

Infrastructure as Code

Zur Automatisierung und Absicherung

Download

- <https://github.com/DominikLandau/HSAN-2025>



Gliederung

- Was ist Infrastructure as Code?
- Welche Anbieter gibt es?
- Wie setzt man Ansible ein?
- Wie setzt man Terraform ein?

Was ist Infrastructure as Code

Infrastructure as Code

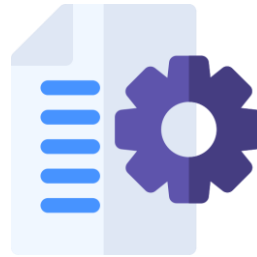
- IaC
- Verwaltung der IT-Infrastruktur mithilfe von Dateien
 - -> Manuelles Erstellen von Ressourcen ist ineffizient und fehleranfällig
- Deklarativer Ansatz
 - -> Wir sagen, was wir wollen und nicht wie es gemacht werden soll

IaC Workflow

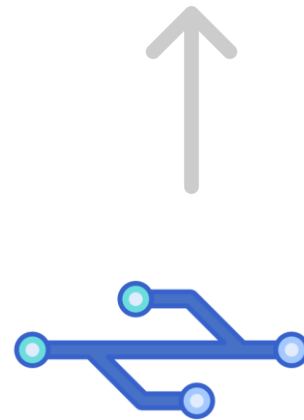
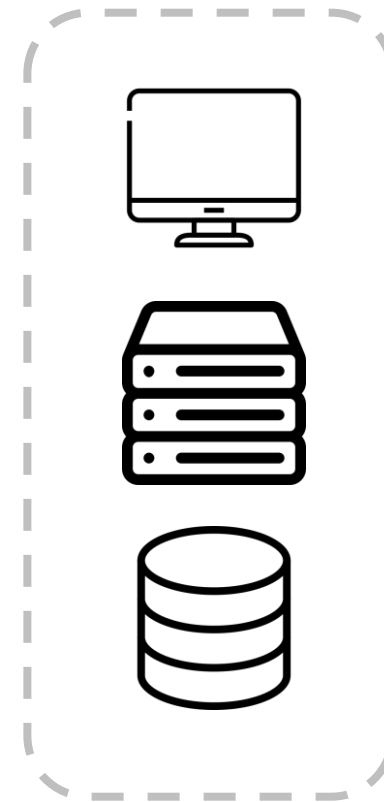
Entwickler



Infrastrukturkonfiguration



Infrastruktur



Versionskontrolle

Kernbereiche

- **Versionierung**
 - Einsatz von Versionskontrollsystemen (Git)
- **Idempotenz**
 - Mehrfaches Ausführen führt zum gleichen Ergebnis
- **Modularität**
 - Wiederverwendbare Code-Bausteine
- **Abstraktion**
 - Vereinfachung komplexer Infrastruktur

Ziele

- Automatisierung
- Nachverfolgbarkeit
- Standardisierung
- Konsistent
- Zuverlässig

Welche Anbieter gibt es?

