Date: 01.08.2025



Copyright @ 2024 PibyThree.com All Rights Reserved

# SaltStack Workflow Orchestration and Logging Optimization

Contents

[SaltStack Workflow Orchestration and Logging Optimization 1](#_Toc204956564)

[1. Git Repository for the Workflow Orchestration 2](#_Toc204956565)

[2. Running the orchestration file 2](#_Toc204956566)

[3. Logging optimization 2](#_Toc204956567)

[4. Orchestration Files vs Top file 2](#_Toc204956568)

# 1. Git Repository for the Workflow Orchestration

This repository contains all the salt state files used for salt operations: [aryan-madhavi/saltstack](https://github.com/aryan-madhavi/saltstack-demo.git)

These files are to be stored in `/srv/salt`.

The required directories for this are: orch/ nginx/ nssm/ services/ and its internal files.

The `orch/nginx\_win\_orch.sls` file orchestrates the following steps in `win-minion`:

# 2. Running the orchestration file

The orchestration file can be run with the following command:  
 $ salt-run state.orchestrate orch.nginx\_win\_orch

# 3. Logging optimization

Whenever a longer state file is being processed, it hogs the terminal without any clear indication of current workflow state. Thus, there is a requirement for proper logging functionality which helps the administrator to understand what is working under the hood.

Thus need to run:

$ salt-run state.orchestrate orch.nginx\_win\_orch -l info

# 4. Orchestration Files vs Top file

|  |  |  |
| --- | --- | --- |
| **Feature / Aspect** | **top.sls (Highstate)** | **Orchestration Files (orchestrate/\*.sls)** |
| **Purpose** | Assign states to minions (i.e. apply configurations) | Coordinate and control execution across multiple minions |
| **Scope** | Per-minion state application | Cluster-wide or multi-minion orchestration and workflows |
| **Location** | Inside your file root (/srv/salt/top.sls) | Typically under /srv/salt/orchestrate/ or similar |
| **Trigger** | Run via state.apply / state.highstate | Run via state.orchestrate |
| **Execution Target** | Runs on the targeted **minions** | Runs **on the master** and calls commands remotely on minions |
| **Example Use Case** | Install nginx, create users, configure SSH, etc. | Provision instance → wait → install → approve key → configure |
| **Can Run Modules?** | Not directly (only states) | Yes — can use salt.state, salt.runner, salt.function, etc. |
| **Common Commands** | salt '\*' state.apply | salt-run state.orchestrate orchestrate.deploy\_cluster |
| **File Example** | top.sls | orchestrate/deploy\_cluster.sls |
| **Visual Comparison** | Declarative Mapping (Minion-focused) | Command-Driven Workflow (Master-focused) |