
Table of Contents

Foreword.....	xv
Preface to the Second Edition.....	xvii
Preface to the First Edition.....	xix
1. Introduction.....	1
A Note About Versions	2
Ansible: What Is It Good For?	2
How Ansible Works	3
What's So Great About Ansible?	5
Easy-to-Read Syntax	5
Nothing to Install on the Remote Hosts	5
Push Based	5
Ansible Scales Down	6
Built-in Modules	6
Very Thin Layer of Abstraction	7
Is Ansible Too Simple?	8
What Do I Need to Know?	9
What Isn't Covered	10
Installing Ansible	10
Setting Up a Server for Testing	11
Using Vagrant to Set Up a Test Server	11
Telling Ansible About Your Test Server	14
Simplifying with the ansible.cfg File	16
Moving Forward	19

2. Playbooks: A Beginning.....	21
Some Preliminaries	21
A Very Simple Playbook	22
Specifying an Nginx Config File	24
Creating a Custom Home Page	25
Creating a Webservers Group	26
Running the Playbook	26
Playbooks Are YAML	28
Start of File	28
Comments	28
Strings	28
Booleans	28
Lists	29
Dictionaries	29
Line Folding	30
Anatomy of a Playbook	30
Plays	32
Tasks	33
Modules	34
Putting It All Together	35
Did Anything Change? Tracking Host State	35
Getting Fancier: TLS Support	36
Generating a TLS Certificate	37
Variables	37
Generating the Nginx Configuration Template	39
Handlers	41
Running the Playbook	42
3. Inventory: Describing Your Servers.....	45
The Inventory File	45
Preliminaries: Multiple Vagrant Machines	46
Behavioral Inventory Parameters	49
ansible_connection	49
ansible_shell_type	50
ansible_python_interpreter	50
ansible_*_interpreter	50
Changing Behavioral Parameter Defaults	50
Groups and Groups and Groups	51
Example: Deploying a Django App	52
Aliases and Ports	54
Groups of Groups	55
Numbered Hosts (Pets versus Cattle)	55

Hosts and Group Variables: Inside the Inventory	56
Host and Group Variables: In Their Own Files	57
Dynamic Inventory	59
The Interface for a Dynamic Inventory Script	60
Writing a Dynamic Inventory Script	61
Preexisting Inventory Scripts	64
Breaking the Inventory into Multiple Files	65
Adding Entries at Runtime with <code>add_host</code> and <code>group_by</code>	65
<code>add_host</code>	65
<code>group_by</code>	67
4. Variables and Facts.....	69
Defining Variables in Playbooks	69
Viewing the Values of Variables	70
Registering Variables	70
Facts	74
Viewing All Facts Associated with a Server	75
Viewing a Subset of Facts	75
Any Module Can Return Facts	76
Local Facts	77
Using <code>set_fact</code> to Define a New Variable	78
Built-in Variables	79
<code>hostvars</code>	79
<code>inventory_hostname</code>	80
Groups	80
Setting Variables on the Command Line	81
Precedence	82
5. Introducing Mezzanine: Our Test Application.....	85
Why Deploying to Production Is Complicated	85
PostgreSQL: The Database	89
Gunicorn: The Application Server	89
Nginx: The Web Server	90
Supervisor: The Process Manager	91
6. Deploying Mezzanine with Ansible.....	93
Listing Tasks in a Playbook	93
Organization of Deployed Files	94
Variables and Secret Variables	95
Using Iteration (with <code>items</code>) to Install Multiple Packages	96
Adding the <code>Become</code> Clause to a Task	98
Updating the Apt Cache	98

