

Pallet

Node provisioning, configuration, command and control

About Pallet

- a library for servers
- uses jclouds
- docs are great ([main docs](#), [lots of examples](#), [uberdoc](#), [api](#))

Supported Providers

```
user=> (pallet.compute/supported-providers)
("hybrid" "openhosting-east1" "serverlove-z1-man" "cloudsigma-zrh" "aws-ec2"
"eucalyptus-partnercloud-ec2" "bluelock-vcloud-zone01" "bluelock-vcdirector"
"cloudservers-us" "elastichosts-lon-p" "skalicloud-sdg-my" "deltacloud" "nova" "trmk-
vcloudexpress" "eucalyptus" "bluelock-vcloud-vcenterprise" "elastichosts-lon-b"
"stratogen-vcloud-mycloud" "cloudservers-uk" "trmk-ecloud" "elastichosts-sat-p"
"slicehost" "gogrid" "stub" "vcloud" "elasticstack" "savvis-symphonyvpdc"
"rimuhosting" "byon" "node-list")
```

Full list at <http://www.jclouds.org/documentation/reference/supported-providers>

session

server state info

action

basic unit of configuration

crate

basically a collection of actions

phase

a set of actions you want to execute (takes in a session)

node-spec

describes the hardware/network/image

server-spec

describes phases for nodes

group-spec

node-specs with server-specs

Operations

converge

adjusts node counts

applies phases

lift

applies phases

```
(require 'pallet.repl)
(require 'pallet.crate.automated-admin-user)
```

```
(pallet.core/converge
  {my-group 2})
```

```
:compute (pallet.compute/service :aws))
```

```
(pallet.compute/nodes (pallet.compute/service :aws))
```

```
(require 'pallet.repl)
(require 'pallet.crate.automated-admin-user)
```

```
(pallet.core/converge
  {load-balancer 2
    web-app 4
    database 2})
```

```
:compute (pallet.compute/service :aws))
```

```
(pallet.compute/nodes (pallet.compute/service :aws))
```

```
(require 'pallet.repl)
(require 'pallet.crate.automated-admin-user)
```

```
(pallet.core/converge
  { (pallet.core/group-spec
      "web-app"
      :extends [with-riak with-wget]
```

```
      :node-spec my-nodes)
```

```
  2}
  :compute (pallet.compute/service :aws))
```

```
(pallet.compute/nodes (pallet.compute/service :aws))
```



```
(require 'pallet.repl)
(require 'pallet.crate.automated-admin-user)
```

```
(pallet.core/converge
  { (pallet.core/group-spec
      "web-app"
      :extends [with-riak with-wget]
```

```
      :node-spec (pallet.core/node-spec
                    :image {:image-id "us-east-1/ami-31814f58"}))
```

```
  2}
  :compute (pallet.compute/service :aws))
```

```
(pallet.compute/nodes (pallet.compute/service :aws))
```

```
(require 'pallet.repl)
(require 'pallet.crate.automated-admin-user)
```

```
(pallet.core/converge
  { (pallet.core/group-spec
      "web-app"
      :extends [with-riak with-wget]
```

```
      :node-spec (pallet.core/node-spec
                    :image { :os-family :ubuntu
                             :os-version-matches "10.10"
                             :os-64-bit true}
                    :hardware { :min-cores 2 :min-ram 512}
                    :network { :inbound-ports [22 80]})
      2}
      :compute (pallet.compute/service :aws))
```

```
(pallet.compute/nodes (pallet.compute/service :aws))
```

```

(require 'pallet.repl)
(require 'pallet.crate.automated-admin-user)

(pallet.core/converge
  { (pallet.core/group-spec
      "web-app"
      :extends [ (pallet.core/server-spec
                    :phases { :bootstrap (pallet.phase/phase-fn (pallet.crate.automated-
admin-user/automated-admin-user))
                          :configure (pallet.phase/phase-fn
                                      (pallet.action.package/package "curl")
                                      riak-action)) })

      with-wget]

      :node-spec (pallet.core/node-spec
                    :image { :os-family :ubuntu
                          :os-version-matches "10.10"
                          :os-64-bit true}
                    :hardware { :min-cores 2 :min-ram 512}
                    :network { :inbound-ports [22 80] })
      2}
    :compute (pallet.compute/service :aws))

(pallet.compute/nodes (pallet.compute/service :aws))

