

Welcome to this lecture! Today, we're going to study some topics about deploying your web information systems.

--

Copyright (C) 2019 Universidad de Sevilla

The use of these slides is hereby constrained to the conditions of the TDG Licence, a copy of which you may download from <http://www.tdg-seville.info/License.html>

## What's deployment about?

---



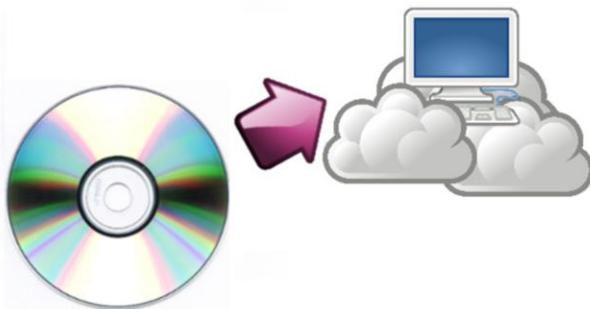
UNIVERSIDAD DE SEVILLA

2

As usual, we start with a question: what's deployment about?

## This is a good definition

---



Deploying a system means installing it on a computer (IaaS) or a platform (PaaS) so that our customer can start using it as soon as possible

Deploying a system means installing it on a computer or a platform so that our customer can start using it as soon as possible. Nowadays, it's very common that the computer or the platform resides on the cloud, that is: they are provided by another company that rents them. Computers on the cloud are also referred to as IaaS; platforms on the cloud are also known as PaaS. A computer's a piece of hardware with an operating system; a platform's a combo with an application server and a database server. Keep reading to learn more about this.

## How is it performed?

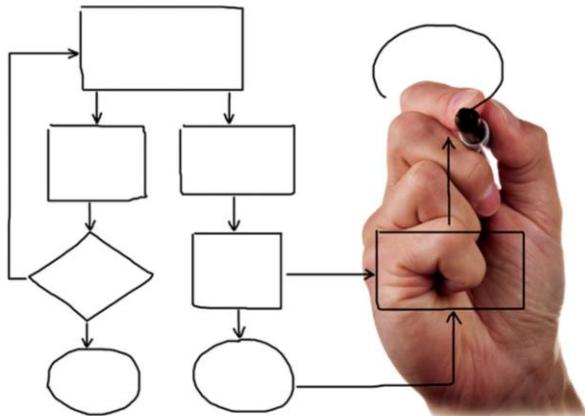
---



It's now time to ask yourself regarding how a typical deployment's performed. As usual, please, try to produce an answer before peeking at the following slides.

## Step 1: learn the foundations

---



First, you have to learn some foundations, that is, some core concepts that we're going to use through the rest of the lesson.

## Step 2: build the deployment artefacts

---



Next, you have to learn how to create the so-called deployment artefacts (or d-artefacts for short).

## Step 3.1: deploy to an IaaS or ...

---



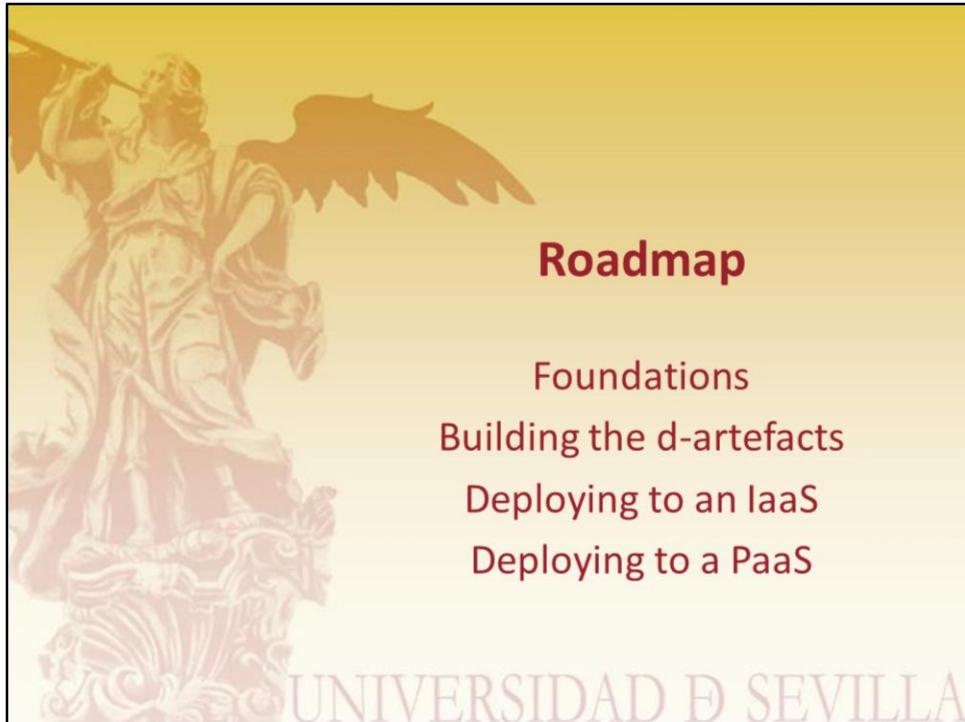
Once the deployment artefacts are ready, you can deploy them to a cloud infrastructure (aka IaaS) that provides a computer, or ...

## Step 3.2: ... or deploy to a PaaS

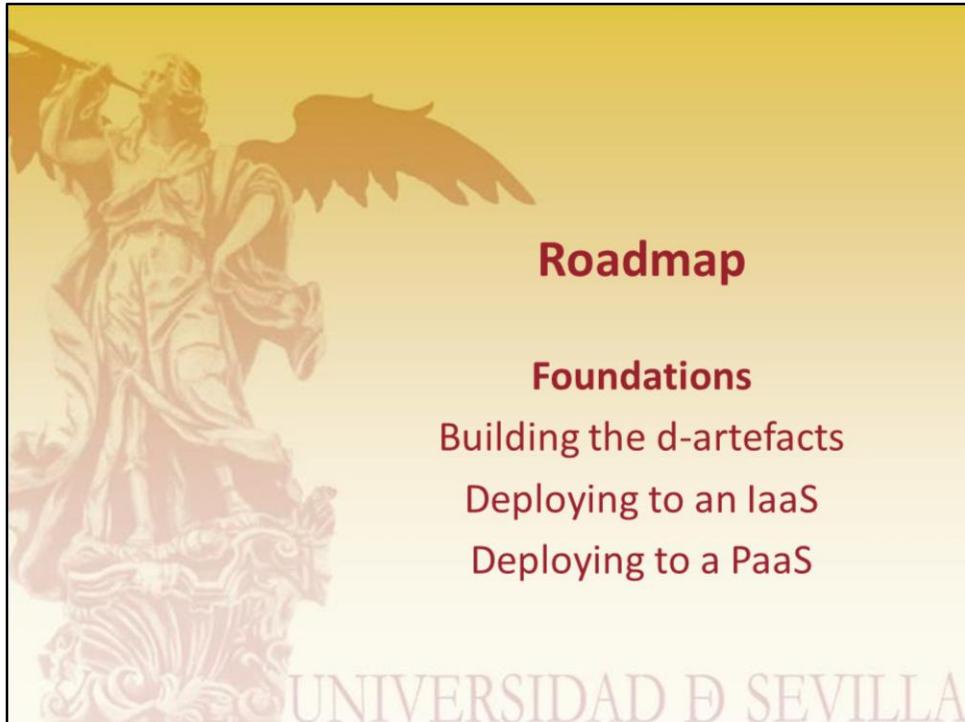
---



... or you can deploy them to a cloud platform (aka PaaS) that provides an application server and a database server.



So, it shouldn't be surprising at all that this is our roadmap today.



Let's start with the foundations.

## The foundations

---



Basic concepts that every software engineer must understand regarding deployment

Simply put: the foundations are basic concepts that every software engineer must understand regarding deployment. That includes you!



## Foundations

Components of a system

Scalability

Architecture

X as a Service

UNIVERSIDAD DE SEVILLA

First, we're going to review the components of every system, then we're going to pay a little attention to scalability, architecture, and X as a service, where X stands for either "Infrastructure" (IaaS) or "Platform" (PaaS).



## Foundations

**Components of a system**

Scalability

Architecture

X as a Service

UNIVERSIDAD DE SEVILLA

Let's start with the components of a system.

## Your system in two words

---

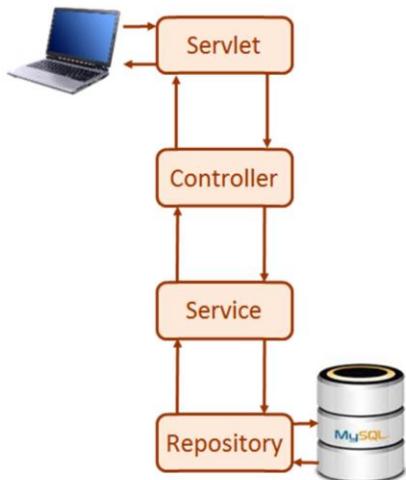


Application

Database

This should not be surprising to you: a typical (web information) system is composed of an application and a database, which must be deployed to the appropriate servers.

## The application



- Servlet
- Controllers
- Services
- Repositories

The application, in turn, is composed of a servlet, several controllers, several services, and several repositories. The servlet cares of getting the HTTP requests from the users and passing them on to the appropriate controllers, which typically delegate processing them to some services that, in turn, use a number of repositories to interact with the database.

# The database



PubID	Publisher	PubAddress
03-4472822	Random House	123 4th Street, New York
04-7733903	Wiley and Sons	45 Lincoln Blvd, Chicago
03-4859223	O'Reilly Press	77 Boston Ave, Cambridge
03-3920886	City Lights Books	99 Market, San Francisco

AuthorID	AuthorName	AuthorBDay
345-28-2939	Halle Selassie	14-Aug-92
392-48-9985	Joe Blow	14-Mar-15
454-22-4012	Sally Hemmings	12-Sep-70
663-59-1254	Hannah Arendt	12-Mar-06

ISBN	AuthorID	PubID	Date	Title
1-345-32-482-1	345-28-2939	03-4472822	1990	Cold Fusion for Dummies
1-38482-995-1	392-48-9985	04-7733903	1980	Macrame and Straw Tyng
2-35921-499-4	454-22-4012	03-4859223	1852	Fluid Dynamics of Aquaducts
1-38278-293-4	663-59-1254	03-3920886	1967	Beads, Baskets & Revolution

- Tables
- Attributes
- Primary keys
- Foreign keys
- Indices
- Stored procedures

And the database is composed of tables that have attributes, some of which are primary keys (the “id” attribute) and some of which are foreign keys (the role attributes in your domain model). We haven’t studied it so far, but the database may also have some indices and some stored procedures.



## Foundations

Components of a system

**Scalability**

Architecture

X as a Service

UNIVERSIDAD DE SEVILLA

There was nothing new in the previous section. Let's now explore scalability.

## A simple definition

---



It's the ability of a system to grow or shrink depending on the workload

Scalability is difficult to implement, but simple to define: it's the ability of a system to grow or shrink depending on the workload. A system's scalable if it's able to attach more computers or detach some computers at run time so that it adapts optimally to the workload.

## Vertical scalability

---



- Very simple
- Just migrate the system
- You know it all

Scalability can be vertical, in which case the system scales from a computer to a more powerful computer or vice versa. Vertical scalability is the easiest way to scale a system because you basically don't have to do anything but moving your system to a more powerful computer or vice versa. Simply put: you know it all regarding vertical scalability.

## Horizontal scaling

---



- More difficult
- Migration + distribution
- You don't know it all

Horizontal scalability is far more difficult to achieve because it requires to migrate your system and distribute it to a cluster of machines. That typically requires your system to be carefully designed so that every instance of the application or the database can serve any request. Simply put: you don't know it all regarding horizontal scalability, but we're going to learn a bit in this lesson.



