

Automatisiertes Deployment mit HashiCorp Terraform

In meinem Praxisprojekt habe ich mich mit dem automatisierten Deployment von Virtuellen Maschinen mit Terraform beschäftigt für den Bereich B4. Dieser benötigt nämlich demnächst mehr als 1000 zusätzliche Server um ein Projekt aufzustemmen.

Terraform ist ein Tool für Infrastructure as a Code. Das bedeutet, dass man in Computercode die Infrastruktur, die man aufgebaut haben möchte, beschreibt und Terraform guckt sich den Ist-Stand an und versucht den Soll-Zustand zu erreichen. Ich habe hierfür aufgrund von Zeitproblemen statt der originell geplanten Vmware Umgebung mit Proxmox gearbeitet. Dabei habe ich in der Terraform-Sprache ein Skript erstellt, welches die Infrastruktur basierend auf Eingangsvariablen erstellt.

Zusätzlich habe ich Ansible, ein Tool für die Automatisierung, dazu genommen. In Ansible habe ich Terraform angesteuert, sodass ich mittels eines Ansible Playbooks Eingangsvariablen validieren und dann an Terraform übergeben kann.

Ziel des Ganzen war es, dieses Ansible Playbook auf die Red Hat Ansible Automation Platform zu portieren, was mir im Endeffekt gelungen ist. Rechts in Bild 1 sehen sie die Ausgabe des Playbooks auf Ansible Automation Platform, nachdem ich dieses ausgeführt hatte.

