

# Multi Cloud mit Terraform - Eine Einführung

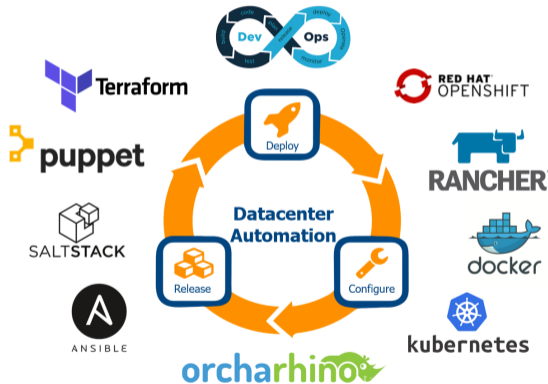


HashiCorp

# Terraform

Martin Grundei

21.08.2022



## ATIX AG

- ▶ Open Source Linux & Automatisierung
- ▶ Consulting
- ▶ Support
- ▶ Engineering
- ▶ Training

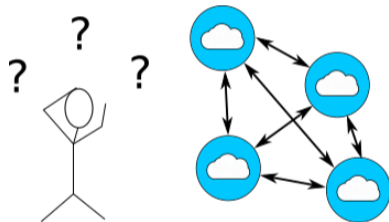
Homepage: [atix.de](http://atix.de)



## Meine Aufgaben als Consultant:

- ▶ Linuxadministration
- ▶ Konfigurationsmanagement (Puppet, Ansible)
- ▶ Cloud (AWS, Azure, Hetzner)
- ▶ Infrastructure as Code

- ▶ Multi-Cloud - Vorteile und Probleme
- ▶ Terraform - Eine kurze Übersicht
- ▶ Szenario 1: Azure VM mit AWS Route 53 DNS
- ▶ Szenario 2: Site-to-Site VPN Azure - AWS
- ▶ Ausblick



# Wechseln in die Cloud - Warum?

- ▶ Flexibilität
- ▶ Agilität
- ▶ Elastizität
- ▶ Verfügbarkeit
- ▶ Kosten

HETZNER



ORACLE



On premise



Cloud

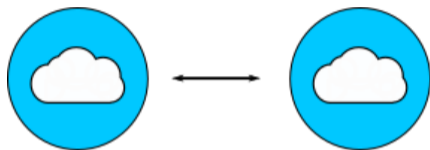


Hybrid



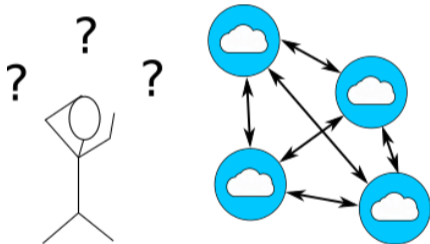
Multi-Cloud

# Warum Multi-Cloud?



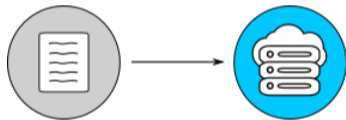
- ▶ Nutzung von Services verschiedener Anbieter
- ▶ Bestehende Infrastruktur
- ▶ Rechtliche Vorgaben
- ▶ Apps in private Clouds
- ▶ Kosten

- ▶ Wie behalte ich den Überblick?
- ▶ Wie verwalte ich meine Toolsets?
- ▶ Wie reagiere ich auf Änderungen?



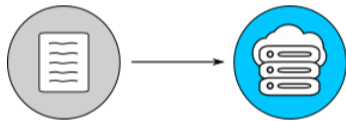


- ▶ Keine manuelle Verwaltung der Infrastruktur
- ▶ Infrastruktur als Dokumentation/Code
- ▶ Automatische Umsetzung der Dokumentation/des Codes



