

# ConPaaS: an integrated runtime environment for elastic cloud applications

Guillaume Pierre  
Vrije Universiteit Amsterdam

July 24th 2012

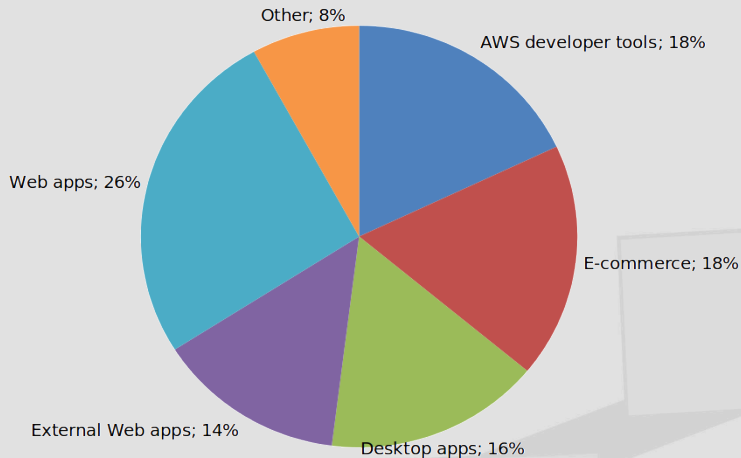


contrail is co-funded by the EC  
7th Framework Programme  
under Grant Agreement nr.  
257438

# 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

- ▶ Web servers
- ▶ Application servers
- ▶ Database servers
- ▶ High-performance frameworks (MapReduce, MPI, Workflows)
- ▶ ... and a few percents of miscellaneous programs

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

# 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
- ▶ **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

# Contrail is composed of three main layers

- ▶ Infrastructure-as-a-Service
  - ▶ Virtual machines, disks, networks
- ▶ Cloud federation
  - ▶ Allow multiple IaaS providers to join forces
- ▶ Platform-as-a-Service (a.k.a. **ConPaaS**)
  - ▶ Web servers — static content and dynamic web applications
  - ▶ MapReduce — for data-intensive computing
  - ▶ TaskFarming — for scientific applications
  - ▶ Databases (SQL and NoSQL) — for everybody

# ConPaaS in a nutshell

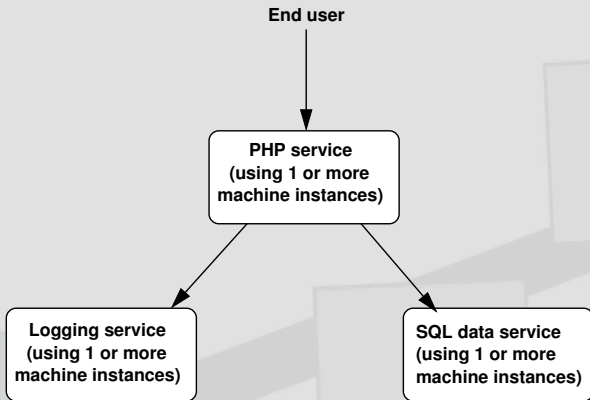
Goal: provide a fully-featured PaaS environment for Contrail

- ▶ **Broad range of functionalities**
  - ▶ Web application hosting (static files, PHP, Java, ...)
  - ▶ Databases (SQL and NoSQL)
  - ▶ High-performance execution frameworks (MapReduce, TaskFarming)
- ▶ **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
- ▶ **Making full use of Contrail's IaaS and federation functionalities**
  - ▶ But also platform-independent

