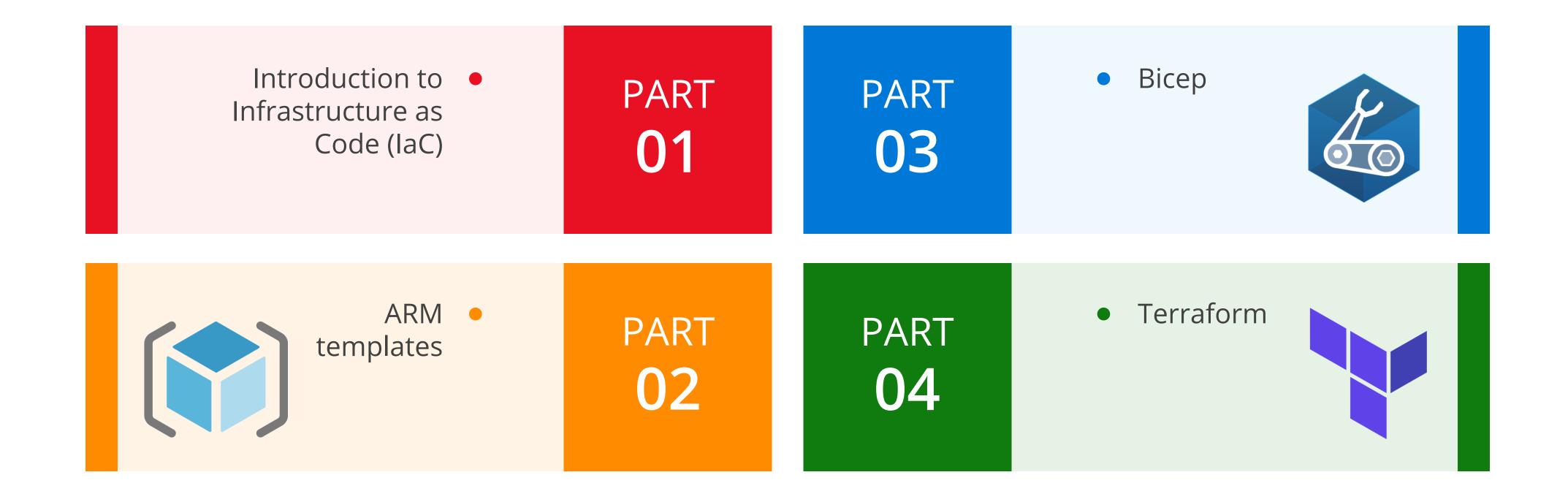


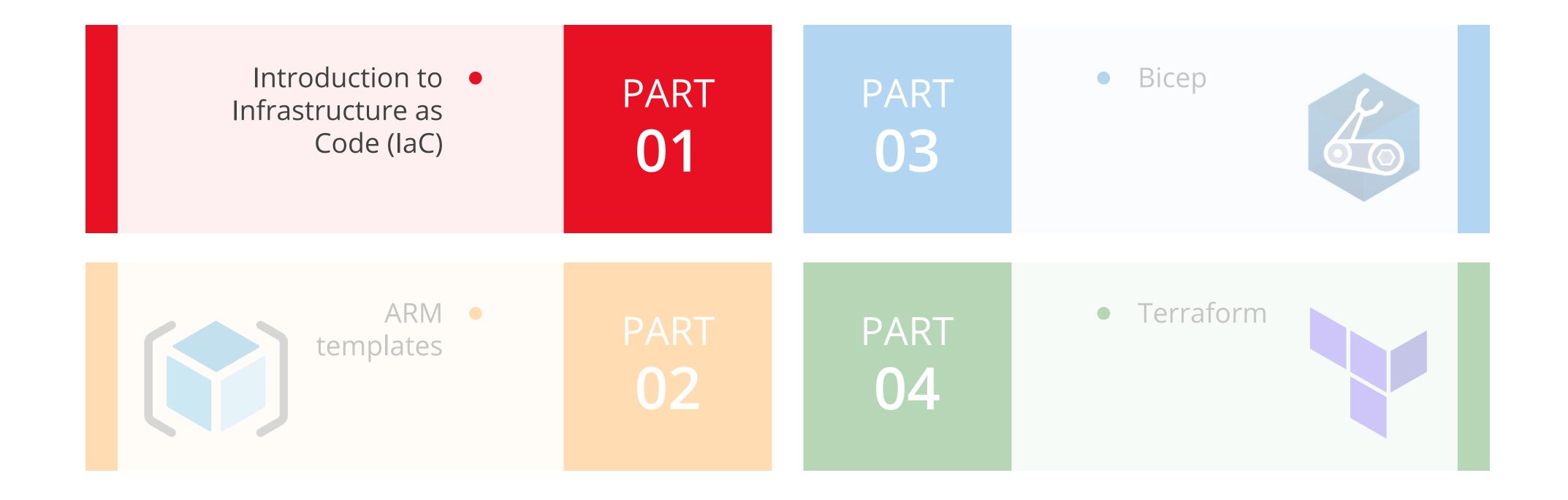
SPEAKER INTRO

TOBIAS FENSTER

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







What is IaC and why should you care?

Intro to IaC

- 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
- 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?"

- Development and operations (the teams creating infrastructure for software created by development) need to work together more closely → DevOps! But how to do that?