IaC Tools



Ansible



Crossplane



Pulumi



Chef



Terraform

IaC Tools



Ansible



Crossplane



Pulumi

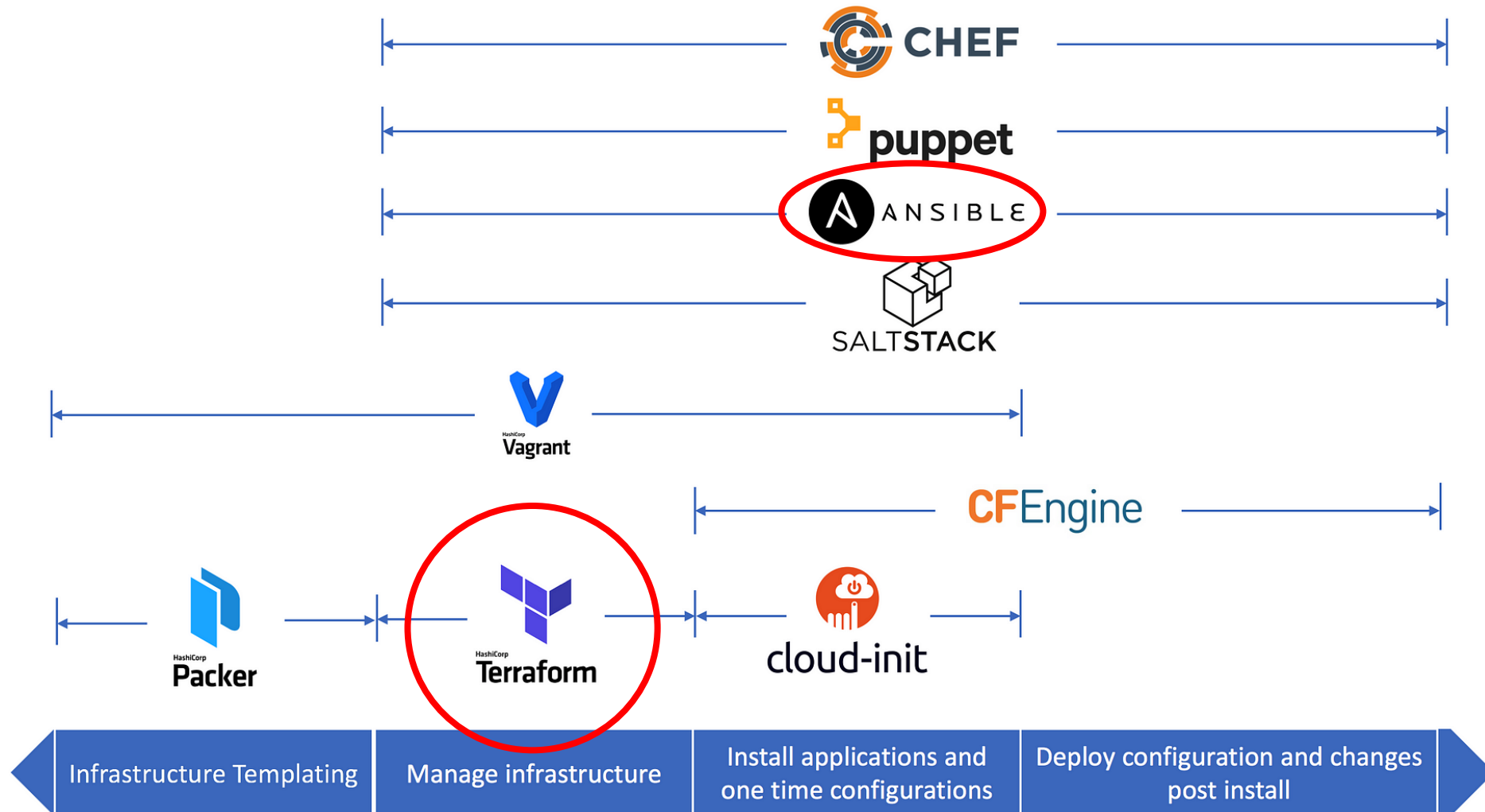


Chef

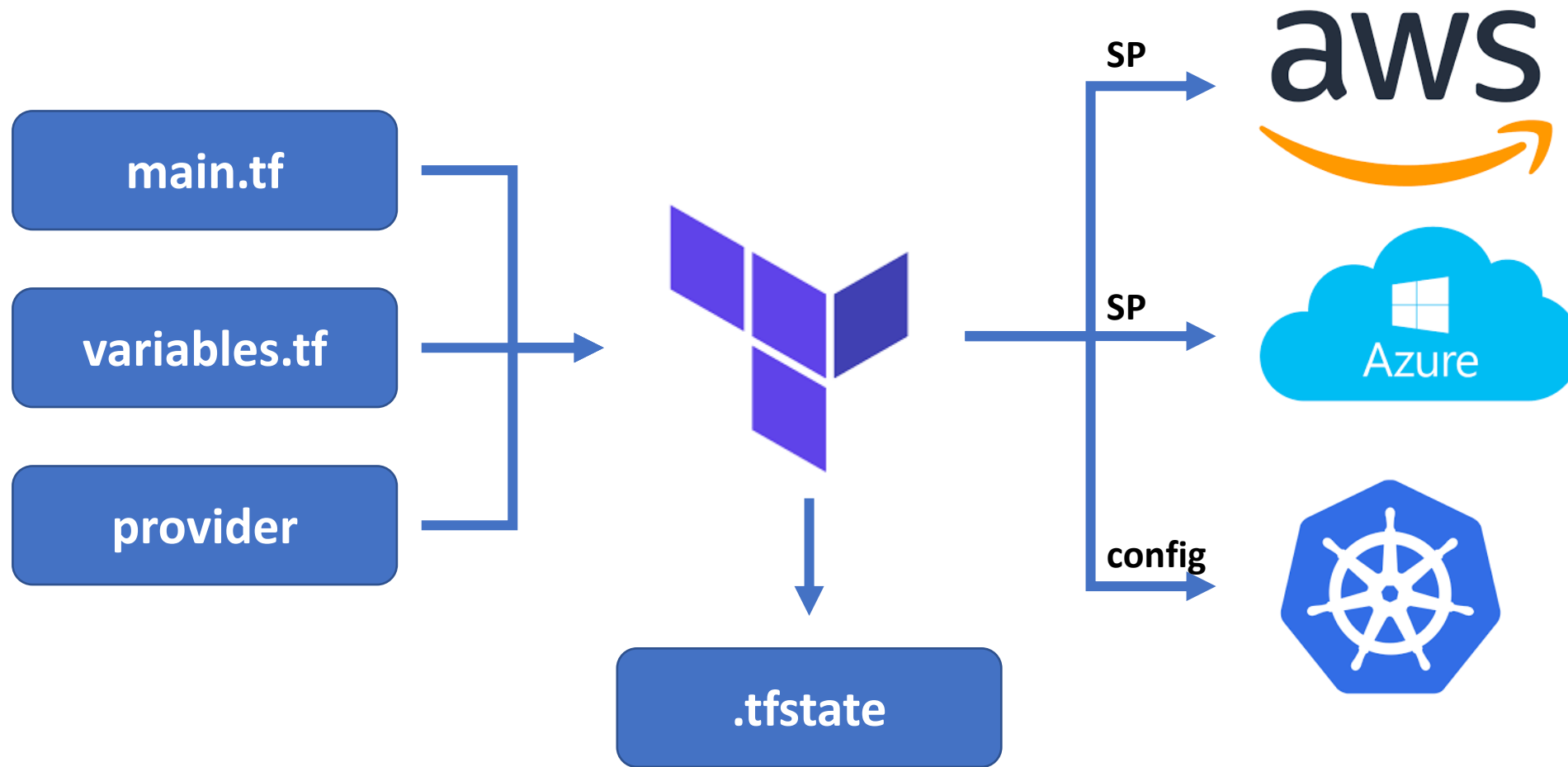


Terraform

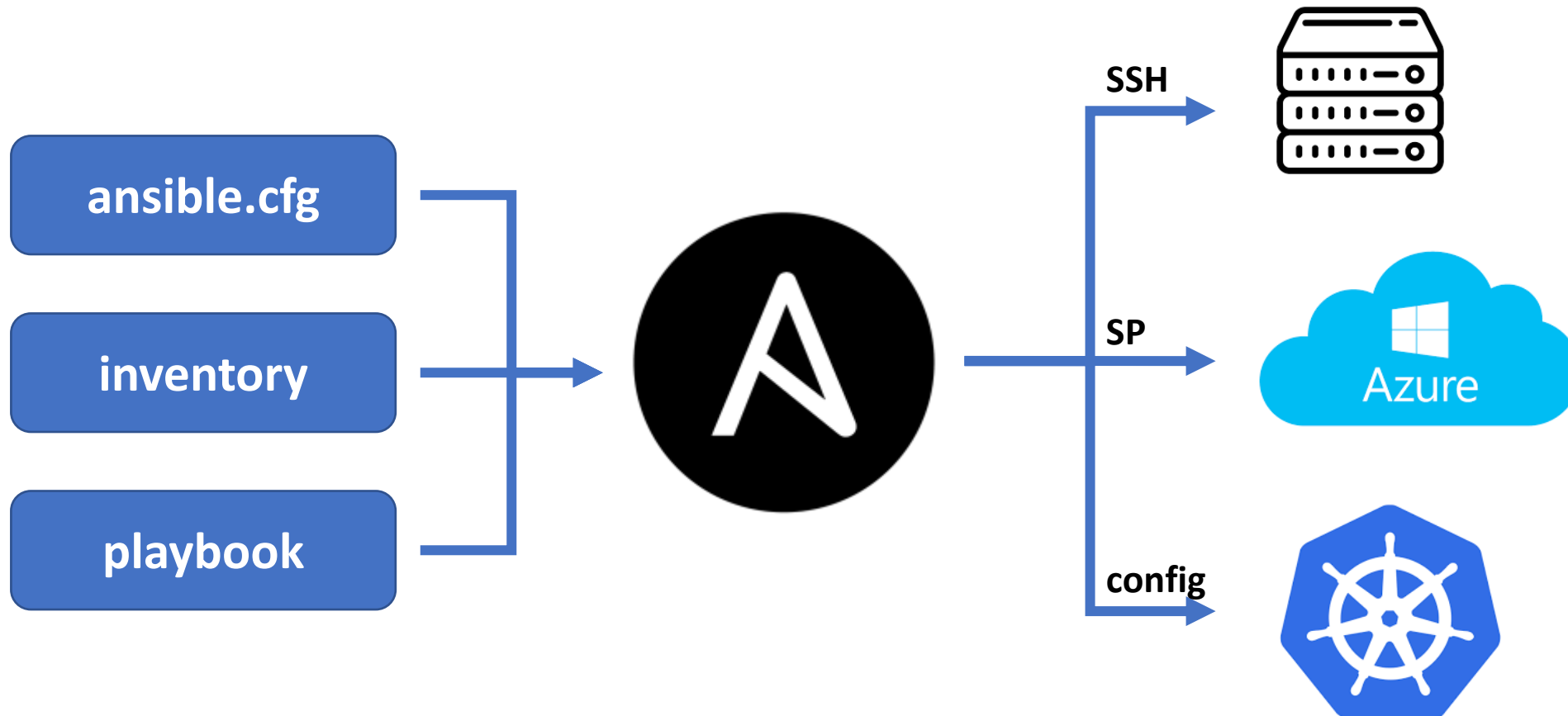
Einsatzgebiet



Terraform



Ansible



Terraform vs. Ansible

