

PROFESSIONAL TESTER

SUBSCRIBE
It's FREE
for testers

Essential for software testers

August 2011 | £4 / €5 | v2.0 | number 10 |

THIS ISSUE OF
PROFESSIONAL TESTER
IS SPONSORED BY



HOW TESTERS COMMUNICATE AND HOW THEY SHOULD

Including articles by:

Bogdan Stefan Bereza
Victo

Brad Higgs
HP

Ian Molyneux
Intechnica

René Tuinhout
Richard Janssen
Logica

George Wilkinson
Aquidium

Virtual reality

by Ian Molyneux

One of empirical testing's biggest problems is about to be solved



Ian Molyneux explains how to build cloud-based test environments

Nearly every tester is too familiar with the problem of obtaining suitable test environments. At Intechnica we believe cloud provisioning of infrastructure is the solution. It offers instant set up, flexible scaling, pay-for-what-you-need pricing and the simplest way to outsource support and maintenance. For many test organizations it has the potential to replace “tin” completely, as it is doing in many businesses. Professional Tester coined the term *test environment as a service* to describe the concept in November 2010 (see <http://professionaltester.com/news/article.asp?id=209>).

Intechnica specializes in web development and performance assurance and has partnered with test consultancy ROQ IT to develop an offering to test organizations working in this and other domains.

CloudFlex is a web application designed to make it easy to set up and manage test environments comprising virtual servers

running on third-party cloud computing capacity. It provides a single environment used to manage server images, data, security, load balancing and geolocation. At present it works with Amazon EC2, but its loosely coupled design makes applying it to other cloud providers straightforward: support for Microsoft Azure will be released soon. Useful new capabilities will be integrated as soon as they are made available by providers, so that they can be used with environments already built. Unlike much cloudware, CloudFlex is not a promise or plan for the future: it's working and available now, and we are seeking contact with potential users to help us understand their needs and improve it fast.

Creating servers

To understand how CloudFlex is used, imagine we are to test a web application. The live environment consists of two load-balanced web servers, an application server which accesses

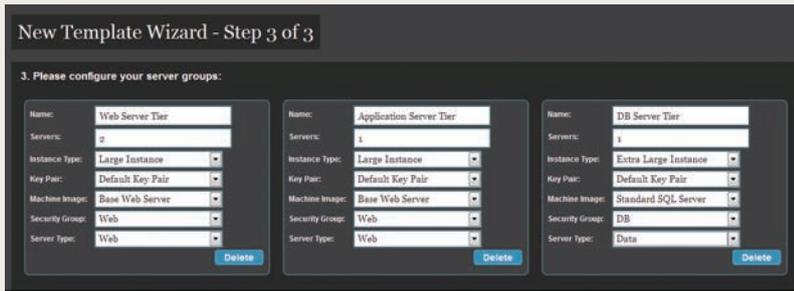


Figure 1

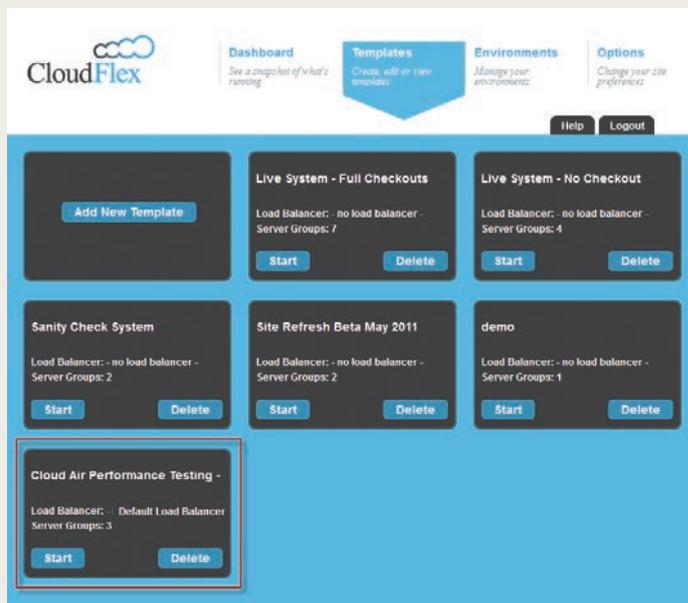


Figure 2

ourselves when the virtual server is running. Once this is done the virtual server can be imaged to create a new AMI easily. EC2 users often share their AMIs, so we might be lucky enough to find some that are very similar to the live servers and can be made sufficiently closely equivalent to them with little work.