Checking Out the Project by Using Git	100
Installing Mezzanine and Other Packages into a virtualenv	102
Complex Arguments in Tasks: A Brief Digression	105
Configuring the Database	107
Generating the local_settings.py File from a Template	108
Running django-manage Commands	111
Running Custom Python Scripts in the Context of the Application	112
Setting Service Configuration Files	115
Enabling the Nginx Configuration	118
Installing TLS Certificates	118
Installing Twitter Cron Job	119
The Full Playbook	120
Running the Playbook Against a Vagrant Machine	124
Troubleshooting	124
Cannot Check Out Git Repository	124
Cannot Reach 192.168.33.10.xip.io	125
Bad Request (400)	125
Deploying Mezzanine on Multiple Machines	125
7. Roles: Scaling Up Your Playbooks.	127
Basic Structure of a Role	127
Example: Database and Mezzanine Roles	128
Using Roles in Your Playbooks	128
Pre-Tasks and Post-Tasks	130
A database Role for Deploying the Database	130
A mezzanine Role for Deploying Mezzanine	133
Creating Role Files and Directories with ansible-galaxy	138
Dependent Roles	138
Ansible Galaxy	139
Web Interface	140
Command-Line Interface	140
Contributing Your Own Role	141
8. Complex Playbooks.	143
Dealing with Badly Behaved Commands: changed_when and failed_when	143
Filters	147
The Default Filter	147
Filters for Registered Variables	147
Filters That Apply to File Paths	148
Writing Your Own Filter	149
Lookups	150
file	151

pipe	152
env	152
password	152
template	153
csvfile	153
dnstxt	154
redis_kv	155
etcd	156
Writing Your Own Lookup Plugin	157
More Complicated Loops	157
with_lines	158
with_fileglob	158
with_dict	158
Looping Constructs as Lookup Plugins	159
Loop Controls	160
Setting the Variable Name	160
Labeling the Output	161
Includes	162
Dynamic Includes	164
Role Includes	164
Blocks	165
Error Handling with Blocks	166
Encrypting Sensitive Data with Vault	169
9. Customizing Hosts, Runs, and Handlers.	171
Patterns for Specifying Hosts	171
Limiting Which Hosts Run	172
Running a Task on the Control Machine	172
Running a Task on a Machine Other Than the Host	173
Running on One Host at a Time	174
Running on a Batch of Hosts at a Time	175
Running Only Once	176
Running Strategies	176
Linear	177
Free	178
Advanced Handlers	180
Handlers in Pre and Post Tasks	180
Flush Handlers	181
Handlers Listen	183
Manually Gathering Facts	189
Retrieving the IP Address from the Host	190

10. Callback Plugins.....	193
Stdout Plugins	193
actionable	194
debug	194
dense	195
json	195
minimal	196
online	196
selective	197
skippy	197
Other Plugins	197
foreman	198
hipchat	198
jabber	198
junit	199
log_plays	200
logentries	200
logstash	200
mail	201
osx_say	201
profile_tasks	201
slack	202
timer	202
11. Making Ansible Go Even Faster.....	203
SSH Multiplexing and ControlPersist	203
Manually Enabling SSH Multiplexing	204
SSH Multiplexing Options in Ansible	205
Pipelining	207
Enabling Pipelining	207
Configuring Hosts for Pipelining	207
Fact Caching	209
JSON File Fact-Caching Backend	210
Redis Fact-Caching Backend	211
Memcached Fact-Caching Backend	212
Parallelism	212
Concurrent Tasks with Async	213
12. Custom Modules.....	215
Example: Checking That We Can Reach a Remote Server	215
Using the Script Module Instead of Writing Your Own	216
can_reach as a Module	216

Where to Put Custom Modules	217
How Ansible Invokes Modules	217
Generate a Standalone Python Script with the Arguments (Python Only)	217
Copy the Module to the Host	217
Create an Arguments File on the Host (Non-Python Only)	217
Invoke the Module	218
Expected Outputs	218
Output Variables that Ansible Expects	219
Implementing Modules in Python	220
Parsing Arguments	221
Accessing Parameters	222
Importing the AnsibleModule Helper Class	222
Argument Options	223
AnsibleModule Initializer Parameters	225
Returning Success or Failure	229
Invoking External Commands	229
Check Mode (Dry Run)	230
Documenting Your Module	231
Debugging Your Module	233
Implementing the Module in Bash	234
Specifying an Alternative Location for Bash	235
Example Modules	236
13. Vagrant	237
Convenient Vagrant Configuration Options	237
Port Forwarding and Private IP Addresses	238
Enabling Agent Forwarding	239
The Ansible Provisioner	239
When the Provisioner Runs	240
Inventory Generated by Vagrant	240
Provisioning in Parallel	241
Specifying Groups	242
Ansible Local Provisioner	243
14. Amazon EC2	245
Terminology	247
Instance	247
Amazon Machine Image	247
Tags	248
Specifying Credentials	248
Environment Variables	249
Configuration Files	249