## Foundations

Components of a system

Scalability

**Architecture**

X as a Service

UNIVERSIDAD DE SEVILLA

Scalability is conceptually simple, right? Let's explore an intimately-related concept: the architecture.

## A simple definition



It's a blueprint that states how your system's going to be deployed

The definition is pretty simple: it's a blueprint that states how your system is going to be deployed.

## A one-computer architecture

---

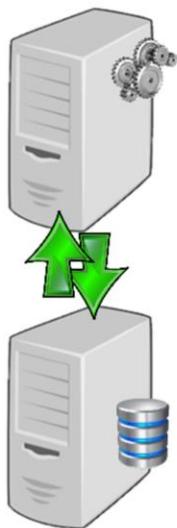


- Very simple
- Small systems (10-100 concurrent users)
- You know it all

The easiest architecture consists of just one computer that you use run both your application server and your database server. That's the architecture that you typically use while you're developing a system, and it's used for small systems that can typically serve from a dozen to a hundred users concurrently. Simply put: you know it all about this architecture.

## A two-computer architecture

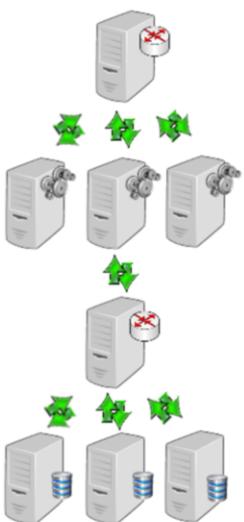
---



- Very simple, too
- Mid systems (100-1000 concurrent users)
- You didn't know that you knew it all

Deploying your systems to two computers so that one of them runs your application server and the other runs the database server is pretty simple: recall that your applications rely on a data source that you have to configure by means of a URL that allows to communicate with your database; there's not a difference between a database that resides on the same machine or a database that resides on another machine. This architecture is used by mid-scale systems that need to serve requests from a hundred to a few hundred concurrent users. The conclusion is that you likely didn't know that you knew it all regarding this architecture, but you did.

## A multi-computer architecture



- Complex till recently
- Large systems (1000+ concurrent users)
- You know little, and you're not going to learn a lot more

Large systems have to serve requests by thousands if not millions of concurrent users. They require to use several computers to run your application server and several computers to run your database server, plus so-called load balancers that distribute the requests amongst the computers so that the overall workload is balanced. Implementing such architectures is far more involved than implementing a one- or a two-computer architecture; it typically requires your software to be designed so that it can run on a fully distributed environment. In other words, we have not studied everything you need to build and deploy such a system and we're not going to delve into these details in this subject, but we can still deploy our systems to such architectures using cloud platforms. Please, keep reading.



## Foundations

Components of a system

Scalability

Architecture

**X as a Service**

UNIVERSIDAD DE SEVILLA

And now, it's about time to learn about X as a service, where X refers to either "Infrastructure" (IaaS) or "Platform" (PaaS).

## A simple definition

---



It's about renting infrastructure and  
platforms from cloud providers

X as a Service refers to renting infrastructure or platforms on the cloud. This typically reduces the cost of ownership since a company doesn't have to care of buying computers and platforms and keep them running; they just rent them to other companies.

## IaaS: renting computers on the cloud

---



- Operating system
- Tomcat server
- MySQL Server
- Security
- Hardware failures
- ...
- Scale the system

IaaS stands for Infrastructure as a Service. It refers to renting a computer on the cloud. Note that a computer's bare metal, which means that you have to care of installing an operating system, the Tomcat server, the MySQL server, configuring security, dealing with hardware failures, and a long etcetera that includes scalability issues.

## PaaS: renting platforms on the cloud

---

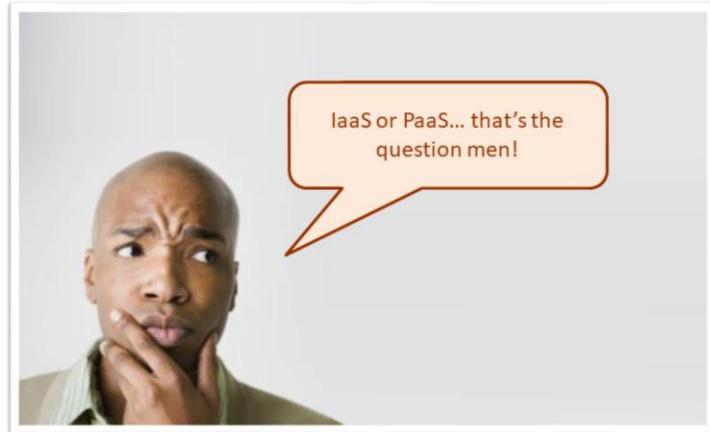


- Tomcat server
- MySQL Server

PaaS means Platform as a Service. It refers to renting a platform to run your systems, that is, a Tomcat server and a MySQL server.

## This is a good question

---



This is definitely a good question: what's the best choice? An IaaS or a PaaS?

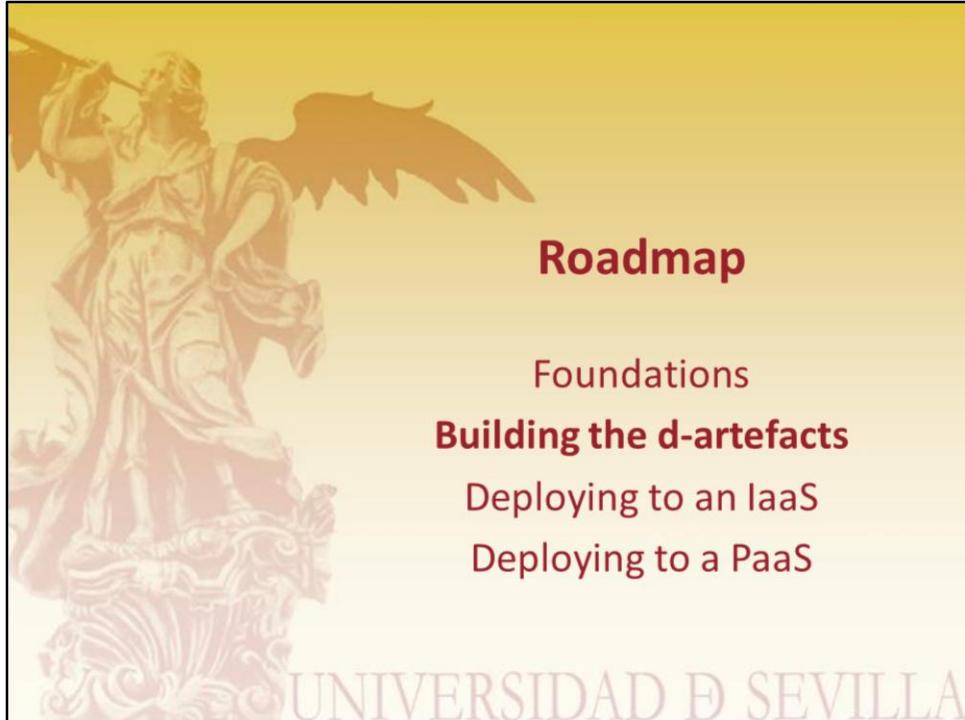
## And this is a good answer

---



PaaS is far more attractive since you can focus on your app, not the system to run it. But there are many companies that prefer an IaaS because they have to keep everything under control or because they can't afford the change.

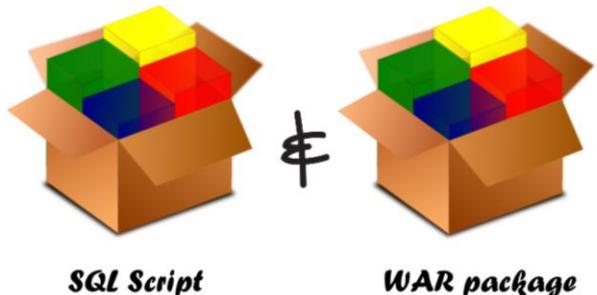
Note that PaaS are far more attractive than IaaS in the context of D&T since they allow you to focus on deploying your application; you can forget about the operating system, the hardware and all other stuff, including scaling your system. But there are many companies that prefer IaaS and the reasons are manifold: many wish to keep everything under control; others can't afford the change because they have to amortise their investment. In practice, both solutions are very common, so we have to learn them both.



That's enough about the foundations. It's time to report on how to build the deployment artefacts (aka d-artefacts) that you need to deploy your system to an IaaS or a PaaS.

## A simple definition

---



They're a SQL script with the schema of your database and a WAR package with the classes and the resources of your application

There are two deployment artefacts, namely: a SQL script with the schema of your database and a WAR package with the classes and the resources of your application.



So, we're going to explore how to create both artefacts separately.



Let's start with the database artefact.

## What's the database artefact?

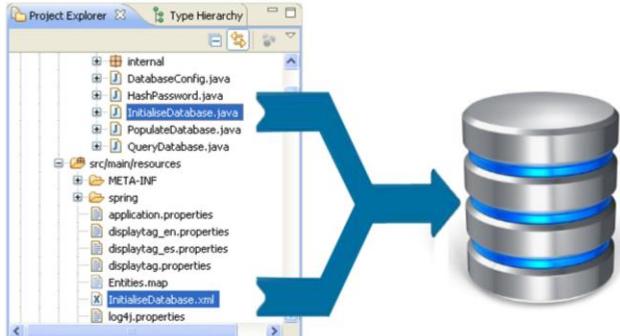
---



It's a SQL script that results from dumping  
your developer's database with the  
minimum data required to start your system

The database artefact's a SQL script that results from dumping your developer's database with the minimum data required to start your system. Typically, an administrator account is enough, but you might well require other data, including: a list of provinces, a list of phone prefixes, a taxonomy of items, or some configuration data (spam words, welcome messages, and the like).

## Initialising the database



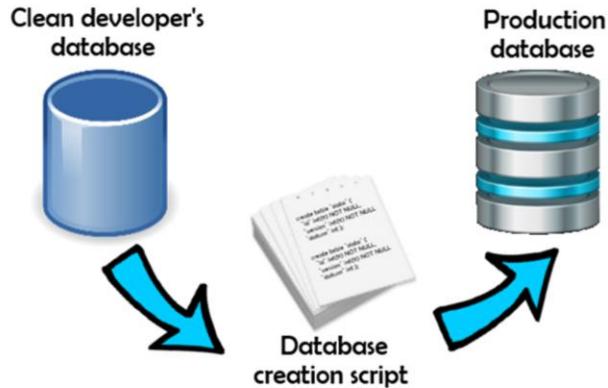
UNIVERSIDAD DE SEVILLA

37

Recall that we provide a couple of utilities with the project template: “InitialiseDatabase”, which is intended to load the minimum data into the database, and “PopulateDatabase”, which is intended to load it with test data. To generate the script, you must first run the “InitialiseDatabase” utility, which reads the minimum data from the “InitialiseDatabase.xml” specification.

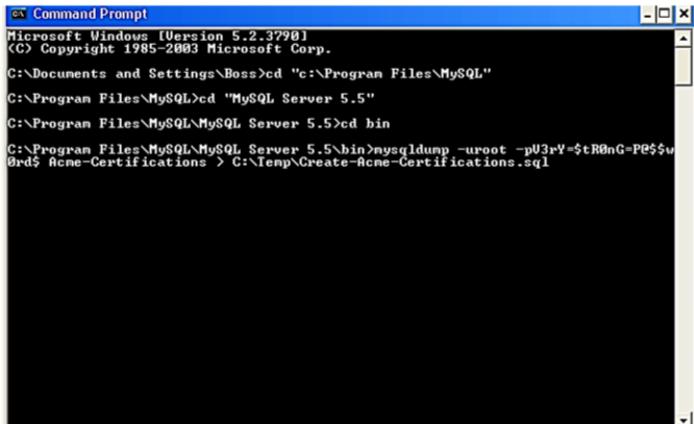
## The process

---



This is the process that we need to implement: first, you must make sure that you have a clean developer's database, that is, a database that doesn't have any test data; then, we'll use a tool to dump it, that is, to create a SQL script that will allow you to recreate it on another configuration. Note that the script must keep only the data that is absolutely necessary for your system to work well; this typically includes an administrator's account information and little else.

