Heterogeneous Cloud Computing for Research

Managing Cloud Services with ConPaaS

Guillaume Pierre

http://www.conpaas.eu/

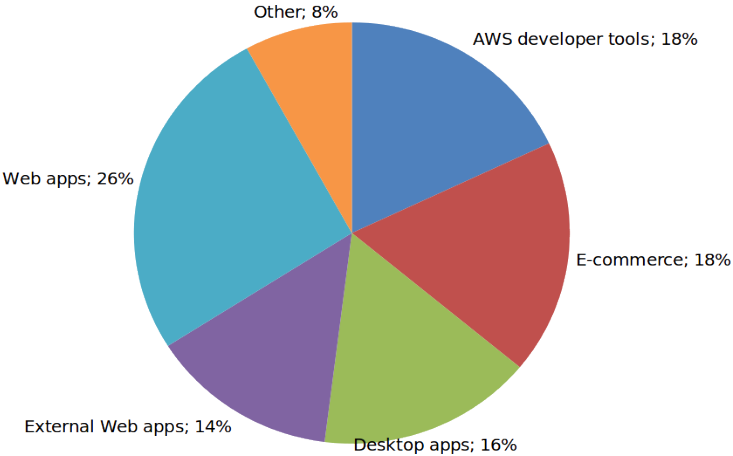
Table of Contents

1. Introduction
2. ConPaaS usage
3. ConPaaS architecture
4. Conclusion

Typical Cloud Applications (according to AWS)

- Application Hosting
- Backup and Storage
- Content Delivery
- E-Commerce
- High Performance Computing
- Media Hosting
- On-Demand Workforce
- Search Engines
- Web Hosting

Applications running at Amazon Web Services



Sample: 50 applications from the AWS Customer App Catalog.

Many Cloud applications are alike

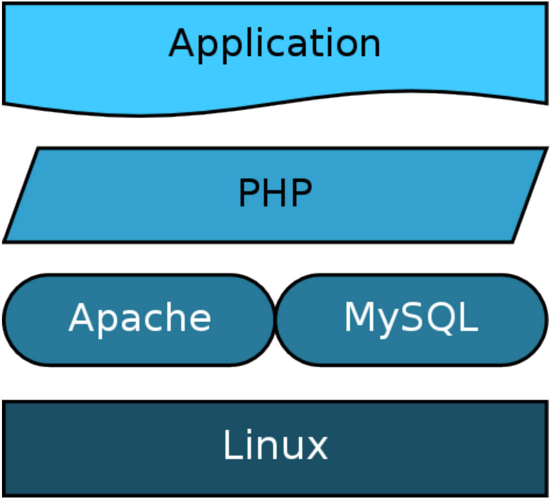
- Many cloud applications are alike
  - Web servers
  - Application servers
  - Database servers
  - High-performance frameworks (MapReduce, MPI, Workflows)
  - . . . and every application has something *different*
- To be deployed on a wide variety of cloud resources

Cloud application developers often rebuild
the same types of frameworks again and again and again. . .

Deploying an application in the Cloud can be easy

1. Choose a cloud provider
2. Start one virtual machine using a ready-made image
3. Install your software in the virtual machine
4. Snapshot the VM

What about a slightly more complex application?



Monitoring
Software upgrades
Fault tolerance
Data safety
M y S Q
PHP
Linux
Auto-scaling
Context-ualization
Xen
KVM
PLANETARY DESIGN,LLC
MISSOULA,MT

Can the Cloud help support common types of applications?

- Infrastructure-as-a-Service provides basic computing resources
  - Absolute flexibility: you can build anything you want
  - But it can be very complex and time consuming
    - Deployment
    - Software upgrades
    - Fault-tolerance
    - Performance monitoring
    - Resource provisioning
    - Dynamic reconfiguration orchestration
    - etc.
- Platform-as-a-Service provides high-level services
  - Each PaaS service targets a specific family of applications
  - Provide a simple deployment environment for applications
  - Provide high-level guarantees for applications using these services

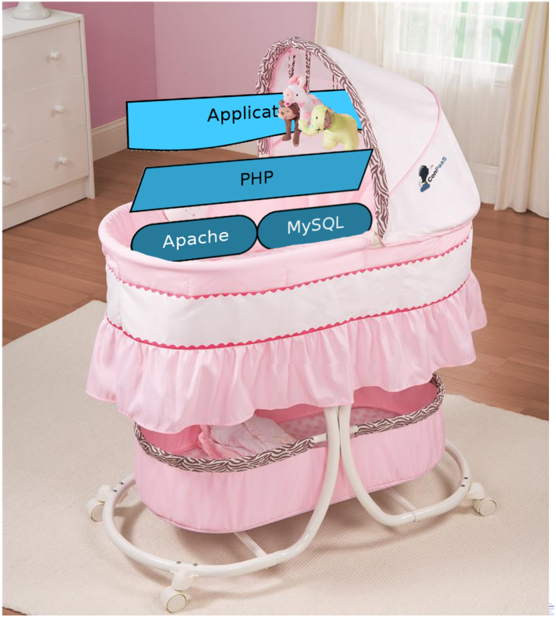
Main PaaS environments

- Google AppEngine
- Microsoft Azure
- Amazon Beanstalk
- RightScale
- OpenShift
- ConPaaS
- etc.

Different PaaS environments address different types of applications

ConPaaS takes care of your applications

ConPaaS takes care of your applications



ConPaaS in a nutshell

- Broad range of functionalities
  - Application servers, databases, high-performance computing, other
- Fully integrated
  - Applications can compose any set of services together
- Easy to use but also very powerful
  - Simple Web GUI + powerful command-line tool
  - Services are highly customizable
- Cutting-edge SLA enforcement technologies
  - Elasticity and resource provisioning techniques to guarantee performance at the lowest possible cost
- Runs over Amazon EC2, OpenNebula, OpenStack and *HARNESS, of course :-)*
- Open-source

`http://www.conpaas.eu`

Table of Contents

1 Introduction

2 ConPaaS usage

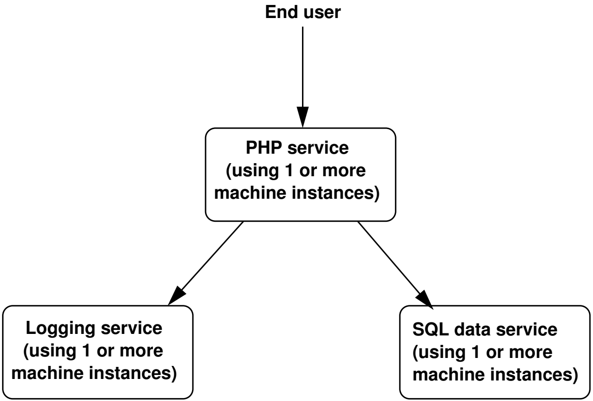
3 ConPaaS architecture

4 Conclusion

ConPaaS Applications

A ConPaaS application is defined as a composition of multiple service instances

- For example: web hosting service + MySQL database + logging service (to store access logs)


End user
PHP service (using 1 or more machine instances)
Logging service (using 1 or more machine instances)
SQL data service (using 1 or more machine instances)

