

Auf einen Blick

1	Einführung und Installation	23
2	Basiseinrichtung und erstes Inventory-Management	41
3	Ad-hoc-Kommandos und Patterns	53
4	Die Konfigurations- und Serialisierungssprache YAML	63
5	Playbooks und Tasks: die Grundlagen	71
6	Playbooks und Tasks: fortgeschrittene Methoden	97
7	Inventory-Management: fortgeschrittene Methoden	147
8	Modularisierung mit Rollen und Includes	167
9	Die Modul-Bibliothek	185
10	Webinterfaces	201
11	Weitere Tools und Techniken	221
12	Ansible als Orchestrierungswerkzeug	251
13	Ansible und Docker	277
14	Windows-Hosts verwalten	303
15	Kochrezepte, Howtos und Best Practices	313
16	Eigene Module erstellen	331
17	Callback-Plugins	345
18	Was könnte noch besser sein, bzw. was fehlt noch?	357

Inhalt

Materialien zum Buch	15
Vorwort	17
Über dieses Buch	19

1 Einführung und Installation 23

1.1 Was ist Ansible?	23
1.2 Was ist Ansible nicht?	25
1.3 Setup/Laborumgebung	26
1.3.1 Zugangsdaten für die Systeme der Laborumgebung	30
1.3.2 Falls Probleme auftreten	30
1.4 Ansible-Installation auf dem Control Host	31
1.5 Authentifizierung und Autorisierung auf den Target Hosts	34
1.6 Einrichten der SSH-Public-Key-Authentifizierung	36
1.7 Ein Ad-hoc-Test ohne jegliche Konfiguration	37
1.8 Noch ein Hinweis zur Migration von älteren Versionen	39

2 Basiseinrichtung und erstes Inventory-Management 41

2.1 Verzeichnisstruktur einrichten	41
2.2 Grundkonfiguration (»ansible.cfg«)	42
2.3 Erstellen und Verwalten eines statischen Inventorys	43
2.4 Inventory-Aliase	46
2.5 Jenseits von Ping	46
2.6 Alternative Inventorys	50

3	Ad-hoc-Kommandos und Patterns	53
3.1	Ad-hoc-Kommandos	53
3.2	Use cases	55
3.3	Idempotenz	56
3.4	Interne Funktionsweise	57
3.4.1	Parallele Ausführung	57
3.4.2	Persistente Verbindungen	58
3.4.3	Was passiert beim Aufruf eines Moduls?	60
3.5	Die Ansible-Konsole	61
3.6	Patterns zum Adressieren von Hosts	62
4	Die Konfigurations- und Serialisierungssprache YAML	63
4.1	Syntax und Struktur	63
4.2	YAML-Files editieren	63
4.3	Listen und Maps	66
4.4	Verschachtelte Strukturen	66
4.5	Block-Ausdrücke	68
4.6	Das Nichts in YAML	69
4.7	anchors und References	69
5	Playbooks und Tasks: die Grundlagen	71
5.1	Hallo Ansible – das allererste Playbook	71
5.2	Formulierung von Tasks	74
5.3	Beenden von Plays	76
5.4	Der problematische Doppelpunkt	77
5.5	Kommandoaufrufe mit den Modulen »command« und »shell«	78
5.6	Fehler und Retry-Files	81
5.7	Tags	84

5.8	Das Kommando »ansible-playbook«	85
5.9	Eine exemplarische Apache-Installation	86
5.9.1	Schritt für Schritt	87
5.9.2	Das komplette Playbook	88
5.9.3	»--start-at-task«, »--check«, »--diff«	89
5.10	Handler: Tasks nur bei Changes durchführen	91
5.10.1	Schritt für Schritt	91
5.10.2	Handler	92
5.10.3	Das komplette Playbook bis hierhin	95

6 Playbooks und Tasks: fortgeschrittene Methoden 97

6.1	Variablen	97
6.1.1	Play Vars	97
6.1.2	Extra Vars	98
6.1.3	Präzedenzen	98
6.1.4	»set_fact«	99
6.1.5	»group_vars«	100
6.1.6	»host_vars«	102
6.1.7	»vars_files«: Variablen in beliebigen externen Dateien	103
6.1.8	Prompting	103
6.1.9	Zugriffe auf komplexe Strukturen	104
6.1.10	Seid ihr alle da?	105
6.2	Facts und implizite Variablen	106
6.2.1	Facts	106
6.2.2	Cachen von Facts	108
6.2.3	Implizite Variablen	109
6.2.4	Ein Beispiel	109
6.2.5	Externe Informationsbeschaffer: »facter« und »ohai«	111
6.2.6	Noch nicht genug Fakten? »/etc/ansible/facts.d«!	111
6.3	Bedingte Ausführung mit »when«	111
6.4	Systemunterschiede ausgleichen – wie denn jetzt?	113
6.4.1	Die plumpe Methode	113
6.4.2	Die solide Methode	114
6.4.3	Die trickreiche Methode	116
6.4.4	Die modulare Methode	116
6.4.5	Das komplette Playbook bis hierhin	118

