

Date: 24.07.2025

Automate Service Management and File Operations using SaltStack States (.sls files)

Contents

Automate Service Management and File Operations using SaltStack States (.sls files).....	1
1. Git Repository for all salt state files	2
2. File Automation.....	2
3. Service Automation	2
Starting a service and stopping another service in Linux	2
Stopping and starting the same service in Windows	2

1. Git Repository for all salt state files

This repository contains all the salt state files used for salt operations: [salt files](#)
These files are to be stored in `/srv/salt`.

2. File Automation

This is a simple .sls file that creates a directory and initializes a file with default text.

The file is stored in the `files_mgmt` directory and is named `init.sls`. When using this naming convention, you can apply the state without specifying the full filename:

```
$ salt '*' state.apply files_mgmt
```

If the file were instead named files.sls and placed in the files_mgmt directory, you would need to specify the filename (without the .sls extension) when applying the state:

```
$ salt '*' state.apply files_mgmt.files
```

This naming convention applies for all files.

3. Service Automation

Starting a service and stopping another service in Linux

```
$ salt 'linux-minion*' state.apply linux_srvc_conf
```

Over here the `linux-minion*` is a targeted expression that tells salt to apply this state to any minion name that starts with `linux-minion`. For eg. `linux-minion-1`, `linux-minion2`, `linux-minion-dev1`, etc. Same applies for all similar targeted expressions.

Stopping and starting the same service in Windows (essentially restarting service)

```
$ salt 'win-minion*' state.apply win_srvc_conf
```