

UD05 • Terraform + Salt Project

Cheat Sheet avanzada con múltiples ejemplos reales de Salt y su integración con Terraform.

1 · Comandos básicos de Salt

- salt '*' test.ping → prueba conexión Minion
- salt '*' grains.items → ver datos del sistema
- salt '*' cmd.run 'uptime' → ejecutar comandos shell
- salt-call state.apply → ejecutar estado localmente (modo minion sin master)
- salt-run jobs.active → ver trabajos activos

2 · Ejemplos de archivos '.sls'

nginx.sls

```
nginx:  
  pkg.installed  
  service.running:  
    - enable: True  
    - require:  
      - pkg: nginx
```

git.sls

```
git:  
  pkg.installed  
  
/clones/mi-repo:  
  git.latest:  
    - name: https://github.com/usuario/repo.git  
    - require:  
      - pkg: git
```

file.sls

```
/etc/mensaje.txt:  
  file.managed:  
    - source: salt://archivos/mensaje.txt  
    - user: root  
    - mode: 644
```

apache.sls (condicional con grains)

```
{% if grains['os_family'] == 'Debian' %}  
apache2:  
  pkg.installed  
  service.running:  
    - name: apache2  
{% else %}  
httpd:  
  pkg.installed  
  service.running:  
    - name: httpd  
{% endif %}
```

3 · Pillars: datos externos y secretos

Top file de pillar

```
base:  
'*':  
  - passwords
```

passwords.sls

```
db_root: "sup3rS3cret"  
user_pass: "salted123"
```

Uso en estado

```
set-pass:  
cmd.run:  
- name: "echo {{ pillar['user_pass'] }} > /tmp/pass.txt"
```

🔗 4 · Integración con Terraform

💻 local-exec

```
resource "null_resource" "salt" {  
  provisioner "local-exec" {  
    command = "salt '*' state.apply"  
  }  
}
```

🌐 remote-exec (desde VM)

```
provisioner "remote-exec" {  
  inline = [  
    "sudo salt-call state.apply nginx"  
  ]  
}
```

📋 outputs → pillars

```
output "web_ip" {  
  value = aws_instance.web.public_ip  
}
```

Puedes usarlo para inyectar dinámicamente a un `.sls` usando templates, env vars o `local-exec` + `pillar.set`.

🎛 5 · Orquestación en Salt

```
# /srv/salt/deploy_stack.sls  
include:
```

```
- docker  
- nginx  
- app  
  
app-deploy:  
  cmd.run:  
    - name: "docker compose up -d"  
    - require:  
      - service: docker
```

Ejecutar:

```
salt-run state.orchestrate deploy_stack
```

6 · Targeting avanzado

- salt -G 'os:Ubuntu' test.ping → por grain
- salt -E 'web[0-9]' cmd.run 'uptime' → por regex
- salt -L 'web1,web2' test.ping → lista separada por coma
- salt -C 'G@os:Debian and web*' test.ping → combinación compuesta

 [Cheat Sheet UD05 – Terraform + Salt · Full Edition](#)