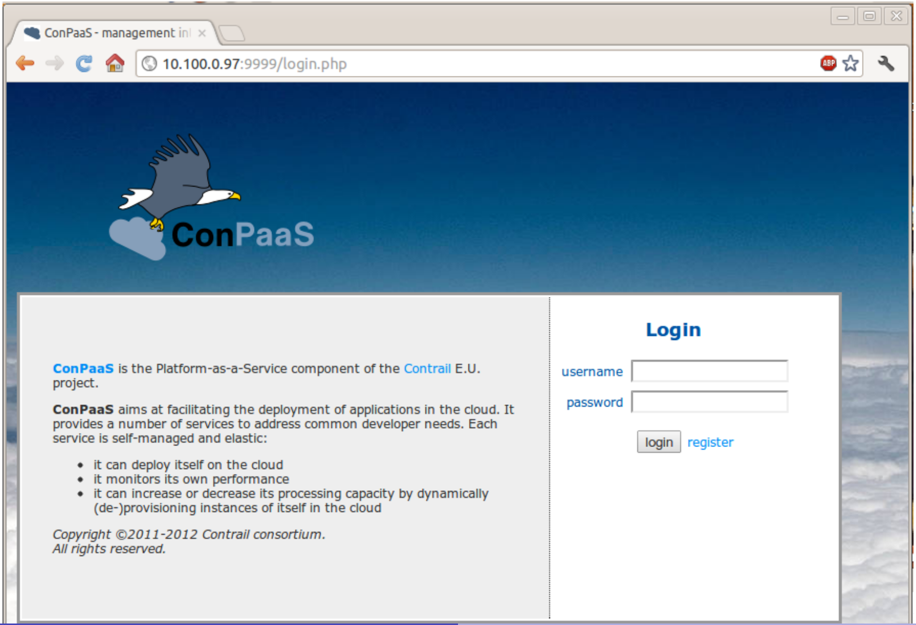
ConPaaS Services

- Web hosting (PHP, Java)
- Databases (MySQL, Scalarix)
- File system (XtreemFS)
- High-performance computations (MapReduce, TaskFarming)
- Functional testing (Selenium)

ConPaaS Operation

- Users access ConPaaS thanks to a Web interface
  - Login
  - Start new services (i.e., start a standard VM image with the service implementation)
  - Manage existing services (i.e., communicate with the service's manager to issue commands)
  - Stop services (i.e., stop all service instances except the service manager)
  - Terminate services (i.e., destroy a service completely)
- An extended set of functionalities is available through a command-line interface
  - All commands from the Web interface are available (except starting a new service)
  - Additional commands may be implemented for expert users
  - The command-line interface makes it easy to script service management

The ConPaaS Front-End



ConPaaS is the Platform-as-a-Service component of the Contrail E.U. project.

ConPaaS aims at facilitating the deployment of applications in the cloud. It provides a number of services to address common developer needs. Each service is self-managed and elastic:

- it can deploy itself on the cloud
- it monitors its own performance
- it can increase or decrease its processing capacity by dynamically (de-)provisioning instances of itself in the cloud

Copyright ©2011-2012 Contrail consortium.
All rights reserved.

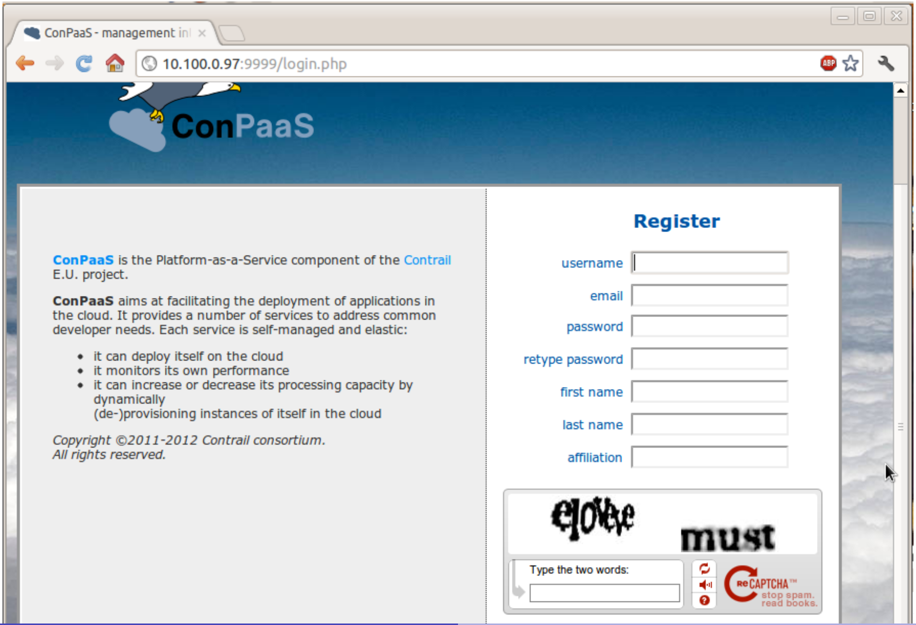
Login

username

password

login register

The ConPaaS Front-End



ConPaaS is the Platform-as-a-Service component of the Contrail E.U. project.

ConPaaS aims at facilitating the deployment of applications in the cloud. It provides a number of services to address common developer needs. Each service is self-managed and elastic:

- it can deploy itself on the cloud
- it monitors its own performance
- it can increase or decrease its processing capacity by dynamically (de-)provisioning instances of itself in the cloud

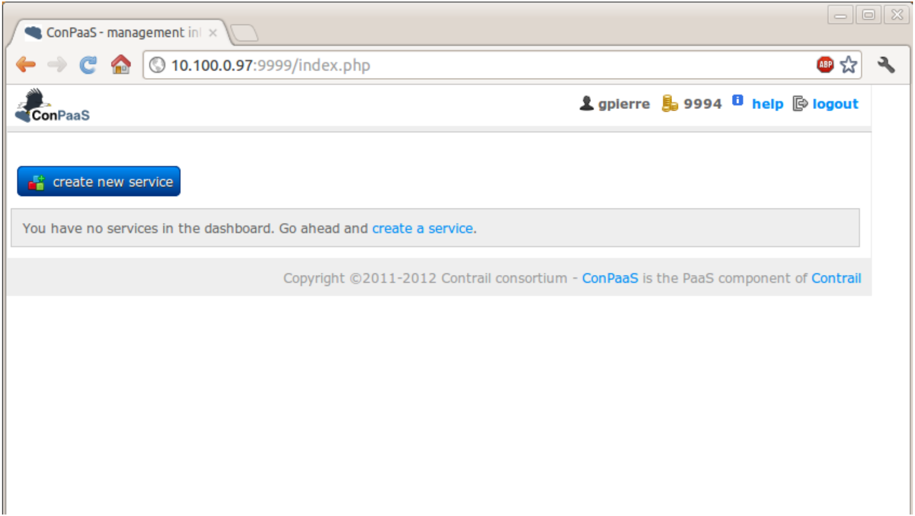
Copyright ©2011-2012 Contrail consortium.
All rights reserved.