## Dumping a database



The screenshot shows a Windows Command Prompt window titled "Command Prompt". The window displays the following command sequence:

```
Microsoft Windows [Version 5.2.3790]
(C) Copyright 1985-2003 Microsoft Corp.

C:\Documents and Settings\Boss>cd "c:\Program Files\MySQL"
C:\Program Files\MySQL>cd "MySQL Server 5.5"
C:\Program Files\MySQL\MySQL Server 5.5>cd bin
C:\Program Files\MySQL\MySQL Server 5.5\bin>mysqldump -uroot -pU3rY=$tR0nG=P@$$w0rd$ Acme-Certifications > C:\Temp\Create-Acme-Certifications.sql
```

Next, you must dump the database, which isn't difficult at all: just open an administrator's shell and execute the following commands to change the current directory to MySQL's bin directory:

```
> cd "C:\Program Files\MySQL"
> cd "MySQL Server 5.5"
> cd "bin"
```

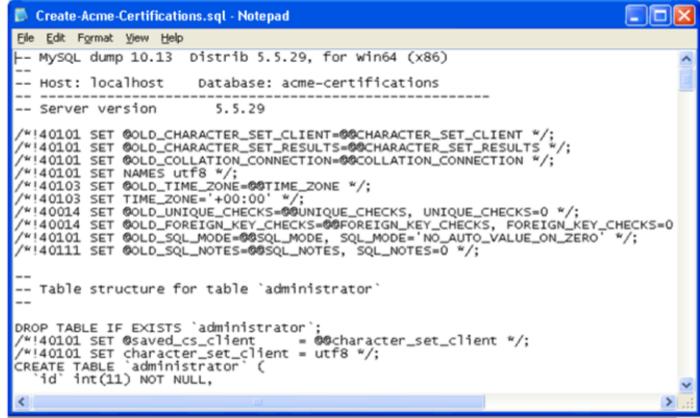
Now, execute the following command to dump your database:

```
> mysqldump -u<root> -p<pass> <database-name> > <script-file.sql>
```

Where `<root>` and `<pass>` refer to the database root credentials, `<database-name>` to the name of the database you wish to dump, and `<script-file.sql>` to the file in which you wish to store the script. In this example, as usual, we're working with the database of project "Acme Certifications".

**NOTE:** in the pre-production machine, the administrator's credentials are "boss/\$!=\$B0\$\$=U\$3=P@\$\$\$", and the database root's credentials are "root"/"V3rY=\$tR0nG=P@\$ \$\$w0rd\$".

# The database creation script



```
File Edit Format View Help
-- MySQL dump 10.13 Distrib 5.5.29, for win64 (x86)
-- Host: localhost      Database: acme-certifications
-- Server version      5.5.29

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8 */;
/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
/*!40103 SET TIME_ZONE='+00:00' */;
/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;

-- Table structure for table `administrator`
-- DROP TABLE IF EXISTS `administrator`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `administrator` (
  `id` int(11) NOT NULL,
```

Please, take a look at the script. It should look more or less like in this slide. It includes a lot of SQL statements to create the data tables, the attributes, the foreign keys, and so on.

## This is a good question!

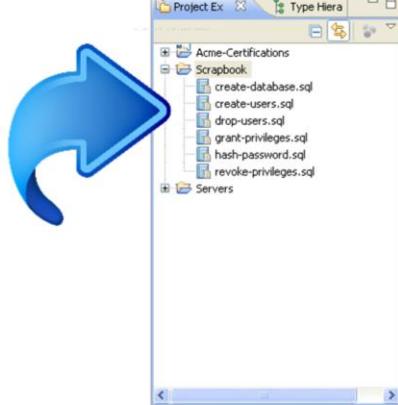
---



Where are the  
statements to create the  
database, the users, or to  
assign privileges?

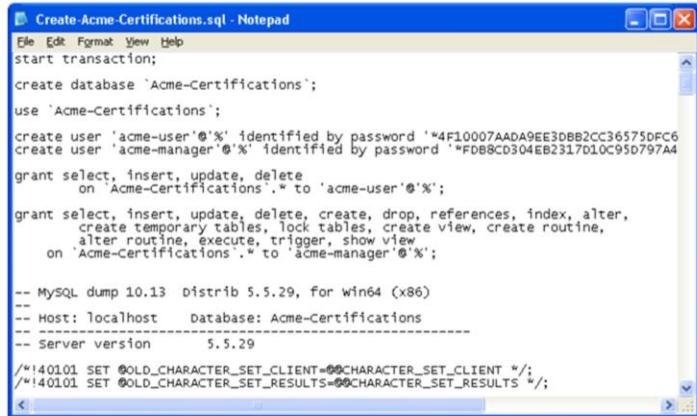
This is a good question: this script is incomplete, isn't it? If you examine the previous script carefully, you'll easily discover that it's incomplete: it includes the sentences that create the schema to persist your objects, but it doesn't include a single sentence to create the database itself, the users, or to assign privileges to them.

## Here are a few useful scripts!



In our workspace template, there's a project called "scrapbook" that provides SQL templates to create databases, to create users, to drop them, to grant privileges to them, to revoke them, or to hash their passwords. We're pretty sure that you're familiar with these scripts since we've used them a lot of times in multiple problems in the previous lessons.

## Completing the script



```
File Edit Format View Help
start transaction;
create database 'Acme-Certifications';
use 'Acme-Certifications';
create user 'acme-user'@'%' identified by password '4F10007AADA9EE3DBB2CC36575DFC6
create user 'acme-manager'@'%' identified by password 'FDB8CD304EB2317D10C95D797A4
grant select, insert, update, delete
on 'Acme-Certifications' .% to 'acme-user'@'%';
grant select, insert, update, delete, create, drop, references, index, alter,
Create temporary tables, lock tables, create view, create routine,
alter routine, execute, trigger, show view
on 'Acme-Certifications' .% to 'acme-manager'@'%';
-- MySQL dump 10.13 Distrib 5.5.29, for win64 (x86)
-- Host: localhost      Database: Acme-Certifications
-- -----
-- Server version      5.5.29
/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
```

You have to edit the script that is generated by “mysqldump” to add the sentences required to create the users, the database, and to grant the users the appropriate privileges on the database. The result should be more or less like in this slide. Please, note that the initial sentence’s “start transaction”, so that no changes are committed to the database if a problem occurs; it’s not shown in the slide, but you need to add a “commit” sentence at the end of the file to close the transaction.

**NOTE:** the MySQL version that we’re using doesn’t support transactions regarding the Schema Manipulation Language; that is, if a problem occurs while inserting data in a table, then the transaction is aborted, which means that the data entered is erased, but not the schema of the database. Please, keep the “start transaction; ... commit;” block since this problem might well be fixed in future releases of MySQL.



Let's go on with the application artefact.

## What's a war package?

---

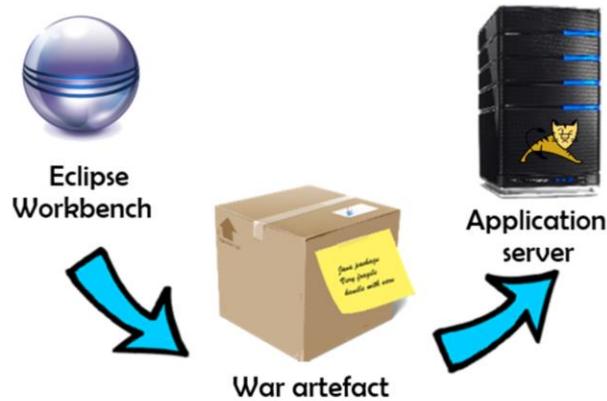


It's a package that contains the classes  
and resources required to run your  
application within a Tomcat server

A war package is a file that contains the classes and resources required to run your application within a Tomcat server. The classes are the binary files that result from compiling your source code and the components on which it relies and the resources include images, scripts, styles, views, i18n&l10n bundles, and configuration files. Simply put, a war package includes almost everything in your project except for the source code of your Java classes and the database.

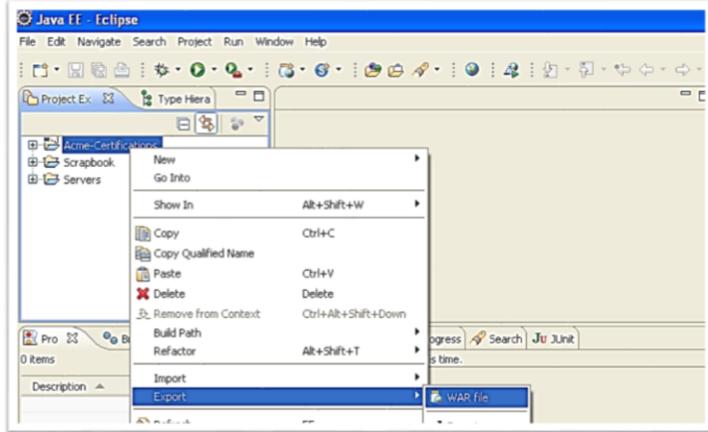
## The process

---



This is the process that we're going to follow: first, we'll use the Eclipse workbench on our developer's virtual machine to create a war package and we'll then upload it to Tomcat.

## Creating a war package (I)

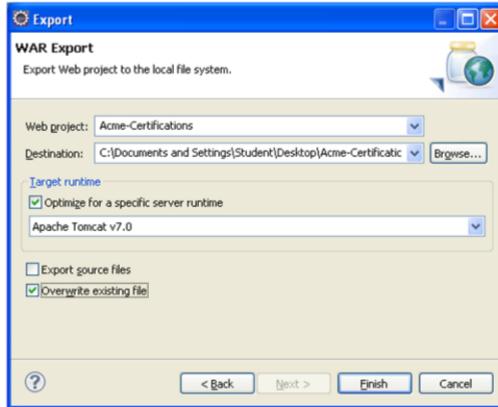


UNIVERSIDAD DE SEVILLA

47

Creating a war package's a simple task! Just right click your project on the project explorer, then select “Export”, and then select “WAR file”.

## Creating a war package (II)



Eclipse will show this dialog in which you only need to select the destination file and click “Finish” to get your war package. We however recommend that you should also check the “Optimise for a specific server runtime” option, since this instructs Eclipse to generate a war package that is specifically tailored to our specific application server, if possible. Note that you can also export your source code into the war package but, as we mentioned earlier, this doesn’t usually make a lot of sense. So you’d better leave this option unchecked.



Once the deployment artefacts are ready, it's time to deploy them. Let's first explore how to deploy them to a cloud computer, that is, an IaaS.

## A simple definition

---



An IaaS is a (virtual) machine that you rent  
from a cloud provider

An IaaS is a machine that you rent from a cloud provider. Typically the machine's not a physical machine, but a virtual machine that provides some processors, some memory, some storage, and some bandwidth. Note that renting a machine has many implications. The most important is that you're responsible for installing the operating system, configuring it, securing it, installing the application and the database server, configuring them, securing them, installing your application, running it, and scaling it. It's not the most attractive choice, but it's a common choice in our local context.