- 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...

Intro to IaC

 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

```
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
```



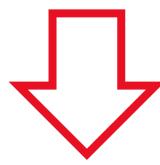
Create from scratch

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

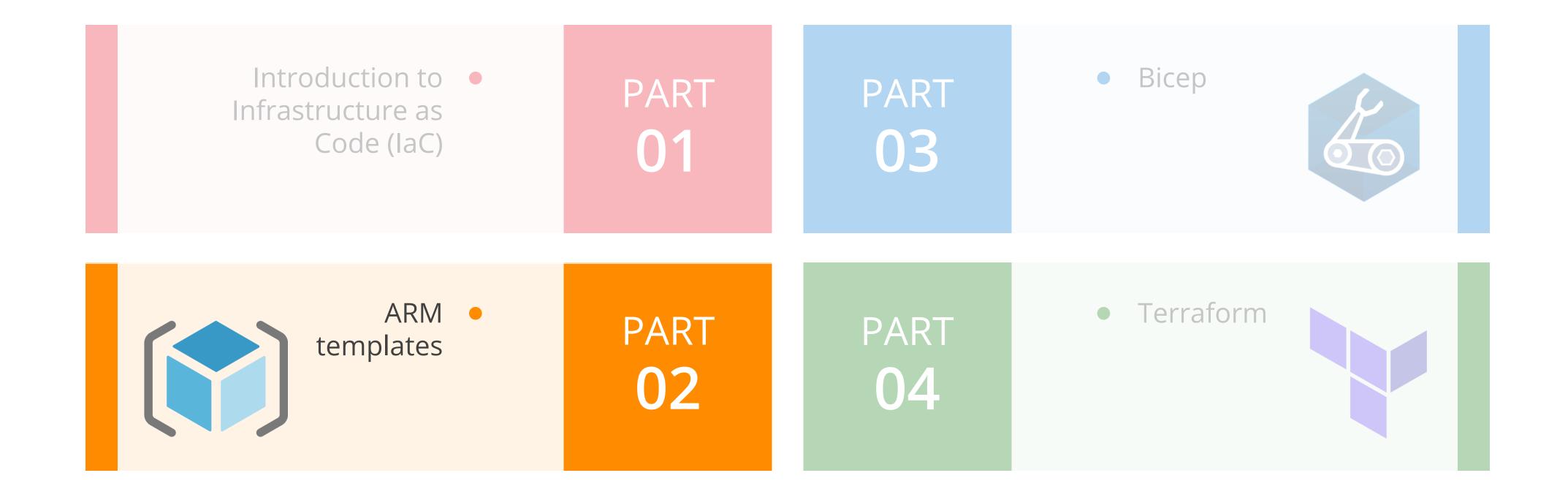
Import existing

- 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...











