



SALTSTACK

Kosher Salt - ST_142

Release 1.4.5

Salt Stack

May 21, 2014

CONTENTS

1	Intro	1
1.1	Kosher Salt	1
1.2	What Are We Going to Cover?	2
1.3	Remote Execution	2
1.4	Salt Cloud	2
1.5	Configuration Management	2
1.6	Extending Salt	2
1.7	Course Format	3
1.8	Introductions	3
1.9	Class Start	3
1.10	Emergencies	3
1.11	Emergency Exits and Restrooms	3
1.12	Lectures, Breaks and Lunches	3
1.13	Lab Environment	4
1.14	Lab Goals	4
1.15	What is Salt?	5
1.16	Not Just Configuration Management	5
1.17	Remote Execution	5
1.18	Open Event System	5
1.19	Layered Configuration Management	5
1.20	Simplicity	5
1.21	Manual Conventions	6
1.22	Manual Conventions	6
2	Diving In	7
2.1	Diving Into Salt - Objectives	7
2.2	Getting The Code	8
2.3	The Code	8
2.4	Installing Salt	9
2.5	Split Packages	9
2.6	Salt Bootstrap	9
2.7	Firewalling Salt	10
2.8	Salt and the Firewall	10
2.9	Basic Master Configuration	11
2.10	Basic Server Configuration	11
2.11	Basic Minion Configuration	13
2.12	Find the Salt Master	13
2.13	Security Settings	13
2.14	Starting the Daemons	14
2.15	Start the Salt	14

2.16	Run Directly	14
2.17	Authentication	15
2.18	RSA Keys	15
2.19	Master Holds Minion Public Keys	15
2.20	Minion Validating the Master	15
2.21	Managing Keys With Salt-Key	16
2.22	Managing Keys With Salt-Key	16
2.23	Navigating the Keys	18
2.24	The pki_dir	18
2.25	Lab 1	19
2.26	Task 1	19
3	Salt Cloud	23
3.1	Salt Cloud - Objectives	23
3.2	Installing Salt Cloud	24
3.3	Installing Salt Cloud	24
3.4	Salt Cloud Configuration	25
3.5	Global Options	25
3.6	Querying Cloud Providers	28
3.7	Running Queries	28
3.8	Gathering Cloud Data	30
3.9	Available Data	30
3.10	Cloud Profiles	32
3.11	Profiles - The VM to Build	32
3.12	Creating Cloud VMs With Profiles	33
3.13	Using a Provider to Create VMs	33
3.14	Destroying VMs	34
3.15	Destruction!	34
3.16	Cloud Maps	35
3.17	Defining a Map of VMs	35
3.18	Actions and Functions	37
3.19	Provider-Specific Commands	37
3.20	Lab 2	38
3.21	Task 1	38
3.22	Task 2	39
3.23	Task 3	39
3.24	Task 4	40
4	Salt Command	43
4.1	The Salt Command - Objectives	43
4.2	Command Structure	44
4.3	Command Structure	44
4.4	Targets	45
4.5	Targeting	45
4.6	Many Ways to Target	45
4.7	Matching by id	46
4.8	Minion id	46
4.9	Glob Targeting	46
4.10	Regular Expression Matching	46
4.11	List Matching	46
4.12	Matching by IP address	47
4.13	Matching by IP Address	47
4.14	Grains	48
4.15	What are Grains?	48

4.16	Matching by Grains	48
4.17	Grains in the Minion Configuration	48
4.18	Globs or Regex	49
4.19	Pillar	50
4.20	What is Pillar?	50
4.21	Targeting with Pillar	50
4.22	Compound Matching	51
4.23	Compound - logical targets	51
4.24	Nodegroups	51
4.25	Functions	52
4.26	Functions, the Salt System Management stdlib	52
4.27	Functions Can Cover Many Areas	52
4.28	Automatic Platform Detection	52
4.29	Self Documenting Functions	53
4.30	Reading the Inline Docs	53
4.31	Calling Docs From the salt Command	53
4.32	Looking in the Man Pages	54
4.33	docs.saltstack.org	54
4.34	Some Core Functions	55
4.35	Command Execution	55
4.36	Package Management	55
4.37	Network Data	55
4.38	Test	55
4.39	User	55
4.40	Bring it Together	56
4.41	Calling Commands	56
4.42	Lab 3	57
4.43	Task 1	57
4.44	Task 2	58
5	Remote Execution in Depth	61
5.1	Remote Execution - Objectives	61
5.2	Returners	62
5.3	Returning Data	62
5.4	Many Executions, One Publication	63
5.5	Compound Commands	63
5.6	Runners	64
5.7	Runners	64
5.8	Available Runners	64
5.9	Reporting	64
5.10	Custom Deployments	64
5.11	Salt Job System	65
5.12	Jobs Runner	65
5.13	Job IDs	65
5.14	Minion proc System	65
5.15	Job Lookup	66
5.16	Lookup Running Jobs	66
5.17	Lookup Old Jobs	66
5.18	Job Cache	67
5.19	Signaling Jobs	68
5.20	Signal Jobs	68
5.21	Kill and Term Jobs	68
5.22	Salt Call	69
5.23	Salt Call Command	69