6.5	Jinja und Templates	120
6.5.1	Begriffsklärung: Templates und Template-Engines	120
6.5.2	Eine individuelle Startseite für unsere Apache-Server	121
6.5.3	Schnelles Testen von Jinja-Templates	123
6.5.4	Jinja-Syntax: Ausgabeausdrücke, Anweisungen, Kommentare	125
6.5.5	Filter	127
6.6	Schleifen	127
6.6.1	Iteration über eine Liste mit »with_items« oder »with_list«	128
6.6.2	Iteration über eine Map mit »with_dict«	130
6.6.3	Iteration über eine generierte Folge mit »with_sequence«	131
6.6.4	Verschachtelte Schleife mit »with_subelements«	132
6.6.5	Tasks wiederholen mit »until«	133
6.6.6	Mehr Kontrolle mit »loop_control«	134
6.7	Fehlerbehandlung mit »failed_when« und »ignore_errors«	137
6.8	Blocks (und noch mal Fehlerbehandlung)	138
6.9	Asynchrone Ausführung	140
6.10	Lokale Tasks	141
6.11	Lookup-Plugins	143
6.12	Umgebungsvariablen setzen	145

7 Inventory-Management: fortgeschrittene Methoden 147

7.1	Das Kommando »ansible-inventory«	147
7.2	Verschachtelte Gruppen	148
7.3	»On-the-fly«-Inventors erstellen mit »add_host«	149
7.4	Dynamische Gruppen mit »group_by«	151
7.5	Dynamische bzw. externe Inventors	153
7.5.1	Beispiel: ein Inventory-Skript in Perl	154
7.5.2	Verwenden von Inventory-Plugins	158
7.5.3	AWS EC2	158
7.5.4	Hetzner Cloud	161
7.5.5	VMware	163
7.5.6	Inventory-Plugins: Caching	164
7.6	Inventory-Verzeichnisse	165

8 Modularisierung mit Rollen und Includes 167

8.1	Erstellung und Verwendung von Rollen	167
8.1.1	Das Rollenkonzept in Ansible	167
8.1.2	Ein einfaches Beispiel einer Rolle	168
8.1.3	Rollen verwenden	169
8.1.4	Abhängigkeiten zwischen Rollen	171
8.1.5	Erstellen neuer Rollen	172
8.2	Ansible Galaxy	173
8.3	Verwendung von Imports/Includes	174
8.4	Noch mal Apache	175
8.5	Dokumentation	179
8.5.1	Code-Konventionen	180
8.5.2	»README.md«	181

9 Die Modul-Bibliothek 185

9.1	Module zur Kommandoausführung	185
9.2	Module zur Paketverwaltung	187
9.3	Module zur Verwaltung von Dateien und Dateiinhalten	188
9.4	Module für weitere typische Verwaltungsaufgaben	193
9.5	Module zur Interaktion mit Netzwerk-Services	195
9.6	Spezialmodule (Kontrollflusssteuerung etc.)	196

10 Webinterfaces 201

10.1	Vorbereitungen zum Betrieb	201
10.2	Ansible Configuration Management Database (ansible-cmdb)	203
10.3	Voraussetzungen für anspruchsvollere Anwendungen	205
10.3.1	Der Reverse-Proxy Traefik	206
10.3.2	Der Git-Server Gitea	207
10.3.3	Einchecken unseres Projekts	208

10.4	Ansible Tower/AWX	210
10.4.1	Test-Setup und erste Anmeldung	211
10.4.2	Exemplarische Verwendung	212
10.5	Polemarch	213
10.5.1	Test-Setup und erste Anmeldung	213
10.5.2	Exemplarische Verwendung	214
10.5.3	(Vorsichtiges) Fazit	215
10.6	Jenkins	215
10.6.1	Test-Setup	215
10.6.2	Exemplarische Verwendung	216
10.6.3	Fazit	217
10.7	Weitere, hier nicht näher betrachtete Möglichkeiten	218
10.8	Laborumgebung: nicht mehr benötigte Anwendungen beenden	218
11	Weitere Tools und Techniken	221
11.1	Die Ansible-Vault	221
11.1.1	Vor aller Technik	221
11.1.2	Erste Schritte	223
11.1.3	Mehrere Vault-Passwörter und weitere Vault-Kommandos	225
11.1.4	Ein Trick zum Wiederfinden von Variablen	225
11.1.5	Mehr Bequemlichkeit bzw. Automatisierbarkeit	226
11.1.6	Bequem und (möglichst) sicher mit GPG + pass	227
11.2	Debugging und Troubleshooting	230
11.2.1	Debug-Mode und Verbosity Level	230
11.2.2	Lesbarkeit von Ausgaben verbessern	231
11.2.3	Gathering Facts dauert zu lange	233
11.2.4	Der Playbook-Debugger	235
11.2.5	Statische Code-Analyse mit »ansible-lint«	237
11.2.6	Überwachen von Dateiänderungen mit »--check« und »--diff«	240
11.2.7	Last, but not least: das »debug«-Modul	242
11.3	Untersuchen von Konfigurationseinstellungen	244
11.4	Playbooks beschleunigen mit Pipelining	244
11.5	Die sprechende Kuh	245