# Vorteile von IaC

- ▶ deklarativ
- ▶ nachvollziehbar
- ▶ vollständig
- ▶ automatisierbar

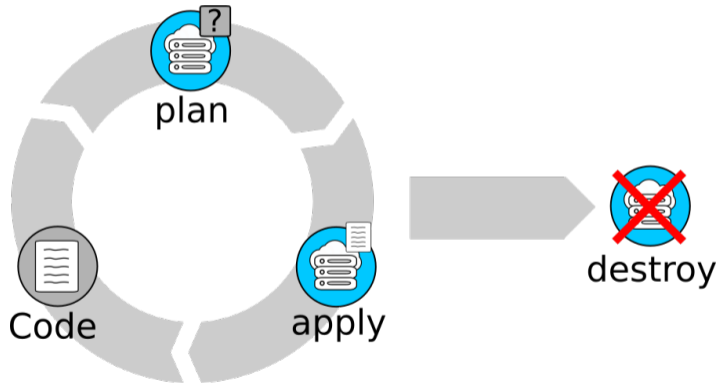


# Terraform - Übersicht

- ▶ Open-Source by HashiCorp
- ▶ Passt interaktiv die bestehende Infrastruktur dem Code an.
- ▶ Interagiert mit einer Vielzahl an Cloud/On-Premise APIs.



1. Code schreiben
2. terraform plan:  
Infrastruktur planen
3. terraform apply:  
Infrastruktur aufsetzen
4. terraform destroy:  
Infrastruktur löschen



- ▶ Ressourcen in 'human-readable' Dateien (.tf)
- ▶ Ressourcen beliebig auf Dateien aufteilbar

Beispiel: Erzeugung von SSH Keys

```
resource "tls_private_key" "this" {  
  algorithm = "RSA"  
  rsa_bits  = 2048  
}
```



- ▶ Reihenfolge implicit aus Abhängigkeiten der Ressourcen
- ▶ Art der Resource beliebig

Beispiel: Ablegen von SSH Key files

```
resource "local_file" "public_key" {  
  file_permission = "0666"  
  content         = tls_private_key.this.public_key_openssh  
  filename       = "${path.root}/local/${var.ssh_key_name}.pub"  
}
```



Terraform 'merkt' sich die erzeugte Infrastruktur.

- ▶ Infrastruktur in `.terraform.tfstate` file
- ▶ Ressourcen rücksetzbar/wiederherstellbar
- ▶ Automatisches Löschen der Infrastruktur möglich



Enthält sensitive Daten im **Klartext**.

`terraform.tfstate` sicher aufbewahren:

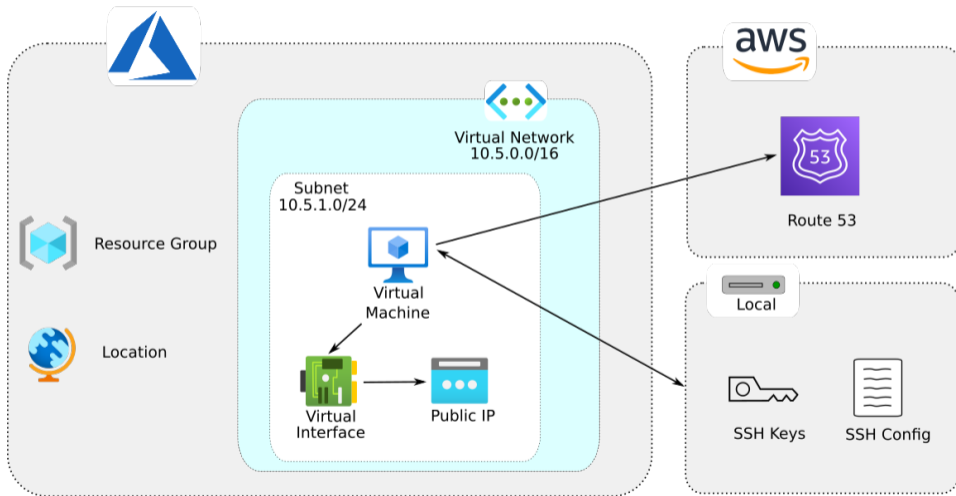
- ▶ Zugang beschränken
- ▶ Verschlüsseln
- ▶ Backend benutzen (v.a. im Team)

Backends für Gitlab, AWS S3 Bucket, Azure, Kubernetes, Consul, etc.