Register

username
email
password
retype password
first name
last name
affiliation

Type the two words:

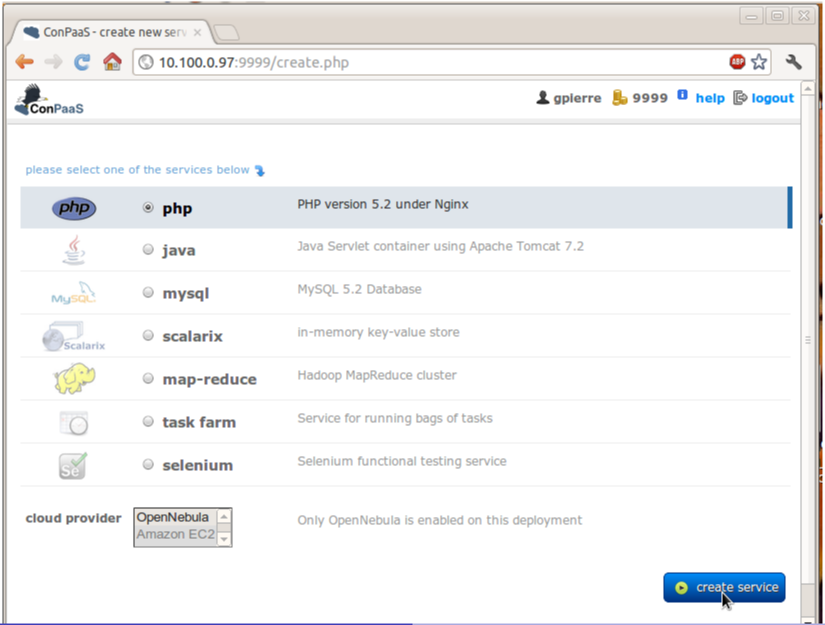
The ConPaaS Front-End



You have no services in the dashboard. Go ahead and create a service.

Copyright ©2011-2012 Contrail consortium - ConPaaS is the PaaS component of Contrail

The ConPaaS Front-End



ConPaaS - create new serv

10.100.0.97:9999/create.php

ConPaaS

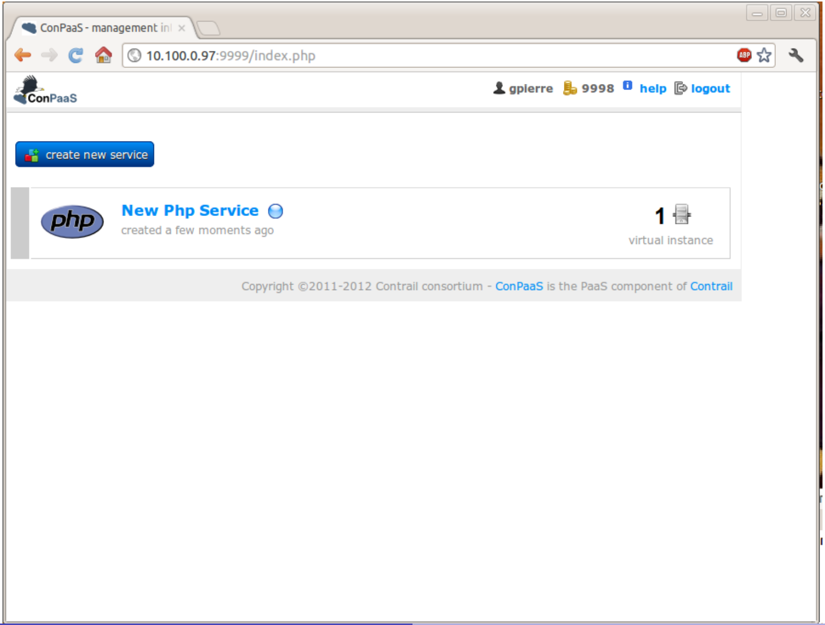
gpierre 9999 help logout

please select one of the services below

php	● php	PHP version 5.2 under Nginx
	○ java	Java Servlet container using Apache Tomcat 7.2
MySQL	○ mysql	MySQL 5.2 Database
Scalarix	○ scalarix	in-memory key-value store
	○ map-reduce	Hadoop MapReduce cluster
	○ task farm	Service for running bags of tasks
Se	○ selenium	Selenium functional testing service

cloud provider: OpenNebula / Amazon EC2 — Only OpenNebula is enabled on this deployment

The ConPaaS Front-End



ConPaaS - management in

10.100.0.97:9999/index.php

ConPaaS

gpierre 9998 help logout

create new service

New Php Service

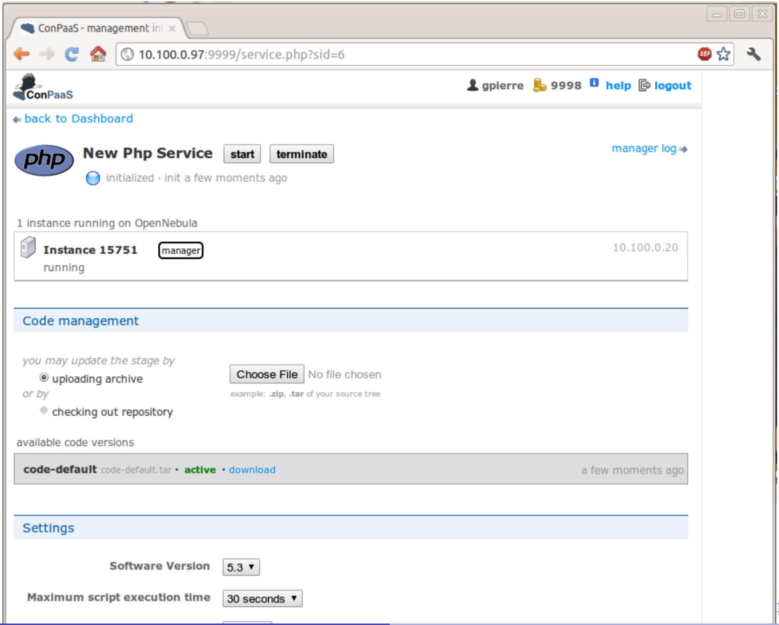
created a few moments ago

1

virtual instance

Copyright ©2011-2012 Contrail consortium - ConPaaS is the PaaS component of Contrail

