

TinyBlog: Deploy

1.1 Previous Week Solution

You can load the full TinyBlog application using the following snippet:

```
Gofers new
  smalltalkhubUser: 'PharoMooC' project: 'TinyBlog';
  package: 'ConfigurationOfTinyBlog';
  load.
#ConfigurationOfTinyBlog asClass loadFinalApp
```

To test it, you should launch the Seaside HTTP server:

```
ZnZincServerAdaptor startOn: 8080.
```

You might also want some initial posts:

```
TBBlog reset ; createDemoPosts
```

You can now complete or finish **your** application and commit it in your Smalltalkhub repository.

1.2 Deploy in the Cloud

Now that you have finished to develop your blog web application, we will see how to deploy it on a server in the cloud.

If you want to deploy your application on your own server, we encourage you to read the last chapter of the book "Enterprise Pharo: a Web Perspective" (<http://books.pharo.org>).

In this tutorial, we will describe a simpler solution provided by PharoCloud (<http://pharocloud.com>).

PharoCloud Hosting

PharoCloud is a hosting service dedicated to Pharo web applications that offers a free testing plan (ephemeric cloud subscription).

Set-up your PharoCloud account:

- Create an account on <http://pharocloud.com>
- Activate your account
- Connect
- Activate your "Ephemeric Cloud" to obtain an **API User ID** and an **API Auth Token**
- Click on "Open Cloud Client" and log in using the above credentials
- Once connected, you should see a page allowing you to upload an archive file (zip) with two files: a Pharo image and its changes file

Preparing Pharo image to Deploy on PharoCloud

Currently, PharoCloud only supports Pharo 4 images.

So, download a Pharo Web 4 image¹. You should also download a virtual machine for Pharo 4 images². Then, launch this image using this VM and we will configure it.

Let's start by configuring Seaside by removing all demo applications and all development tools:

```
"Seaside Deployment configuration"
WAAdmin clearAll.
WAAdmin applicationDefaults removeParent: WADevelopmentConfiguration
instance.
WAFileHandler default: WAFileHandler new.
WAFileHandler default
  preferenceAt: #fileHandlerListingClass
  put: WAHtmlFileHandlerListing.
WAAdmin defaultDispatcher
  register: WAFileHandler default
  at: 'files'.
```

Now, you can load the TinyBlog application:

```
"Load TinyBlog"
Gofer new
  smalltalkhubUser: 'PharoMooc' project: 'TinyBlog';
  package: 'ConfigurationOfTinyBlog';
```

¹<https://ci.inria.fr/pharo-contribution/job/PharoWeb/PHARO=40,VERSION=stable,VM=vm/lastSuccessfulBuild/artifact/PharoWeb.zip>

²<http://get.pharo.org/vm40>

1.2 Deploy in the Cloud

```
load.  
#ConfigurationOfTinyBlog asClass loadFinalApp.  
  
"Create Demo posts if needed"  
#TBBlog asClass createDemoPosts.
```

or **your** application directly from your Smalltalkhub code repository:

```
"Load TinyBlog"  
Gofer new  
  smalltalkhubUser: 'XXXX' project: 'TinyBlog';  
  package: 'TinyBlog';  
  load.  
  
"Create Demo posts if needed"  
#TBBlog asClass createDemoPosts.
```

We now configure Seaside to use TinyBlog as the default application and we launch the HTTP server.

```
"Tell Seaside to use TinyBlog as default app"  
WADispatcher default defaultName: 'TinyBlog'.  
  
"Start HTTP server"  
ZnZincServerAdaptor startOn: 8080.  
  
"Register TinyBlog on Seaside"  
TBApplicationRootComponent initialize.
```

You can now save the Pharo image (World Menu > save) and test locally with a web browser on the URL: <http://localhost:8080>.

Deploy on Ephemeric Cloud of PharoCloud

You just need to create an archive (zip file) containing both the `PharoWeb.image` and `PharoWeb.changes`.

Then, drag'n drop this archive file on the Ephemeric Cloud web page and activate this Pharo image (play button) as shown in figure 1.1.

If you click on the public URL provided Ephemeric Cloud, you should see your web application as displayed on figure 1.2.

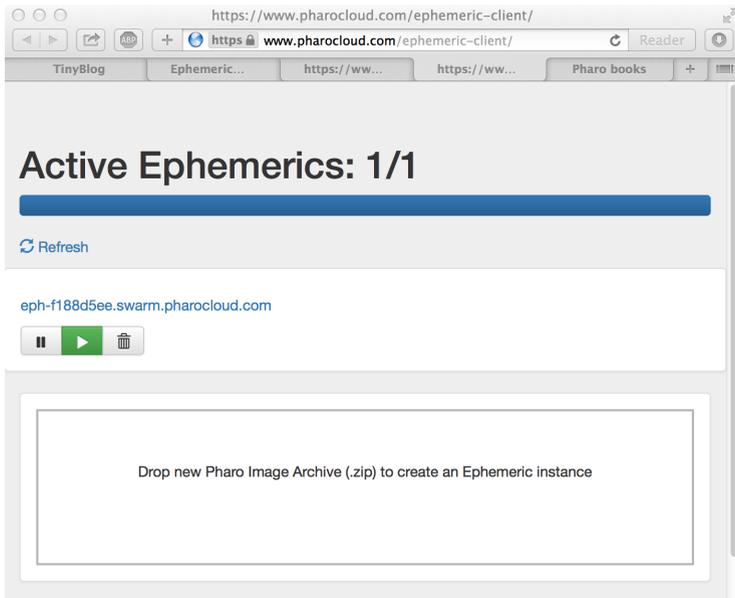


Figure 1.1 Managing Pharo images on Ephemeric Cloud.

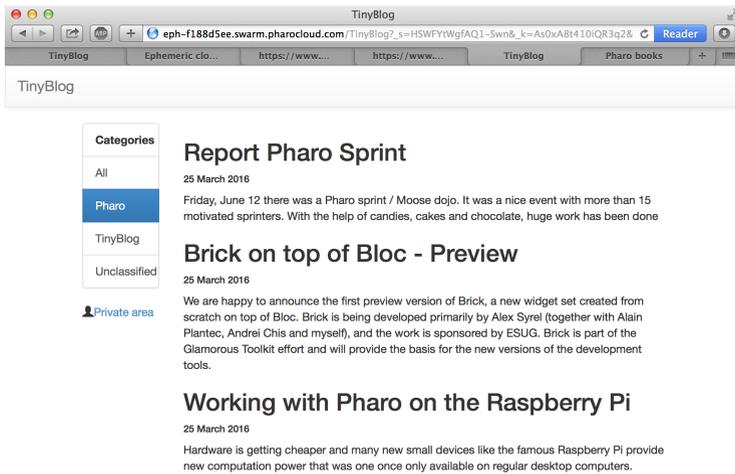


Figure 1.2 Your TinyBlog Application Hosted on PharoCloud.