# Azure - Route 53



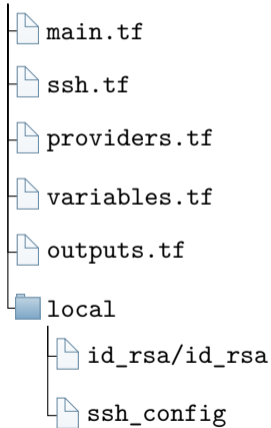
## Simple Beispiel-Szenario:

- ▶ Linux VM in Azure
- ▶ DNS mit Route 53
- ▶ Login über SSH

## Was wird benötigt:

- ▶ SSH Key
- ▶ Azure Virtual Network
- ▶ Azure Resource Group
- ▶ Azure Subnet
- ▶ Azure Linux VM
- ▶ Azure Network Interface
- ▶ Azure Public IP
- ▶ AWS Route 53 A Record

## Azure - Route53



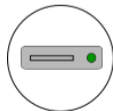
- ▶ `main.tf`: Hauptteil des Codes, Cloud Ressourcen
- ▶ `providers.tf`: Provider, Terraform Konfiguration, Grundlegende Datentypen
- ▶ `variables.tf`: Variablen (Mehr dazu später)
- ▶ `outputs.tf`: Output (Auch mehr dazu später)
- ▶ `local`: Lokale Dateien (SSH Keys, SSH Config File)

# Provider Block



Provider interagieren mit der API.

```
provider "azurerm" {  
  features {}  
  skip_provider_registration = true  
}  
  
provider "aws" {  
  region = var.aws_provider_region  
}  
  
provider "local" {}  
  
provider "tls" {}
```



# Providers



## hashicorp/azurerem

- ▶ Resource Manager API
- ▶ Computing
- ▶ Storage
- ▶ Networking
- ▶ ...

(Für Users/Groups/AD/... s.  
hashicorp/azuread)



## hashicorp/aws

- ▶ Computing
- ▶ Datenbanken
- ▶ IAM
- ▶ Networking
- ▶ Container
- ▶ ...



## hashicorp/local

- ▶ Lokale Dateien



## hashicorp/tls

- ▶ SSH keys



## terraform plan/terraform apply

- ▶ Vergleich Ist-Zustand mit Soll-Zustand mit Übersicht
- ▶ Bestätigung für das Deployment nötig

```
[user@laptop]$ terraform apply
# tls_private_key.this will be created
+ resource "tls_private_key" "this" {
  + algorithm           = "RSA"
  + ecdsa_curve         = "P224"
  ...
}
...
Plan: 10 to add, 0 to change, 0 to destroy.
...
Do you want to perform these actions?
```



# terraform plan/terraform apply



## ► Output nach terraform apply

```
Apply complete! Resources: 10 added, 0 changed, 0 destroyed
```

```
Outputs:
```

```
azure_admin_username = <sensitive>
```

```
azure_vm_fqdn = "testubuntuvms.atix-training.de"
```

```
public_ip_ip_address = "13.80.22.203"
```



Nach dem Deployment erhält man folgendes SSH Config File

```
Host azure_vm_fqdn_login
  HostName testubuntuvm.atix-training.de
  User atix
  IdentityFile id_rsa
```

login ist möglich mit

```
user@laptop:~$ ssh -F ssh_config azure_vm_fqdn_login
Welcome to Ubuntu 20.04.4 LTS (GNU/Linux 5.15.0-1014-azure x86_64)
...
atix@TestUbuntuVM:~$
```



Microsoft Azure

[Home](#) >

## Virtual machines

ATIX AG

[+](#) Create [↔](#) Switch to classic [🕒](#) Reservations [⚙️](#) Manage view [🔄](#) Refresh

Filter for any field...

Subscription equals **all**

Type equals **all**

[+](#) Add filter

[More \(2\)](#)

No grouping

List view

| <input type="checkbox"/> | Name ↑↓      | Type ↑↓         | Subscription ↑↓          | Resource group ↑↓ | Location ↑↓ |
|--------------------------|--------------|-----------------|--------------------------|-------------------|-------------|
| <input type="checkbox"/> | TestUbuntuVM | Virtual machine | Microsoft Partner Net... | MartinGDemo       | West Europe |

Page 1 of 1 Showing 1 to 1 of 1 records.