## 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)

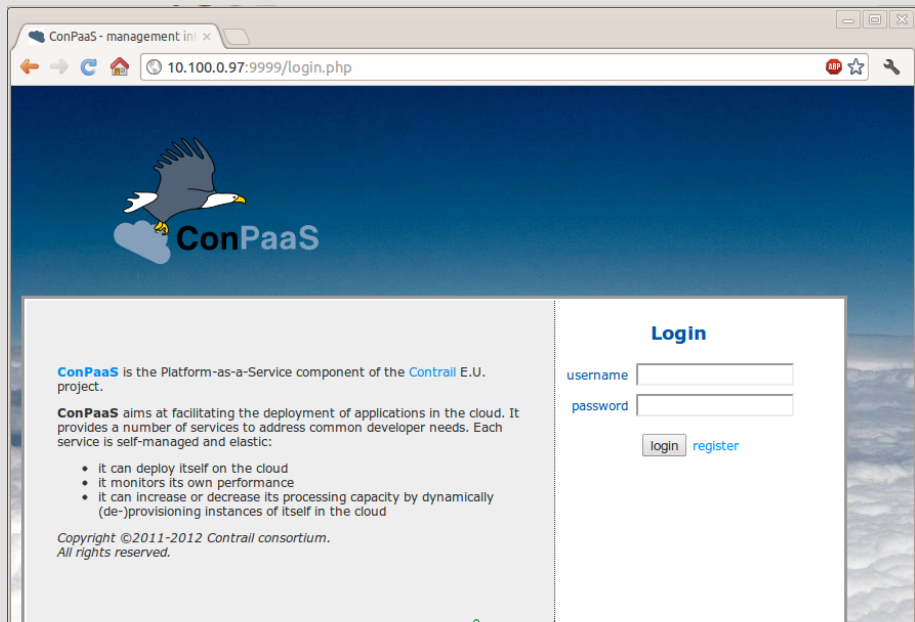




# ConPaaS Operation

- ▶ 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



The screenshot shows a web browser window with the address bar containing "10.100.0.97:9999/login.php". The page features the ConPaaS logo, which includes a stylized eagle and the text "ConPaaS". The main content area is divided into two sections: a descriptive text block on the left and a login form on the right.

**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

[register](#)

# The ConPaaS Front-End

ConPaaS - management in | x

10.100.0.97:9999/login.php



## ConPaaS

**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

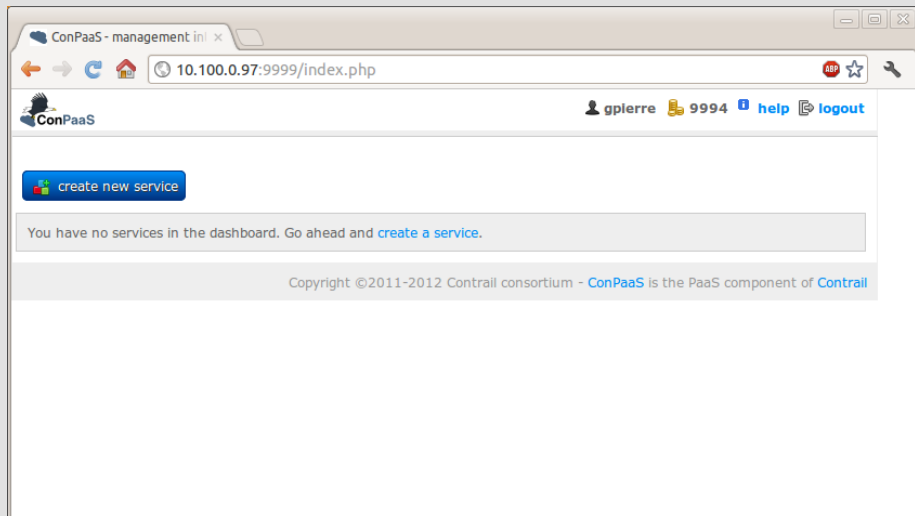


Type the two words:

 reCAPTCHA™  
stop spam.  
read books.

register

# The ConPaaS Front-End



The screenshot shows a web browser window with the following elements:

- Browser Tab:** ConPaaS - management inl x
- Address Bar:** 10.100.0.97:9999/index.php
- Page Header:** ConPaaS logo on the left; user profile 'gpierre', session '9994', and links for 'help' and 'logout' on the right.
- Main Content:** A blue button labeled 'create new service' with a small icon.
- Message Box:** A light gray box containing the text: "You have no services in the dashboard. Go ahead and [create a service](#)."
- Footer:** Copyright ©2011-2012 Contrail consortium - ConPaaS is the PaaS component of [Contrail](#)

# The ConPaaS Front-End

The screenshot shows a web browser window with the URL `10.100.0.97:9999/create.php`. The page title is "ConPaaS - create new serv". The user is logged in as "gpierre" with a balance of "9999". There are navigation links for "help" and "logout".

The main content area displays a list of services to choose from, with "php" selected. Below the services is a "cloud provider" dropdown menu.

please select one of the services below

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

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

[create service](#)

javascript: void(0);

# The ConPaaS Front-End

The screenshot displays a web browser window titled "ConPaaS - management inl x". The address bar shows the URL "10.100.0.97:9999/index.php". The page header includes the "ConPaaS" logo on the left and user information "gplerre" with a balance of "9998" and links for "help" and "logout" on the right. A blue button labeled "create new service" is positioned above a service card. The card features the PHP logo, the text "New Php Service" with a globe icon, and "created a few moments ago". To the right of the card, it indicates "1" virtual instance with a server icon. The footer contains the text "Copyright ©2011-2012 Contrail consortium - ConPaaS is the PaaS component of Contrail".

