



Roboconf in less than 10 Slides

By the [Roboconf team](#) / [@Roboconf](#)

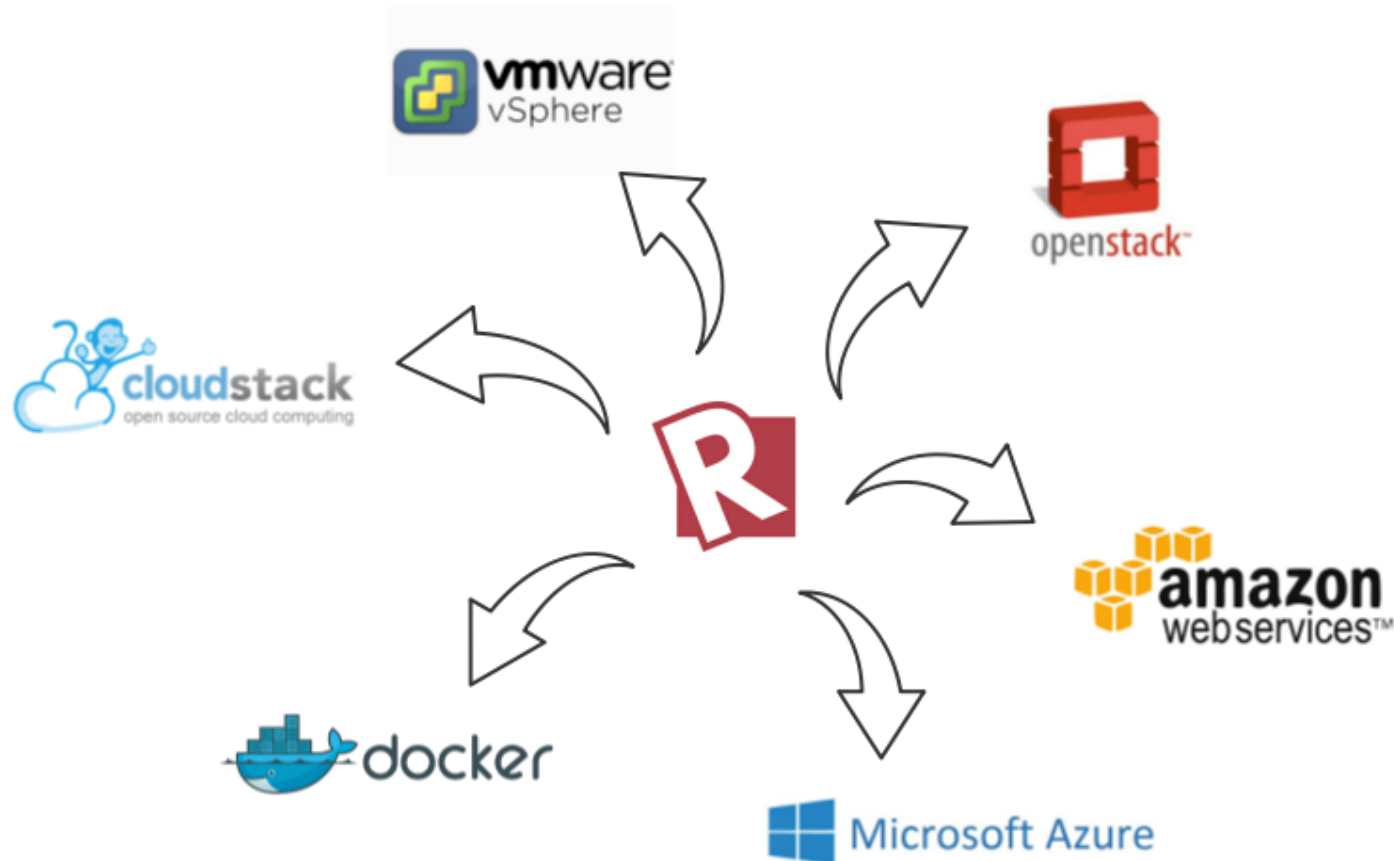
Roboconf

- ... is open source Software (Apache v2)
- ... is hosted on GitHub
- ... started in fall 2013
- ... is powered by [Linagora](#) and [Joseph Fourier University](#)

The project focuses on adaptable (or elastic) deployments.
It is currently used in the [Open PaaS](#) project.

Roboconf covers several possible uses cases.

Cloud Deployment



Roboconf addresses public, private and hybrid cloud deployments. Several cloud providers are supported, as well as virtualization platforms (like VMWare).