[Give feedback](#)

terraform destroy löscht die komplette Infrastruktur.

```
Plan: 0 to add, 0 to change, 10 to destroy.
```

```
...
```

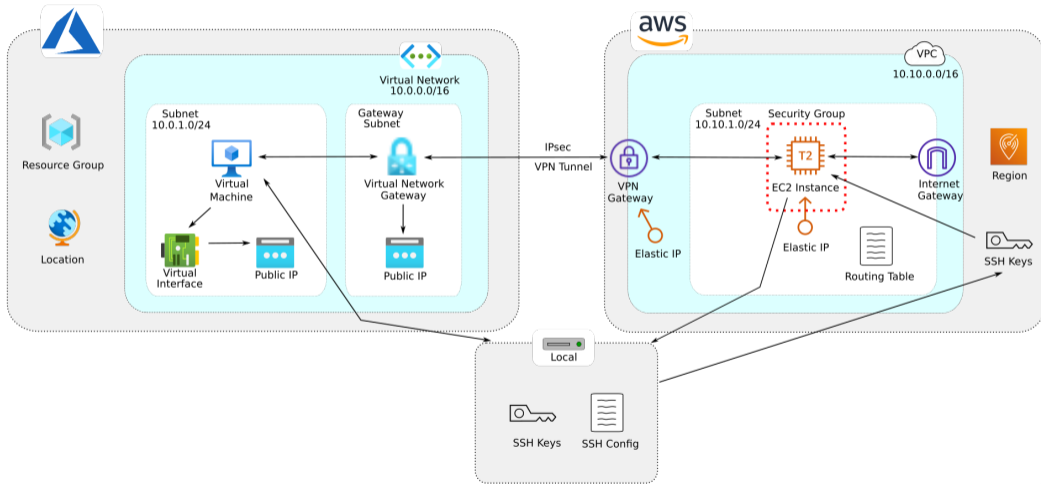
```
Do you really want to destroy all resources?
```

```
Terraform will destroy all your managed infrastructure, as shown above.  
There is no undo. Only 'yes' will be accepted to confirm.
```

```
Enter a value:
```

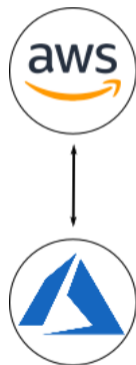
```
Destroy complete! Resources: 10 destroyed.
```

# Azure - AWS - Site-to-Site - VPN



## Szenario:

- ▶ Linux VM in Azure subnet 10.0.1.0/24
- ▶ Linux VM in AWS subnet 10.10.1.0/24
- ▶ VPN Site-to-Site Connection zwischen beiden Subnetzen
- ▶ ICMP und SSH zwischen beiden VMs über lokales Netzwerk



- ▶ Setup Azure VM:
  - ▶ s. Szenario 1
- ▶ Setup AWS VM:
  - ▶ VPC
  - ▶ Subnet
  - ▶ Elastic IP
  - ▶ Security Group
  - ▶ Internet Gateway
  - ▶ EC2 Instance
  - ▶ SSH Keys
- ▶ AWS VPN Setup:
  - ▶ VPN Gateway
  - ▶ Elastic IP
  - ▶ Routing Table
- ▶ Azure VPN Setup:
  - ▶ Gateway Subnet
  - ▶ Virtual Network Gateway
  - ▶ Public IP
- ▶ Azure-AWS-connection:
  - ▶ 2 Azure Local Gateways
  - ▶ AWS Customer Gateway

terraform apply:

```
Apply complete! Resources: 29 added, 0 changed, 0 destroyed.
```

```
Outputs:
```

```
aws_elastic_ip_ip_address = "3.74.131.215"  
awsvm_local_ip           = "10.10.1.220"  
azure_admin_username    = <sensitive>  
azure_public_ip_ip_address = "20.229.216.53"  
azurevm_local_ip        = "10.0.1.4"
```