# The ConPaaS Front-End

The screenshot displays the ConPaaS management interface in a web browser. The browser's address bar shows the URL `10.100.0.97:9999/service.php?sid=6`. The page header includes the ConPaaS logo, a user profile for 'gpierre', a user ID '9998', and links for 'help' and 'logout'. A navigation link 'back to Dashboard' is present.

The main content area features a 'New Php Service' section with a 'php' logo. It shows a service that has been 'Initialized - init a few moments ago'. There are 'start' and 'terminate' buttons, and a 'manager log' link.

Below this, it indicates '1 Instance running on OpenNebula'. A table lists the instance:

Instance ID	Status	IP Address
Instance 15751	running	10.100.0.20

The 'Code management' section provides options to update the stage: 'uploading archive' (selected), 'checking out repository', and 'Choose File' (with 'No file chosen' and an example: 'example: .zip, .tar of your source tree').

The 'available code versions' section shows a table with one entry:

Code Version	Code File	Status	Action	Time
code-default	code-default.tar	active	download	a few moments ago

The 'Settings' section includes three dropdown menus: 'Software Version' set to '5.3', 'Maximum script execution time' set to '30 seconds', and 'Memory limit' set to '128M'.

# The ConPaaS Front-End

The screenshot displays the ConPaaS management interface in a web browser. The browser's address bar shows the URL `10.100.0.97:9999/service.php?sid=6`. The page header includes the ConPaaS logo, the user name `gplerre`, the ID `9997`, and links for `help` and `logout`. A navigation link `back to Dashboard` is present at the top left.

The main content area features a section for a **New Php Service**. It includes a `php` logo, a `stop` button, and a status indicator showing a green dot and the text `running - started a few moments ago`. To the right of this section are links for `access application` and `manager log`.

Below this, it states `2 Instances running on OpenNebula` and lists two instances:

	<b>Instance 15751</b> running	<code>manager</code>	10.100.0.20
	<b>Instance 15752</b> running	<code>proxy</code> <code>web</code> <code>php</code>	10.100.0.21

Below the instance list, there is a section for adding or removing instances to the deployment, featuring buttons for `proxy`, `web`, `php`, and a `submit` button.

The **Code management** section includes a link for `access application` and options for updating the stage by `uploading archive` (selected) or `checking out repository`. A `Choose File` button is present, with the text `No file chosen` and an example: `example: .zip, .tar of your source tree`.

Under **available code versions**, there is a list with `code-default` selected, followed by `code-default.tar`, `active`, and `download`. A timestamp `a few moments ago` is shown to the right.

The page number `9` is centered at the bottom, and a **Settings** link is visible at the very bottom.



# The ConPaaS Front-End

The screenshot displays the ConPaaS management interface in a browser window. The page title is "ConPaaS - management" and the URL is "10.100.0.97:9999/service.php?sid=6". The user is logged in as "gpierre" with ID "9997".

The main content area shows a "New Php Service" deployment. It is currently "running" and was started "a few moments ago". There are two instances running on OpenNebula:

Instance ID	Status	IP Address
Instance 15751	running	10.100.0.20
Instance 15752	running	10.100.0.21

Below the instances, there are buttons to "add or remove instances to your deployment": proxy, +2 web, php, and submit. A modal dialog titled "The page at 10.100.0.97:9999" is open, asking for the "no. of instances (e.g. +1, -2)". The input field contains "2".

The "Code management" section allows updating the stage by "uploading archive" or "checking out repository". The "available code versions" section shows "code-default" as the active version.

The "Settings" section includes:

- Software Version: 5.3
- Maximum script execution time: 30 seconds
- Memory limit: 128M

A "save" button is located at the bottom of the settings section.

# The ConPaaS Front-End

The screenshot displays the ConPaaS management interface in a browser window. The page title is "ConPaaS - management" and the URL is "10.100.0.97:9999/service.php?sid=6". The user is logged in as "gplerre" with ID "9994".