The ConPaaS Front-End



New Php Service

initialized · init a few moments ago

1 instance running on OpenNebula

Instance 15751 manager running 10.100.0.20

Code management

you may update the stage by

● uploading archive

or by

○ checking out repository

Choose File No file chosen

example: .zip, .tar of your source tree

available code versions

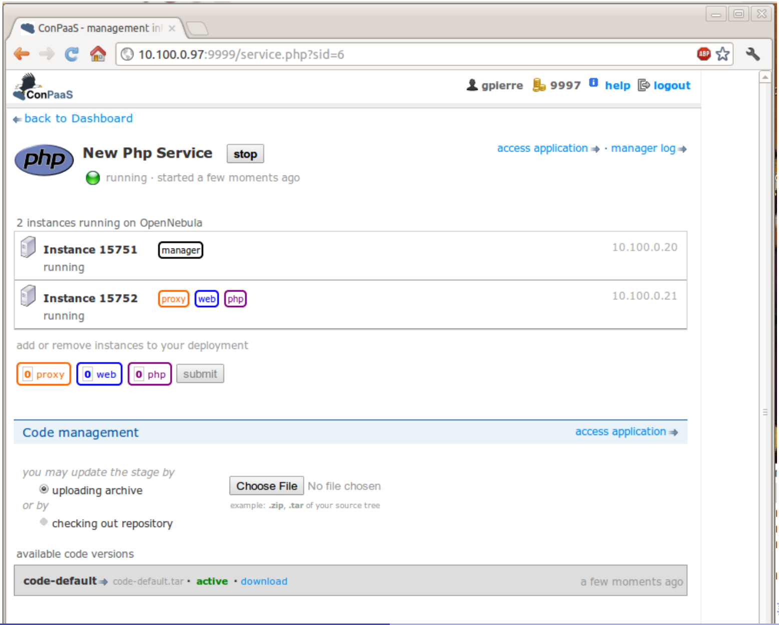
code-default code-default.tar • active • download a few moments ago

Settings

Software Version: 5.3

Maximum script execution time: 30 seconds

The ConPaaS Front-End



ConPaaS - management in ×

10.100.0.97:9999/service.php?sid=6

ConPaaS

gpierre 9997 help logout

back to Dashboard

New Php Service stop

running · started a few moments ago

access application · manager log

2 instances running on OpenNebula

Instance 15751 manager running		10.100.0.20
Instance 15752 proxy web php running		10.100.0.21

add or remove instances to your deployment

0 proxy 0 web 0 php submit

Code management

access application

you may update the stage by

● uploading archive

or by

○ checking out repository

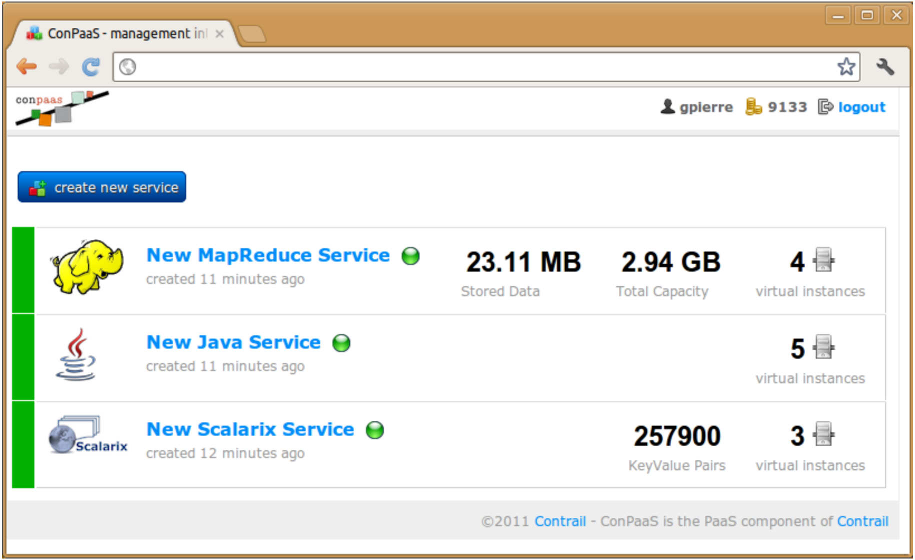
Choose File No file chosen

example: **.zip**, **.tar** of your source tree

available code versions

code-default code-default.tar · **active** · download a few moments ago

The ConPaaS Front-End



New MapReduce Service ●
created 11 minutes ago

23.11 MB Stored Data | 2.94 GB Total Capacity | 4 virtual instances

New Java Service ●
created 11 minutes ago

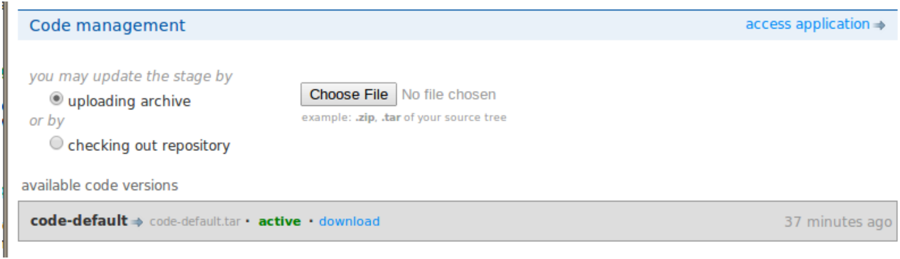
5 virtual instances

New Scalarix Service ●
created 12 minutes ago

257900 KeyValue Pairs | 3 virtual instances

©2011 Contrail - ConPaaS is the PaaS component of Contrail

Uploading new application versions



Code management

access application

you may update the stage by

● uploading archive

or by

○ checking out repository

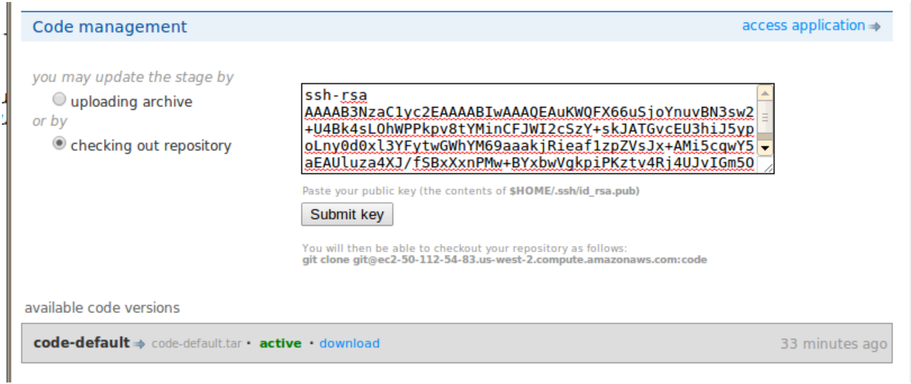
Choose File No file chosen

example: .zip, .tar of your source tree

available code versions

code-default code-default.tar • **active** • download 37 minutes ago

Uploading new application versions



Code management

access application ➜

you may update the stage by

○ uploading archive

or by

● checking out repository

```
ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAQEAuKWQFX66uSjoYnuvBN3sw2
+U4Bk4sLOhWPPkpv8tYMinCFJWI2cSzY+skJATGvcEU3hiJ5yp
oLny0d0xl3YFytwGWhYM69aaakjRieaf1zpZVsJx+AMi5cqwY5
aEAUluza4XJ/fSBxXxnPMw+BYxbwVgkpiPKztv4Rj4UJvIGm5O
```

Paste your public key (the contents of **$HOME/.ssh/id_rsa.pub**)