# Azure - AWS - Site-to-Site - VPN



The screenshot shows the AWS Management Console interface for EC2 instances. The top navigation bar includes the AWS logo, a search bar, and the region 'Frankfurt'. The main content area is titled 'Instances (1) Info' and features a search bar and a filter 'Instance state = running'. Below this is a table with the following data:

| Name           | Instance ID         | Instance state | Instance type | Status check      | Alarm status | Availability Zone | Public IPv4 DNS          | Public IPv4 ... |
|----------------|---------------------|----------------|---------------|-------------------|--------------|-------------------|--------------------------|-----------------|
| demo_amazon... | i-004b4b5bf3a4a6467 | Running        | t2.micro      | 2/2 checks passed | No alarms    | eu-central-1a     | ec2-3-74-131-215.eu-c... | 3.74.131.215    |

The screenshot shows the Microsoft Azure portal interface for virtual machines. The top navigation bar includes the Microsoft Azure logo and a search bar. The main content area is titled 'Virtual machines' and features a search bar and a filter 'Subscription equals all'. Below this is a table with the following data:

| Name     | Type            | Subscription            | Resource group | Location    | Status  | Operating system | Size         | Public IP address | Disks |
|----------|-----------------|-------------------------|----------------|-------------|---------|------------------|--------------|-------------------|-------|
| UbuntuVM | Virtual machine | Microsoft Partner Net-- | MartinGDemo    | West Europe | Running | Linux            | Standard_B1s | 20.229.216.53     | 1     |

# Azure - AWS - Site-to-Site - VPN



The screenshot shows the AWS Management Console interface for a VPN connection. The breadcrumb navigation is VPC > VPN connections > vpn-03d54eeb3932baaed. The page title is vpn-03d54eeb3932baaed / azure\_to\_aws\_connection. There are buttons for 'Download configuration' and 'Actions'. The 'Details' section is divided into four columns of key-value pairs:

- VPN ID:** vpn-██████████
- State:** Available
- Virtual private gateway:** [vgw-01a4d00463aee409c](#)
- Customer gateway:** [cgw-03a689e4b2bb686b7](#)
- Transit gateway:** -
- Customer gateway address:** 13.93.14.95
- Type:** ipsec.1
- Category:** VPN
- VPC:** -
- Routing:** Static
- Acceleration enabled:** False
- Authentication:** Pre-shared key
- Local IPv4 network CIDR:** 0.0.0.0/0
- Remote IPv4 network CIDR:** 0.0.0.0/0
- Local IPv6 network CIDR:** -
- Remote IPv6 network CIDR:** -
- Core network ARN:** -
- Core network attachment ARN:** -
- Gateway association state:** associated
- Outside IP address type:** PublicIpv4

Below the details are tabs for 'Tunnel details', 'Static routes', and 'Tags'. The 'Tunnel state' table is shown below:

| Tunnel number | Outside IP address | Inside IPv4 CIDR   | Inside IPv6 CIDR | Status | Last status change                 | Details | Certificate ARN |
|---------------|--------------------|--------------------|------------------|--------|------------------------------------|---------|-----------------|
| Tunnel 1      | 18.194.132.6       | 169.254.82.112/30  | -                | Up     | July 17, 2022, 9:36:02 (UTC+02:00) | -       | -               |
| Tunnel 2      | 35.157.140.182     | 169.254.129.196/30 | -                | Up     | July 17, 2022, 9:37:02 (UTC+02:00) | -       | -               |



Microsoft Azure

Home > Virtual network gateways > VirtualNetworkGateway

## VirtualNetworkGateway | Connections ✦ ☆ ... ✕

Virtual network gateway

+ Add  Refresh

| Name               | ↑↓ | Status    | ↑↓ | Connection type      | ↑↓ | Peer               | ↑↓  |
|--------------------|----|-----------|----|----------------------|----|--------------------|-----|
| AzuretoAWS         |    | Connected |    | Site-to-site (IPsec) |    | AWSLocalGateway    | ... |
| AzureToAWSFailover |    | Connected |    | Site-to-site (IPsec) |    | AWSFailoverGateway | ... |

ssh configuration file:

```
Host azure_vm
  HostName 20.229.216.53
  User atix
  IdentityFile id_rsa

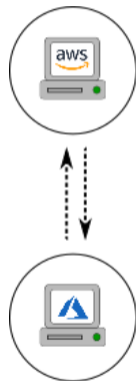
Host aws_vm_jump
  HostName 10.10.1.220
  User atix
  IdentityFile id_rsa
  ProxyJump azure_vm
```