Instance type and size

A virtual server created from an AMI is called an instance. EC2 offers a range of instance types (see <http://aws.amazon.com/ec2/instance-types>), differing in processing power, word size (32 or 64 bits), memory and storage. The one most similar to its counterpart tin server in the live environment is chosen.

There are two approaches to deploying the software under test: starting the virtual servers and using our current deployment process, or including the deployed application in the image. Obviously the second is more efficient, but the design of the SUT often makes it difficult, especially when changing the number of instances in use. CloudFlex offers various tools to help. Once the images are ready it makes using them to build various environments simple.

third-party web services (availability, ticketing, payment processing etc) and a database server which stores account and transaction information.

To build an equivalent cloud-based test environment an obvious approach would be to image the live servers then upload and deploy the images. Unfortunately this cannot be done in EC2, so the images must be created, starting from a generic “base” image called in EC2 an *Amazon Machine Image*.

AMIs are available with many combinations of the most common OSs and applications installed. However these tend to be “latest and greatest”: if our application requires older versions we will probably need to install them

When testing requires a lot of data online it is not always necessary to increase the size or number of instances. The data can be stored in Amazon's storage web service, S3, and attached to database instances from there, or in its SimpleDB or Relational Database Service (both currently in beta) where it can be queried directly by application instances.

Repeating environments

Whether tin or cloud, the hardest part of managing test environments is changing them, especially when it is necessary to return to previously used ones. Starting, stopping and configuring instances in EC2 requires work-intensive and complex command line operations where mistakes could lead to inaccurate test repetition and missed defects.



Figure 3

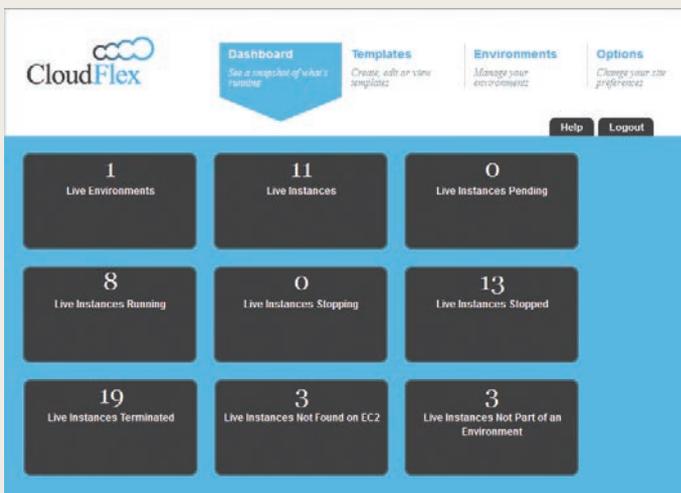


Figure 4



CloudFlex replaces these with a clear user interface and automates the process of switching between environments.

This is achieved by creating environment templates using a simple wizard (see figure 1). All templates are accessed from a single screen (figure 2) and can be edited or adapted to create new ones easily (figure 3). Any templated environment can then be started and terminated with a few clicks. CloudFlex's dashboard (figure 4) displays the current status of all instances.

Identical to live

In this example we have considered using cloud to create test environments for an application that is running on tin. This works

well for functional and usability testing, although some care must be taken to ensure the test environment is sufficiently "like live" to avoid test results differing from those were the tests run on the live site.

Suppose, however, that the site owner makes the increasingly common decision to use EC2 or another cloud service to host it. Now test environments can be built

which are exactly the same as the live environment in all respects necessary for the testing at hand. As well as removing any risk of inaccurate functional testing results caused by lack of realism, this means accurate performance, reliability and scalability testing can be done by creating additional instances to run performance testing tools injecting load to the test environment ■

Ian Molyneaux is head of performance at Int Technica (<http://inttechnica.co.uk>)