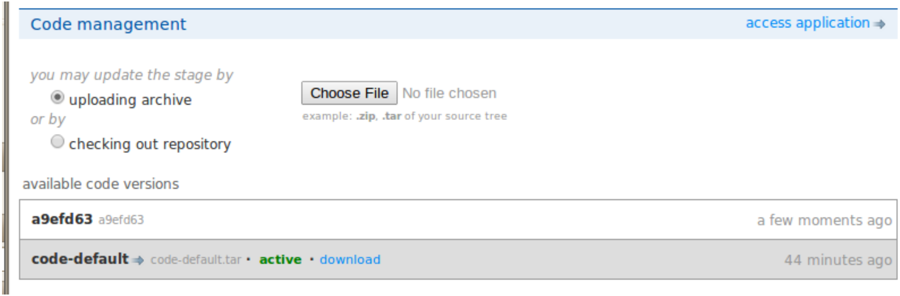
Submit key

You will then be able to checkout your repository as follows:
git clone git@ec2-50-112-54-83.us-west-2.compute.amazonaws.com:code

available code versions

code-default ➜ code-default.tar • **active** • download — 33 minutes ago

Uploading new application versions



Code management

access application

you may update the stage by

● uploading archive

or by

○ checking out repository

Choose File No file chosen

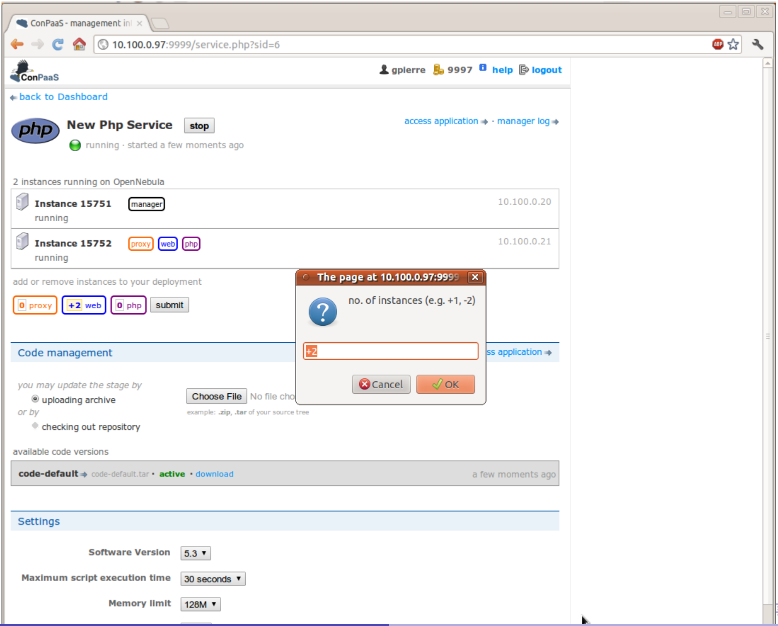
example: **.zip**, **.tar** of your source tree

available code versions

a9efd63 a9efd63 a few moments ago

code-default code-default.tar · **active** · download 44 minutes ago

Scaling a service



ConPaaS - management in ×

10.100.0.97:9999/service.php?sid=6

ConPaaS

gpierre 9997 help logout

back to Dashboard

New Php Service stop

running · started a few moments ago

access application · manager log

2 instances running on OpenNebula

Instance 15751 manager running 10.100.0.20

Instance 15752 proxy web php running 10.100.0.21

add or remove instances to your deployment

0 proxy +2 web 0 php submit

The page at 10.100.0.97:9999

no. of instances (e.g. +1, -2)

+2

Cancel OK

Code management

you may update the stage by

● uploading archive

or by

○ checking out repository

Choose File No file cho

example: .zip, .tar of your source tree

available code versions

code-default code-default.tar • active • download a few moments ago

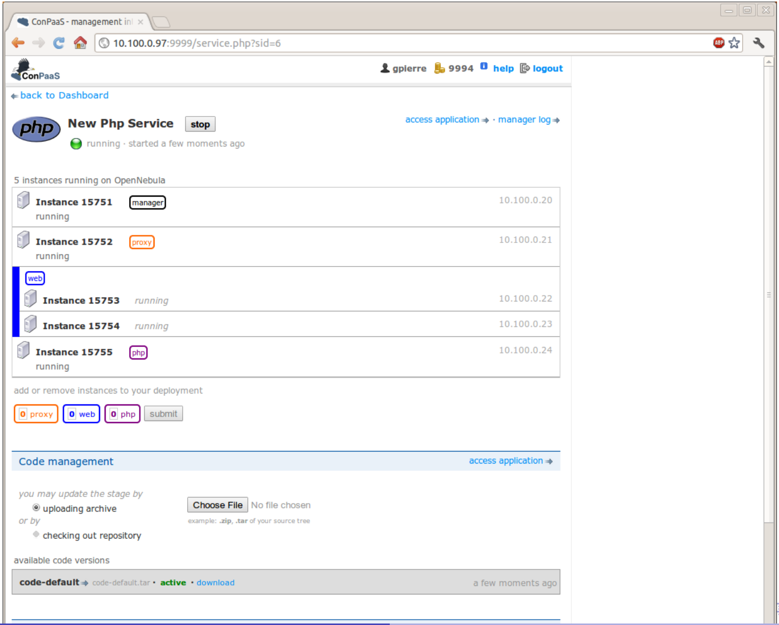
Settings

Software Version 5.3

Maximum script execution time 30 seconds

Memory limit 128M

Scaling a service



New Php Service

running · started a few moments ago

5 instances running on OpenNebula

Instance 15751 manager running		10.100.0.20
Instance 15752 proxy running		10.100.0.21
web		
Instance 15753	*running*	10.100.0.22
Instance 15754	*running*	10.100.0.23
Instance 15755 php running		10.100.0.24