**ConPaaS** [back to Dashboard](#)

**New Php Service**  [access application](#) · [manager log](#)

running · started a few moments ago

5 instances running on OpenNebula

<b>Instance 15751</b> <input type="button" value="manager"/>	10.100.0.20
running	
<b>Instance 15752</b> <input type="button" value="proxy"/>	10.100.0.21
running	
<input type="button" value="web"/> <b>Instance 15753</b> <i>running</i>	10.100.0.22
<b>Instance 15754</b> <i>running</i>	10.100.0.23
<b>Instance 15755</b> <input type="button" value="php"/>	10.100.0.24
running	

add or remove instances to your deployment

**Code management** [access application](#)

you may update the stage by

- uploading archive  No file chosen  
example: .zip, .tar of your source tree
- checking out repository

available code versions

**code-default** [code-default.tar](#) · **active** · [download](#) a few moments ago

**Settings**

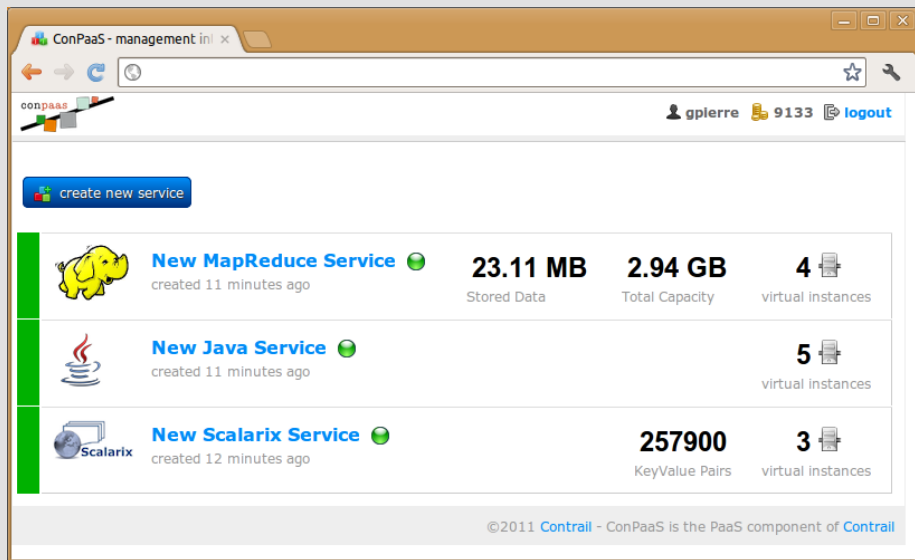
# The ConPaaS Front-End

The screenshot displays the ConPaaS management interface in a browser window. The address bar shows the URL `10.100.0.97:9999/service.php?sid=6`. The user is logged in as 'gpierre' with 9994 credits. The main content area features a 'New Php Service' deployment, which is currently 'running' and was started a few moments ago. Below this, a table lists 5 instances running on OpenNebula:




Instance ID	Role	Status	IP Address
Instance 15751	manager	running	10.100.0.20
Instance 15752	proxy	running	10.100.0.21
Instance 15753	web	running	10.100.0.22
Instance 15754	web	running	10.100.0.23
Instance 15755	php	running	10.100.0.24

Below the instance list, there are controls to 'add or remove instances to your deployment', including buttons for 'proxy', 'web', 'php', and a 'submit' button. The 'Code management' section allows updating the stage by 'uploading archive' (with a 'Choose File' button) or 'checking out repository'. The 'available code versions' section shows two versions: 'code-y3gOfi' (active) and 'code-default' (active).

# The ConPaaS Front-End



The screenshot shows a web browser window titled "ConPaaS - management in | x". The browser's address bar is empty. The page header includes the "conpaas" logo on the left and the user "gpierre" with a balance of "9133" and a "logout" button on the right. A blue button labeled "create new service" is positioned above a table of services. The table lists three services, each with a status indicator (green circle) and a printer icon.

Service Name	Created	Stored Data	Total Capacity	Virtual Instances
 <b>New MapReduce Service</b>	created 11 minutes ago	23.11 MB	2.94 GB	4
 <b>New Java Service</b>	created 11 minutes ago			5
 <b>New Scalarix Service</b>	created 12 minutes ago		257900	3

©2011 [Contrail](#) - ConPaaS is the PaaS component of [Contrail](#)

