

# Saltstack

## Table of Contents

Documentation sur le système .....	1
Documentation sur le web .....	1
Minions .....	1
Statut de serviteur .....	1
Grains cible avec des fichier Salt .....	2
Grains .....	2
Liste tous les grains sur tous les serveurs .....	2
Manipuler les grains.....	2
Emplois à Salt .....	2
Sysadmin specific .....	3
Système et statut .....	3
Packages .....	3
Vérifier l'état d'un service et manipuler des services .....	3
Réseau .....	3
Salt cloud .....	4

## Documentation sur le système

```
salt '*' sys.doc          # sortie sys.doc (= toute la documentation)
salt '*' sys.doc pkg      # uniquement sys.doc pour le module pkg
salt '*' sys.doc network  # uniquement sys.doc pour le module réseau
salt '*' sys.doc system   # uniquement sys.doc pour le module système
salt '*' sys.doc status   # uniquement sys.doc pour le module d'état
```

## Documentation sur le web

Documentation SaltStack: <http://docs.saltstack.com/en/latest/> Salt-Cloud: <http://docs.saltstack.com/en/latest/topics/cloud/> Emplois: <http://docs.saltstack.com/en/latest/topics/jobs/>

## Minions

### Statut de serviteur

Vous pouvez également utiliser plusieurs commandes pour vérifier si les sbires sont vivants et donner un coup de pied, mais je préfère gérer.status / haut / bas.

```
salt-run manage.status      # Quel est le statut de tous mes serveurs? (à la fois
vers le haut et vers le bas)
salt-run manage.up          # Des serveurs en place?
salt-run manage.down        # Des serveurs qui sont en panne?
salt-run manage.alived      # Afficher tous les serveurs vivants
salt '*' test.version       # Afficher la version du salt
salt '*' test.ping          # Utilisez le module de test pour vérifier si le serveur est
actif et répond.

                           # (Pas un ping ICMP!)
```

## Grains cible avec des fichier Salt

Appliquer un fichier d'état spécifique à un (groupe de ..) serveur (s). N'utilisez pas l'extension .sls. (comme dans les fichier Salt!)

```
salt '*' state.sls mystatefile      # mystatefile.sls sera appliqué à *
salt 'minion1' state.sls prod.somefile  # prod / somefile.sls sera appliqué à
minion1
```

## Grains

### Liste tous les grains sur tous les serveurs

Voir un seul élément de grains pour répertorier les valeurs.

```
salt '*' grains.ls
```

```
salt '*' grains.item os           # Affiche la valeur du grain OS pour chaque séide
salt '*' grains.item rôles       # Affiche la valeur des rôles grain pour chaque
serveur
```

### Manipuler les grains.

```
salt 'minion1' grains.setval mygrain True      # Définissez mygrain sur True (créez
s'il n'existe pas encore)
salt 'minion1' grains.delval mygrain           # Supprimer la valeur du grain
```

## Emplois à Salt

Certaines opérations de travaux qui sont souvent utilisées. (<http://docs.saltstack.com/en/latest/topics/jobs/>)

```
salt-run jobs.active          # obtenir la liste des travaux actifs
salt-run jobs.list_jobs      # récupère la liste des travaux historiques
salt-run jobs.lookup_jid <numéro d'identification du travail>      # obtenir les
détails de ce travail spécifique
```

## Sysadmin specific

Quelques trucs qui sont spécifiquement intéressants pour les administrateurs système.

## Système et statut

```
salt 'minion-x- *' system.reboot      # Redémarrons tous les serveurs qui
correspondent à minion-x- *
salt '*' status.uptime                # Obtenez la disponibilité de tous nos serveurs
```

## Packages

```
salt '*' pkg.list_upgrades      # récupère la liste des packages à mettre à niveau
salt '*' pkg.upgrade           # Met à niveau tous les packages via apt-get dist-upgrade
(ou similaire)
```

```
salt '*' pkg.version bash       # récupère la version actuelle du paquet bash
salt '*' pkg.install bash       # installe ou met à niveau le paquet bash
salt '*' pkg.install bash refresh = True      # install ou upgrade package bash mais
# actualisez la base de données du package avant l'installation.
```

## Vérifier l'état d'un service et manipuler des services

```
salt '*' service.status <nom du service>
salt '*' service.available <nom du service>
salt '*' service.start <nom du service>
salt '*' service.restart <nom du service>
salt '*' service.stop <nom du service>
```

## Réseau

Faites des trucs en réseau sur vos serveurs.

```
salt 'minion1' network.ip_addrs          # Obtenez l'IP de votre serveur
salt 'minion1' network.ping <nom d'hôte> # Ping un hôte depuis votre serveur
salt 'minion1' network.traceroute <nom d'hôte> # Traceroute un hôte depuis
votre serveur
salt 'minion1' network.get_hostname      # Récupère le nom d'hôte
salt 'minion1' network.mod_hostname      # Modifier le nom d'hôte
```

## Salt cloud

Salt Cloud est utilisé pour provisionner des machines virtuelles dans le cloud. (surprise!) (<http://docs.saltstack.com/en/latest/topics/cloud/>)

```
salt-cloud -p profile_do my-vm-name -l debug          # Provision utilisant profile_do
comme profil
    # et my-vm-name comme nom de machine virtuelle tout en
    # en utilisant l'option de débogage.
salt-cloud -d my-vm-name          # détruit la machine virtuelle my-vm-name.
salt-cloud -u          # Mettre à jour salt-bootstrap vers la dernière version de
développement sur GitHub.
```