add or remove instances to your deployment

Code management

you may update the stage by

● uploading archive

or by

○ checking out repository

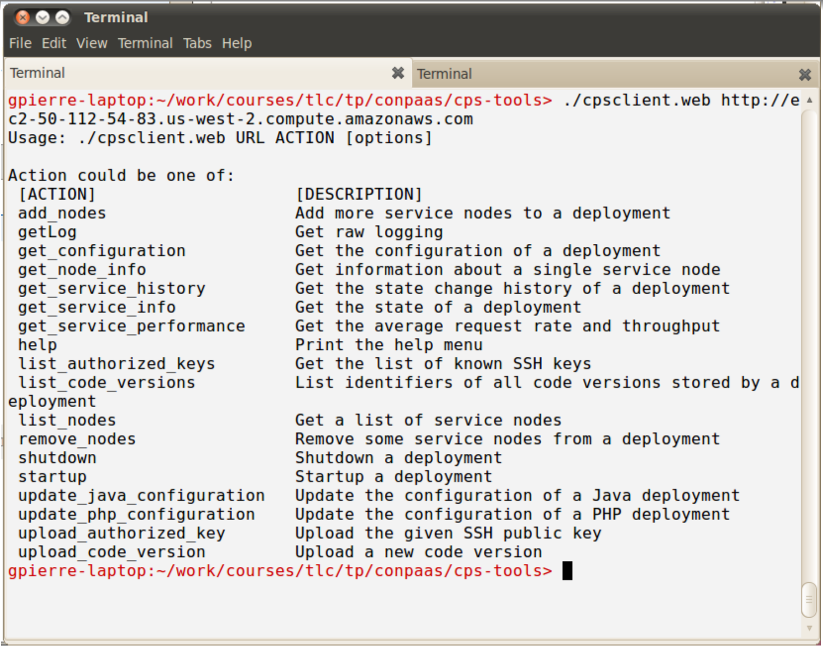
No file chosen

example: .zip, .tar of your source tree

available code versions

code-default code-default.tar • **active** • download — a few moments ago

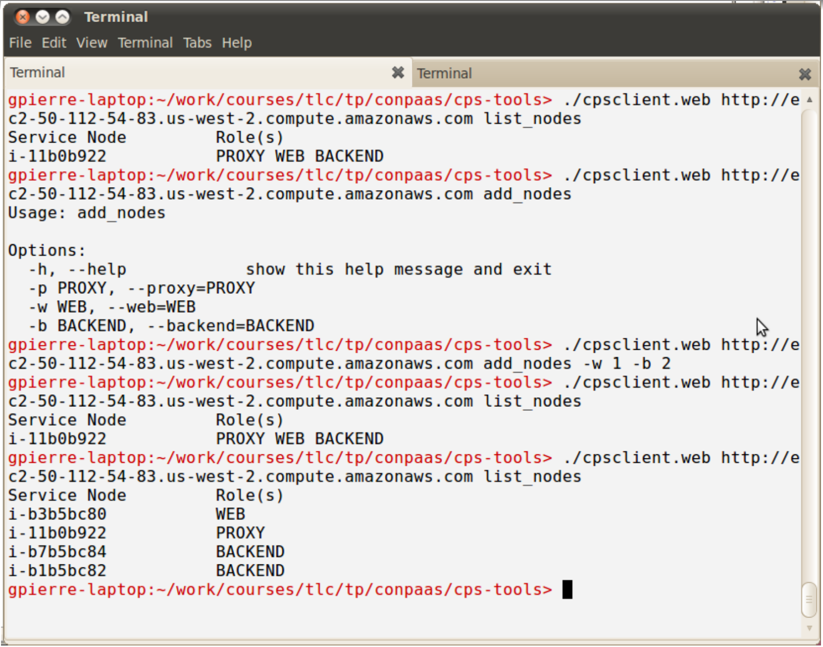
The ConPaaS command-line interface



```
gpierre-laptop:~/work/courses/tlc/tp/conpaas/cps-tools> ./cpsclient.web http://e
c2-50-112-54-83.us-west-2.compute.amazonaws.com
Usage: ./cpsclient.web URL ACTION [options]

Action could be one of:
 [ACTION]                      [DESCRIPTION]
 add_nodes                     Add more service nodes to a deployment
 getLog                        Get raw logging
 get_configuration             Get the configuration of a deployment
 get_node_info                 Get information about a single service node
 get_service_history           Get the state change history of a deployment
 get_service_info              Get the state of a deployment
 get_service_performance       Get the average request rate and throughput
 help                          Print the help menu
 list_authorized_keys          Get the list of known SSH keys
 list_code_versions            List identifiers of all code versions stored by a d
eployment
 list_nodes                    Get a list of service nodes
 remove_nodes                  Remove some service nodes from a deployment
 shutdown                      Shutdown a deployment
 startup                       Startup a deployment
 update_java_configuration     Update the configuration of a Java deployment
 update_php_configuration      Update the configuration of a PHP deployment
 upload_authorized_key         Upload the given SSH public key
 upload_code_version           Upload a new code version
gpierre-laptop:~/work/courses/tlc/tp/conpaas/cps-tools> █
```

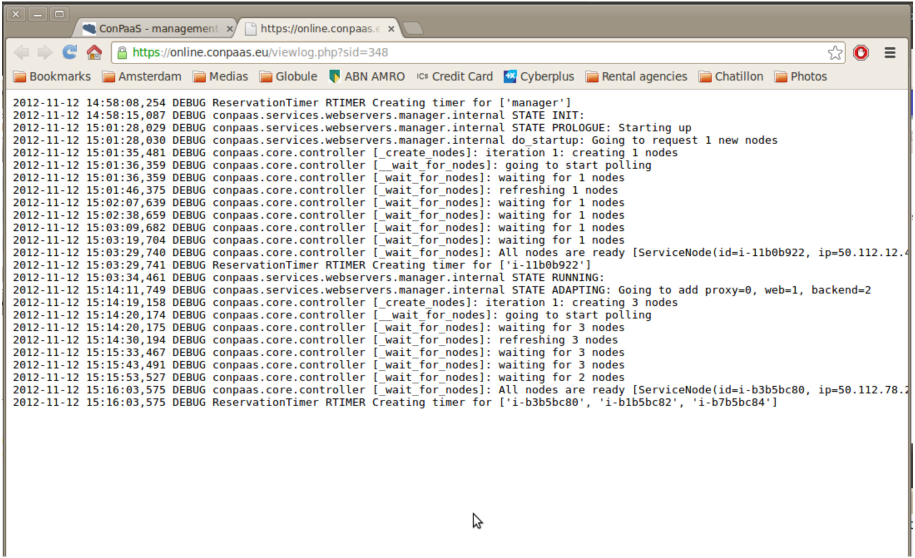
The ConPaaS command-line interface



```
gpierre-laptop:~/work/courses/tlc/tp/conpaas/cps-tools> ./cpsclient.web http://e
c2-50-112-54-83.us-west-2.compute.amazonaws.com list_nodes
Service Node          Role(s)
i-11b0b922            PROXY WEB BACKEND
gpierre-laptop:~/work/courses/tlc/tp/conpaas/cps-tools> ./cpsclient.web http://e
c2-50-112-54-83.us-west-2.compute.amazonaws.com add_nodes
Usage: add_nodes

Options:
  -h, --help            show this help message and exit
  -p PROXY, --proxy=PROXY
  -w WEB, --web=WEB
  -b BACKEND, --backend=BACKEND
gpierre-laptop:~/work/courses/tlc/tp/conpaas/cps-tools> ./cpsclient.web http://e
c2-50-112-54-83.us-west-2.compute.amazonaws.com add_nodes -w 1 -b 2
gpierre-laptop:~/work/courses/tlc/tp/conpaas/cps-tools> ./cpsclient.web http://e
c2-50-112-54-83.us-west-2.compute.amazonaws.com list_nodes
Service Node          Role(s)
i-11b0b922            PROXY WEB BACKEND
gpierre-laptop:~/work/courses/tlc/tp/conpaas/cps-tools> ./cpsclient.web http://e
c2-50-112-54-83.us-west-2.compute.amazonaws.com list_nodes
Service Node          Role(s)
i-b3b5bc80            WEB
i-11b0b922            PROXY
i-b7b5bc84            BACKEND
i-b1b5bc82            BACKEND
gpierre-laptop:~/work/courses/tlc/tp/conpaas/cps-tools> 
```