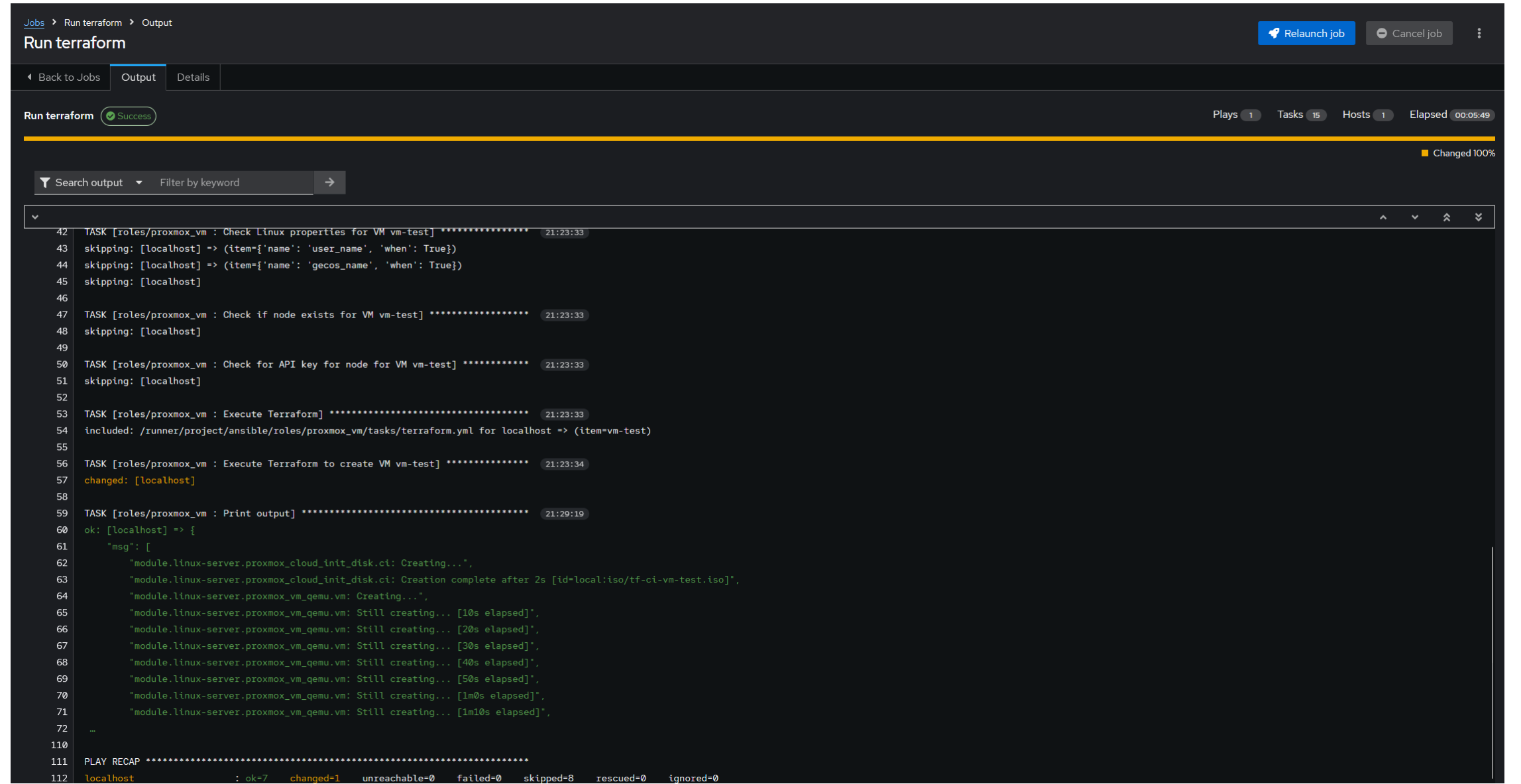


Bild 1: Erfolgreiche Ausführung vom Ansible Playbook

Links zu sehen ist die Proxmox-Weboberfläche meines Testsystems. Proxmox ist eine Virtualisierungsumgebung. Das Ansible-Playbook und Terraform-Skript haben am Ende diese VM hervorgebracht. Hierbei wurde automatisch ein Ubuntu installiert und mit SSH-Keys für den Remote-Zugang ausgestattet. Bis zum Ende der Praxisphase und danach werde ich das Ansible-Playbook und Terraform Skript weiter ausbauen und flexibler gestalten. Zum Beispiel möchte ich Windows-VMs möglich machen, welche einen anderen Installationsweg als Linux haben.

