
Table of Contents

Preface to the Third Edition.....	xvii
1. Introduction.....	1
A Note About Versions	2
Ansible: What Is It Good For?	2
How Ansible Works	4
What’s So Great About Ansible?	5
Simple	5
Powerful	8
Secure	11
Is Ansible Too Simple?	13
What Do I Need to Know?	13
What Isn’t Covered	14
Moving Forward	14
2. Installation and Setup.....	15
Installing Ansible	15
Loose Dependencies	16
Running Ansible in Containers	16
Ansible Development	17
Setting Up a Server for Testing	17
Using Vagrant to Set Up a Test Server	17
Telling Ansible About Your Servers	20
Simplifying with the ansible.cfg File	22
Kill Your Darlings	25
Convenient Vagrant Configuration Options	25
Port Forwarding and Private IP Addresses	25

Enabling Agent Forwarding	27
The Docker Provisioner	28
The Ansible Local Provisioner	28
When the Provisioner Runs	29
Vagrant Plug-ins	29
Hostmanager	29
VBGuest	30
VirtualBox Customization	30
Vagrantfile Is Ruby	30
Production Setup	33
Conclusion	34
3. Playbooks: A Beginning	35
Preliminaries	35
A Very Simple Playbook	36
Specifying an NGINX Config File	37
Creating a Web Page	38
Creating a Group	38
Running the Playbook	39
Playbooks Are YAML	41
Start of Document	41
End of File	41
Comments	41
Indentation and Whitespace	41
Strings	42
Booleans	42
Lists	43
Dictionaries	44
Multiline Strings	44
Pure YAML Instead of String Arguments	45
Anatomy of a Playbook	45
Plays	46
Tasks	47
Modules	48
Viewing Ansible Module Documentation	49
Putting It All Together	49
Did Anything Change? Tracking Host State	49
Getting Fancier: TLS Support	50
Generating a TLS Certificate	51
Variables	51
Quoting in Ansible Strings	52

Generating the NGINX Configuration Template	53
Loop	54
Handlers	55
A Few Things to Keep in Mind About Handlers	55
Testing	56
Validation	56
The Playbook	57
Running the Playbook	58
Conclusion	60
4. Inventory: Describing Your Servers.....	61
Inventory/Hosts Files	62
Preliminaries: Multiple Vagrant Machines	62
Behavioral Inventory Parameters	65
Changing Behavioral Parameter Defaults	66
Groups and Groups and Groups	67
Example: Deploying a Django App	68
Aliases and Ports	71
Groups of Groups	71
Numbered Hosts (Pets Versus Cattle)	72
Hosts and Group Variables: Inside the Inventory	72
Host and Group Variables: In Their Own Files	74
Dynamic Inventory	76
Inventory Plug-ins	77
Amazon EC2	77
Azure Resource Manager	77
The Interface for a Dynamic Inventory Script	77
Writing a Dynamic Inventory Script	79
Breaking the Inventory into Multiple Files	82
Adding Entries at Runtime with add_host and group_by	83
add_host	83
group_by	85
Conclusion	86
5. Variables and Facts.....	87
Defining Variables in Playbooks	87
Defining Variables in Separate Files	87
Directory Layout	88
Viewing the Values of Variables	88
Variable Interpolation	88
Registering Variables	89

Facts	93
Viewing All Facts Associated with a Server	94
Viewing a Subset of Facts	95
Any Module Can Return Facts or Info	95
Local Facts	96
Using set_fact to Define a New Variable	97
Built-In Variables	98
hostvars	98
inventory_hostname	99
groups	99
Extra Variables on the Command Line	100
Precedence	102
Conclusion	103
6. Introducing Mezzanine: Our Test Application.....	105
Why Is Deploying to Production Complicated?	105
Postgres: The Database	108
Gunicorn: The Application Server	108
NGINX: The Web Server	109
Supervisor: The Process Manager	110
Conclusion	110
7. Deploying Mezzanine with Ansible.....	111
Listing Tasks in a Playbook	111
Organization of Deployed Files	112
Variables and Secret Variables	113
Installing Multiple Packages	114
Adding the Become Clause to a Task	115
Updating the apt Cache	115
Checking Out the Project Using Git	117
Installing Mezzanine and Other Packages into a Virtual Environment	118
Complex Arguments in Tasks: A Brief Digression	121
Configuring the Database	123
Generating the local_settings.py File from a Template	124
Running django-manage Commands	127
Running Custom Python Scripts in the Context of the Application	128
Setting Service Configuration Files	130
Enabling the NGINX Configuration	133
Installing TLS Certificates	134
Installing Twitter Cron Job	135
The Full Playbook	136

