



SALTSTACK

Salt Administration II

Training Syllabus

This is the second course in the SaltStack Administration training series. It builds on the concepts of the previous course by presenting additional topics above the fundamentals of Salt administration. The topics are presented with scenario-based labs.

PREREQUISITES

Salt Administration I

OVERVIEW

This course begins with presenting the different aspects of the provisioning and deploying Salt minions. It presents cloud integration with Salt, custom deployments of Linux, Windows, and Proxy minions. Alternative topologies such as implementing master failover strategies. Salt syndic and other hierarchical solutions are presented outlining the requirements, benefits, and caveats of each type.

Building on the fundamental administrative tasks presented in the Admin-I course, this course shows how to address administrative tasks in large environments. It provides a detailed outline of various execution paths in Salt architecture. Execution topics include Salt events, state and file rendering, and other concepts important to Salt architecture. Security topics include running as a non-root user, using external authentication, and troubleshooting Salt.

The course adds to configuration management concepts by outlining best practices for state formulas. There will be scenarios presenting showing the best use of templating with Jinja. It shows how to utilize the Salt state system with a deep-dive into the stages of execution. Salt orchestration scenarios will show how to automate enterprise-wide deployments. The content also includes information about creating a dynamic environment with Salt reactors and beacons.

CERTIFICATION

Completing “Admin I” and “Admin II” will prepare an attendee to sit the **SaltStack Certified Engineer (SSCE)** exam.

OFFERING

All students are provided:

- A printed course training manual
- A Salt cloud environment for hands-on labs during training

FORMAT

This course is offered in the following instructor-led formats:

- 3 full-days
- 5 half-day days

This training is offered publically at SaltStack headquarters in Lehi, UT with virtual instructor-led (vILT) access for remote attendees on the 5 half-day format.

When offered privately onsite or virtually the training follows the 3-day format.

COURSE TECHNICAL REQUIREMENTS

The technical requirements for SaltStack Training are:

Video Conference

For remote attendees we offer remote virtual instructor-led (vILT) access. Remote access is typically uses Join.me or Google Hangouts. It works best in a Chrome browser with the Google Hangouts plugin. You can download it from <https://www.google.com/tools/dlpage/hangoutplugin>

Sometimes a video conferencing technology may be used. In such cases technical requirements will be given.

Labs

Each student will be given a group virtual machines containing a Salt master with minions hosted in the Amazon cloud. You will need to be able to SSH (destination port 22) from your network to access them. Detailed information about accessing the lab environment will be provided at the beginning of class.

Class Portal:

The class portal is a website located at <http://training.saltstack.com> contains links to resources and end of chapter knowledge checks. A login to this site will be given at the beginning of class.



COURSE OUTLINE

Introduction

- Welcome
- Objectives
- Getting Started
- Topics Covered
- Summary

Salt Cloud

- Objectives
- What is Salt Cloud?
- Salt Cloud Components
- The salt-cloud Command
- Configuring Salt Cloud
- Querying Available Data
- Defining Virtual Machine Profiles
- Salt Cloud Options
- Creating VMs with Profiles
- The Salt Cloud Provisioning Process
- Salt Cloud Command Output
- Querying for VMs
- Destroying Virtual Machines in the Cloud
- Managing Multiple VMs Instances
- Provider-Specific Commands
- Salting Existing Systems
- The Class Setup
- Your Salt Lab Environment
- Summary
- Lab - Accessing the Lab Environment
- Lab - Use the EC2 Salt Cloud Driver
- Lab - Create and Destroy Cloud VMs
- Lab - Create a Cloud Map
- Lab - Add Existing Systems to the Salt Master
- Cleanup

Windows Minions

- Objectives
- Salt Windows Minion Management
- The Salt Windows Management Framework
- Deploying Windows Minions
- User and Group Management
- Managing Windows Permissions
- Software Management
- Managing Windows Features and Roles
- Summary
- Lab - Provision Windows Minions
- Lab - Software Management
- Lab - Managing Windows Roles and Features
- Lab - Windows Updates

