

Date: 01.08.2025

# SaltStack Workflow Orchestration and Logging Optimization

## Contents

SaltStack Workflow Orchestration and Logging Optimization .....	1
1. Git Repository for the Workflow Orchestration .....	2
2. Running the orchestration file .....	2
3. Logging optimization .....	2
4. Orchestration Files vs Top file .....	2

## 1. Git Repository for the Workflow Orchestration

This repository contains all the salt state files used for salt operations: [aryan-madhavi/saltstack](#)  
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)
<b>Purpose</b>	Assign states to minions (i.e. apply configurations)	Coordinate and control execution across multiple minions
<b>Scope</b>	Per-minion state application	Cluster-wide or multi-minion orchestration and workflows
<b>Location</b>	Inside your file root (/srv/salt/top.sls)	Typically under /srv/salt/orchestrate/ or similar
<b>Trigger</b>	Run via state.apply / state.highstate	Run via state.orchestrate
<b>Execution Target</b>	Runs on the targeted <b>minions</b>	Runs <b>on the master</b> and calls commands remotely on minions
<b>Example Use Case</b>	Install nginx, create users, configure SSH, etc.	Provision instance → wait → install → approve key → configure
<b>Can Run Modules?</b>	Not directly (only states)	Yes — can use salt.state, salt.runner, salt.function, etc.
<b>Common Commands</b>	salt '*' state.apply	salt-run state.orchestrate orchestrate.deploy_cluster
<b>File Example</b>	top.sls	orchestrate/deploy_cluster.sls
<b>Visual Comparison</b>	Declarative Mapping (Minion-focused)	Command-Driven Workflow (Master-focused)