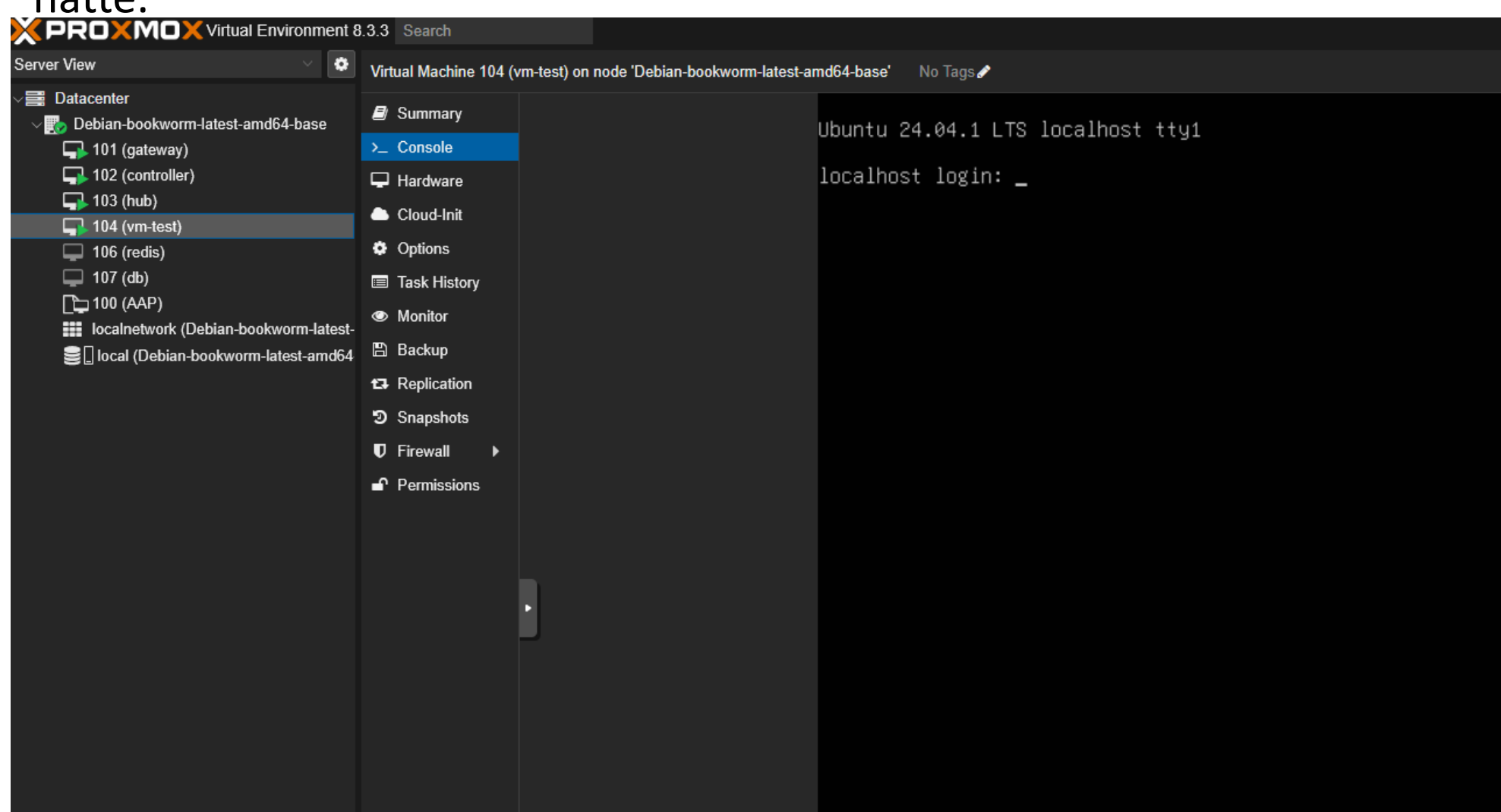


Bild 2: Proxmox-Weboberfläche mit erfolgreich erstellter VM

Während meiner Praxisphase habe ich neben der Konzeption, dem Testen und der Entwicklung weitere Erkenntnisse gewonnen und viel gelernt. Für einen anderen Studenten in meinem Unternehmen habe ich mit alter Hardware eine Testumgebung aufgebaut und diese über ein Smartphone mit dem Internet verbunden. Das Smartphone diente dabei als Modem und leitete seine Internetverbindung an einen der Server weiter. Anschließend haben wir die Internetverbindung mithilfe der Software-Firewall des Servers und NAT an die restlichen Server weitergegeben. Um den externen Zugriff für uns zu ermöglichen, habe ich ein VPN mit der Software „Netbird“ eingerichtet. Diese erstellt ein WireGuard-VPN, über das sowohl die Server als auch wir verbunden waren, wodurch ein eigenständiges Netzwerk entstand.