Jedes Tool hat seine Stärken – es gibt nicht immer ein Entweder-oder

Terraform

- + tfstate
- + Plan zeigt Änderungen
- + Infrastruktur Management

Ansible

- + Sehr flexibel
- + Einfach erweiterbar
- + Konfiguration Management

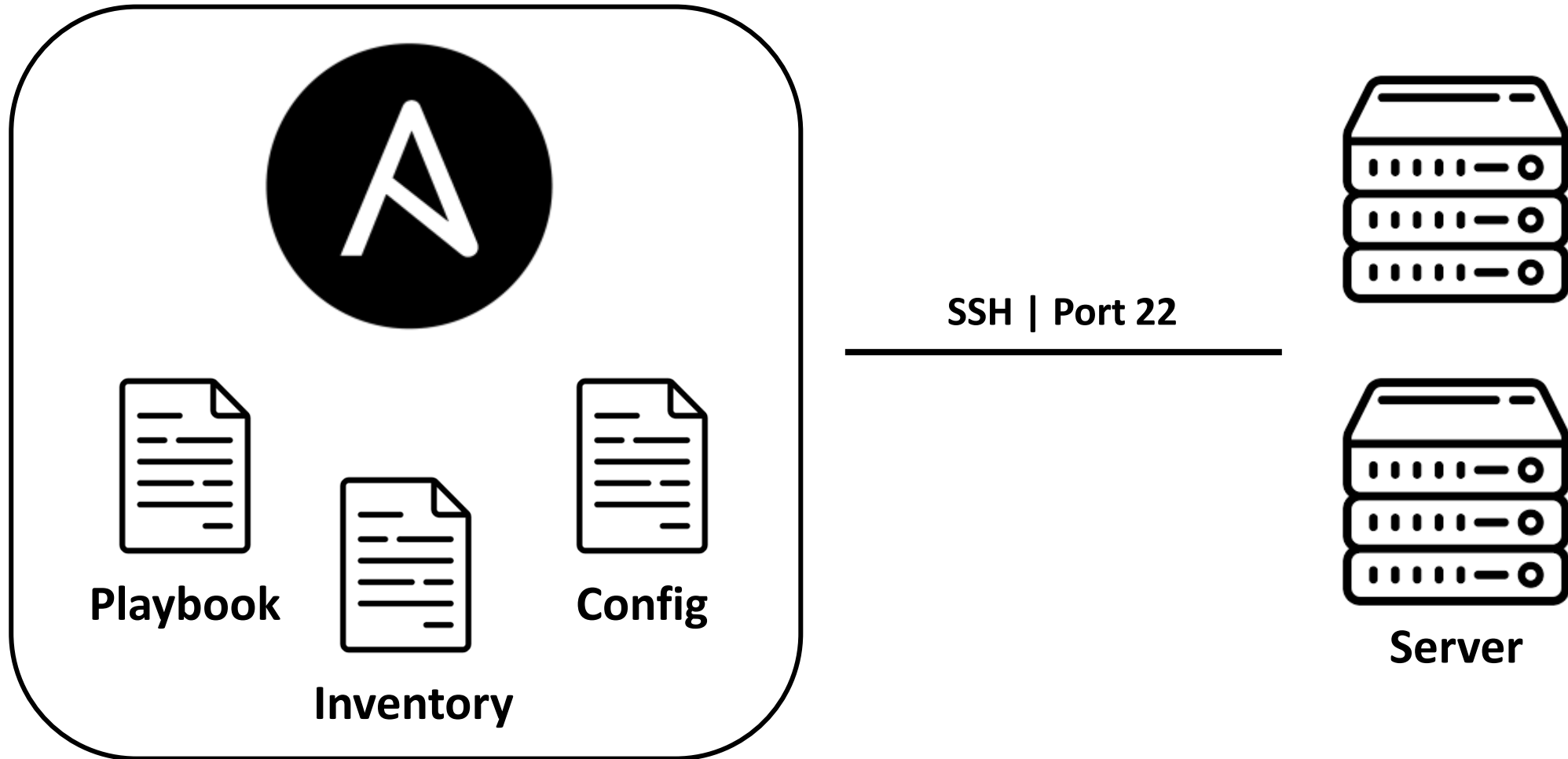
-> Oft ist die Lösung eine Kombination

Abisicherung

