



# SALT CONF

TWENTY ONE

# Adding workflow approval to Salt

---

SALTCONF



# Cleber Paiva de Souza

**DIRECTOR**  
*SSYS Sistemas*

---



@cleberps



[linkedin.com/in/cleberps](https://linkedin.com/in/cleberps)

# Lucas Sanches

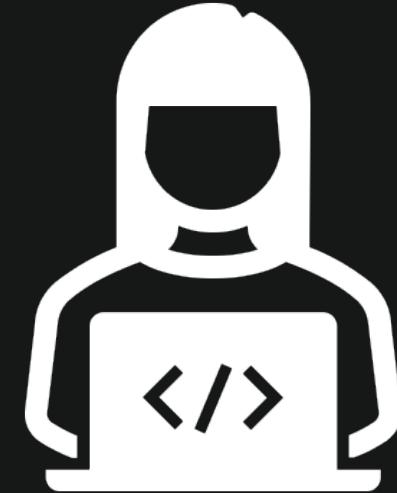
**SOFTWARE ENGINEER**  
*SSYS Sistemas*

---



[linkedin.com/in/lucascosanches](https://linkedin.com/in/lucascosanches)

# Problem



- 
- No workflow approval in the Salt Project. It is a requirement for many business customers.
  - No persistency. Past actions are not reapplied if the system is reinstalled.
  - Simple actions require human intervention, such as rebooting, restarting services, installing packages, and many more.

# Workflow approval

## BUSINESS REQUIREMENT

- Chain of approval for mapped service request types.
- Async actions with intermediate database to export pillar data.
- Custom workflow for each type of service request.



# No persistency

## COMPLIANCE

- Salt does not keep historic data for further consumption, so our approach was to make usage of an intermediate database to export pillar data to the environment.
- Situations such as system reinstallation will return machines to the latest known configuration based on pillar data.



# Personnel

## OPERATIONS

- Operators do not need to be in charge of mapped and repetitive tasks in the infrastructure.
- Optimization of operators' time for more skillful demands of the business.
- Reduces the problem of contracting more IT professionals to the company for simple tasks.



# ITSM (Information technology service management)

- Tools that connect business processes to IT demands.
- Each one has its own key points and specific methods for extending and integrating it. Select your preferred tool.
- Our work is based on Jira, but we have also experimented with ServiceNow.



# Technologies used

- Salt
  - <https://saltproject.io>
  - <https://github.com/saltstack/pepper>
- Jira
  - <https://atlassian.com/software/jira>
  - <https://github.com/pycontribs/jira>
- Python
  - <https://python.org>
- Flask
  - <https://flask.palletsprojects.com>



# Technologies used

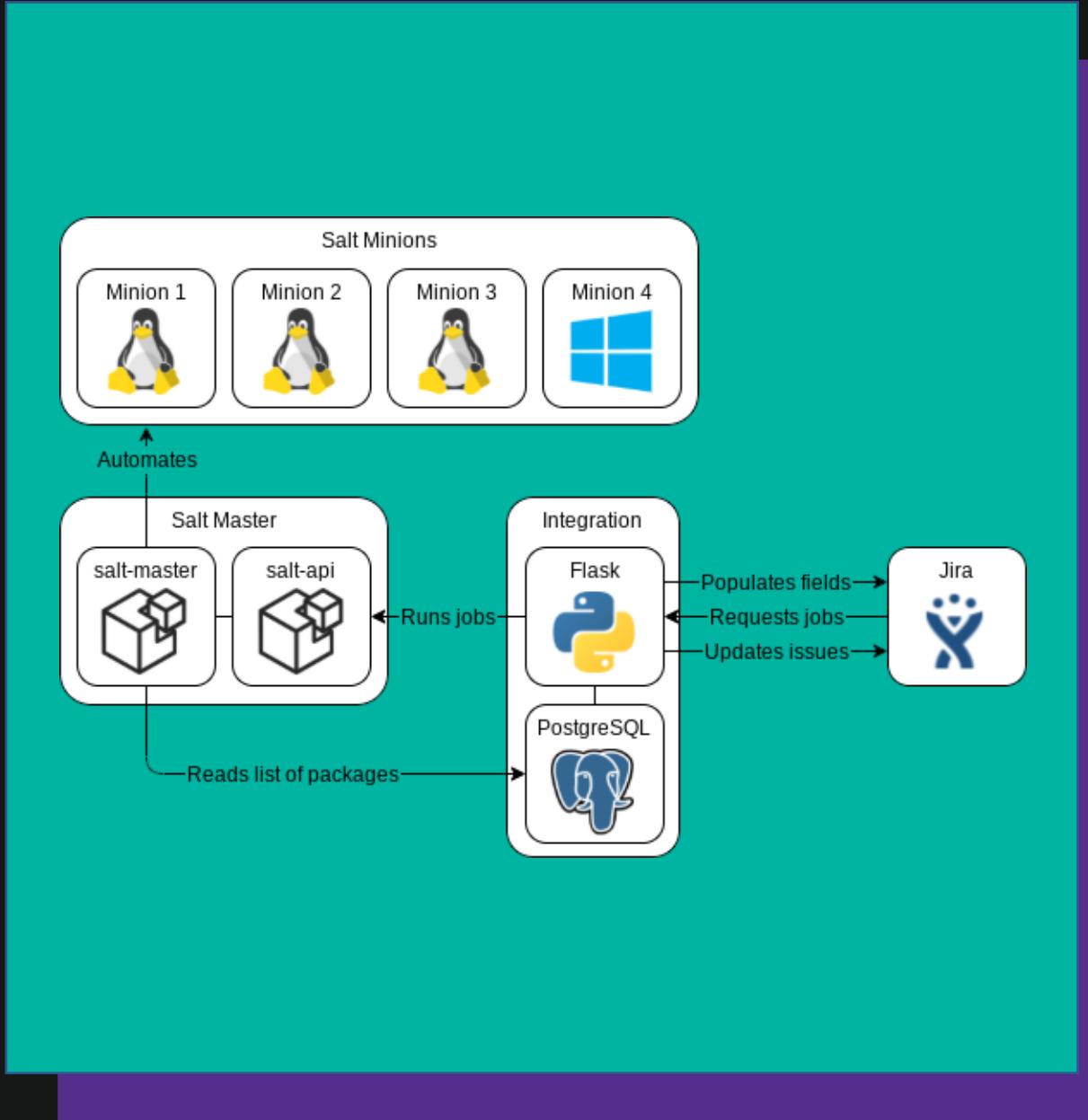
- Docker
  - <https://docker.com>
- PostgreSQL
  - <https://postgresql.org>
  - <https://psycopg.org>
- Gunicorn
  - <https://gunicorn.org>
- Supervisor
  - <https://supervisord.org>
- Code available on GitHub
  - <https://github.com/s-sys/saltconf21>