5.24	Peer Communication	70
5.25	Minions Sending Commands	70
5.26	Configuring the Peer Interface	70
5.27	Configuring the Peer Runner	70
5.28	Executing Peer Commands	71
5.29	Running Peer Commands	71
5.30	Executing Peer Runners	71
5.31	Salt SSH	72
5.32	Salt SSH	72
5.33	Minion Optional	72
5.34	Calling Salt SSH	72
5.35	Rosters	73
5.36	Rosters	73
5.37	Flat Roster	73
5.38	Scan Roster	73
5.39	Lab 4	74
5.40	Task 1	74
5.41	Task 2	74
5.42	Task 3	75
5.43	Task 4	75
6	Salt States	77
6.1	Salt States - Objectives	77
6.2	What Are States?	78
6.3	Salt States	78
6.4	State System in Layers - Low Layers	79
6.5	State System in Layers - Low	79
6.6	High Layers	81
6.7	High Data - Easily Define Complex Low Data	81
6.8	SLS Files	81
6.9	High State Data	81
6.10	The State Tree	82
6.11	The State Tree	82
6.12	State Tree Configuration	82
6.13	The Top File	83
6.14	The Top File	83
6.15	Defining the Top File	83
6.16	Making States	85
6.17	Getting the Right Data	85
6.18	Represent!	85
6.19	The SLS File	86
6.20	Salt SLS Files	86
6.21	Basic States	88
6.22	Basic States	88
6.23	Bring it Together	90
6.24	Simplicity!	90
6.25	Example to Merge it All Together	90
6.26	Running States	92
6.27	Managing State Runs	92
6.28	Over State System	94
6.29	Orchestrate States	94
6.30	Ordering in Groups	94
6.31	The overstate.sls	94
6.32	Executing the Over State	95

6.33	Lab 5	96
6.34	Task 1	96
6.35	Task 2	96
6.36	Task 3	97
6.37	Task 4	98
7	Power in States	101
7.1	Power in States - Objectives	101
7.2	High Data Structure	102
7.3	The Data	102
7.4	The Include Statement	103
7.5	Including other SLS Files	103
7.6	Extend	105
7.7	Extending External SLS Data	105
7.8	The Extend Declaration	105
7.9	Extend is a Top Level Declaration	105
7.10	Rules to Extend By	106
7.11	Requisites	107
7.12	Requisite Declaration	107
7.13	Multiple Requisites	107
7.14	The <code>require</code> Requisite	108
7.15	The <code>watch</code> Requisite	108
7.16	<code>watch</code> and the <code>mod_watch</code> Function	109
7.17	Finite Ordering	111
7.18	Ordering States	111
7.19	The Order	111
7.20	The Order Option	111
7.21	Name and Names!	112
7.22	ID vs Name	112
7.23	One ID to Many Names	112
7.24	Requisite In	114
7.25	The Requisite “in” Declaration	114
7.26	Using Prereq	115
7.27	Using the Prereq Requisite	115
7.28	How Prereq Works	115
7.29	Use and <code>use_in</code>	116
7.30	<code>use</code> and <code>use_in</code> Requisites	116
7.31	Startup States	117
7.32	Startup States	117
7.33	Providers	118
7.34	State Providers	118
7.35	Render Data How You Want	119
7.36	Renderers	119
7.37	Multiple Renderers	119
7.38	Python Renderer	120
7.39	Using the Python Renderer	120
7.40	Render Pipes	121
7.41	Using Render Pipes	121
7.42	LAB 6	122
7.43	Task 1	122
7.44	Task 2	123
7.45	Task 3	123
7.46	Task 4	124