Running the Playbook Against a Vagrant Machine	140
Troubleshooting	141
Cannot Check Out Git Repository	141
Cannot Reach 192.168.33.10.nip.io	142
Bad Request (400)	142
Conclusion	142
8. Debugging Ansible Playbooks.....	143
Humane Error Messages	143
Debugging SSH Issues	144
Common SSH Challenges	147
PasswordAuthentication no	147
SSH as a Different User	148
Host Key Verification Failed	148
Private Networks	149
The debug Module	149
Playbook Debugger	150
The assert Module	152
Checking Your Playbook Before Execution	153
Syntax Check	154
List Hosts	154
List Tasks	154
Check Mode	155
Diff (Show File Changes)	155
Tags	156
Limits	156
Conclusion	157
9. Roles: Scaling Up Your Playbooks.....	159
Basic Structure of a Role	159
Example: Deploying Mezzanine with Roles	161
Using Roles in Your Playbooks	161
Pre-Tasks and Post-Tasks	162
A database Role for Deploying the Database	163
A mezzanine Role for Deploying Mezzanine	166
Creating Role Files and Directories with ansible-galaxy	171
Dependent Roles	172
Ansible Galaxy	173
Web Interface	173
Command-Line Interface	173
Role Requirements in Practice	174

Contributing Your Own Role	175
Conclusion	175
10. Complex Playbooks.....	177
Dealing with Badly Behaved Commands	177
Filters	181
The default Filter	181
Filters for Registered Variables	181
Filters That Apply to Filepaths	182
Writing Your Own Filter	183
Lookups	184
file	186
pipe	187
env	187
password	187
template	187
csvfile	188
dig	189
redis	190
Writing Your Own Lookup Plug-in	191
More Complicated Loops	191
With Lookup Plug-in	192
with_lines	193
with_fileglob	193
with_dict	194
Looping Constructs as Lookup Plug-ins	195
Loop Controls	195
Setting the Variable Name	195
Labeling the Output	196
Imports and Includes	198
Dynamic Includes	199
Role Includes	200
Role Flow Control	200
Blocks	201
Error Handling with Blocks	201
Encrypting Sensitive Data with ansible-vault	205
Multiple Vaults with Different Passwords	207
Conclusion	207
11. Customizing Hosts, Runs, and Handlers.....	209
Patterns for Specifying Hosts	209

Limiting Which Hosts Run	210
Running a Task on the Control Machine	210
Manually Gathering Facts	211
Retrieving an IP Address from the Host	211
Running on One Host at a Time	213
Running on a Batch of Hosts at a Time	215
Running Only Once	216
Limiting Which Tasks Run	216
step	216
start-at-task	216
Running Tags	216
Skipping Tags	218
Running Strategies	218
Linear	218
Free	219
Advanced Handlers	221
Handlers in Pre- and Post-Tasks	221
Flush Handlers	223
Meta Commands	223
Handlers Notifying Handlers	224
Handlers Listen	224
The SSL Case for the listen Feature	225
Conclusion	231
12. Managing Windows Hosts.....	233
Connection to Windows	233
PowerShell	234
Windows Modules	237
Our Java Development Machine	237
Adding a Local User	239
Windows Features	239
Installing Software with Chocolatey	240
Configuration of Java	241
Updating Windows	241
Conclusion	242
13. Ansible and Containers.....	243
Kubernetes	244
Docker Application Life Cycle	245
Registries	245
Ansible and Docker	246

Connecting to the Docker Daemon	246
Example Application: Ghost	246
Running a Docker Container on Our Local Machine	247
Building an Image from a Dockerfile	247
Pushing Our Image to the Docker Registry	249
Orchestrating Multiple Containers on Our Local Machine	250
Querying Local Images	252
Deploying the Dockerized Application	253
Provisioning MySQL	254
Deploying the Ghost Database	254
Frontend	255
Frontend: Ghost	256
Frontend: NGINX	257
Cleaning Out Containers	258
Conclusion	258
14. Quality Assurance with Molecule.....	259
Installation and Setup	259
Configuring Molecule Drivers	260
Creating an Ansible Role	261
Scenarios	262
Desired State	262
Configuring Scenarios in Molecule	262
Managing Virtual Machines	263
Managing Containers	264
Molecule Commands	266
Linting	266
YAMLint	267
ansible-lint	267
ansible-later	268
Verifiers	269
Ansible	269
Goss	269
TestInfra	271
Conclusion	272
15. Collections.....	273
Installing Collections	274
Listing Collections	275
Using Collections in a Playbook	275
Developing a Collection	276