## Main players

---

- Amazon Web Services
- AT&T Private Cloud
- Cicos's Cloud Infrastructure
- Citrix
- Google Cloud Platform
- HP's Public Cloud
- IBM's Smart Cloud
- Microsoft's Windows Azure
- Verizon Cloud
- VMware vCloud Air

In this slide, you can find some of the main companies that rent computers on the cloud. Select some of them and browse their web sites to learn about their offers.

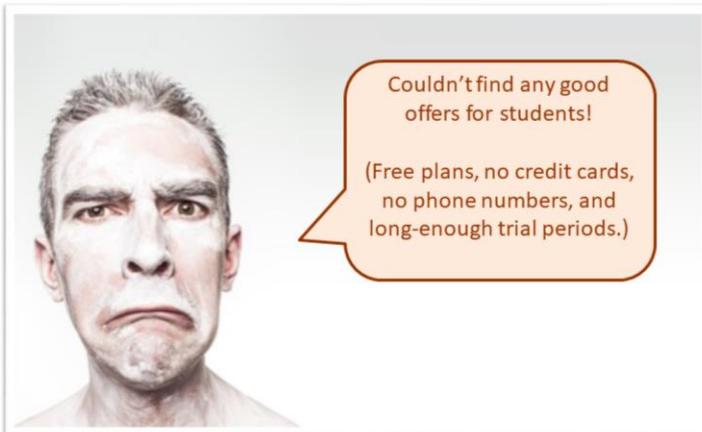
## Secondary players

1&1	DediServer	Liquid Web
Alibaba Cloud	Digital Ocean	LogicWrks
Bit Refinery	Dimension Data	Lunacloud
BlueLock	Do Daddy	NaviSite
Calligo	Elastic Host	Open Nebula
CBTS	Enki	OpenStack
CDNify	Exabytes	OVH
Century Link	GoGrid's Web hosting	Profit Bricks
Cirrcore Enterprise Cloud	Green Cloud Technologies	Ostack
Cirrity	Green House Data	RackSpace Open Cloud
Cloud Sigma	Host 1 Plus	Skytap
CloudForge	Internap	SoftLayer's Cloud Layer
CloudStack	Joyent Triton	The Virtual Machine Company
Datapipe	Linode	Vultr

But there are many secondary players, some of which are listed in this slide. Again, browse some of their web sites to learn about their offers.

## It's a pity!

---



Couldn't find any good  
offers for students!

(Free plans, no credit cards,  
no phone numbers, and  
long-enough trial periods.)

Unfortunately, we haven't found a good offer for students, that is, an offer with free plans, no credit cards, no phone numbers, and long-enough trial periods.

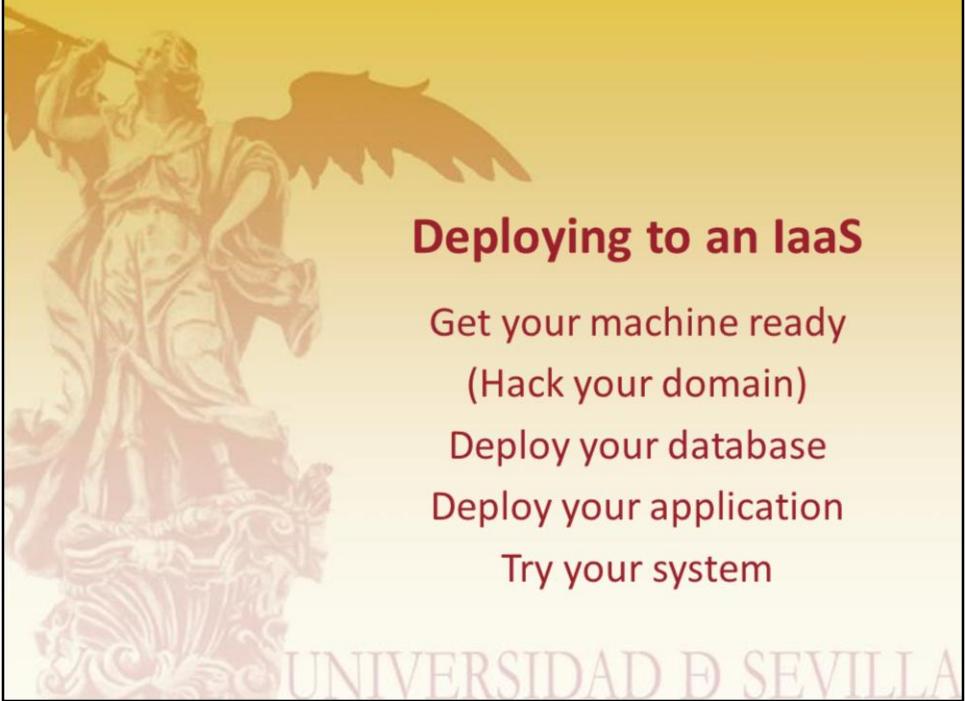
## Don't worry!

---



Don't worry, we provide you with a pre-production machine, which is a virtual machine that you might deploy to any of those cloud providers... but you don't have to!

That's the reason why we provide you with a pre-production machine, which is a virtual machine that you can use to learn how to deploy to an IaaS without actually deploying to any of them. You can run this virtual machine on your laptop and the lecturers will have exactly the same virtual machine to evaluate your results.



## Deploying to an IaaS

Get your machine ready

(Hack your domain)

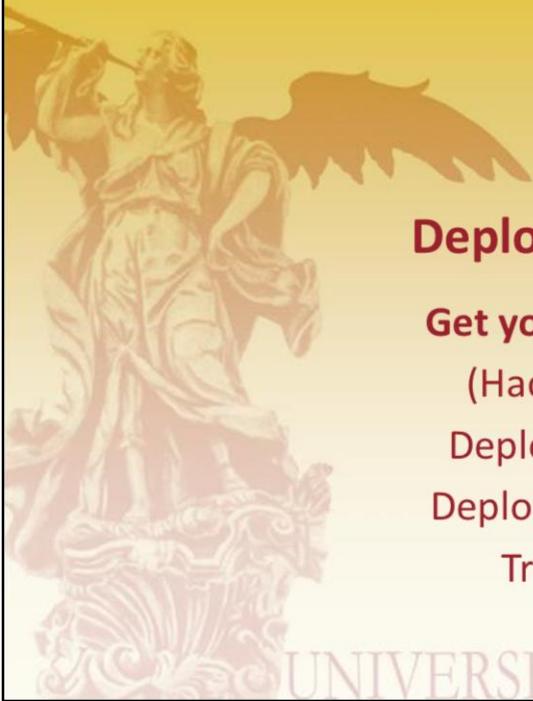
Deploy your database

Deploy your application

Try your system

UNIVERSIDAD DE SEVILLA

This is what we're going to study regarding deploying to an IaaS: first, we'll explore how to get your machine ready, then we'll explore how to hack your domain (which is something that you must perform on the pre-production machine that we provide only), then we'll learn how to deploy your database, how to deploy your application, and, finally, how to give a try to your system.



## Deploying to an IaaS

**Get your machine ready**

(Hack your domain)

Deploy your database

Deploy your application

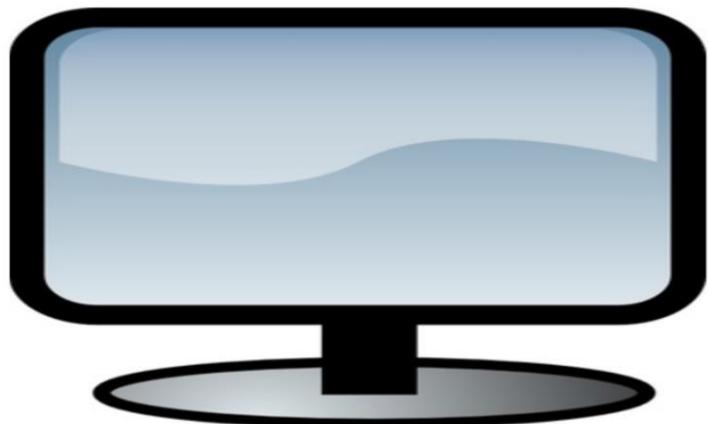
Try your system

UNIVERSIDAD DE SEVILLA

Let's start with getting your machine ready.

## What you get: bare metal

---



This is what you typically get from your provider: bare metal.

## Install a server operating system

---



The first thing that you have to do is to install an operating system. Most providers have images that they can copy to your machine in a matter of minutes. Windows or GNU/Linux are good choices; unfortunately, Apple does not have any server version of their operating systems. Please, recall that you should almost never work as an administrator in your developer's virtual machine; in the pre-production machine, however, the only few operations that you need to perform are commonly administrative ones, so you must use your administrator's account all the time. Please, do not create any other user accounts except for the non-privileged accounts to run your database server and your application server, if necessary.

## Configure it

---



It's very important that you configure the operating system for maximum performance and security. Please, review the following general guidelines:

- a) Have a single administration account; make sure the username's not "administrator", "administrador", "root", or something like that. These username's are far too obvious for hackers.
- b) Remove every unnecessary component, e.g., desktop accessories, mail clients, media players, messaging utilities, and the like; just keep a browser, if actually necessary.
- c) Remove every unnecessary service, e.g., multi-media services, theme services, DNS servers, and the like.
- d) Configure the firewall for maximum protection; only port 80 should be available to external customers.
- e) Search for other security guidelines that are applicable to your operating system and apply them.

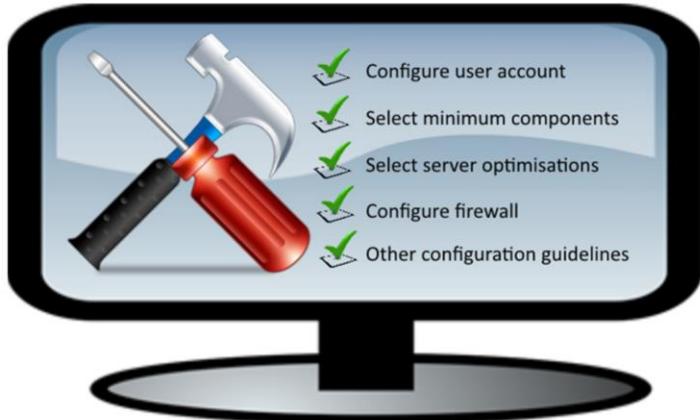
## Install your servers

---



Install a MySQL Server and a Tomcat server.

## Configure them



And configure them. Please, review the following general guidelines:

- a) Configure the user account used to run them. Don't use the administrator's account! Windows provides a couple of non-privileged user accounts to run services, namely: "Local Service" and "Network Service"; most servers are installed so that they run on the appropriate user account automatically. GNU/Linux doesn't have a standard user account to run services; please, consult your distribution's documentation to find out which the most appropriate account is or to create a new one.
- b) Select the minimum number of components required to run your system. Typically, the servers have a number of components that are very useful in some projects, but completely useless in others. Keeping the components that you install to a minimum increases the security of your system.
- c) Select server optimisations where available so that they perform as fast as possible. For instance, Tomcat runs a lot faster when the so-called Apache Portable Runtime (APR) is available in the class path; similarly, MySQL has some installation options that optimise it for production.
- d) Configure the firewall to allow the minimum number of exceptions for your servers. Tomcat requires port 80 to be open and MySQL requires port 3306 to be open; that's all!
- e) And don't forget to review the documentation of your servers and to follow their specific guidelines.



## Deploying to an IaaS

Get your machine ready

**(Hack your domain)**

Deploy your database

Deploy your application

Try your system

UNIVERSIDAD DE SEVILLA

Let's now report on how to hack your domain. We wrote this section in rounded parenthesis because this is something that you have to perform on the pre-production machine that we provide; it's not the kind of thing that you're likely to perform in a real project.

## Our domain: [www.acme.com](http://www.acme.com)

---



The machine on which you're going to deploy your system has to serve an application for a customer of yours, who is very likely to have an internet domain and a DNS. In such cases, your customer has to enter a new record in his or her DNS to redirect the traffic to his or her domain to the machine to which you've deployed his or her system..