# Solution architecture

## HOW IT WORKS

- 1) Integration requests list of packages and minions from Salt.
- 2) Integration stores the data into the database and populates fields on Jira.
- 3) Jira requests reboots or package management from the integration.
- 4) Integration stores the data into the database and runs the appropriate jobs on Salt.
- 5) Integration updates the issue status on Jira.



# Jira service requests

## REBOOT

- Runs immediately after approval.

Create issue

Project\*  
SaltConf21 (SC21)

Type\*  
Reboot Request

Select an issue type or request type from the list.

Summary\*  
Reboot minions

Minions\*  
minion1.internal.cloudapp.net.ssys.lab  
minion2.internal.cloudapp.net.ssys.lab  
minion3.internal.cloudapp.net.ssys.lab  
minion4.internal.cloudapp.net.ssys.lab

The ID of the Salt minions.

Description

Style

Create another

# Jira service requests

## LINUX PACKAGE MANAGEMENT

- Packages can be removed.
- Can be scheduled after a specified date and time.
- Installation could be reverted.
- Uses pkg.installed and pkg.purged.

Create issue

Project\*  
SaltConf21 (SC21)

Type\*  
Linux Package Managem...

Select an issue type or request type from the list.

Summary\*  
Install VLC

Linux Minions\*  
minion1.internal.cloudapp.net.ssys.lab  
minion2.internal.cloudapp.net.ssys.lab  
minion3.internal.cloudapp.net.ssys.lab

The ID of the Salt minions.

Linux Package\*  
vlc-bin 3.0.9.2-1

The system package and version to install, upgrade or downgrade to, or remove.

After\*  
1/Jan/21 12:00 AM

Enter when the request is scheduled to be applied.

Description

Style B I U A <sup>A</sup> <sub>A</sub> A ~~A~~

Create another Create Cancel

# Jira service requests

## WINDOWS PACKAGE MANAGEMENT

- Packages can be removed.
- Can be scheduled after a specified date and time
- Can be reverted.
- Uses Chocolatey.

Create issue

Project\*  
SaltConf21 (SC21)

Type\*  
Windows Package Mana...

Select an issue type or request type from the list.

Summary\*  
Install Firefox

Windows Minions\*  
minion4.internal.cloudapp.net.ssys.lab

The ID of the Salt minions.

Windows Package\*  
firefox 93.0.0.20211014

The system package and version to install, upgrade or downgrade to, or remove.

After\*  
1/Jan/21 12:00 AM

Enter when the request is scheduled to be applied.

Description

Style B I U A A ~~A~~ <sup>A</sup> <sub>A</sub>

Create another Create Cancel

# Demo

LET'S SEE IT WORKING



# Limitations

## WHAT WE COULDN'T DO

- Jira can only store 10,000 field options.
- Jira can't populate field options on-the-fly.
- We can't separate packages per minion with Jira.
- Other ITSM tools can solve those issues, but have their own limitations.



# Future work

## WHAT COULD BE IMPROVED

- Make it more robust for usage in production.
- Use patches instead of deleting all options when populating Jira.
- Allow reboots to be scheduled.
- Allow packages to be reverted to a clean state.
- Validate job results before updating tickets.



# Thank you

Cleber Paiva de Souza  
[cleber@ssys.com.br](mailto:cleber@ssys.com.br)

Lucas Sanches  
[lucas.sanches@ssys.com.br](mailto:lucas.sanches@ssys.com.br)

<https://github.com/s-sys/saltconf21>