Prerequisite: Boto Python Library	250
Dynamic Inventory	250
Inventory Caching	252
Other Configuration Options	253
Autogenerated Groups	253
Defining Dynamic Groups with Tags	254
Applying Tags to Existing Resources	254
Nicer Group Names	255
EC2 Virtual Private Cloud and EC2 Classic	256
Configuring ansible.cfg for Use with ec2	257
Launching New Instances	257
EC2 Key Pairs	259
Creating a New Key	259
Uploading an Existing Key	260
Security Groups	261
Permitted IP Addresses	262
Security Group Ports	263
Getting the Latest AMI	263
Adding a New Instance to a Group	264
Waiting for the Server to Come Up	266
Creating Instances the Idempotent Way	268
Putting It All Together	268
Specifying a Virtual Private Cloud	270
Dynamic Inventory and VPC	275
Building AMIs	275
With the ec2_ami Module	276
With Packer	276
Other Modules	280
15. Docker.....	281
The Case for Pairing Docker with Ansible	282
Docker Application Life Cycle	283
Example Application: Ghost	284
Connecting to the Docker Daemon	284
Running a Container on Our Local Machine	285
Building an Image from a Dockerfile	285
Orchestrating Multiple Containers on Our Local Machine	287
Pushing Our Image to the Docker Registry	288
Querying Local Images	290
Deploying the Dockerized Application	292
Backend: Postgres	292
Frontend	293

Frontend: Ghost	294
Frontend: Nginx	294
Cleaning Out Containers	295
Connecting Directly to Containers	295
Ansible Container	297
The Conductor	297
Creating Docker Images	298
Running Locally	301
Publishing Images to Registries	303
Deploying Containers to Production	304
16. Debugging Ansible Playbooks.....	305
Humane Error Messages	305
Debugging SSH Issues	306
The Debug Module	308
Playbook Debugger	308
The Assert Module	310
Checking Your Playbook Before Execution	312
Syntax Check	312
List Hosts	312
List Tasks	312
Check Mode	313
Diff (Show File Changes)	313
Limiting Which Tasks Run	314
Step	314
Start-at-Task	314
Tags	314
17. Managing Windows Hosts.....	317
Connection to Windows	317
PowerShell	318
Windows Modules	321
Our First Playbook	321
Updating Windows	322
Adding Local Users	324
Conclusion	327
18. Ansible for Network Devices.....	329
Status of Network Modules	330
List of Network Vendors Supported	330
Preparing Our Network Device	330
Enable SSH Authentication	331

How the Modules Work	334
Our First Playbook	334
Inventory and Variables for Network Modules	336
Local Connection	337
Host Connection	337
Authentication Variables	338
Save the Config	339
Use Configs from a File	340
Templates, Templates, Templates	344
Gathering Facts	346
Conclusion	347
19. Ansible Tower: Ansible for the Enterprise.....	349
Subscription Models	350
Try Ansible Tower	350
What Ansible Tower Solves	351
Access Control	351
Projects	352
Inventory Management	353
Run Jobs by Job Templates	355
RESTful API	357
Ansible Tower CLI	358
Installation	358
Create a User	359
Launch a Job	361
Onward	362
A. SSH.....	363
B. Using IAM Roles for EC2 Credentials.....	373
Glossary.....	377
Bibliography.....	381
Index.....	383