# The ConPaaS Front-End

```
Terminal
File Edit View Search Terminal Help
renard:~/php/tmp/ConPaaSWeb/src> export PYTHONPATH=$PWD
renard:~/php/tmp/ConPaaSWeb/src> ./managerc.py http://ec2-50-17-94-105.compute-1.amazonaws.com:5555/
Usage: ./managerc.py URL ACTION [options]

Action could be one of:
[ACTION]                [DESCRIPTION]
add                     Add more service nodes to a deployment
downloadCodeVersion    Download a code version
getConfiguration        Get the configuration of a deployment
getHighLevelMonitoring Get the average request rate and throughput
getLog                  Get raw logging
getServiceNode          Get information about a single service node
getState                Get the state of a deployment
getStateChanges         Get the state change history of a deployment
help                    Print the help menu
listCodeVersions        List identifiers of all code versions stored by a deployment
listServiceNodes        Get a list of service nodes
remove                  Remove some service nodes from a deployment
shutdown                Shutdown a deployment
startup                 Startup a deployment
updateConfiguration    Update the configuration of a deployment
uploadCodeVersion       Upload a new code version
renard:~/php/tmp/ConPaaSWeb/src> █
```

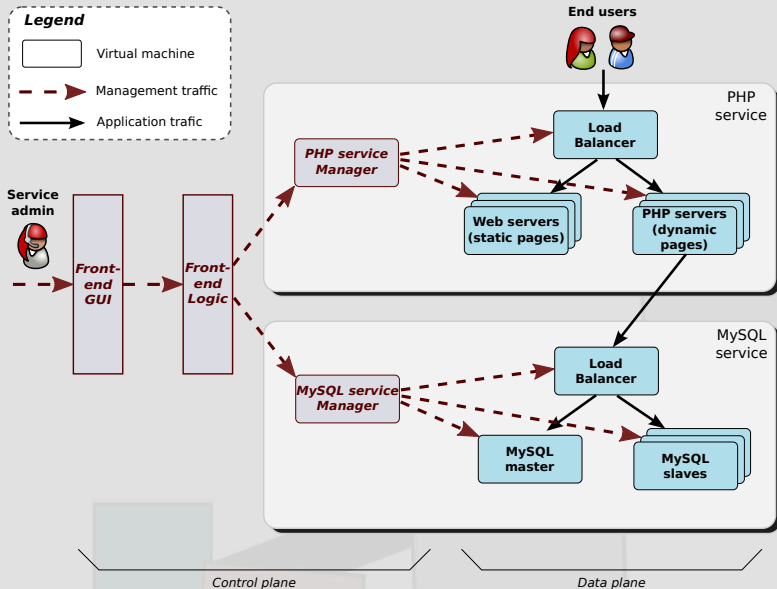
# The ConPaaS Front-End

```
Terminal
File Edit View Search Terminal Help
renard:~/php/tmp/ConPaaSWeb/src> ./managerc.py http://ec2-50-17-94-105.compute-1.amazonaws.com:5555/ listServiceNodes
Service Node      Role(s)
i-3d386153        PHP
i-e3065f8d        PROXY
i-3f386151        WEB
i-2138614f        PHP
i-2338614d        WEB
i-2538614b        WEB
renard:~/php/tmp/ConPaaSWeb/src> ./managerc.py http://ec2-50-17-94-105.compute-1.amazonaws.com:5555/ getServiceNode i-3f386151
Service Node      Address          Role(s)
i-3f386151        ec2-184-73-93-153.compute-1.amazonaws.com WEB
renard:~/php/tmp/ConPaaSWeb/src> █
```

# Architecture of a ConPaaS service

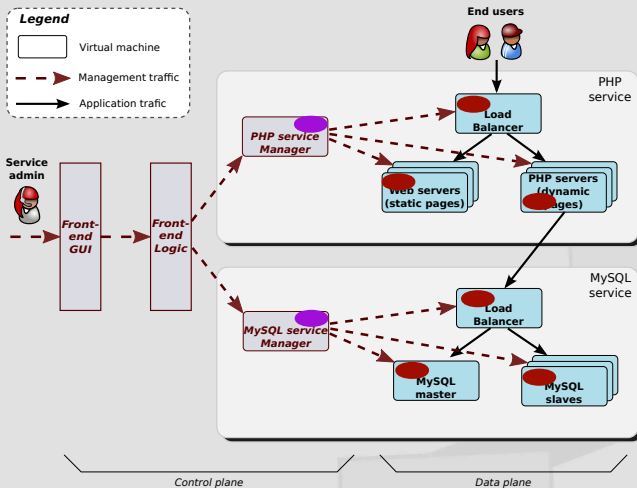
- ▶ A ConPaaS service is implemented as **one or more virtual machine instances** dedicated to a single user
  - ▶ Single-tenant: each VM belongs to a single user
  - ▶ No VM sharing between services (even for the same user)
- ▶ ConPaaS services are **elastic**: we can grow/shrink their capacity at runtime with no service disruption
  - ▶ Horizontal provisioning: add/remove virtual machines
- ▶ ConPaaS services will support **dynamic resource provisioning**: automatic capacity adjustment to support performance guarantees at minimum cost