Conclusion	278
16. Creating Images.....	279
Creating Images with Packer	279
Vagrant VirtualBox VM	279
Combining Packer and Vagrant	282
Cloud Images	283
Google Cloud Platform	283
Azure	285
Amazon EC2	287
The Playbook	288
Docker Image: GCC 11	288
Conclusion	291
17. Cloud Infrastructure.....	293
Terminology	297
Instance	297
Amazon Machine Image	297
Tags	297
Specifying Credentials	298
Environment Variables	298
Configuration Files	299
Prerequisite: Boto3 Python Library	300
Dynamic Inventory	301
Inventory Caching	302
Other Configuration Options	303
Defining Dynamic Groups with Tags	303
Applying Tags to Existing Resources	303
Nicer Group Names	304
Virtual Private Clouds	305
Configuring ansible.cfg for Use with ec2	305
Launching New Instances	306
EC2 Key Pairs	307
Creating a New Key	308
Uploading Your Public Key	308
Security Groups	308
Permitted IP Addresses	309
Security Group Ports	310
Getting the Latest AMI	310
Create a New Instance and Add It to a Group	312
Waiting for the Server to Come Up	313

Putting It All Together	314
Specifying a Virtual Private Cloud	316
Dynamic Inventory and VPC	319
Conclusion	319
18. Callback Plug-ins.....	321
Stdout Plug-ins	321
ARA	322
debug	323
default	323
dense	323
json	324
minimal	324
null	324
online	324
Notification and Aggregate Plug-ins	324
Python Requirements	325
foreman	325
jabber	326
junit	326
log_plays	327
logentries	327
logstash	327
mail	328
profile_roles	328
profile_tasks	328
say	329
slack	329
splunk	329
timer	330
Conclusion	330
19. Custom Modules.....	331
Example: Checking That You Can Reach a Remote Server	332
Using the Script Module Instead of Writing Your Own	332
can_reach as a Module	333
Should You Develop a Module?	334
Where to Put Your Custom Modules	334
How Ansible Invokes Modules	334
Generate a Standalone Python Script with the Arguments (Python Only)	334
Copy the Module to the Host	335

Create an Arguments File on the Host (Non-Python Only)	335
Invoke the Module	335
Expected Outputs	336
Output Variables That Ansible Expects	336
Implementing Modules in Python	337
Parsing Arguments	339
Accessing Parameters	339
Importing the AnsibleModule Helper Class	340
Argument Options	340
AnsibleModule Initializer Parameters	343
Returning Success or Failure	346
Invoking External Commands	347
Check Mode (Dry Run)	348
Documenting Your Module	349
Debugging Your Module	350
Implementing the Module in Bash	352
Specifying an Alternative Location for Bash	353
Conclusion	354
20. Making Ansible Go Even Faster.....	355
SSH Multiplexing and ControlPersist	355
Manually Enabling SSH Multiplexing	356
SSH Multiplexing Options in Ansible	358
More SSH Tuning	359
Algorithm Recommendations	359
Pipelining	361
Enabling Pipelining	361
Configuring Hosts for Pipelining	361
Mitogen for Ansible	363
Fact Caching	364
JSON File Fact-Caching Backend	366
Redis Fact-Caching Backend	366
Memcached Fact-Caching Backend	367
Parallelism	368
Concurrent Tasks with Async	368
Conclusion	370
21. Networking and Security.....	371
Network Management	371
Supported Vendors	372
Ansible Connection for Network Automation	372

Privileged Mode	373
Network Inventory	374
Network Automation Use Cases	374
Security	375
Comply with Compliance?	376
Secured, but Not Secure	376
Shadow IT	379
Sunshine IT	380
Zero Trust	380
Conclusion	381
22. CI/CD and Ansible.....	383
Continuous Integration	383
Elements in a CI System	384
Jenkins and Ansible	389
Running CI for Ansible Roles	392
Staging	394
Ansible Plug-in	395
Ansible Tower Plug-in	397
Conclusion	398
23. Ansible Automation Platform.....	399
Subscription Models	402
Ansible Automation Platform Trial	403
What Ansible Automation Platform Solves	404
Access Control	404
Projects	405
Inventory Management	406
Run Jobs by Job Templates	408
RESTful API	410
AWX.AWX	412
Installation	412
Create an Organization	413
Create an Inventory	413
Running a Playbook with a Job Template	414
Using Containers to Run Ansible	416
Creating Execution Environments	416
Conclusion	417
24. Best Practices.....	419
Simplicity, Modularity, and Composability	419

Organize Content	420
Decouple Inventories from Projects	420
Decouple Roles and Collections	420
Playbooks	421
Code Style	421
Tag and Test All the Things	421
Desired State	421
Deliver Continuously	422
Security	422
Deployment	422
Performance Indicators	423
Benchmark Evidence	423
Final Words	424
Bibliography.....	425
Index.....	427