8	Salt Extras	127
8.1	Salt Extras	127
8.2	Jinja Basics	128
8.3	Jinja - Templating SLS	128
8.4	Jinja Variables	128
8.5	Salt Data in Jinja	129
8.6	Data passed into Jinja	129
8.7	Pillar of Salt	131
8.8	Pillar of Salt	131
8.9	Building Pillar	132
8.10	Declaring the Master Pillar	132
8.11	Pillar Configuration	132
8.12	Setting up the Pillar	132
8.13	Minion Specific Pillar	134
8.14	Private Pillars	134
8.15	External Pillar	135
8.16	External Pillar	135
8.17	Parameterize Everything	136
8.18	Parameterizing	136
8.19	Using the “get” Functions	136
8.20	Client ACLs	137
8.21	Client ACLs	137
8.22	Client ACL Example	137
8.23	External Job Cache	138
8.24	External Job Cache	138
8.25	Salt File Server	139
8.26	Salt File Server	139
8.27	Asynchronous	139
8.28	Environment Aware	139
8.29	Gzip Files	139
8.30	Getting Files	139
8.31	Salt Wheel System	141
8.32	Salt Wheel System	141
8.33	External Authentication	142
8.34	External Authentication System	142
8.35	External Authentication Example	142
8.36	LAB 7	143
8.37	Task 1	143
8.38	Task 2	144
8.39	Task 3	145
9	Extending Salt	149
9.1	Extending Salt - Objectives	149
9.2	Writing Modules	150
9.3	Exposing Functions	150
9.4	Return data	150
9.5	Args and KWargs	151
9.6	Modules and kwargs	151
9.7	States and kwargs	151
9.8	Built in Components	152
9.9	Access to Grains	152
9.10	Cross Calling Modules	152
9.11	Unified Module Configuration	153
9.12	The __virtual__ function	153

9.13	Loader System	154
9.14	Module Loader	154
9.15	Loader Configuration	154
9.16	Distributing Modules Via <code>file_roots</code>	154
9.17	Writing Grains	155
9.18	Functions	155
9.19	Static Data	155
9.20	Return Data	155
9.21	Writing States	157
9.22	Matching Components	157
9.23	State Return Data	157
9.24	Test Interface	157
9.25	Writing Returners	159
9.26	Getting Configuration Data	159
9.27	Returner Function	159
9.28	Writing Renderers	162
9.29	Single Function Loader	162
9.30	Passing the Right Data	162
9.31	Loading the Data	162
9.32	Writing External Pillars	165
9.33	Configuration to Python	165
9.34	Example	165
9.35	LAB 8	166
9.36	Task 1	166
9.37	Task 2	167
10	Topology	169
10.1	Salt Topologies	169
10.2	Multi-Master	170
10.3	Multi Master	170
10.4	Syndic	172
10.5	The Salt Syndic	172
10.6	Logical Groups	172
10.7	Transparent	172
10.8	Configure the Syndic	173
10.9	Configuring the Syndic	173
10.10	Standalone Minion	174
10.11	Standalone Minion	174
10.12	Configuring the File Client	174
10.13	Minion Swarm Tests	175
10.14	Minion Swarming	175
10.15	Using the Swarm	175
10.16	Operations For the Swarm	175
10.17	Event System	176
10.18	The Event System	176
10.19	Tags and Data	176
10.20	Minions Fire Master Events	176
10.21	Listening to Events	177
10.22	Using the Event API	177
10.23	React to Events	178
10.24	Reactor System	178
10.25	Mapping Events to Reactor SLS Files	178
10.26	Reactor Logic	179
10.27	Reactor File Logic	179

10.28 Schedule Recurring Executions	180
10.29 Salt Scheduling	180
10.30 Git File Server Backend	181
10.31 GitFS File Server Backend	181
10.32 Simple Configuration	181
10.33 Extended Fileserver Backends	182
10.34 Extended Backends	182
10.35 Multiple Backends	182
10.36 Salt API	183
10.37 Salt API	183
10.38 Turning on the API	183
10.39 Lab 9	184
10.40 Task 1	184