## We need to hack it

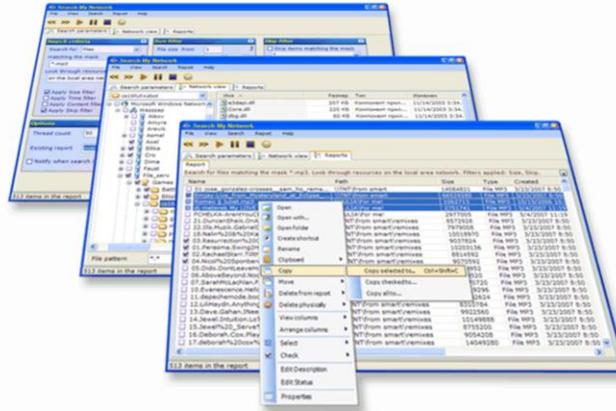
---



Unfortunately, we don't have an actual customer with an actual internet domain, so we can't route external requests to "www.acme.com" to our pre-production machine. Don't worry, we'll hack the domain. It's not difficult, but it's very low-level. It's kind of entering the Matrix!

**NOTE:** hacking a system so that it emulates an internet domain's highly dependent on your operating system. We provide you with a few hints if you're using GNU/Linux, but we can't provide a general solution that works with every distribution of this operating system. Please, consult your operating system's documentation or search the Web for information when necessary.

## Set up your hosts file

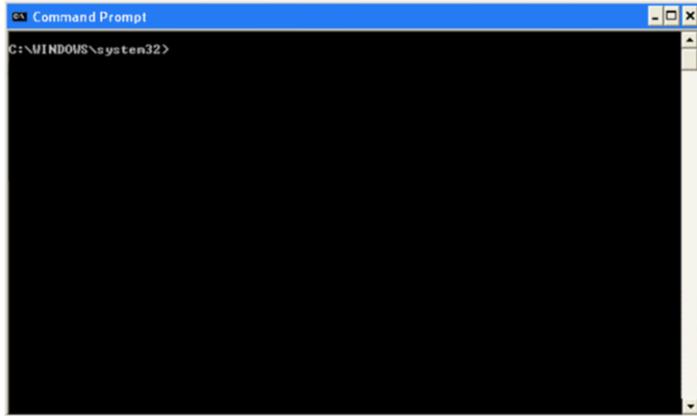


UNIVERSIDAD DE SEVILLA

65

Have you ever heard of a file called “hosts”? It provides your operating system with a local DNS translation table by means of which you can override the records in public DNS servers. Simply put, you can configure your operating system so that every time you make your browser for “www.acme.com” it locally translates this address into “localhost”.

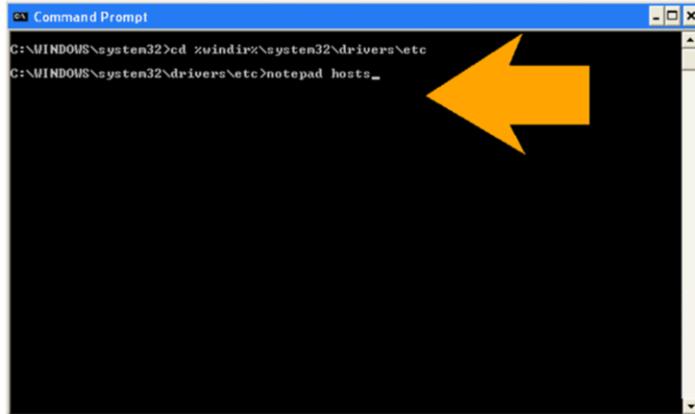
## Open an administrator's shell



To change your “hosts” file, you need to work on an administrator’s shell. Just press Windows+R and type “cmd.exe”, which will result in a shell like the one in this slide.

**NOTE:** from now on, unless otherwise stated, we’ll assume that you’re working on a pre-production machine in which there’s only an administrator’s account. Please, consult your operating system’s documentation if you’re not using Windows. In the pre-production machine, the administrator’s credentials are “boss/\$I=B0\$\$=U\$3=P@\$\$”.

## Locate file hosts

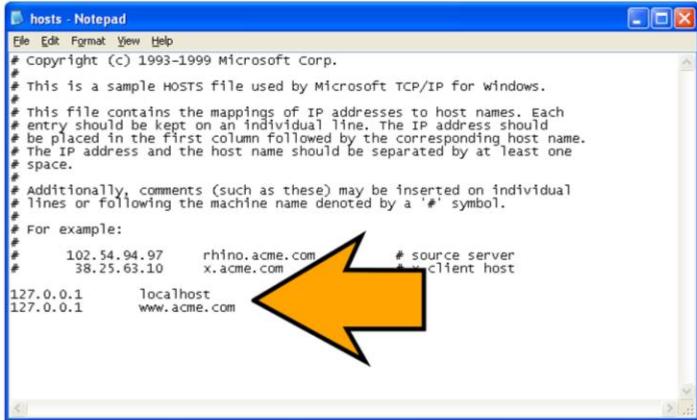


Now, find the “hosts” file and open it in a text editor using the following commands:

```
> cd %windir%\system32\drivers\etc  
> notepad hosts
```

**NOTE:** please, consult your operating system’s documentation to find out where your “hosts” file resides. There’s a document at [http://en.wikipedia.org/wiki/Hosts\\_file](http://en.wikipedia.org/wiki/Hosts_file) with information regarding this file’s path in many operating systems.

## Edit and save it

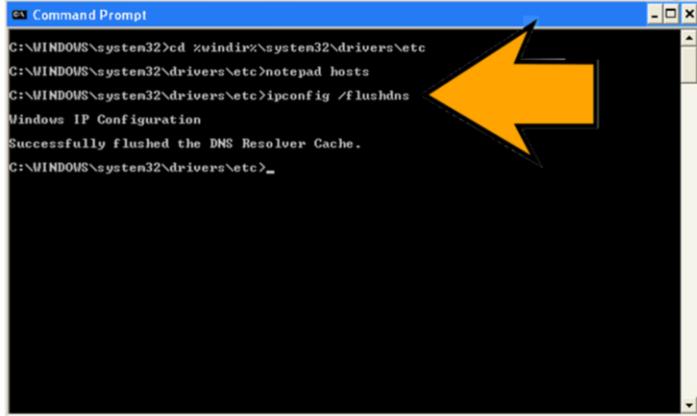


This is what the file looks like. It has a lot of comments that start with a hash sign.  
Add the following two lines at the end:

```
127.0.0.1      localhost
127.0.0.1      www.acme.com
```

You should know that “127.0.0.1” is your computer’s loopback address. These two lines define two hosts called “localhost” and “www.acme.com” that have exactly your computer’s loopback address. Save the file when you’re done. Thanks to the “hosts” file, when your browser dereferences “www.acme.com”, it actually gets IP “127.0.0.1”, that is, your local host.

## Refresh your DNS cache



```
Command Prompt  
C:\WINDOWS\system32>cd %windir%\system32\drivers\etc  
C:\WINDOWS\system32\drivers\etc>notepad hosts  
C:\WINDOWS\system32\drivers\etc>ipconfig /flushdns  
Windows IP Configuration  
Successfully flushed the DNS Resolver Cache.  
C:\WINDOWS\system32\drivers\etc>
```

A large orange arrow points from the left towards the command "ipconfig /flushdns" in the screenshot.

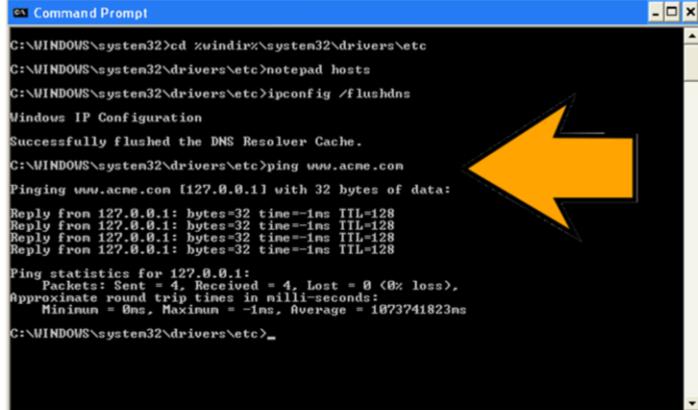
Now, you need to refresh your DNS cache by means of the following command:

> ipconfig /flushdns

This instructs your operating system to reload the “hosts” file.

**NOTE:** please, consult your operating system’s documentation to find out how the DNS host cache is refreshed. Unfortunately, there’s not a standard way to flush the cache in Unix-based operating systems. You may try the hints that are available at <https://help.dreamhost.com/hc/en-us/articles/214981288-Flushing-your-DNS-cache-in-Mac-OS-X-and-Linux>.

## Ping www.acme.com



```
C:\> Command Prompt  
C:\> cd %windir%\system32\drivers\etc  
C:\> notepad hosts  
C:\> ipconfig /flushdns  
Windows IP Configuration  
Successfully flushed the DNS Resolver Cache.  
C:\> ping www.acme.com  
Pinging www.acme.com [127.0.0.1] with 32 bytes of data:  
Reply from 127.0.0.1: bytes=32 time=1ms TTL=128  
Ping statistics for 127.0.0.1:  
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),  
    Approximate round trip times in milli-seconds:  
        Minimum = 0ms, Maximum = 1ms, Average = 1073741823ms  
C:\>
```

UNIVERSIDAD DE SEVILLA

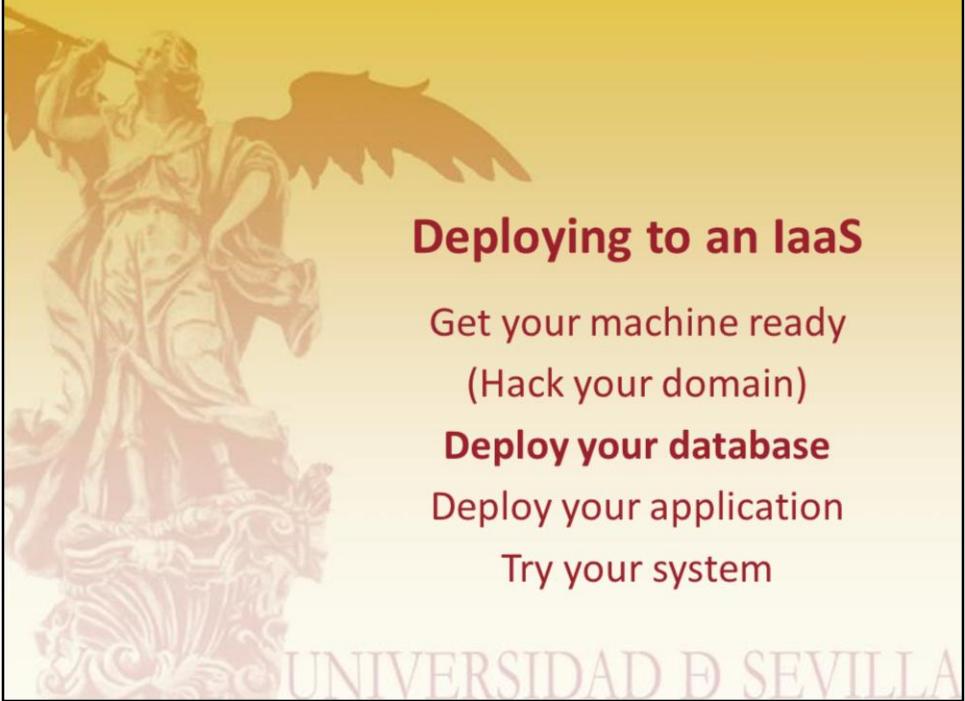
70

You may try immediately your new internet domain. Enter the following command:

> ping www.acme.com

You should get replies from “127.0.0.1”, which indicates that requests to “www.acme.com” are being routed to your computer. That’s enough so far.

