

Deploy own Runbot to build a dashboard

Seinlet Nicolas, Technical consultant

   @nseinlet

Schedule

- 1 Introduction
- 2 Deploy a Runbot
- 3 Configure Runbot
- 4 Create own dashboard
- 5 Conclusion

Build a dashboard





1 Introduction

What is Runbot

- Runbot is a continuous integration server
- Runbot is an Odoo module
- Runbot is a communication tool between functional and technical guys
- Live preview with 2 databases (base, all)
- Provide build status to GitHub

Why Runbot

- Automate test phase
- Let test instances leaving
- Ease communication between technical and functional guys
- Easy to use, even for non-technical guys
- Do not replace a Q&A on customer side
- Other CI softwares exists and may be already used by technical team
- Designed for Odoo by Odoo
- Integrated with Git and GitHub

Who should use Runbot

Any member, functional or technical, member of a company which :

- Deploy Odoo
- Deploy custom module
- Deploy custom theme

And also:

- Developer who wants to test a new module
- Reviewer to check the commit
- Functional to check a use case
- Anybody to test a new app

When use Runbot

- Since the very beginning of the process
- Before every merge
- For every functional test
- For a new feature test

2

Deploy Runbot in 4
steps

Step 1: PostgreSQL

As runbot is an Odoo module, we need PostgreSQL.

As runbot is not a standard deployment, due to the number of running instances, frequent databases creations and deletions, we must adapt it's configuration accordingly.

- # connections
- Allocated memory
- User authentication
- fsync=False

Step 2: Odoo

As runbot is an Odoo module, we need Odoo itself. version 8 is used on official runbot.

- Install python libraries
- Install Odoo 8 from the github repo
- Install runbot from the odoo-extra github repo
- Install any runbot custom module

If you forget the dbfilter parameter, your runbot server will access plenty of databases.





- Don't forget the dbfilter parameter

Step 3: SSH

Your runbot will need read access to all your repositories, as well as read access to the odoo repository.

Create a dedicated user for this purpose, with only read access to your repos.

If you build on top of enterprise version, ask read access for this runbot user.

- Generate an ssh key for the linux user running runbot
- Create a  GitHub,  BitBucket, ... user for your runbot
- Add your ssh key to the  GitHub,  BitBucket, ... user
- Give read access for this user to your repositories