The ConPaaS command-line interface



```
2012-11-12 14:58:08,254 DEBUG ReservationTimer RTIMER Creating timer for ['manager']
2012-11-12 14:58:15,087 DEBUG conpaas.services.webservers.manager.internal STATE INIT:
2012-11-12 15:01:28,029 DEBUG conpaas.services.webservers.manager.internal STATE PROLOGUE: Starting up
2012-11-12 15:01:28,030 DEBUG conpaas.services.webservers.manager.internal do_startup: Going to request 1 new nodes
2012-11-12 15:01:35,481 DEBUG conpaas.core.controller [_create_nodes]: iteration 1: creating 1 nodes
2012-11-12 15:01:36,359 DEBUG conpaas.core.controller [__wait_for_nodes]: going to start polling
2012-11-12 15:01:36,359 DEBUG conpaas.core.controller [_wait_for_nodes]: waiting for 1 nodes
2012-11-12 15:01:46,375 DEBUG conpaas.core.controller [_wait_for_nodes]: refreshing 1 nodes
2012-11-12 15:02:07,639 DEBUG conpaas.core.controller [_wait_for_nodes]: waiting for 1 nodes
2012-11-12 15:02:38,659 DEBUG conpaas.core.controller [_wait_for_nodes]: waiting for 1 nodes
2012-11-12 15:03:09,682 DEBUG conpaas.core.controller [_wait_for_nodes]: waiting for 1 nodes
2012-11-12 15:03:19,704 DEBUG conpaas.core.controller [_wait_for_nodes]: waiting for 1 nodes
2012-11-12 15:03:29,740 DEBUG conpaas.core.controller [_wait_for_nodes]: All nodes are ready [ServiceNode(id=i-11b0b922, ip=50.112.12.4
2012-11-12 15:03:29,741 DEBUG ReservationTimer RTIMER Creating timer for ['i-11b0b922']
2012-11-12 15:03:34,461 DEBUG conpaas.services.webservers.manager.internal STATE RUNNING:
2012-11-12 15:14:11,749 DEBUG conpaas.services.webservers.manager.internal STATE ADAPTING: Going to add proxy=0, web=1, backend=2
2012-11-12 15:14:19,158 DEBUG conpaas.core.controller [_create_nodes]: iteration 1: creating 3 nodes
2012-11-12 15:14:20,174 DEBUG conpaas.core.controller [__wait_for_nodes]: going to start polling
2012-11-12 15:14:20,175 DEBUG conpaas.core.controller [_wait_for_nodes]: waiting for 3 nodes
2012-11-12 15:14:30,194 DEBUG conpaas.core.controller [_wait_for_nodes]: refreshing 3 nodes
2012-11-12 15:15:33,467 DEBUG conpaas.core.controller [_wait_for_nodes]: waiting for 3 nodes
2012-11-12 15:15:43,491 DEBUG conpaas.core.controller [_wait_for_nodes]: waiting for 3 nodes
2012-11-12 15:15:53,527 DEBUG conpaas.core.controller [_wait_for_nodes]: waiting for 2 nodes
2012-11-12 15:16:03,575 DEBUG conpaas.core.controller [_wait_for_nodes]: All nodes are ready [ServiceNode(id=i-b3b5bc80, ip=50.112.78.2
2012-11-12 15:16:03,575 DEBUG ReservationTimer RTIMER Creating timer for ['i-b3b5bc80', 'i-b1b5bc82', 'i-b7b5bc84']
```

Table of Contents

1 Introduction

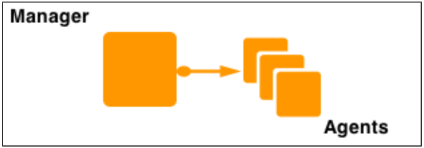
2 ConPaaS usage

3 ConPaaS architecture

4 Conclusion

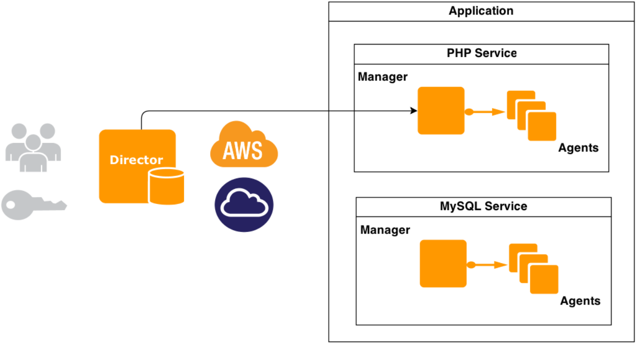
ConPaaS Services

- A ConPaaS service is composed of 1 Manager and *n* Agents
  - One Python class running in the Manager VM
  - One Python class running in the Agent VM



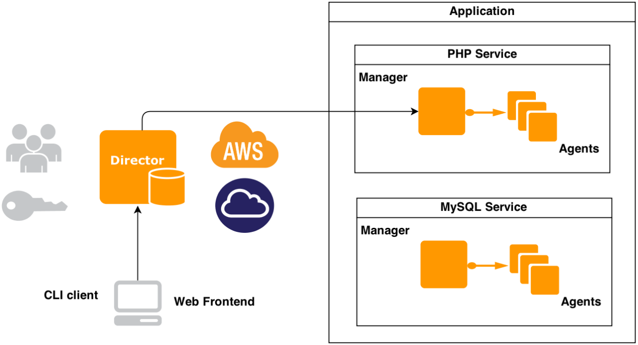
The ConPaaS Director

- The single entry point to the system
- Keeps tracks of users, credentials, services
- Handles the life-cycle of ConPaaS applications

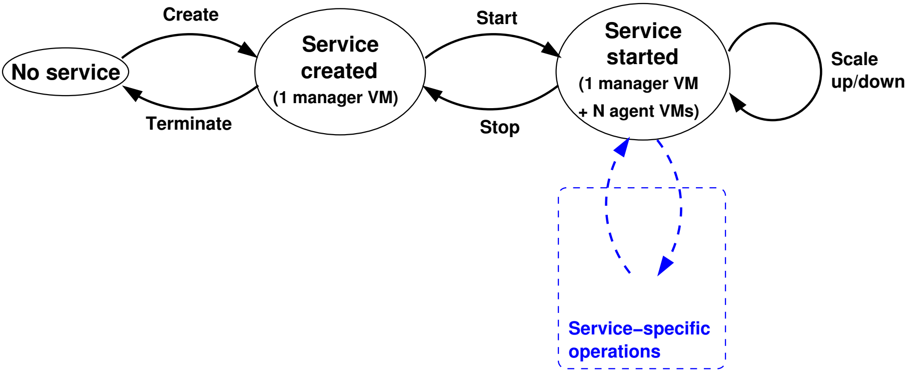


ConPaaS clients

- Clients make it easy to send commands to the Director
  - Command-line interface
  - Web-based GUI

