

Praxishandbuch Terraform

Infrastructure as Code für DevOps,
Administration und Entwicklung

DAS INHALTS- VERZEICHNIS

» Hier geht's
direkt
zum Buch

Vorwort	9
1 Warum Terraform?	19
Was ist DevOps?	19
Was ist Infrastructure as Code?	21
Ad-hoc-Skripte	22
Konfigurationsmanagementtools	23
Server-Templating-Tools	25
Orchestrierungstools	28
Provisionierungstools	31
Was sind die Vorteile von Infrastructure as Code?	32
Wie arbeitet Terraform?	34
Wie verhält sich Terraform im Vergleich mit anderen IaC-Tools?	36
Konfigurationsmanagement versus Provisionierung	37
Veränderbare Infrastruktur versus immutable Infrastruktur	37
Prozedurale Sprache versus deklarative Sprache	38
Allzwecksprache versus domänenspezifische Sprache	41
Mit oder ohne Master	43
Mit oder ohne Agenten	44
Bezahlt versus kostenlos	46
Große versus kleine Community	47
Ausgereift versus topaktuell	49
Gemeinsamer Einsatz mehrerer Tools	50
Zusammenfassung	52
2 Einstieg in Terraform	55
Ihr AWS-Konto aufsetzen	56
Terraform installieren	59
Einen einzelnen Server deployen	60
Einen einzelnen Webserver deployen	68
Einen konfigurierbaren Webserver deployen	75
Ein Cluster mit Webservern deployen	81

Einen Load Balancer deployen	84
Aufräumen	92
Zusammenfassung	93
3 Wie Sie den Terraform-Status managen	95
Was ist der Terraform-Status?	95
Gemeinsamer Storage für Statusdateien	97
Einschränkungen bei Terraform-Backends.	105
Statusdateien isolieren	107
Isolieren über Workspaces.	108
Isolieren über das Dateilayout	113
Die Data Source terraform_remote_state	118
Zusammenfassung	125
4 Wie man wiederverwendbare Infrastruktur mit Terraform-Modulen erzeugt.	127
Modulgrundlagen	130
Moduleingaben.	132
Lokale Werte in Modulen.	136
Modulausgaben	138
Fallstricke bei Modulen	140
Dateipfade	140
Inline-Blöcke	141
Versionierung von Modulen	144
Zusammenfassung	149
5 Tipps und Tricks zu Terraform: Schleifen, if-Anweisungen, Deployment und Fallstricke	151
Schleifen	152
Schleifen mit dem count-Parameter.	152
Schleifen mit for_each-Ausdrücken	159
Schleifen mit for-Ausdrücken	165
Schleifen bei der String-Direktive.	168
Bedingte Anweisungen	169
Bedingte Anweisungen mit dem count-Parameter.	169
Bedingungen mit for_each und Ausdrücken	174
Bedingte Anweisungen mit der if-String-Direktive	175
Zero-Downtime-Deployment	177
Fallstricke bei Terraform.	185
Einschränkungen bei count und for_each	186
Einschränkungen bei Zero-Downtime-Deployments	187
Gültige Pläne können fehlschlagen	190
Refaktorisieren kann schwierig sein	192
Zusammenfassung	195

6	Secrets mit Terraform managen	197
	Grundlagen des Secret-Managements	198
	Tools für das Secret-Management	199
	Arten von abzulegenden Secrets	199
	Wege, Secrets abzulegen	200
	Schnittstellen für den Zugriff auf Secrets	201
	Vergleich verschiedener Secret-Management-Tools	201
	Secret-Management-Tools in Terraform einsetzen	202
	Provider	202
	Ressourcen und Data Sources	212
	Statusdateien und Plandateien	222
	Zusammenfassung	224
7	Arbeiten mit mehreren Providern	227
	Arbeiten mit einem Provider	227
	Was ist ein Provider?	228
	Wie installieren Sie Provider?	229
	Wie nutzen Sie Provider?	231
	Arbeiten mit mehreren Kopien des gleichen Providers	232
	Mit mehreren AWS-Regionen arbeiten	232
	Mit mehreren AWS-Konten arbeiten	243
	Module erstellen, die mit mehreren Providern arbeiten können ...	250
	Arbeiten mit mehreren verschiedenen Providern	253
	Ein Crashkurs zu Docker	254
	Ein Crashkurs zu Kubernetes	257
	Docker-Container in AWS mit dem Elastic Kubernetes	
	Service deployen	269
	Zusammenfassung	277
8	Produktiv nutzbarer Terraform-Code	279
	Warum dauert es so lange, dem Produktivbetrieb gewachsene	
	Infrastruktur zu bauen?	281
	Die Checkliste für produktiv nutzbare Infrastruktur	283
	Dem Produktivbetrieb gewachsene Infrastrukturmodule	284
	Kleine Module	285
	Zusammensetzbare Module	289
	Testbare Module	294
	Versionierte Module	301
	Module, die über Terraform hinausgehen	308
	Zusammenfassung	316

9	Wie Sie Terraform-Code testen	317
	Manuelle Tests	318
	Grundlagen manueller Tests	319
	Aufräumen nach dem Testen	321
	Automatisierte Tests	322
	Unit Tests	323
	Integrationstests	348
	End-to-End-Tests	362
	Andere Testansätze	364
	Zusammenfassung	372
10	Wie Sie Terraform im Team verwenden	377
	Infrastructure as Code in Ihrem Team übernehmen	377
	Überzeugen Sie Ihren Chef oder Ihre Chefin	378
	Arbeiten Sie inkrementell	381
	Geben Sie Ihrem Team Zeit zum Lernen	383
	Ein Workflow für das Deployen von Anwendungscode	384
	Versionsverwaltung verwenden	384
	Code lokal ausführen	385
	Coding-Änderungen vornehmen	386
	Änderungen zum Reviewen einchecken	386
	Automatisierte Tests ausführen	387
	Mergen und Releasen	388
	Deployen	389
	Ein Workflow für das Deployen von Infrastrukturcode	393
	Versionsverwaltung verwenden	393
	Code lokal ausführen	397
	Coding-Änderungen vornehmen	398
	Änderungen zum Reviewen einchecken	399
	Automatisierte Tests ausführen	401
	Mergen und Releasen	402
	Deployen	403
	Alles zusammenbringen	414
	Zusammenfassung	415
	Anhang Empfehlenswerte Lektüre	417
	Index	421