- CIS (Center for Internet Security)
 - <https://downloads.cisecurity.org/#/>
- Automatisierte Sicherheitsüberprüfungen
 - Statische Analyse
- Ansible
 - https://ansible-lockdown.readthedocs.io/en/latest/CIS/CIS_table.html

Wie setzt man Ansible ein?

Ansible Basics



Ansible Dateien



ansible.cfg

```
[defaults]
inventory = ./inventory
remote_user = dominik
private_key_file = .ssh/azure
host_key_checking = False
```



inventory

```
[webserver]
23.96.11.119
```



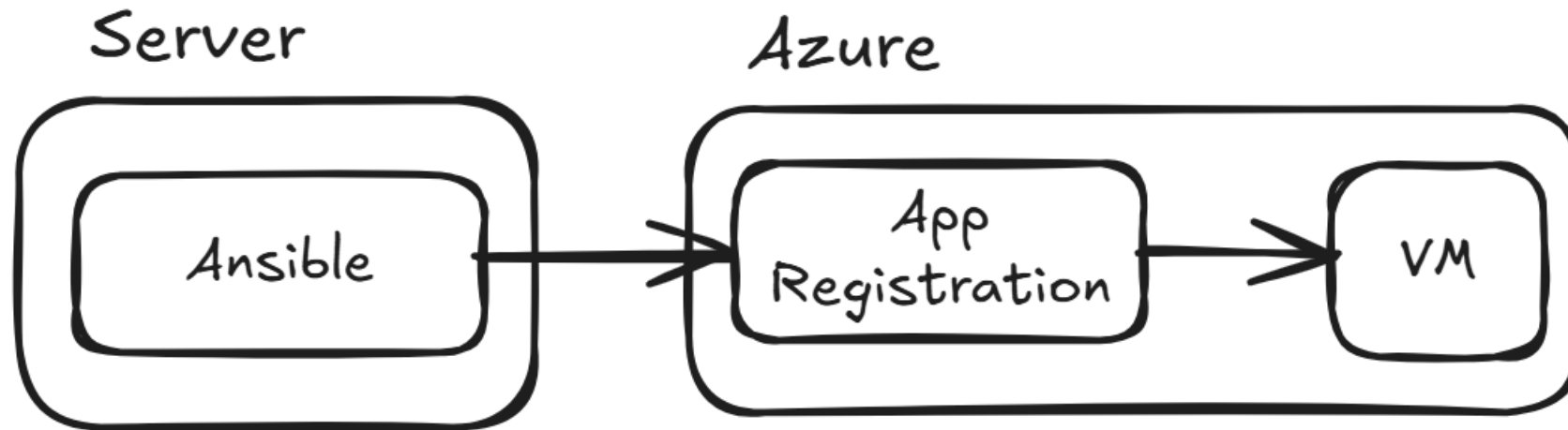
main.yaml

```
- name: Simple Playbook
  hosts: all

  tasks:
    - name: One Task
      ansible.builtin.debug:
        msg: "Hallo"
```