11.6	Ansible im Pull-Mode	246
11.6.1	»ansible-pull«: Technik und Voraussetzungen	247
11.6.2	Erste Schritte	248
11.6.3	Die ganze Lösung	249
11.6.4	Was fehlt noch?	250

12 Ansible als Orchestrierungswerkzeug 251

12.1	Administrierst du noch oder orchestrierst du schon?	251
12.2	Ausflug in die Cloud: Vorbereitungen	252
12.3	Erste Server in der Cloud	254
12.4	Die Abarbeitungsreihenfolge beeinflussen	256
12.4.1	»serial«	257
12.4.2	Fehlerhafte Hosts im »serial«-Betrieb	258
12.4.3	Strategy-Plugins	259
12.5	Delegierung	264
12.6	Ein komplexeres Beispiel	267
12.7	Löschen von Workern	274
12.8	Rolling-Updates	275

13 Ansible und Docker 277

13.1	Installation von Docker	277
13.2	Docker-Module	279
13.2.1	Vorbereitungen und Vorüberlegungen	279
13.2.2	Überblick	281
13.3	Eine Beispielanwendung	286
13.4	Ansible und Docker Compose	290
13.5	Erstellen von Images	295
13.5.1	Erstellen von Images mit »docker build«	295
13.5.2	»ansible-bender«	296
13.5.3	Erstellen von Images mit »ansible-bender«	298
13.5.4	Erklärungen und Fazit	301

14	Windows-Hosts verwalten	303
14.1	WinRM	303
14.2	Vorbereitungen auf dem Control Host	304
14.3	Voraussetzungen auf der Windows-Seite und WinRM-Setup	305
14.4	Setup mit Active Directory/Kerberos	306
14.5	WinRM-Troubleshooting	308
14.6	Windows-Module	309
15	Kochrezepte, Howtos und Best Practices	313
15.1	Eine empfehlenswerte »ansible.cfg«	313
15.2	Suche in einer Liste von Maps	314
15.3	Erweiterung von Maps oder Listen während der Laufzeit	315
15.4	Zufallspasswörter generieren	316
15.5	Einfache Installer bauen	318
15.6	IP-Adresse eines Target Hosts bestimmen	320
15.7	firewalld managen (falls vorhanden)	323
15.8	Linux-Software-Updates einspielen	324
15.9	Ansible über einen Gateway- bzw. Jump host	327
15.10	Host-spezifische Ressourcen verwalten	328
16	Eigene Module erstellen	331
16.1	Erste Schritte	331
16.2	Modul-Parameter	334
16.3	Module in Python	337
16.3.1	Ein Beispiel in Python	338
16.3.2	Erklärungen und weitere Möglichkeiten	340
16.3.3	Eingebettete Dokumentation	343

17 Callback-Plugins 345

17.1	Stdout-Plugins	345
17.1.1	»default«	346
17.1.2	»yaml«	347
17.1.3	»json«	348
17.1.4	»unixy«	348
17.1.5	»dense«	349
17.1.6	»minimal«	349
17.1.7	»online«	350
17.1.8	»debug«	350
17.1.9	»selective«	350
17.1.10	»counter_enabled«	351
17.2	Sonstige Plugins	351
17.3	Entwickeln eigener Callback-Plugins	353

18 Was könnte noch besser sein, bzw. was fehlt noch? 357

18.1	Lang laufende Tasks verfolgen	357
18.2	Abarbeitung einer Rolle beenden	358
18.3	Schleifen über Blöcke	360
18.4	Locking bei konkurrierenden Playbook-Aufrufen	361
18.5	Fazit	363

Anhang 365

A	SSH (Secure Shell)	367
A.1	Voraussetzungen zur Nutzung von SSH	367
A.2	SSH-Client-Programme	368
A.3	Public-Key-Authentifizierung	369
A.4	SSH-Agenten	372

A.5	Fortgeschrittene Konfiguration und Nutzung	373
A.6	Windows als SSH-Client	379
A.7	pssh bzw. parallel-ssh	380
B	Reguläre Ausdrücke	383
B.1	Motivation	383
B.2	Dialekte	384
B.3	Reguläre Ausdrücke – Basics	384
B.4	Reguläre Ausdrücke – erweiterte Möglichkeiten (PCRE)	386
B.5	Reguläre Ausdrücke – erweiterte Möglichkeiten (POSIX Extended RE)	388
Index		391