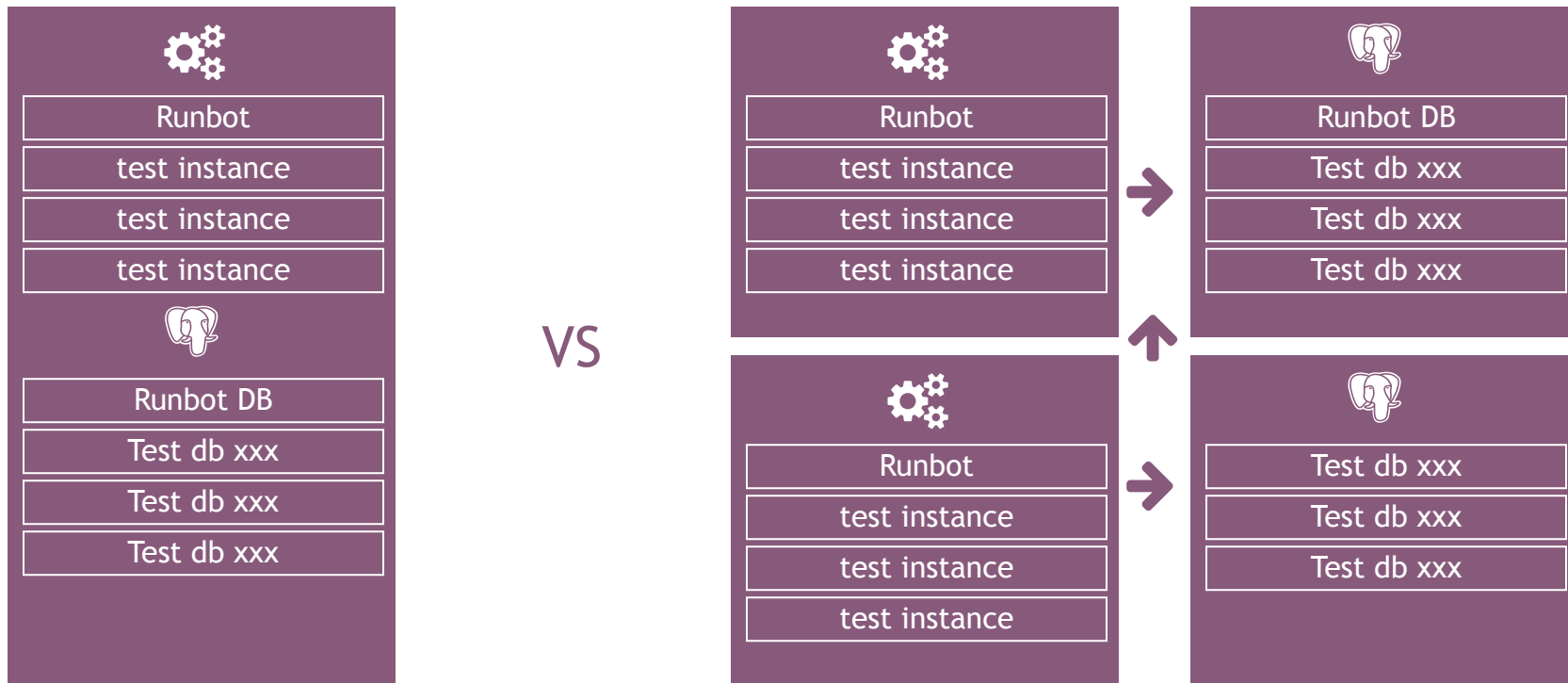
Step 4: External tools

Your runbot will need to test every of your business process, and will need access to every external library or software to be able to test it.

- DNS : redirect *.runbot.mycompany.com
- NGINX : runbot is running on port 8080
- Fake SMTP server (MailCatcher, ...)
- Any external software you link to (ftp, edi, ...)

Scaling

Most of the time, a single server do the job. When team grow, runbot scale.





3 Configure Runbot

How it's made

Server	Scheduler
✓ Repositories	✓ Update repositories
✓ Branches	✓ Create missing branches
✓ Builds	✓ Self assign builds
✓ Jobs	✓ Advance jobs (base, all, run)
	✓ Update NGINX config

Runbot

Repositories

Branches

Builds

Global parameters

Runbot

Settings

<input type="text" value="5"/>	Total Number of Workers
<input type="text" value="40"/>	Maximum Number of Running Builds
<input type="text" value="1800"/>	Default Timeout (in seconds)
<input type="text" value="2000"/>	Starting Port for Running Builds
<input type="text" value="runbot.mycompany.com"/>	Runbot Domain

- Parameters are shared by all instances
- Timeout vs Max CPU time
- Runbot domain vs FQDN

Add authentication

As runbot is generally behind a public IP, it's better to create groups and users

- Use groups per team, project, ...
- Add users in groups
- Manage default user profile

It's possible to use other authentication methods (nginx, ...)

Add your first repo : Odoo

- The easiest one
- Necessary for other repositories
- Find missing libraries
- Check Nginx
- Check smtp
- Check dns
- Check ssh access to GitHub
- Check PostgreSQL config
- Check system load
- **New since last year:** Webhook instead of poll

Add your first repo : Odoo

Repository

github.com:odoo/odoo

Update

Cron

Killall

Params

Mode

Hook

Nginx



Duplicate repo

Extra dependencies

Modules to install

Do not build



Other modules to install automatically

Repository modules (excluding dependencies)

Github token

Limited to groups

Last hook time

08/24/2016 12:49:25

Add your own repo : petstore

Repository

`git@github.com:nseinle`

Update

Cron

Killall

Params

Mode

Hook

Nginx



Duplicate repo

Extra dependencies

`git@github.com:odoo/odoo` `git@github.com:odoo/enterprise`

Modules to install

Do not build



Other modules to install automatically

Repository modules (excluding dependencies)

Github token

Limited to groups

`my_project`

Last hook time

Use branch name matching

- Same branch name
- A PR whose head name match
- Match a branch which is the dashed-prefix of current branch name
- Common ancestors (git merge-base)

Use sticky branches

- Always on top
- No build skipped
- Last build not killed

github.com/odoo/odoo ▾

runbot10.odoo.com: 0 testing, 75 running runbot5.odoo.com: 0 testing, 75 running runbot8.odoo.c