Ansible und Azure

- Modul
 - <https://galaxy.ansible.com/ui/repo/published/azure/azcollection/>



Ansibel Code

playbook.yaml

```
- name: Create a Resource Group
hosts: localhost
connection: local
```

tasks:

```
- name: Create a resource group
  azure_rm_resourcegroup:
    name: myResourceGroup
    location: westus
```

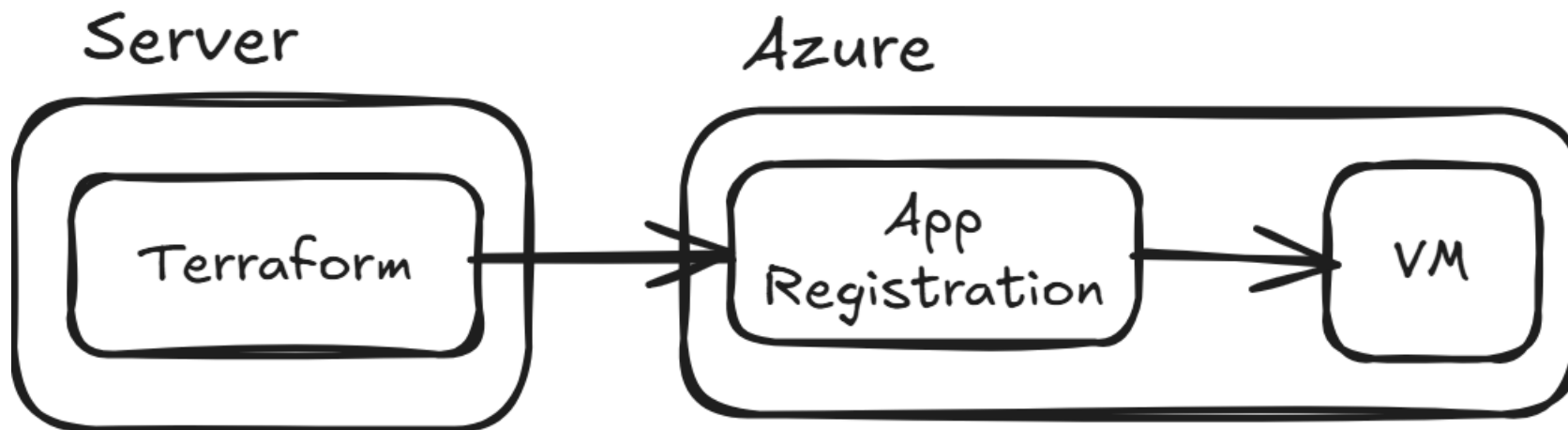
app-reg-creds.sh

```
export AZURE_SUBSCRIPTION_ID=<id>
export AZURE_CLIENT_ID=<id>
export AZURE_SECRET=<pwd>
export AZURE_TENANT=<id>
```

Wie setzt man Terraform ein?

