Systems & Network Training

Definitive Salt for engineers

A 2 day Hands on training course

Save on expenses. This course is also available on your premises.

Description

Salt is a remote execution framework and configuration management system. This course covers Salt from the basics. After a quick first taste the course moves onto execution modules, salt states, minion and master data, jinja, Salt extensions and then topology and configuration options. Hands on sessions are used to reinforce the theory rather than teach specific manufacturer equipment.

Key outcomes

By the end of the course delegates will be able to:

- ✔ Install and use Salt.
- ✔ Describe the architecture of Salt.
- ✔ Manage configurations with Salt.
- ✔ Extend Salt.

Training Approach

This structured course uses Instructor Led Training to provide the best possible learning experience. Small class sizes ensure students benefit from our engaging and interactive style of teaching with delegates encouraged to ask questions throughout the course. Quizzes follow each major section allowing checking of learning. Hands on sessions are used throughout to allow delegates to consolidate their new skills.

Details

Who will benefit?
Anyone working with Salt.

Prerequisites
Linux fundamentals.

Duration: 2 days

Overall rating: ★★★★★

Generic Training	Small Class Sizes	Hands On Training	Our Courseware	Customise Your Course
Generic training compliments product specific courses covering the complete picture of all relevant devices including the protocols "on the wire".	We limit our maximum class size to 8 delegates; often we have less than this. This ensures optimal interactivity between delegates and instructor.	The majority of our courses use hands on sessions to reinforce the theory.	We write our own courses; courseware does not just consist of slides and our slides are diagrams not bullet point text.	Please contact us if you would like a course to be customised to meet your specific requirements. Have the course your way.
"*Friendly environment with expert teaching that teaches the why before the how.*" G.C. Fasthosts	*"Excellent course. The small class size was a great benefit..."* M.B. IBM	*"Not many courses have practice added to it. Normally just the theoretical stuff is covered."* J.W. Vodafone	*"Comprehensive materials that made the course easy to follow and will be used as a reference point."* V.B. Rockwell Collins	*"I was very impressed by the combination of practical and theory. Very informative. Friendly approachable environment, lots of hands on."* S.R. Qinetiq

Definitive Salt for engineers

Course Content

Introduction

What is Salt? High- level architecture, Some quick examples, system management, configuration management, A brief history, Topology options, Extending Salt.

Quick start: First taste of Salt

Single-master setup, from packages, bootstrap scripts, Starting up, Basic commands, salt: the main workhorse, salt-key: key management, salt-call: execution on the minion, salt-run: co-ordination of jobs on the master, summary of commands, Key management, viewing keys, accepting keys, rejecting keys, key files, Minion targeting, minion ID, list (-L), glob, regular expressions (-E), grains (-G), compound (-C), targeting summary, Additional remote execution details, Conclusion.

Execution modules: The functional foundation

sys: information and documentation about modules, sys.doc basic documentation, sys.list_modules, sys.list_functions: simple listings, cmd: execute via shell, cmd.run: run any command, pkg: manage packages, virtual modules, pkg.lists_pkgs: list all installed packages, pkg.available version: see what version will be installed, pkg.install: install packages, user: manage users, user.add: add users, user.list_users, user info: get user info, saltutil: access various Salt utilities, Summary.

Configuration management: Salt states

Salt files overview, SLS example: adding a user, working with the multi-layered state system, Highstate and the top file, the top file, State ordering, require: depend on another state, watch: run based on other changes, odds and ends, Summary.

Minion data / master data

Grains are minion data, performing basic grain operations, setting grains, targeting with grains in the top file, Pillars are data from the master, querying pillar data, querying other sources with external pillars, Renderers give data options.

Extending Salt: part I

Introduction to Jinja, Jinja basics, Templating with Jinja, filtering by grains, Custom execution module, Custom state modules, Custom grains, External pillars, Summary.

More on the matter

Runners, manage minions, manage jobs, The orchestrate runner, The event system, The reactor system, Summary.

Extending Salt: part II

Python client API, reading configuration data on a master and minion, using the master client (localclient) API, Using the caller client API, Custom runners, writing a custom runner, using the runnerclient API, Summary.

Topology and configuration options

Master configuration, directories and files, logging, access control, files server options, Topology variations, masterless minions, peer systems, syndication masters, multiple masters.

Brief introduction to salt-cloud

Overview, Setup AWS and salt-cloud, installing salt-cloud, cloud providers, cloud profiles, cloud maps, Introspection via salt cloud, Creating infrastructure, More information.

Using vagrant to run Salt examples

YAML.