Branch		
★ 10.0 2h Branch or pull ↻ ▶▶	[FIX] sale: tour, remove tip whi... Denis Ledoux 174601-10-0-3babd2 on runbot5.odoo.com 👍 age 2h time 18m	[FIX] website: set the correct h... qsm-odoo 174582-10-0-b970ea on runbot9.odoo.com 👍 age 14h time 16m
★ 7.0 8d Branch or pull ↻ ▶▶	[I18N] Last 7.0 update translati... Martin Trigaux 173651-7-0-b431e6 on runbot8.odoo.com 👍 age 8d time 5m	[I18N] Last 7.0 update translati... Martin Trigaux 172941-7-0-b431e6 on runbot5.odoo.com 👍 age 13d time 4m

Link to

- Set GitHub token to use GitHub API
- Bi directional link



Create own
dashboard

A first simple custo

As a partner, I do not need to build Odoo, Odoo s.a. already do it, but

- Official Odoo repo needed as fallback
- Official Odoo repo can be used to test new features

Only build sticky branches of odoo official repo

Add a field, Override update_git

```
class runbot_repo(models.Model):
    _inherit = "runbot.repo"

    nobuild = fields.Boolean("Do not build")

    @api.model
    def update_git(self, repo):
        super(runbot_repo, self).update_git(repo=repo)
        if repo.nobuild:
            self.env['runbot.build'].search([('repo_id', '=', repo.id),
                                             ('state', '=', 'pending'),
                                             ('branch_id.sticky', '=', False)
                                             ]).write({'state': 'done', 'result': 'skipped'})
```

```
<?xml version="1.0" encoding="utf-8"?>
<openerp><data>
  <record id="view_repo_form" model="ir.ui.view">
    <field name="model">runbot.repo</field>
    <field name="inherit_id" ref="runbot.view_repo_form"/>
    <field name="arch" type="xml">
      <field name="modules" position="after">
        <field name="nobuild" />
      </field>
    </field>
  </record>
</data></openerp>
```

Add a new controller

```
class RunbotCustomController(RunbotController):
    @http.route(['/runbot/ir'], type='http', auth="public", website=True)
    def irdashboard(self, refresh=None, **post):
        [...]
        return request.render("runbot_ir.ir-dashboard", qctx)
```

And the corresponding template

```
<template id="ir-dashboard">
  <t t-call='website.layout'>
    <script id="status_bootstrap" type="text/javascript">
      (function () {
        "use strict";
        window['statuses'] = <t t-raw='json.dumps(statuses)'/>;
      }) ();
    </script>

    <t t-set="head">
      <t t-if="refresh">
        <meta http-equiv="refresh" t-att-content="refresh"/>
      </t>
    </t>
    <div class="container-fluid">
      <div class="row">
        <div class='col-md-3'>
          <div class="row">
            <div class='col-md-12'>
              <h1>Builds status</h1>
              <div id="status_chart"></div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </t>
</template>
```

Add a widget for a graph

```
(function () {  
    "use strict";  
  
    var website = openerp.website;  
    var _t = openerp._t;  
  
    var StatusPie = openerp.Widget.extend({  
        init: function (parent, options) {  
            this._super.apply(this, arguments);  
            this.statuses = options;  
        },  
        start: function () {  
            [...]  
        }  
    });  
})
```

Add libraries in assets, and a theme

```
<template id="theme_slate" name="Slate" inherit_id="website.theme" active="False" customize_show="True">
  <xpath expr="//link[@id='bootstrap_css']" position="replace">
    <link rel='stylesheet' href='/website/static/src/css/bootswatch/slate.min.css' t-ignore="true"/>
    <link rel='stylesheet' href='/website/static/src/css/bootswatch/slate.fix.css' t-ignore="true"/>
  </xpath>
</template>

<template id="assets_frontend" inherit_id="website.assets_frontend" name="irrunbot.assets.frontend">
  <xpath expr="." position="inside">
    <script type="text/javascript" src="/runbot_ir/static/src/js/runbotir.js"/>
    <script src="//d3js.org/d3.v3.min.js"></script>
  </xpath>
</template>
```

Final result

Code available on
https://github.com/nseinlet/runbot_ir





5 Conclusions

Your own runbot

- Deployable on your own infrastructure
- Linked to your GitHub account
- Access to your repos
- Part of the dev process
- Easy to use for everyone
- A communication tool
- A dev tool
- A quick test tool
- Customizable to fit your needs

Any question?



#odooexperience

Nicolas Seinlet (nse@odoo.com)

   @nseinlet