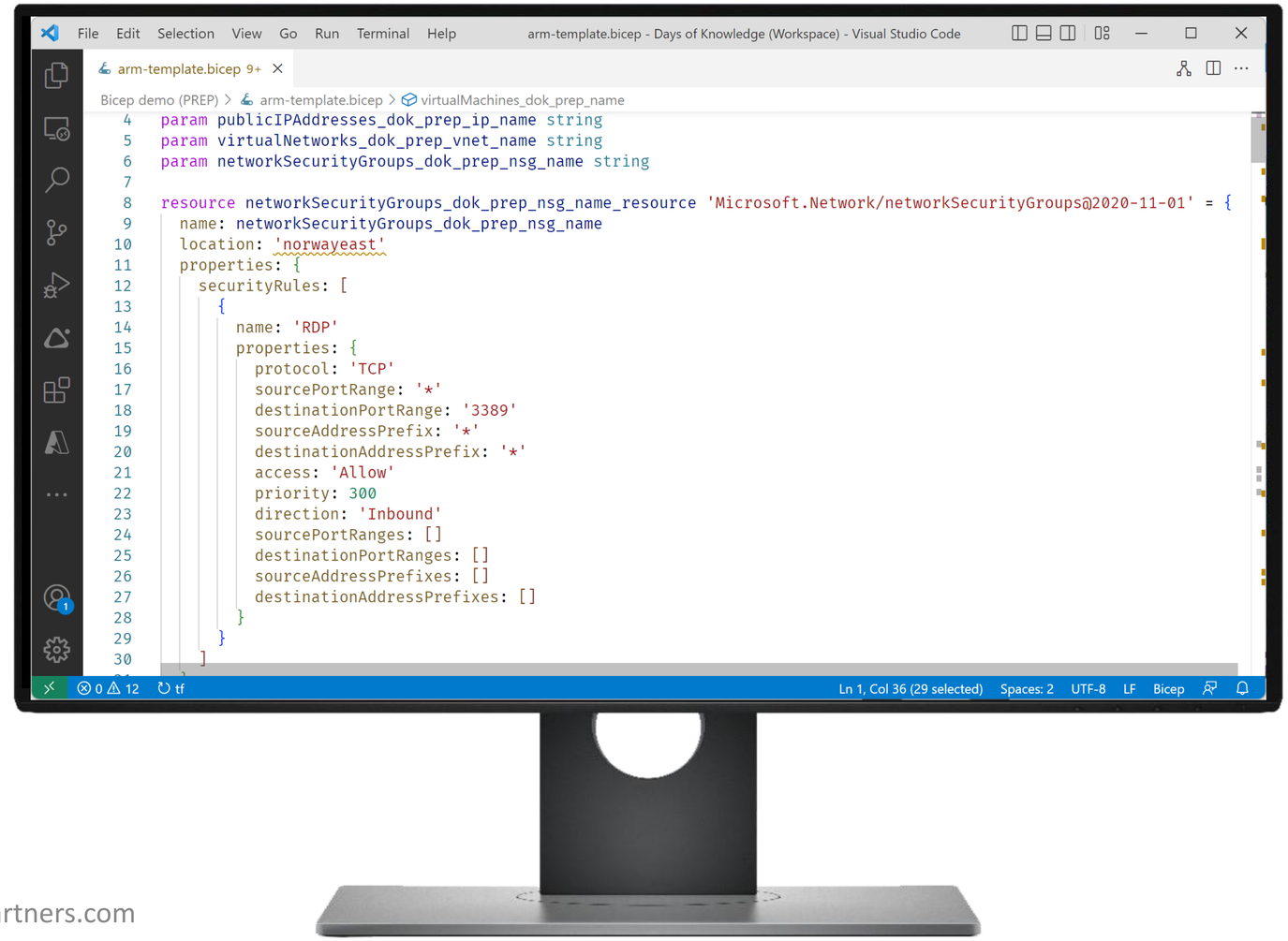
- IaC tool native to Azure: Might be a limitation, but also ensures quick and comprehensive support for Azure services
- Simple syntax (JSON-like but shorter)
- Good support in VS Code through an extension
- Repeatable deployments with auto-managed orchestration
- Support for modules and “what if” deployments
- Open source
- Very good documentation: https://docs.microsoft.com/en-us/azure/azure-resource-manager/bicep/overview

What does it look like?