PaaS Framework

Software as a Service
(SaaS)

Applicative PaaS
(Platform as a Service)



Low-level PaaS
(Platform as a Service)



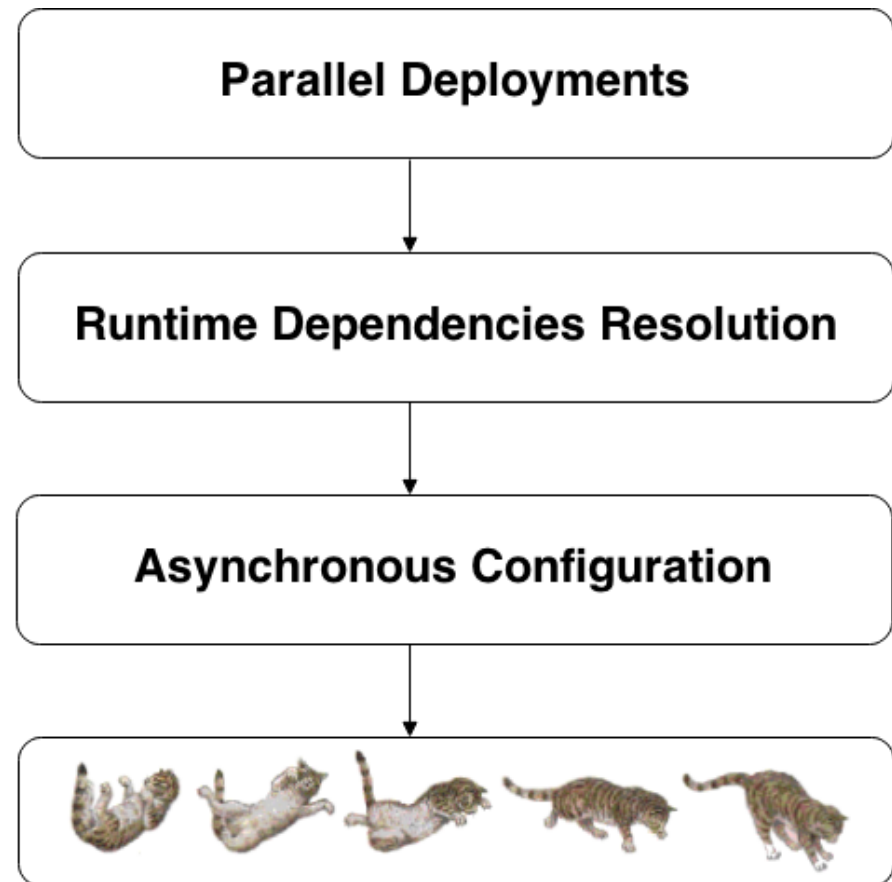
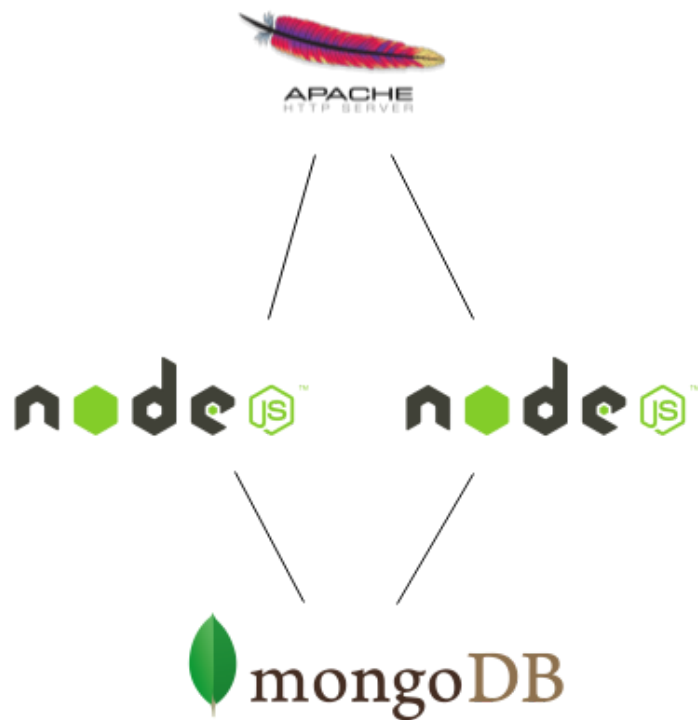
Infrastructure as a Service
(IaaS)



Roboconf can be used to build tailored PaaS. It manages deployment, monitoring and reconfiguration of potentially any application.