**NOTE:** some systems are configured not to reply to ping requests. Please, consult your operating system’s and your firewall’s documentation if you can’t ping “www.acme.com”.



## Deploying to an IaaS

Get your machine ready

(Hack your domain)

**Deploy your database**

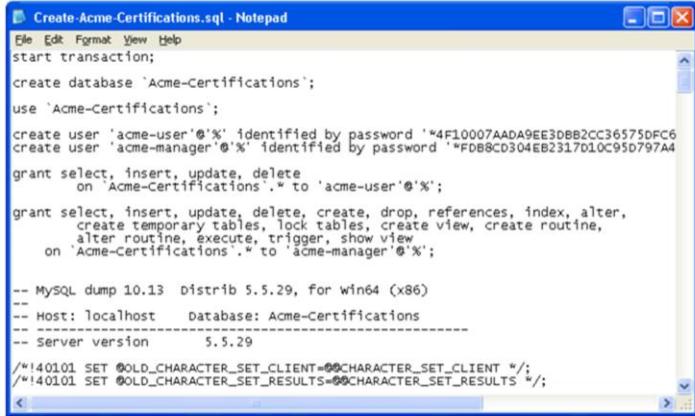
Deploy your application

Try your system

UNIVERSIDAD DE SEVILLA

Entering the matrix wasn't difficult, was it? Let's now delve into how to deploy your database.

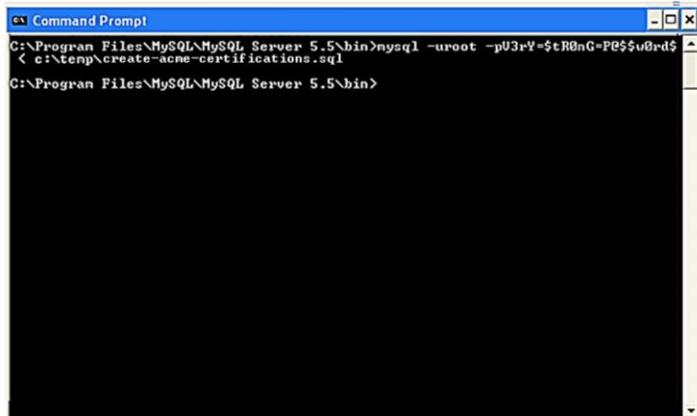
## Transfer the database artefact and ...



```
File Edit Format View Help
start transaction;
create database `Acme-Certifications`;
use `Acme-Certifications`;
create user 'acme-user'@'%' identified by password '4F10007AADA9EE3DBB2CC36575DFC6
create user 'acme-manager'@'%' identified by password 'FDB8CD304EB2317D10C95D797A4
grant select, insert, update, delete
    on `Acme-Certifications`.* to 'acme-user'@'%';
grant select, insert, update, delete, create, drop, references, index, alter,
    create temporary tables, lock tables, create view, create routine,
    alter routine, execute, trigger, show view
    on `Acme-Certifications`.* to 'acme-manager'@'%';
-- MySQL dump 10.13 Distrib 5.5.29, for win64 (x86)
-- Host: localhost      Database: Acme-Certifications
-- -----
-- Server version      5.5.29
/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
```

It's pretty simple: transfer the database artefact to the pre-production machine.

## ... And execute it!



And execute it! You need to open an administrator's shell on your pre-production configuration, change the working directory to MySQL's bin directory and execute the following command:

```
> mysql -u<root> -p<pass> < <script-file.sql>
```

Where `<root>` and `<pass>` refer to the database root credentials and `<script-file.sql>` refers to the file that provides the database creation script. If everything goes well, then you shouldn't see any messages on the screen; if there are any problems, please, interpret the error messages and correct them.

**NOTE:** in the pre-production machine, the administrator's credentials are "boss/\$I=BO\$\$=U\$3=P@\$@", and the database root's credentials are "root"/"V3rY=\$tR0nG=P@\$@\$\$w0rd\$".

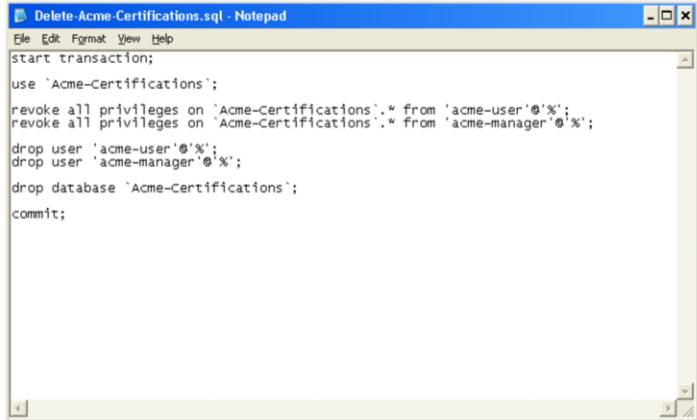
## That's a good request

---



Before concluding this section, we'd like to provide you with a few hints that might help you if you get in trouble.

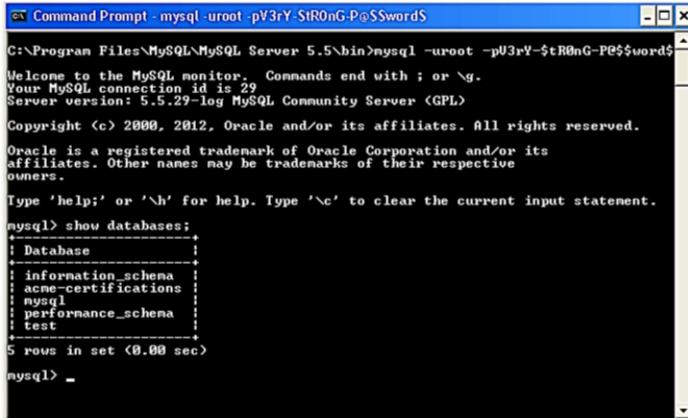
## Hint #1: how to drop everything?



```
 Delete-Acme-Certifications.sql - Notepad
File Edit Format View Help
start transaction;
use `Acme-Certifications`;
revoke all privileges on `Acme-Certifications`.* from 'acme-user'@'%';
revoke all privileges on `Acme-Certifications`.* from 'acme-manager'@'%';
drop user 'acme-user'@'%';
drop user 'acme-manager'@'%';
drop database `Acme-Certifications`;
commit;
```

The first hint's that you should also create a script to completely remove your database from the server. It's very common that you make mistakes and that you have to deploy your database several times before you command the procedure. Every time you do that, you need to remove your database from the server. In this slide, we show a typical script to remove every trace of your database from a configuration: first you have to revoke your users' privileges, then you have to drop them, and, finally, you have to drop the database.

## Hint #2: listing databases



```
C:\> Command Prompt - mysql -uroot -pV3rY-$tR0nG-P@$$word$  
C:\> Program Files\MySQL\MySQL Server 5.5\bin>mysql -uroot -pV3rY-$tR0nG-P@$$word$  
Welcome to the MySQL monitor. Commands end with ; or \g.  
Your MySQL connection id is 29  
Server version: 5.5.29-log MySQL Community Server (GPL)  
Copyright (c) 2000, 2012, Oracle and/or its affiliates. All rights reserved.  
Oracle is a registered trademark of Oracle Corporation and/or its  
affiliates. Other names may be trademarks of their respective  
owners.  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
mysql> show databases;  
+ Database +  
| information_schema |  
| acne-certifications |  
| mysql |  
| performance_schema |  
| test |  
5 rows in set <0.00 sec>  
mysql> -
```

The second hint's regarding the command you need to know in order to list the databases that are available at your server. First you have to open a MySQL shell by means of the following command:

```
> mysql -u<root> -p<pass>
```

Where <root> and <pass> refer to the database root credentials. There are a few interesting MySQL commands that you can run in this shell. The first one is “show databases”, which reports on the names of the databases that are available at the server, including the system databases “information\_schema”, “mysql”, “performance\_schema”, and “test”.

**NOTE:** in the configuration that we provide, the root credentials are “root”/”V3rY=\$tR0nG=P@\$\$w0rd\$”.

## Hint #3: listing users

```
mysql> select user, host, password from mysql.user;
+-----+-----+-----+
| user | host | password          |
+-----+-----+-----+
| root | localhost | *BBC130FB08F15ED0EDC89D492B7F388952AC8E |
| root | 127.0.0.1 | *BBC130FB08F15ED0EDC89D492B7F388952AC8E |
| root | ::1 | *BBC130FB08F15ED0EDC89D492B7F388952AC8E |
| acme-user | % | *4F10007A0AYEE3DBB2CC36575DFC6F4FDE27577 |
| acme-manager | % | *FD88CD304EB2317D10C95D797A0BD7492560F55F |
+-----+-----+-----+
5 rows in set (0.00 sec)

mysql>
```

UNIVERSIDAD DE SEVILLA

77

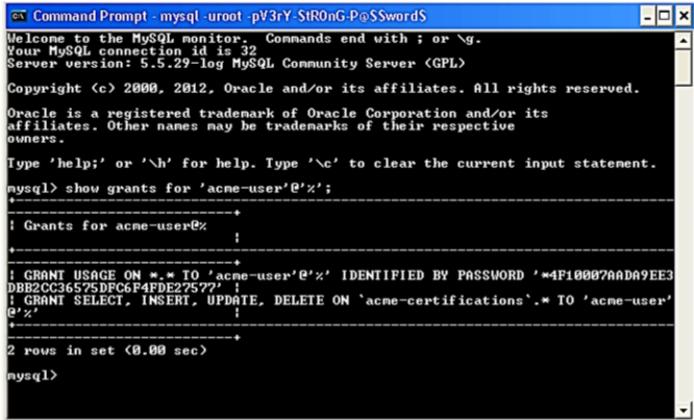
You can also use the following command to list the users that are registered in the server:

```
> select user, host, password from mysql.user;
```

It lists their usernames, the computers from which they are allowed to connect to the server, and the hashes of their passwords. Please, recall that no serious application should store passwords in clear text.

**NOTE:** In MySQL's parlance, a host is a computer and "%" means any host.

## Hint #4: listing privileges



The screenshot shows a Windows Command Prompt window titled "Command Prompt - mysql -uroot -pV3rY STROnG-P@SSwordS". The window displays the MySQL monitor welcome message, copyright information, and a command-line session. The user has run the command "show grants for 'acme-user'@'%';" which lists the grants for the user 'acme-user'. The output shows two grants: one for usage on all databases and another for SELECT, INSERT, UPDATE, and DELETE on the 'acme-certifications' database. The session ends with "2 rows in set <0.00 sec>".