Terraform und Azure

- Beispiel
 - <https://learn.microsoft.com/en-us/azure/developer/terraform/create-resource-group?tabs=azure-cli>



Terraform Code

app-reg-creds.sh

```
export ARM_SUBSCRIPTION_ID=<id>
export ARM_CLIENT_ID=<id>
export ARM_CLIENT_SECRET=<pwd>
export ARM_TENANT_ID=<id>
```

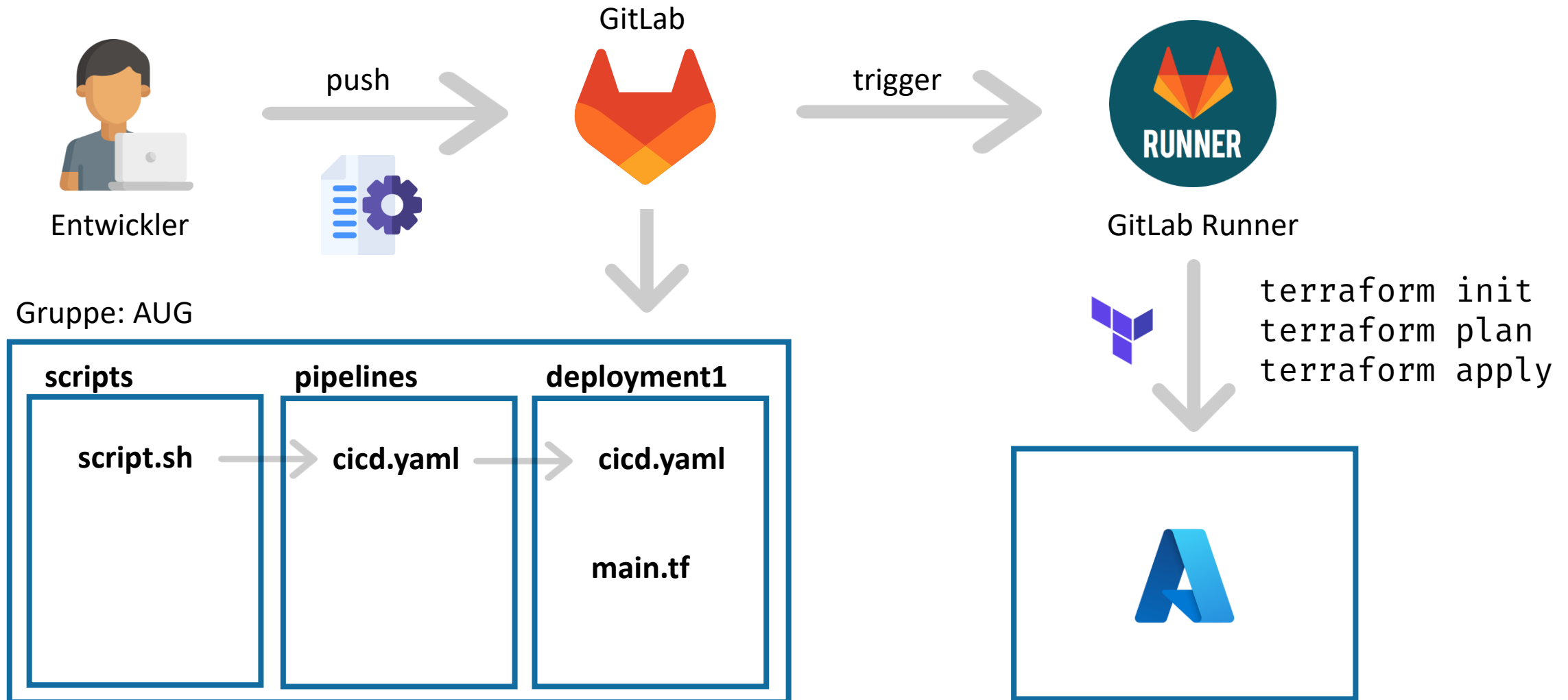
Terraform.tf

```
terraform {
  required_providers {
    azurerm = {
      source = "hashicorp/azurerm"
      version = "~>4.0"
    }
  }

  provider "azurerm" { features {} }

  resource "azurerm_resource_group" "example" {
    location = "germanywestcentral"
    name     = "Tagderoffenentuer"
  }
```


Terraform, Azure und GitLab



Cloud Computing

Erste Schritte mit der Cloud
(Azure)