Verwaltung von Firewall Regeln mit Hilfe von Infrastructure as Code

Studiengang: MAS | Vertiefung: MAS Information Technology

Die anstehende Payment Card Industry Data Security Standard (PCI-DSS) Zertifizierung stellt PostFinance in vielen Bereichen vor neue Herausforderungen. Dazu gehört auch die Handhabung von Firewall Regeln. Unter anderem müssen zusätzliche Anforderungen im Bereich Nachvollziehbarkeit für die PCI-DSS Zertifizierung erfüllt werden. Mit Hilfe eines IaC (Infrastructure as Code) Ansatzes soll die bestehende Lösung durch eine neue ersetzt werden.

Ausgangslage

PostFinance betreibt heute mehrere hundert Virtuelle Maschinen (VMs) mit Windows oder Linux Betriebssystem. Die Systeme werden mit Hilfe von IaC und Terraform (Software zur Infrastrukturautomation) auf der on-premises Infrastruktur provisioniert und verwaltet. Firewall Regeln werden zum aktuellen Zeitpunkt noch nicht mit dieser Lösung verwaltet. Durch zusätzliche Anforderungen an die Nachvollziehbarkeit im Bereich „Firewall Rule Management“ und zur weiteren Steigerung der Automatisierung soll eine neue Lösung entwickelt werden, welche sich in die bestehende Infrastrukturautomation integriert.

Zielsetzung

Mit dieser Master-Thesis entwickle ich eine Software, in der sich Firewall Regeln abbilden lassen und die sich mit Hilfe eines „Provider Plugins“ und entsprechenden APIs in Terraform integriert. Dazu wird ein eigenes Modell zur Verwaltung der Regeln entwickelt und in der Software implementiert. Die Softwarelösung wir dabei als „Minimum Viable Product“ entwickelt und die Basis für das weitere Vorgehen in diesem Bereich bilden.

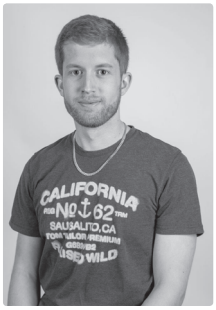
Lösung

Das entwickelte Modell zur Verwaltung der Firewall Regeln arbeitet mit Labels in der Form „app=web“, welche jedem Server automatisch oder manuell via Infrastrukturautomation zugeordnet werden. Ähnliche Modelle findet man heute bereits bei Public Cloud Providern wie Google oder Microsoft.
Firewall Regeln werden neu mit Hilfe von IaC verwaltet. Quell- und Zielsystem(e) werden dabei mit einer Kombination aus einem oder mehreren solcher Labels adressiert und via Infrastrukturautomation in der Softwarelösung persistiert. Die Labels der entsprechenden Regeln werden durch die Software zu einer normalen Firewall Regel (mit Quell- und Ziel IP-Adressen/Ports anstelle von Labels) übersetzt.

Realisiert wurde die Applikation in einem Microservice Ansatz mit der Programmiersprache go. Schnittstellen wurden „Contract first“ mit Hilfe von gRPC definiert und anschliessend implementiert. Als Laufzeitumgebung wurde die PostFinance interne Platform as a Service (PaaS) auf Basis von Kubernetes gewählt.

Fazit

Mit dieser Arbeit wurde die Grundlage geschaffen für eine moderne, nachvollziehbare Verwaltung von Firewall Regeln bei PostFinance. Es hat sich gezeigt, dass durch IaC ein grosser Teil der Anforderungen in Bezug auf die Nachvollziehbarkeit gelöst werden kann. Ausserdem wird die Automatisierung in der Infrastruktur weiter gesteigert. Die gewonnen Erkenntnisse werden nun in einem Projekt weiterverarbeitet und eingeführt.
Für die Entwicklung konnte ich einiges an Wissen aus besuchten CAS Modulen verwenden und weiter vertiefen. Besonders herausfordernd und lehrreich empfand ich die Entwicklung der Applikation von A-Z ohne Hilfe anderer Entwickler/Architekten. Das Lösen kniffliger Probleme hat mich herausgefordert, angetrieben und schliesslich in meiner Arbeitsweise bestätigt.



Thomas Gosteli

```
module "jira.example.com" {
  source           = "git::ssh://git@git.example.com/terraform/modules/linux.git//vm?ref=v1.1.0"
  usage            = "IDOKU"
  node_platform    = "Entwicklung"
  node_description = "JIRA"
  node_domain      = "example.com"
  node_ip = [
    {
      interface = true
      managed   = true
      name      = "jira"
      vlan      = "1234"
    },
  ]
  sap_cost_center           = "API03532"
  foreman_major             = "7"
  foreman_minor             = "6"
  foreman_media             = "Redhat 7.6 - 201906"
  vsphere_template          = "rhel-7.6-20190618-20190702-1748"
  vsphere_datacenter_name   = "Layer 2"
  vsphere_datastore_cluster = "L2_2_DSCL"
  vsphere_resource_pool     = "RP_ITD - 2"
  vsphere_node_vcpu         = "12"
  vsphere_node_memory       = "16384"
  vsphere_node_disks = [
    {
      size = "128"
      unit = "0"
    },
    {
      size = "250"
      unit = "1"
    },
    {
      size = "100"
      unit = "2"
    },
  ]
}
```

Terraform Definition einer VM