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 32
Server version: 5.5.29-log MySQL Community Server (GPL)

Copyright (c) 2000, 2012, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

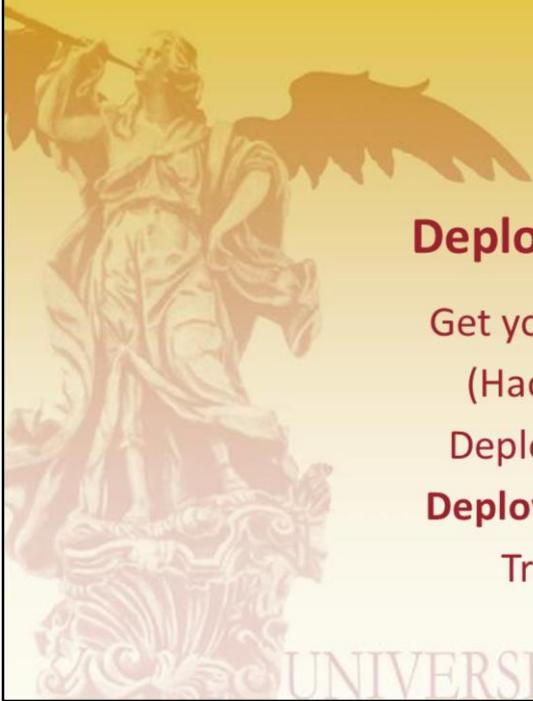
mysql> show grants for 'acme-user'@'%';
+-----+
| Grants for acme-user@%          |
+-----+
| GRANT USAGE ON *.* TO 'acme-user'@'%' IDENTIFIED BY PASSWORD '*4F10007AAD9EE3DBB2CC36575DFCF4FDE27577' |
| GRANT SELECT, INSERT, UPDATE, DELETE ON `acme-certifications`.* TO 'acme-user'@'%'           |
+-----+
2 rows in set <0.00 sec>