# ConPaaS Organization



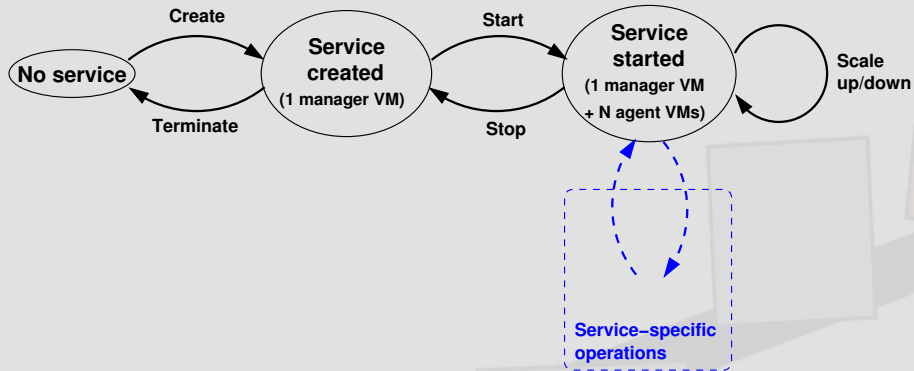


# ConPaaS Organization



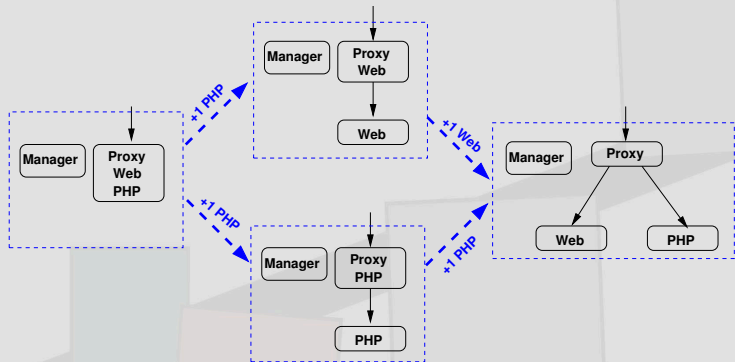
- ▶ Each manager VM contains a **manager process**
- ▶ Each agent VM contains an **agent process**

# Lifecycle of a ConPaaS service



# 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**
- ▶ 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

# 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 is HARD
  - ▶ Build a proper VM image with contextualization
  - ▶ Develop new **manager** and **agent** daemons
  - ▶ 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 daemons**
  - ▶ The service core implements **shared functionality** between all services
    - ▶ Start/stop/contextualize virtual machines
    - ▶ Secure communication primitives
  - ▶ 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 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

# 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
- ▶ **Future plans**:
  - ▶ Automatic performance control
  - ▶ Application manifests
  - ▶ Better developer support
  - ▶ More services :-)



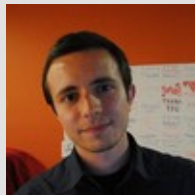
# Credits



Adriana Szekeres  
(server side)



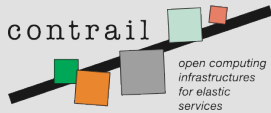
Ismail El Helw  
(server side)



Claudiu Gheorghe  
(Web GUI)

## About the hands-on session

- ▶ All support documents can be found at <http://bit.ly/MCKtil>
  - ▶ Presentation slides, exercises, support programs etc.
- ▶ Please work in **groups of two**
- ▶ Make sure your Web browser is configured to use the **SOCKS proxy** at <http://130.37.30.108:80/>
- ▶ The ConPaaS front-end is located at <http://10.100.0.97:9999/>



**contrail** is co-funded by the  
EC 7th Framework Programme

Funded under: FP7 (Seventh Framework Programme)  
Area: Internet of Services, Software & Virtualization  
(ICT-2009.1.2)

Project reference: FP7-IST-257438

Total cost: 11.29 million euro

EU contribution: 8.3 million euro

Execution: From 2010-10-01 till 2013-09-30

Duration: 36 months

Contract type: Collaborative project (generic)