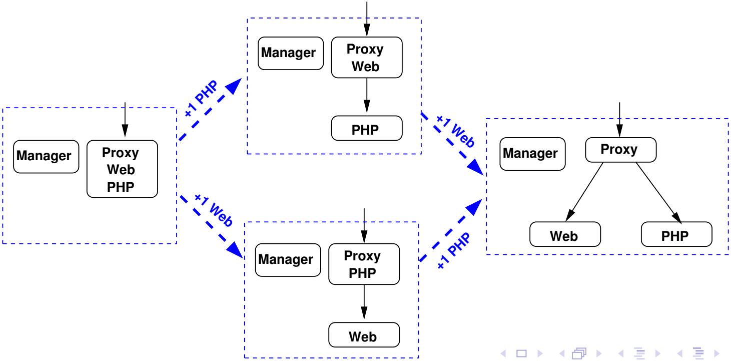


Lifecycle of a ConPaaS service



Example: the Web hosting service

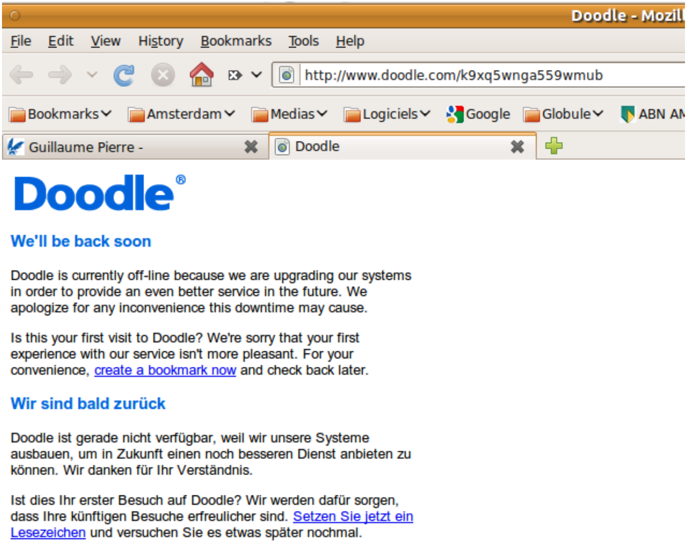
- The service exists in two versions: PHP and Java
- Initially the service has 2 VMs
  - 1 VM running the manager
  - 1 VM running a load balancer, a web server and a PHP backend
- When adding VMs each VM becomes specialized (load balancer VMs, web server VMs, PHP backend VMs)



Session handling in the PHP service

- PHP has built-in support for sessions
  - Normally stored in main memory of the PHP server
- We must share session state between multiple PHP backends (otherwise users would logout at each request)
  - We use the Scalaris key-value store for that
  - One Scalaris server inside the manager VM
- Making use of the Scalaris session storage is totally transparent to the applications

What we want to avoid



Doodle - Mozil
File Edit View History Bookmarks Tools Help
http://www.doodle.com/k9xq5wnga559wmub
Bookmarks Amsterdam Medias Logiciels Google Globule ABN AM
Guillaume Pierre -
Doodle

Doodle®

We'll be back soon

Doodle is currently off-line because we are upgrading our systems in order to provide an even better service in the future. We apologize for any inconvenience this downtime may cause.

Is this your first visit to Doodle? We're sorry that your first experience with our service isn't more pleasant. For your convenience, create a bookmark now and check back later.

Wir sind bald zurück

Doodle ist gerade nicht verfügbar, weil wir unsere Systeme ausbauen, um in Zukunft einen noch besseren Dienst anbieten zu können. Wir danken für Ihr Verständnis.

Ist dies Ihr erster Besuch auf Doodle? Wir werden dafür sorgen, dass Ihre künftigen Besuche erfreulicher sind. Setzen Sie jetzt ein Lesezeichen und versuchen Sie es etwas später nochmal.

Service reconfiguration

- When the user scales the service up:
  1. The front-end sends a request to the service manager to scale up
  2. The service manager creates a new VM with proper contextualization information, then starts polling
  3. The agent VM boots, then starts its manager process
  4. When the manager establishes a connection with the agent, it requests it to start one or more roles
  5. The manager uploads code/data as necessary
  6. The manager reconfigures other VMs as necessary

- When scaling down:
  - Same story in opposite order

Building new ConPaaS services

- Building new ConPaaS services from scratch was HARD
  - Build a proper VM image with contextualization
  - Develop new manager and agent deamons
  - Implement a standardized protocol between the front-end and the agents
  - All communication goes over SSL with custom security checks

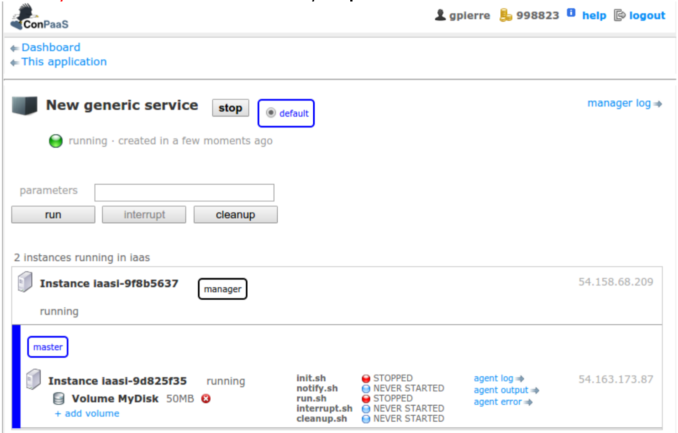
- Solution: the service core
  - All ConPaaS services use a single VM image
  - All ConPaaS services use the same manager and agent deamons
  - The service core implements shared functionality between all services
    - Start/stop/contextualize virtual machines
    - Secure communication primitives
    - Performance monitoring mechanisms
  - Each service can specialize the service core
    - Implement the service-specific parts

Structure of a service implementation

- Building a new ConPaaS service from the service core is now EASY:
  - (optional) Provide shell scripts to be executed when VMs start and stop
  - Write a manager and an agent class in Python
  - Extend one Python file to register the new service
  - Extend the front-end with one service-specific page in PHP
- Adding functionality to all services is quite easy as well
  - Secure internal communication
  - Support for new clouds
  - Performance monitoring

Our latest baby: the *Generic service*

- Package *any* application, provide 5 scripts:
  - init.sh: install the application
  - run.sh: start the application
  - interrupt.sh: stop the application
  - cleanup.sh: uninstall the application
  - notify.sh: handle the arrival/departure of another VM in this service



ConPaaS

gpierre 998823 help logout

Dashboard
This application

New generic service stop default

manager log

running · created in a few moments ago

parameters

run interrupt cleanup

2 instances running in iaas

Instance iaasi-9f8b5637 manager 54.158.68.209

running

master

Instance iaasi-9d825f35 running 54.163.173.87

Volume MyDisk 50MB

+ add volume

init.sh	STOPPED
notify.sh	NEVER STARTED
run.sh	STOPPED
interrupt.sh	NEVER STARTED
cleanup.sh	NEVER STARTED

agent log
agent output
agent error

Table of Contents

1 Introduction

2 ConPaaS usage

3 ConPaaS architecture

4 Conclusion

Conclusion

- ConPaaS is a platform-as-a-service environment
  - Designed to facilitate elastic application hosting in the cloud
  - Designed to be easily extensible

- ConPaaS addresses two major classes of applications:
  - Web applications
  - Scientific applications
  - Combinations of both

- We will use the public ConPaaS installation for practical assignments:

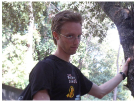
`https://online.conpaas.eu`

Credits

Ismail El Helw (services, managers, agents)

Adriana Szekeres (security)

Emanuele Rocca (director, networking)



Francesco Allertsen (manifests)

Claudiu Gheorghe (web frontend)

Yann Radenac (multi-cloud)

Genc Tato (nutshell, app managers)

Teodor Crivăț (nutshell, disk volumes)