## SSH Proxy Jump:

```
[mg@mg-XPS-15-9500 local (develop)]$ ssh -F ssh_config
azure_vm_jump
Welcome to Ubuntu 20.04.4 LTS (GNU/Linux 5.15.0-1014-azure
x86_64)
...
Last login: Sun Jul 17 10:24:11 2022 from 10.10.1.220
To run a command as administrator (user "root"), use "sudo
<command>".
See "man sudo_root" for details.

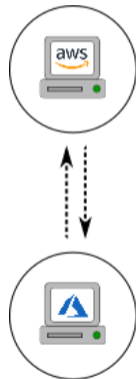
atix@UbuntuVM:~$
```



## ICMP:

```
atix@UbuntuVM:~$ ping 10.0.1.4 -c 3
PING 10.0.1.4 (10.0.1.4) 56(84) bytes of data.
64 bytes from 10.0.1.4: icmp_seq=1 ttl=64 time=0.029 ms
64 bytes from 10.0.1.4: icmp_seq=2 ttl=64 time=0.050 ms
64 bytes from 10.0.1.4: icmp_seq=3 ttl=64 time=0.051 ms

--- 10.0.1.4 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time
 2050ms
```

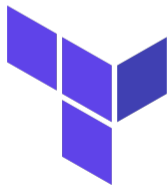
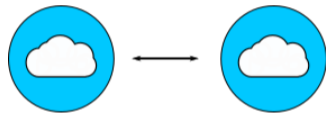


## Multi-Cloud

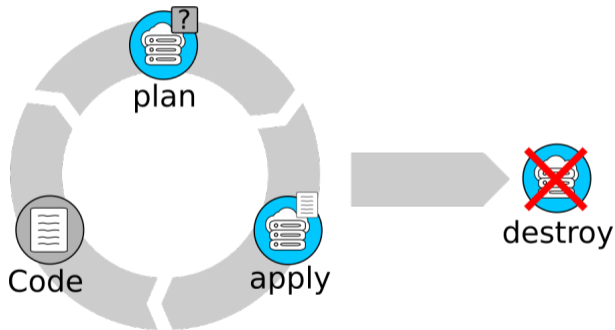
- ▶ Ursache: Services, Rechtliche Vorgaben, Kosten,...
- ▶ Problem: Wie manage ich meine Umgebungen?
- ▶ Lösung: IaC

## Terraform

- ▶ Ressourcen einfach deployen
- ▶ Bestimmt automatisch die Reihenfolge
- ▶ 'Merkt' sich die Infrastruktur



## Workflow



## Produktiv gehen

- ▶ Versionsverwaltung
- ▶ Pipelines
- ▶ Terraform Backend(!)



Vielen Dank fürs Zuhören



HETZNER



ORACLE



OSAD  
OPEN SOURCE AUTOMATION DAYS

Science Congress Center  
Garching  
04.10. – 06.10.2022

2 Conference Days  
1 Workshop Day

The poster features the text 'OSAD' in large orange letters with a blue gear icon to the left. Below it, 'OPEN SOURCE AUTOMATION DAYS' is written in white. The location 'Science Congress Center Garching' and dates '04.10. – 06.10.2022' are in orange. At the bottom, '2 Conference Days' and '1 Workshop Day' are listed in orange. The background is black with several colorful, glowing gear icons in blue, purple, yellow, and green.

- ▶ Documentation: <https://www.terraform.io/docs/>
- ▶ HCL specific: <https://www.terraform.io/docs/language/>
- ▶ Providers and Modules: <https://registry.terraform.io/>