Proxy Minions

- Objectives
- What is a Proxy Minion?
- Salt Proxy Minion Topology
- Using Proxy Minions
- Summary
- Lab - Provision Proxy Minion

Salt Execution and Architecture

- Objectives
- Salt Execution
- Salt Components
- Calling Modules Locally on a Minion
- Sending Jobs to the Salt Master
- The Event System
- Salt Master Processes
- The "salt" Execution Architecture
- Summary
- Lab - View Master Processes
- Lab - Listening to Events
- Lab - Analyze Salt Activity
- Cleanup

The Salt State System

- Objectives
- Uses of the Salt State System
- SLS Files
- Salt State Processing
- State Rendering
- State Compile
- State Runtime
- Summary
- Lab - Analyzing State Execution

Salt Renderers

- Objectives
- Salt State Rendering
- Using the Python Renderer
- Using the Jinja Renderer
- Conditional "if" Statements
- Jinja Variable Assignments
- Leveraging Lookup Lists and Dictionaries
- Jinja Filters
- Calling Execution Modules with Jinja
- Using the GPG Render
- Summary
- Lab - Create Users and Groups Management
- State



Building Salt Formulas

- Objectives
- Salt Formulas
- Salt State Formulas Best Practices
- Using Inline Pillar Data
- Use Lookup Dictionaries Based on Grains
- The Online Salt Formula Repository
- A Complete State Example
- Summary
- Lab - Create States with Salt Formulas
- Lab - Webstack: Web Server State
- Lab - Webstack: Configuring an HAProxy Load Balancer
- Lab - Webstack: Configuring the "top.sls" File

Salt Orchestration Formulas

- Objectives
- Running Jobs on the Salt Master
- Salt Orchestration
- Orchestration Declarations
- Using Inline Pillar with Orchestration
- Using State Modules in Orchestration
- The Salt Mine
- Summary
- Lab - Webstack: Create the Pillar Data for the Web Servers
- Lab - Webstack: Configuring Web Server Mine Functions
- Lab - Webstack: Creating the Web Servers Orchestration State
- Lab - Windows Orchestration

Reactors and Beacons Formulas

- Objectives
- Reactor System
- Beacons
- Reactor Best Practices
- Summary
- Lab - Windows Beacons
- Lab - Windows Reactors
- Lab - Webstack: Configure the Load Beacon
- Lab - Webstack: Creating the Web Server Reactors
- Lab - Webstack: Triggering the Scale Up Event

Supporting and Troubleshooting Salt

- Objectives
- Introduction
- Viewing Salt Activity
- Network Settings
- Salt Keys
- Salt States
- Summary
- Overview
- Loading Lab Troubleshooting Scenarios
- Lab - Troubleshooting Scenario 1
- Lab - Troubleshooting Scenario 2
- Lab - Troubleshooting Scenario 3
- Lab - Troubleshooting Scenario 4
- Lab - Troubleshooting Scenario 5
- Lab - Troubleshooting Scenario 6
- Lab - Troubleshooting Scenario 7
- Lab - Troubleshooting Scenario 8
- Lab - Troubleshooting Scenario 9
- Lab - Troubleshooting Scenario 10
- Lab - Troubleshooting Scenario 11
- Lab - Troubleshooting Scenario 12

Salt Security

- Objectives
- More About Keys
- The "pki_dir"
- Increasing Security
- Decreasing Security
- Client ACLs
- External Authentication System
- Hardening Salt
- Security Issues
- Summary
- Lab - Understanding Salt Keys
- Lab - Configuring Client ACLs
- Lab - External Authentication Cleanup

Multi-Master

- Objectives
- Multi-Master
- Sharing Files Between Masters
- Summar



ADDITIONAL INFORMATION

For additional information please contact:

SaltStack, Inc.

3400 N. Ashton Blvd,
Suite 110
Lehi, UT 84043
T +1 801.207.7440

Email:

training@saltstack.com

Web:

<http://www.saltstack.com/training>
<http://www.saltstack.com/certification>
<http://www.saltstack.com/services>

