

# Salt - Conductor Framework specific salt grains

coming soon.....

- [Overview](#)

## Overview

This page describes the list of salt grains that are created and used throughout the salt ecosystem when provisioning and managing vm instances with the SaltStack conductor runner. This runner uses salt-cloud to provisioning instance in the cloud with specific configurations (role based etc..) with lots of hooks to provide and end to end delivery of product, systems and services.

The conductor framework consists of multiple aspects. Conductor runner is the central automation tool, but there is a specific Pillar and State data model that goes hand in hand with the conductor to provide an enterprise scalable framework for Saltstack.

The model and the runner together create and consume quite a few [salt grains](#) specific to the implementation. Salt itself creates a lot of grains on each minion that is registered to the master. Salt-cloud also creates grains when used to provision new instances.

Grains are a very important part of the Salt ecosystem and are used in many ways when targeting minions.

The conductor creates and uses grains on all instances that it will create and uses them for targeting in numerous situations like finding cluster members, identifying cloud, cluster and provision id's, orchestration hooks etc...

Each product or technology specific salt state may use grains to learn what tasks to do. This is a recommendation in the Conductor framework.

A minion can have almost unlimited grains of any type (string, int, list, dictionary). This page documents the static and dynamic salt grains that will be applied to minions created within the conductor tool with a brief explanation. This is not intended to be a deep dive on the topic of Saltstack grains and the extend to what they can be used beyond how the conductor is using them in the framework.