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

```
File Edit Selection View Go Run Terminal Help
                                                        template.json - Visual Studio Code
                                                                                                            deploy.ps1
                     {} template.json 8 ×
      C: > Users > tfenster > OneDrive - 4PS Group BV > Dokumente > events > 2022-06 Days of Knowledge > arm-demo-prep > {} template.json > 🖭 $sch
                  "resources": [
        81
        82
                          "name": "[parameters('networkInterfaceName')]",
        83
                          "type": "Microsoft.Network/networkInterfaces",
        84
                          "apiVersion": "2021-03-01",
        85
                          "location": "[parameters('location')]",
        86
                          "depends0n":
                              "[concat('Microsoft.Network/networkSecurityGroups/', parameters('networkSecurity
                              "[concat('Microsoft.Network/virtualNetworks/', parameters('virtualNetworkName')
                              "[concat('Microsoft.Network/publicIpAddresses/', parameters('publicIpAddressName
        91
                          "properties": {
        92
                              "ipConfigurations": [
        93
        94
                                       "name": "ipconfig1",
        95
                                       "properties":
                                           "subnet":
        97
                                               "id": "[variables('subnetRef')]"
        98
                                           "privateIPAllocationMethod": "Dynamic",
      100
                                           "publicIpAddress":
      101
   8 1 0 ⊗
                                                                        Ln 2, Col 42 Spaces: 4 UTF-8 LF {} JSON 🔊 🚨
```



Bicep

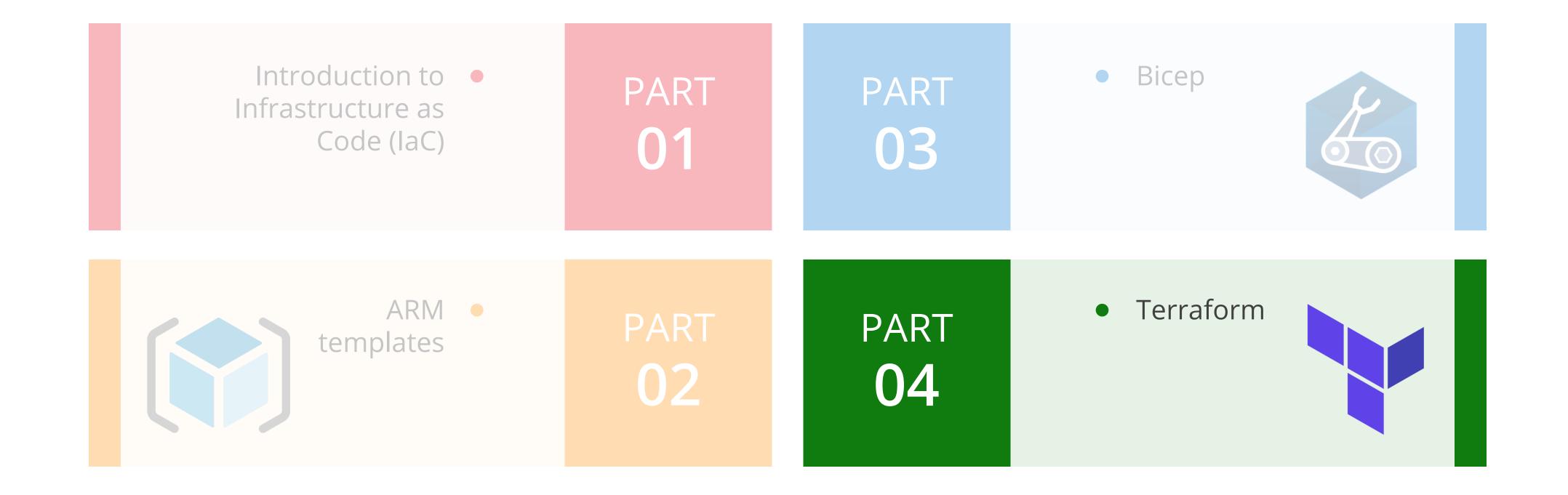
- IaC tool native to Azure: Might be a limitation, but also ensures quick and comprehensive support for Azure services
- Simple syntax (JSON-like but shorter)
- Good support in VS Code through an extension
- Repeatable deployments with auto-managed orchestration
- Support for modules and "what if" deployments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