Deployment Orchestration

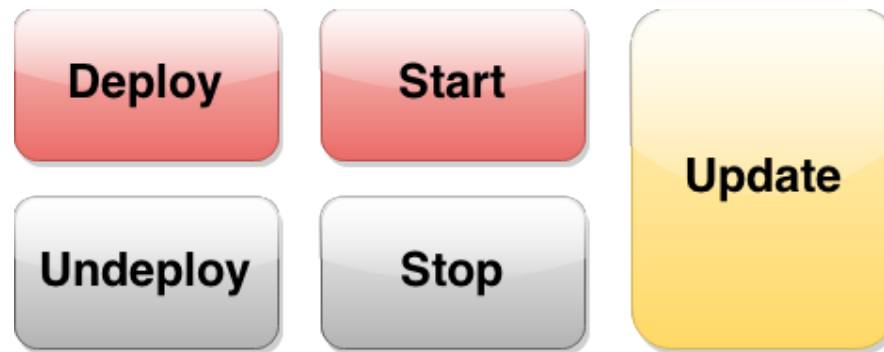
Suppose we want to orchestrate the deployment of the following application.



By using predefined relations between Software components, Roboconf resolves runtime dependencies and updates configuration files so that everything works.

Automating Procedures

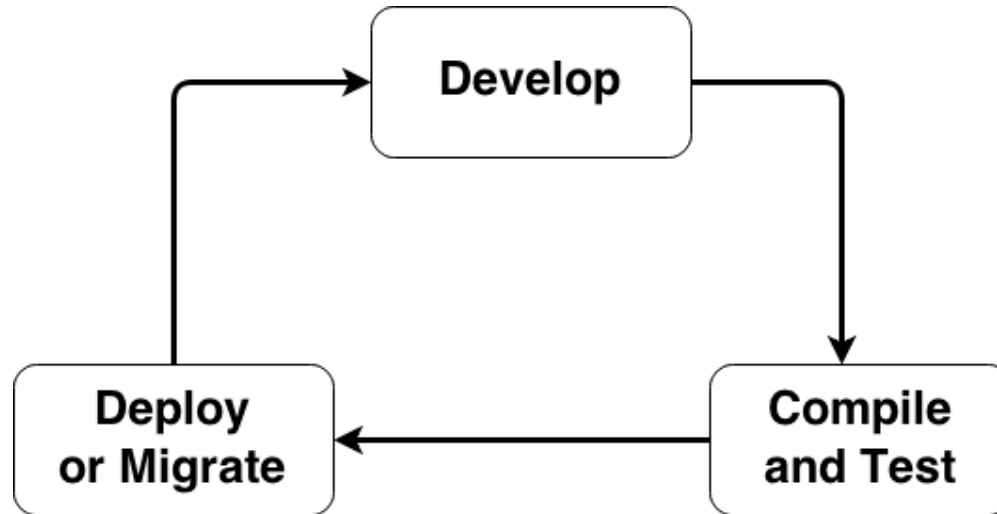
For every Software component that Roboconf manages, normalized actions can be specified. These callback actions are called « recipes ». They can be written as scripts, as a Puppet module, ...



The **update** phase is specific to Roboconf. It is invoked when a runtime dependency appears, is modified or removed. Reconfiguration is thus handled automatically.

Continuous Deployment

Continuous deployments are part of Roboconf's roadmap.



Several migration strategies are planned.

- Full replacement (with service interruption).
- Invariant-based, for continuous development (devOps).
- Progressive upgrade (with architecture change).

Do NOT reinvent the Wheel

Roboconf plugs dynamic in existing deployment solutions.



```
#!/bin/bash
```

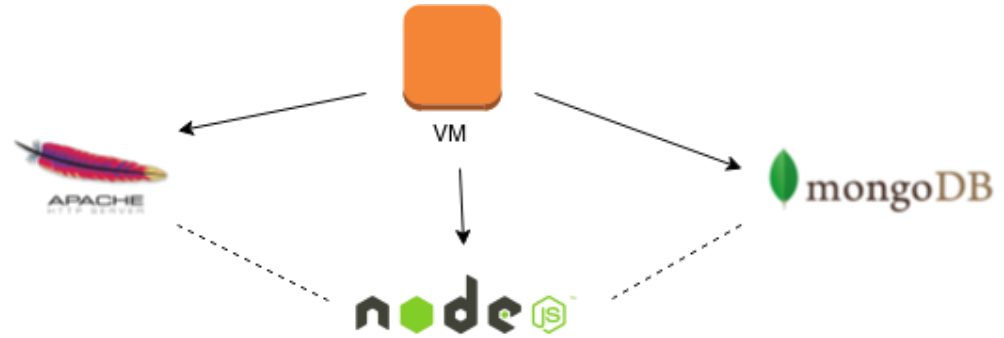


Additional solutions can be supported.



Working with Roboconf

1. Define Software components and their relations



2. Write or Reuse Recipes



`#!/bin/bash`

...

3. Plug Monitoring Rules / Tools

Nagios



...



That's it, you can start deploying with Roboconf.