```

```

(require 'pallet.repl)
(require 'pallet.crate.automated-admin-user)

(pallet.core/converge
  { (pallet.core/group-spec
      "web-app"
      :extends [ (pallet.core/server-spec
                    :phases { :bootstrap (pallet.phase/phase-fn (pallet.crate.automated-
admin-user/automated-admin-user))
                          :configure (pallet.phase/phase-fn
                                        (pallet.action.package/package "curl")
                                        (pallet.action.exec-script/exec-script
                                          (mkdir "/tmp/riak-package")
                                          (cd "/tmp/riak-package")
                                          (wget "http://downloads.basho.com/riak/riak-1.0.2/riak\_1.0.2-1\_amd64.deb")
                                          (dpkg "-i" "riak_1.0.2-1_amd64.deb")))))]
      with-wget]

      :node-spec (pallet.core/node-spec
                    :image { :os-family :ubuntu
                          :os-version-matches "10.10"
                          :os-64-bit true}
                    :hardware { :min-cores 2 :min-ram 512}
                    :network { :inbound-ports [22 80]})
      2}
    :compute (pallet.compute/service :aws))

(pallet.compute/nodes (pallet.compute/service :aws))

```

```

(require 'pallet.repl)
(require 'pallet.crate.automated-admin-user)

(pallet.core/converge
  { (pallet.core/group-spec
      "web-app"
      :extends [ (pallet.core/server-spec
                    :phases { :bootstrap (pallet.phase/phase-fn (pallet.crate.automated-
admin-user/automated-admin-user))
                          :configure (pallet.phase/phase-fn
                                        (pallet.action.package/package "curl")
                                        (pallet.action.exec-script/exec-script
                                          (mkdir "/tmp/riak-package")
                                          (cd "/tmp/riak-package")
                                          (wget "http://downloads.basho.com/riak/riak-1.0.2/riak\_1.0.2-1\_amd64.deb")
                                          (dpkg "-i" "riak_1.0.2-1_amd64.deb")))))]
                    (pallet.core/server-spec
                      :phases { :configure (fn [session]
                                              (->
                                                session
                                                (pallet.action.package/package "wget")))))]
      :node-spec (pallet.core/node-spec
                    :image { :os-family :ubuntu
                          :os-version-matches "10.10"
                          :os-64-bit true}
                    :hardware { :min-cores 2 :min-ram 512}
                    :network { :inbound-ports [22 80]})
                    2}
      :compute (pallet.compute/service :aws))

(pallet.compute/nodes (pallet.compute/service :aws))

```

Configuration

```
; in ~/.pallet/config.clj
(defpallet
  :services {:aws {:provider "aws-ec2"
                    :identity "key"
                    :credential "secret-key"
                    :environment {:user
                                  { :username "bmaddy"
                                    :private-key-path "/Users/bmaddy/.ssh/id_rsa"
                                    :public-key-path  "/Users/bmaddy/.ssh/id_rsa.pub" } } } } })
```

(demo)



VS.



images borrowed from <http://earth911.com/recycling/construction/pallets/> and <http://mytornadoalley.com/tag/swedish-chef/>

Things people are doing with Pallet

- Building full environments out of a data structure
- Deployments with Hudson
- Setup ephemeral cloud clusters

Thanks @tbatchelli

Who's using it?

- Twitter
- Paper.il
- EMI Music
- LonoCloud
- Others

More info

palletops.com

Questions?

Brian Maddy
@bmaddy