Demo flow

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

```
File Edit Selection View Go Run Terminal Help
                                                     arm-template.bicep - Days of Knowledge (Workspace) - Visual Studio Code
                                                                                                                           况 □ …
      ♣ arm-template.bicep 9+ X
      Bicep demo (PREP) > ♣ arm-template.bicep > ❤ virtualMachines_dok_prep_name
            param publicIPAddresses_dok_prep_ip_name string
            param virtualNetworks_dok_prep_vnet_name string
            param networkSecurityGroups_dok_prep_nsg_name string
             resource networkSecurityGroups_dok_prep_nsg_name_resource 'Microsoft.Network/networkSecurityGroups@2020-11-01' = {
              name: networkSecurityGroups_dok_prep_nsg_name
               location: 'norwayeast'
       10
              properties: {
       11
       12
                securityRules: [
       13
                     name: 'RDP'
       14
       15
                     properties: {
       16
                       protocol: 'TCP'
                       sourcePortRange: '*'
       17
       18
                       destinationPortRange: '3389'
       19
                       sourceAddressPrefix: '*'
       20
                       destinationAddressPrefix: '*'
       21
                       access: 'Allow'
       22
                       priority: 300
       23
                       direction: 'Inbound'
       24
                       sourcePortRanges: []
       25
                       destinationPortRanges: []
       26
                       sourceAddressPrefixes: []
       27
                       destinationAddressPrefixes: []
       28
       29
                                                                                      Ln 1, Col 36 (29 selected) Spaces: 2 UTF-8 LF Bicep 🔊
```





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

```
File Edit Selection View Go Run Terminal Help
                                                       main.tf - Days of Knowledge (Workspace) - Visual Studio Code
                                                                                                                              □ …
      main.tf X
      Terraform demo > tf > 😭 main.tf > 😭 resource "azurerm_network_security_group" "res-0"
            resource "azurerm_subnet" "res-5" {
       59
              address prefixes = ["10.1.0.0/24"]
       60
                                    = "default"
              resource_group_name = "dok-prep"
              virtual_network_name = "dok-prep-vnet"
       63
               depends_on = [
       64
                azurerm_virtual_network.res-2,
       65
       66
            resource "azurerm_windows_virtual_machine" "dok-prep" {
       67
                                     = "ignored-as-imported"
               admin_password
       69
              admin_username
                                     = "vmadmin"
       70
              location
                                     = "norwayeast"
       71
                                     = "dok-prep"
              network_interface_ids = ["/subscriptions/94670b10-08d0-4d17-bcfe-e01f701be9ff/resourceGroups/dok-prep/providers/Mici
       73
               resource_group_name = "dok-prep"
               size
                                     = "Standard_D4s_v5"
       75
               boot_diagnostics {
                storage_account_uri = "https://dokprepdiag.blob.core.windows.net/"
       76
       77
       78
               os_disk {
       79
                caching
                                      = "ReadWrite"
       80
                storage_account_type = "Premium_LRS"
       81
              source_image_reference {
                offer = "WindowsServer"
                publisher = "MicrosoftWindowsServer"
   ⊗ 0 <u>A</u> 0 ∪ tf
                                                                                            Ln 1, Col 1 Spaces: 2 UTF-8 LF Terraform 🔊 🚨
```

Which questions can I answer?

Creating and maintaining Azure infrastructure professionally



Tobias Fenster
Managing Partner, 4PS Germany





www.directions4partners.com