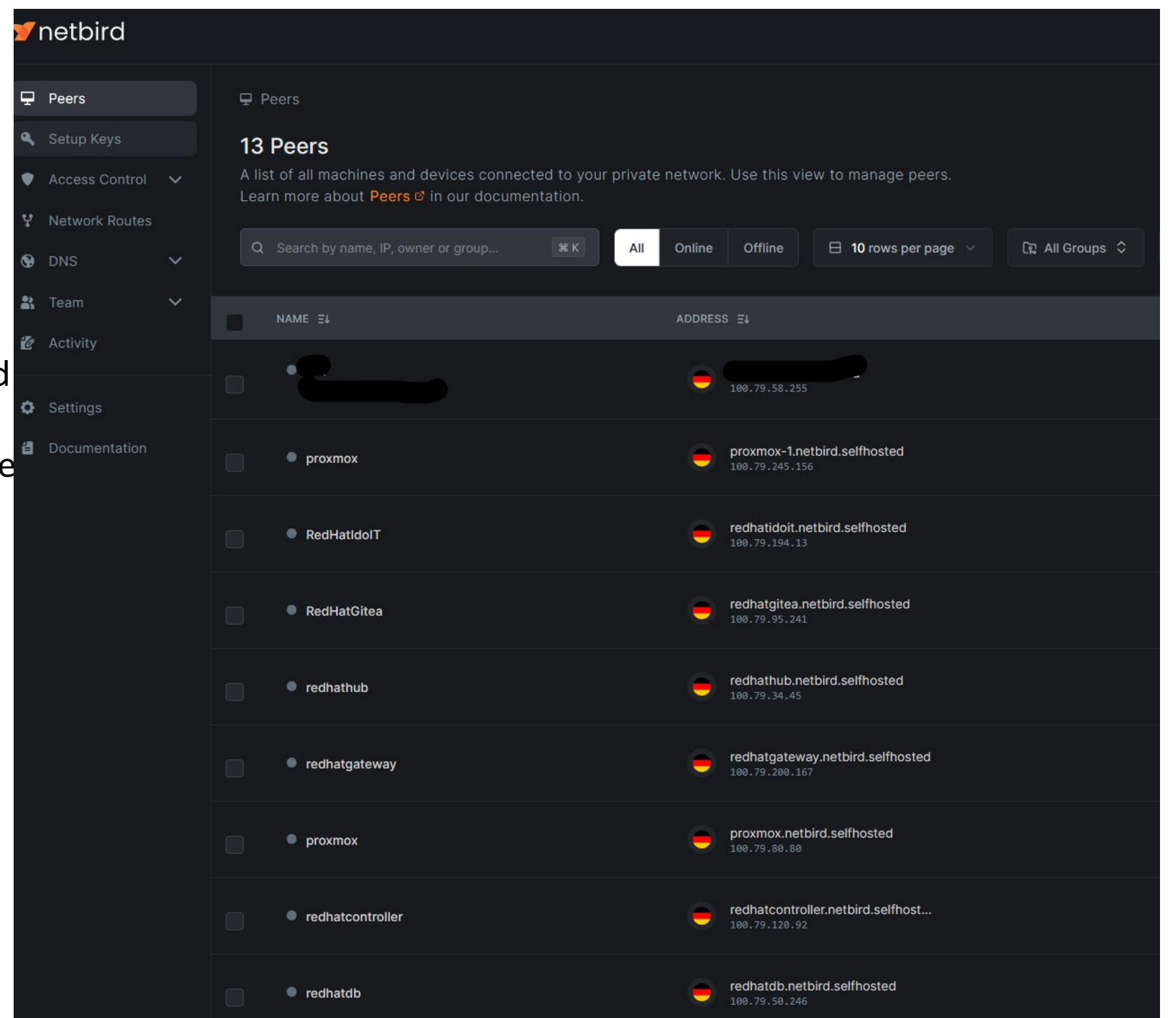


Bild 3: Netbird-VPN Weboberfläche

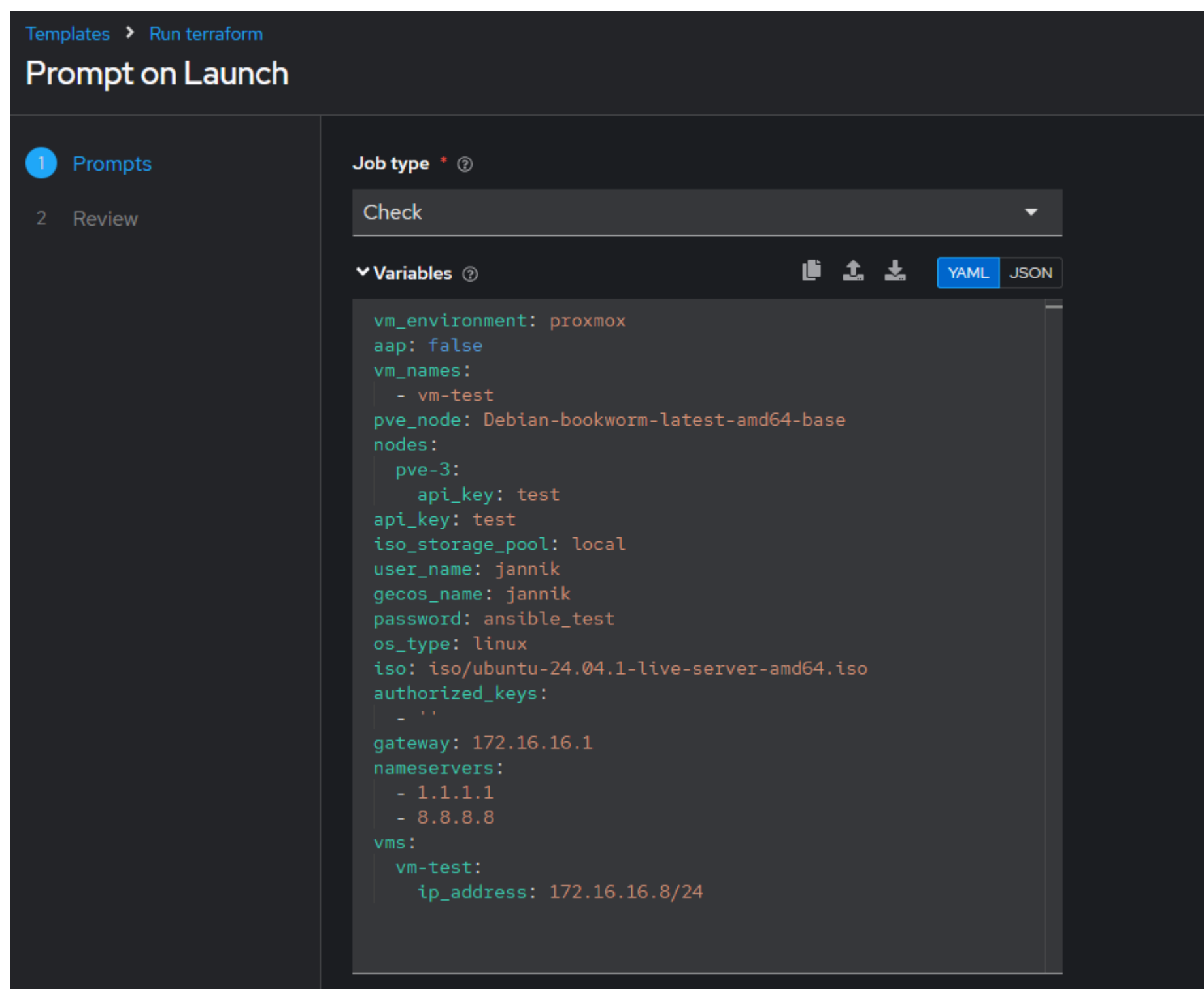


Bild 4: Eingabe der Eingangsvariablen in Ansible Automation Platform



Bild 5: Ansible Playbook: Terraform Ausführung