Bicep

Demo flow

- Export existing resources:
  - Single
  - Full resource group
- Deploy



Agenda

PART 01
- Introduction to Infrastructure as Code (IaC)

PART 02
- ARM templates

PART 03
- Bicep

PART 04
- Terraform

Terraform

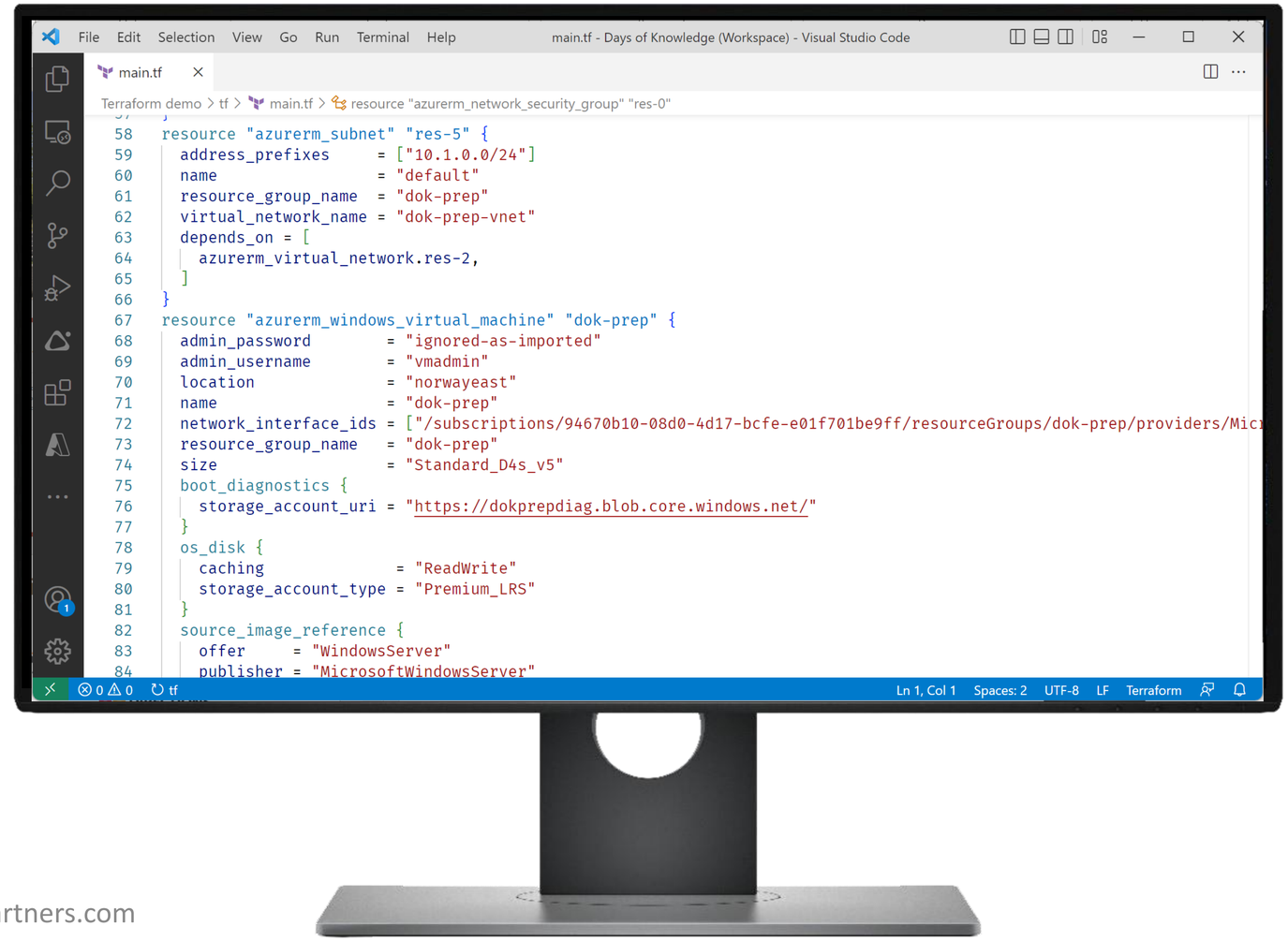
- Multi-cloud IaC tool: Especially handy if you also target other cloud vendors
- Own language HCL, can be a bit of a learning curve but ideally suited for the task
- Good support in VS Code through an extension
- Separate state file
- Support for modules and deployment preview (“plan”)
- Very active community
- Good documentation: https://www.terraform.io/intro

What does it look like?

Terraform

Demo flow

- Export resource group
- Deploy



Which questions can I answer?

Creating and maintaining Azure infrastructure professionally

Tobias Fenster
Managing Partner, 4PS Germany

Followwind

DIRECTIONS 4 PARTNERS

www.directions4partners.com