From zero to hero

Infrastructure as Code

Daniel Hasselwander

ti&m

Goals for this session

1. Understand how you can iterate through objects or arrays

2. Understand how to filter on arrays

3. Understand the usage of specific functions

ti&m

How a array looks like?

```
participants = [
  "rolf",
  "manuel",
  "gianluca",
  "wolfgang",
  "matthias",
  "daniel"
]


teacher = [
  "daniel"
]
```

ti&m

Terraform

Iterate over a array

```
participants = [
  "rolf",
  "manuel",
  "gianluca",
  "wolfgang",
  "matthias",
  "daniel"
]

participants_with_uppercase = [
  for participant in local.participants : upper(participant)
]
```

```
[
  "ROLF",
  "MANUEL",
"GIANLUCA",
"WOLFGANG",
"MATTHIAS",
"DANIEL"
]
```

ti&m

Iterate over a array with condition

```
locals {
  participants =
  [
    {
      name="rolf",
      teacher = false
    },
    {
      name="manuel",
      teacher = false
    },
    {
      name="gianluca",
      teacher = false
    },
    {
      name="wolfgang",
      teacher = false
    },
    {
      name="matthias",
      teacher = false
    },
    {
      name="daniel",
      teacher = true
    }
  ]

  participants_with_uppercase = [
    for participant in local.participants : upper(participant.name) if participant.teacher == false
  ]

}
```

```
[
    "ROLF",
    "MANUEL",
    "GIANLUCA",
    "WOLFGANG",
    "MATTHIAS",
]
```

ti&m

Iterate over a array with condition and objects

```
participants = [
  "rolf",
  "manuel",
  "gianluca",
  "wolfgang",
  "matthias",
  "daniel"
]

teacher = ["daniel"]

participants_with_uppercase = [
  for participant in local.participants : upper(participant) if contains(local.teacher, participant)
]
```

```
[
  "ROLF",
  "MANUEL",
  "GIANLUCA",
  "WOLFGANG",
  "MATTHIAS",
]
```

ti&m

Warum Arrays ?

ti&m

Generate Ressource in Loops

```
[

    "ROLF",

    "MANUEL",

    "GIANLUCA",

    "WOLFGANG",

    "MATTHIAS",

]
```

```hcl
data "ad_user" "participants" {
  for_each = toset(local.participants_with_uppercase)
  user_id = each.value # each.key is the same
}


resource "azurerm_resource_group" "rg" {
  location = "switzerlandnorth"
  name     = "rg-participients"
}


resource "azurerm_storage_account" "storagaeaccount" {
  account_replication_type = "LRS"
  account_tier             = "Standard"
  location                 = "switzerlandnorth"
  name                     = "stoparticipants"
  resource_group_name      = azurerm_resource_group.rg.name
}


resource "azurerm_storage_container" "stcontainer" {
  count = length(local.participants_with_uppercase)
  name                  = local.participants_with_uppercase[count.index]
  storage_account_name  = azurerm_storage_account.storagaeaccount.name
  container_access_type = "blob"
}
```

ti&m

Generate Ressource in Loops

Nachteile Count => Ändert sich die Reihenfolge muss alles neugeneriert werden

Tipp: Erst einsetzen wenn es sich lohnt

ti&m

Generate Ressource Blocks in Loops

```
locals {
  ports = [22, 80, 8080, 8081, 7080, 7081]
}


resource "azurerm_resource_group" "rg" {
  location = "switzerlandnorth"
  name     = "rg-participients"
}


resource "azurerm_network_security_group" "nsg" {
  name                = "nsg"
  location            = azurerm_resource_group.rg.location
  resource_group_name = azurerm_resource_group.rg.name

  dynamic "security_rule" {
    for_each = toset(local.ports)
    content {
      name                       = "inbound-rule-${security_rule.key}"
      description                = "Inbound Rule ${security_rule.key}"
      priority                   = sum([1000, security_rule.key])
      direction                  = "Inbound"
      access                     = "Allow"
      protocol                   = "Tcp"
      source_port_range          = security_rule.value
      destination_port_range     = security_rule.value
      source_address_prefix      = "*"
      destination_address_prefix = "*"
    }
  }
}
```

—Wird n-mal erzeugt in der Ressource beim Plan und Apply

ti&m

We digitalize your company.

Practice

ti&m

We digitalize your company.

Functions - Configuration Language | Terraform | HashiCorp Developer

ti&m

Specific Methods - Networking

```
resource "azurerm_subnet" "ssis_agent" {
  name                 = "subnet-sql-ssis-agent"
  resource_group_name  = module.snf_store_azure_group.name
  virtual_network_name = azurerm_virtual_network.store.name
  address_prefixes     = [cidrsubnet("172.17.30.0/24", 4, 0)]
}
```

Subnet 172.17.30.0/28 = 16 IPs = 11 Available

ti&m

We digitalize your company.

Practice

ti&m

Specific Methods - Networking

```
resource "azurerm_subnet" "ssis_agent" {
  name                 = "subnet-sql-ssis-agent"
  resource_group_name  = module.snf_store_azure_group.name
  virtual_network_name = azurerm_virtual_network.store.name
  address_prefixes     = [cidrsubnet("172.17.30.0/24", 4, 0)]
}
```

Subnet 172.17.30.0/28 = 16 IPs = 11 Available

ti&m

We digitalize your company.

Practice

ti&m

Specific Methods - Networking

```
resource "azurerm_subnet" "ssis_agent" {
  name                 = "subnet-sql-ssis-agent"
  resource_group_name  = module.snf_store_azure_group.name
  virtual_network_name = azurerm_virtual_network.store.name
  address_prefixes     = [cidrsubnet("172.17.30.0/24", 4, 0)]
}
```

Subnet 172.17.30.0/28 = 16 IPs = 11 Available

ti&m