mysql>
```

Finally, you can use the following command to list the privileges (aka grants) that a user has:

> `show grants for <username>@<host>`

where `<username>@<host>` refers to the username and the host in which we're interested.



## Deploying to an IaaS

Get your machine ready

(Hack your domain)

Deploy your database

**Deploy your application**

Try your system

UNIVERSIDAD DE SEVILLA

Let's now delve into deploying your application.

## Enter the app manager



UNIVERSIDAD DE SEVILLA

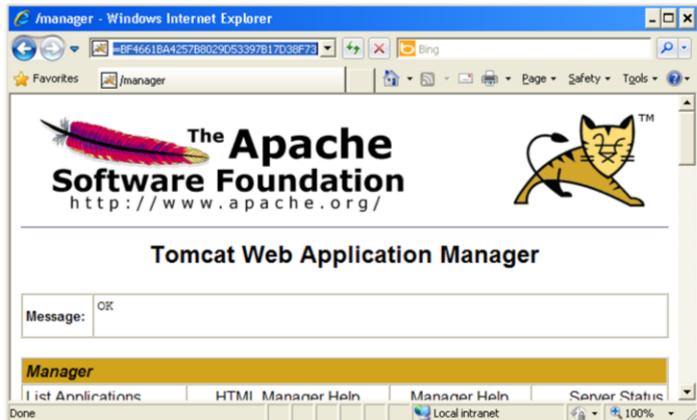
80

Open a browser and make it for “<http://localhost/manager>”. “manager” is a pre-defined application context in which Tomcat’s service installs a simple application manager that will help us deploy our sample system. An authentication window will popup. Just key in Tomcat’s administrator’s credentials and press the “OK” button.

**NOTE:** in the virtual machine that we provide, the administrator’s credentials are “boss/\$!=\$0\$\$=U\$3=P@\$\$”, and Tomcat’s administrative credentials are “admin”/“T0mC@t=Adm1n1\$trat0R”.

## This is what it looks like

---



UNIVERSIDAD DE SEVILLA

81

This is what the application manager looks like. It's quite a simple web application.

## Scroll to see the applications

The screenshot shows the 'Applications' section of the Tomcat Manager interface. It lists two application contexts: 'Welcome to Tomcat' and 'Tomcat Documentation'. Both contexts are running, have 0 sessions, and are associated with a session timeout of 30 minutes. The 'Welcome to Tomcat' context is highlighted with a green background.

Path	Version	Display Name	Running	Sessions	Commands
/	None specified	Welcome to Tomcat	true	0	<button>Start</button> <button>Stop</button> <button>Reload</button> <button>Undeploy</button>  <button>Expire sessions</button> with idle ≥ 30 minutes
/docs	None specified	Tomcat Documentation	true	0	<button>Start</button> <button>Stop</button> <button>Reload</button> <button>Undeploy</button>  <button>Expire sessions</button> with idle ≥ 30 minutes

UNIVERSIDAD DE SEVILLA

82

Scroll a little down to see the applications that are installed by default: a welcome application, documentation, examples, and the manager itself.

**NOTE:** the vocabulary is a little confusing here since “application contexts” are referred to as “paths” in the application manager. (See the header of the first column).

## Tidy your server up!

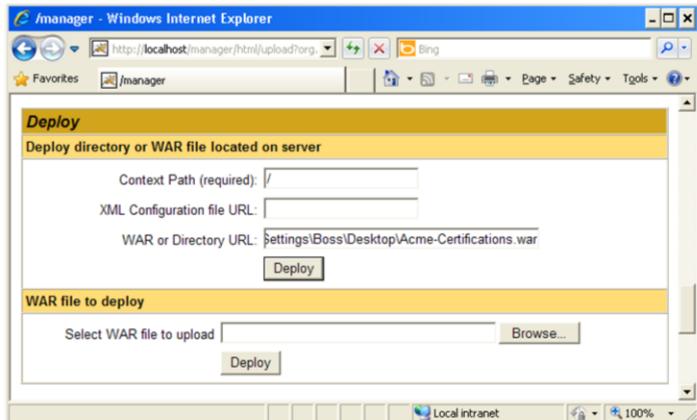
The screenshot shows a Windows Internet Explorer window titled '/manager - Windows Internet Explorer'. The address bar contains the URL 'http://localhost:8080/manager'. The main content area displays the 'Applications' section of the Tomcat Manager. A table lists one application:

Path	Version	Display Name	Running	Sessions	Commands
/manager	None specified	Tomcat Manager Application	true	1	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes

Below the applications table, there is a 'Deploy' section with a sub-section 'Deploy directory or WAR file located on server'. It includes a 'Context Path (required):' input field and a 'Done' button.

The first thing you must do is to tidy your server a little up. Please, undeploy every default application, but the application manager.

## Deploy your war package



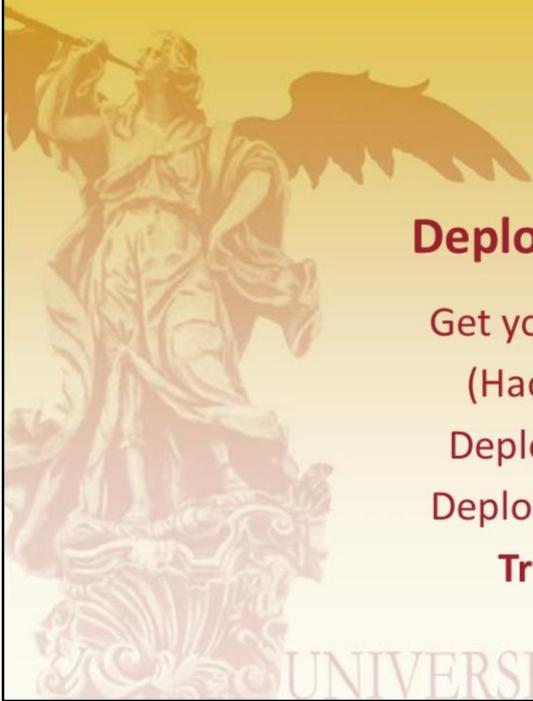
We're ready to deploy our war package. Scroll down until you get to a section entitled "Deploy". There are two subsections, and you have to use the first one: "Deploy directory or war file located on server". You just need to provide an application context (a path in the manager's parlance) and the path to the war package in your file system; please, ignore the XML configuration file. The application context should be "/", since we wish our sample application to be available from the root context of our web domain. Note that the subsection entitled "WAR file to deploy" also allows you to deploy a war package and it offers a typical "Choose file" button by means of which you can easily locate your war package in the file system; unfortunately, if you deploy your artefacts using this choice, you won't be able to indicate the application context to which you wish to deploy them. They'll be deployed to a context whose name's the name of your war package; if your package is named "ROOT.war", then it'll be deployed to the root context..

# Is everything OK?

The screenshot shows the 'Applications' section of the Tomcat Manager interface. It lists two applications:

Path	Version	Display Name	Running	Sessions	Commands
/	None specified	Acme-Certifications	true	0	<button>Start</button> <button>Stop</button> <button>Reload</button> <button>Undeploy</button>  <button>Expire sessions</button> with idle ≥ 30 minutes
/manager	None specified	Tomcat Manager Application	true	1	<button>Start</button> <button>Stop</button> <button>Reload</button> <button>Undeploy</button>  <button>Expire sessions</button> with idle ≥ 30 minutes

If everything's OK, then you should see your application in the “Applications” section. There you can stop, reload, start, or undeploy it whenever necessary.



## Deploying to an IaaS

Get your machine ready

(Hack your domain)

Deploy your database

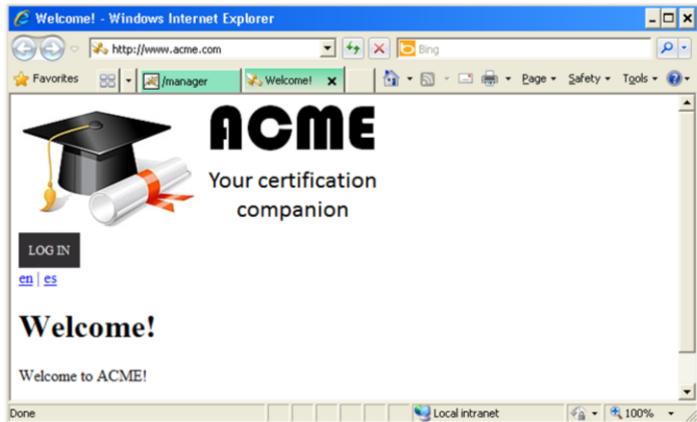
Deploy your application

**Try your system**

UNIVERSIDAD DE SEVILLA

It's the right moment to try your system.

Try <http://www.acme.com> now!

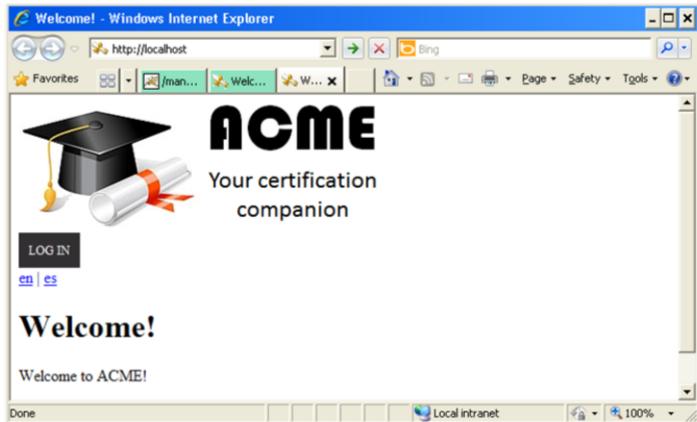


UNIVERSIDAD DE SEVILLA

87

Make your browser for “<http://www.acme.com>” and your application should show up in a few seconds. Note that “[www.acme.com](http://www.acme.com)” works locally because we hacked our pre-production machine so that it overrides the public DNS record for this internet domain; on other computers, “[www.acme.com](http://www.acme.com)” will show you customer’s current web site.

## Try `http://localhost`, too



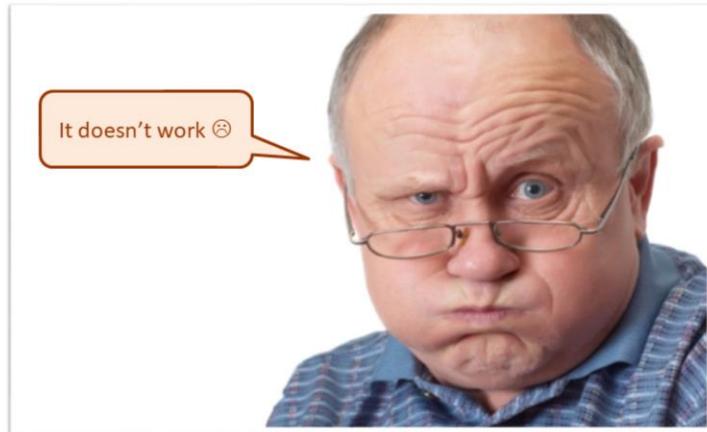
UNIVERSIDAD DE SEVILLA

88

Try making your browser for “`http://localhost`”, too. The result should be the same. Please, recall that both “`localhost`” and “`www.acme.com`” redirect to your computer’s IP address.

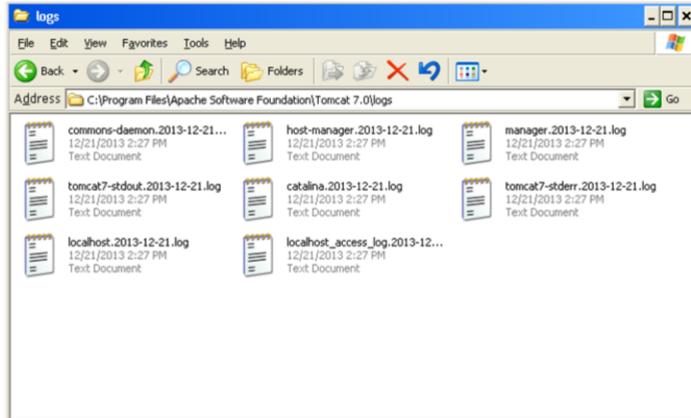
## Ooohhh, sorry!

---

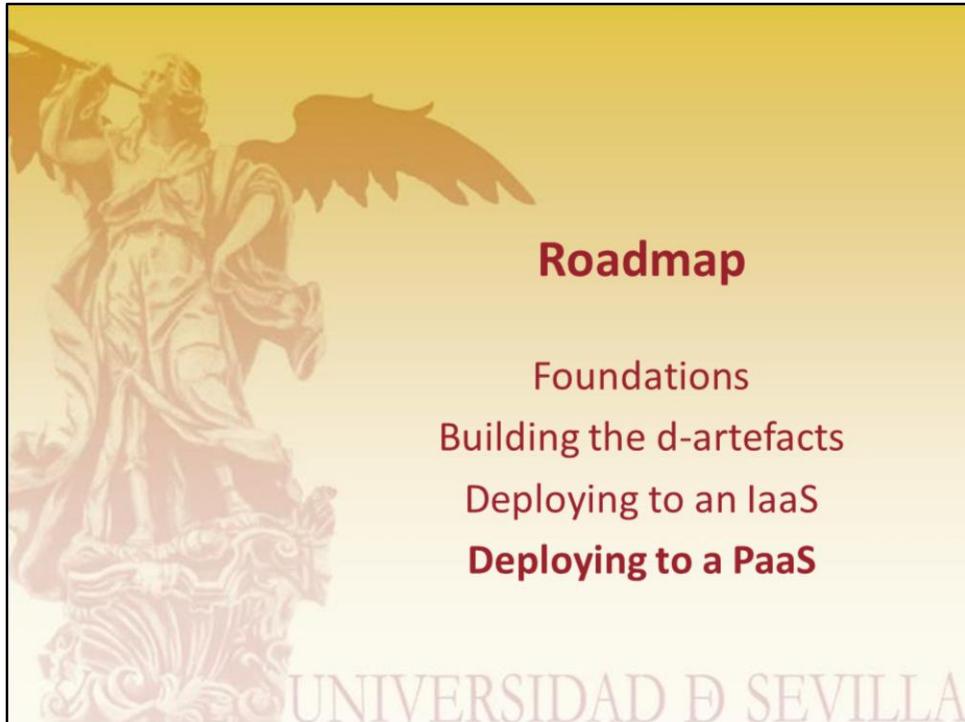


Doesn't it work? We're sorry. In that case you need to check every step and repeat them until you find the problem that's preventing you from deploying and using your web information system.

## A good hint: check Tomcat's logs



In case of trouble, there's very little you can do to debug your system. We suggest that you should take a look at Tomcat's logs. Please, search for the log files that correspond to the day on which you're working and analyse them. The logs typically show exceptions that may give you a hint regarding what the problem is. Typically, there are many logs, so it makes sense to remove them, start the system up, reproducing the problem, and then analysing the logs again since this guarantees that every thing that you see in the log files is related to the problem.



Let's now explore how to deploy your system to a PaaS.

## A simple definition

---



A PaaS is a platform (application server plus database server) that you rent from a cloud provider

A PaaS is a platform that you rent from a cloud provider. By platform, we mean an application server and a database server. Obviously, they both run on physical or virtual machines; the key here is that you don't have to care about them. You just care of your servers and forget about the operating system, user accounts, firewalls, and the like.

## Main players

---

- Amazon Web Services
- Force.com
- Google App Engine
- IBM Bluemix
- Microsoft Azure
- Oracle Cloud
- Redhat OpenShift
- SAP Hana Cloud
- Software AG Live

In this slide, you can find some of the main players. Select some of them and browse their webs to find out about their offers.

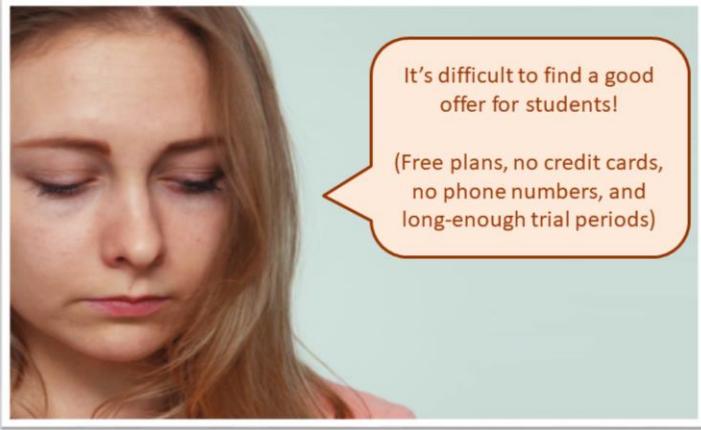
## Secondary players

Acquia Cloud	Cloud Ways	MoPaaS
Anynines	CloudNote	mOSAIC
App 42	ConPaaS	Now
App Agile	Convoy	Nuvla
App Fog	Dokku	Orange Scape
App Harbor	Dokkur	OutSystems
App Scale	Engine Yard	Pagoda Box
Apprenda	Flynn	Pivotal Web Services
Apprenda	FortRabbit	Platform.sh
Appuio	Getup Cloud	PodSpace
Bright Box	Gigalixir	Predix
Clever Cloud	Heirloom PaaS	Scalingo
Clodify	Heroku	Standing Cloud
Cloud 66	Jelastic	Tsuru
Cloud Bees	Mendix	Zoho Creator
Cloud Foundry	MeshCloud	

But there are many other secondary players, some of which are listed in this slide. Again, browse some of their web sites to learn about their offers.

## It's a pity!

---



It's difficult to find a good offer for students!

(Free plans, no credit cards, no phone numbers, and long-enough trial periods)

Unfortunately, it's difficult to find a good offer for students, that is: free plans available, no credit cards, no phone numbers, and long-enough trial periods.

## Don't worry!

---



But don't worry, we have found a good-enough provider that is appropriate for our students. It's Clever Cloud. It's not full of features, but it's enough to learn the foundations about deploying to a PaaS.



## Deploying to a PaaS

Create an account

Create your system

Deploy your database

Deploy your application

Try your system

UNIVERSIDAD DE SEVILLA

This is our roadmap in this section: we'll report on how to create an account, how to create your system, how to deploy your database, how to deploy your application, and, finally, how to give it a try.



## Deploying to a PaaS

**Create an account**

Create your system

Deploy your database

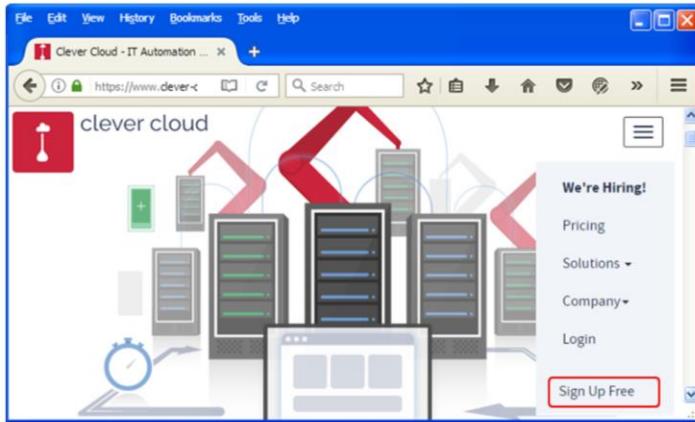
Deploy your application

Try your system

UNIVERSIDAD DE SEVILLA

Let's start with how to create an account.

## The welcome page



UNIVERSIDAD DE SEVILLA

99

Make your browser for <http://www.clever-cloud.com> and you'll get a simple welcome page with a menu with which you can sign up for free.

## Create an account

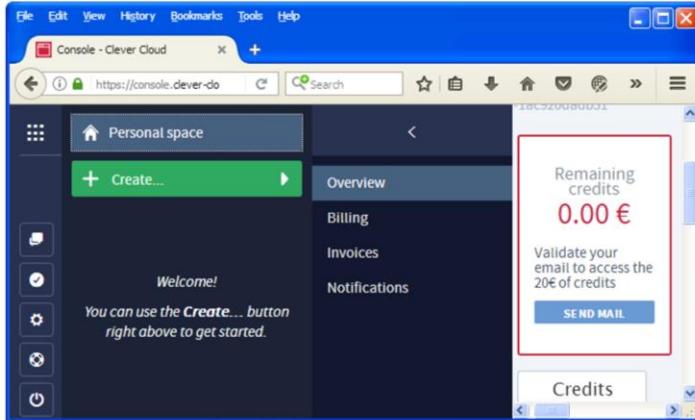
The screenshot shows a web browser window with a blue header bar containing the title "Sign up · Clever Cloud" and the URL "https://api.clever-cloud.o...". Below the header is a search bar and a toolbar with various icons. The main content area is a form titled "Sign up with Email". It has three input fields: "EMAIL ADDRESS" containing "corchu@mail.com", "PASSWORD" containing a series of asterisks, and "VERIFY PASSWORD" also containing a series of asterisks. A blue button at the bottom right of the form area says "SIGN UP WITH EMAIL".

UNIVERSIDAD DE SEVILLA

100

Fill in the sign up form, and hit the submission button.

## Get your free credit



UNIVERSIDAD DE SEVILLA

101

When you log in for the first time, you'll get a page like the one in this slide. The initial credit is 0.00€, but you may easily increase it to 20.00€ by registering your email. 20.00€ does not seem a lot of credit, but it's enough for the purposes of this subject.



## Deploying to a PaaS

Create an account

**Create your system**

Deploy your database

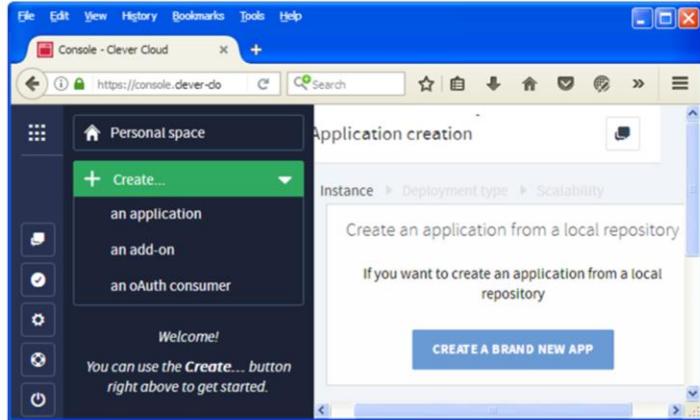
Deploy your application

Try your system

UNIVERSIDAD DE SEVILLA

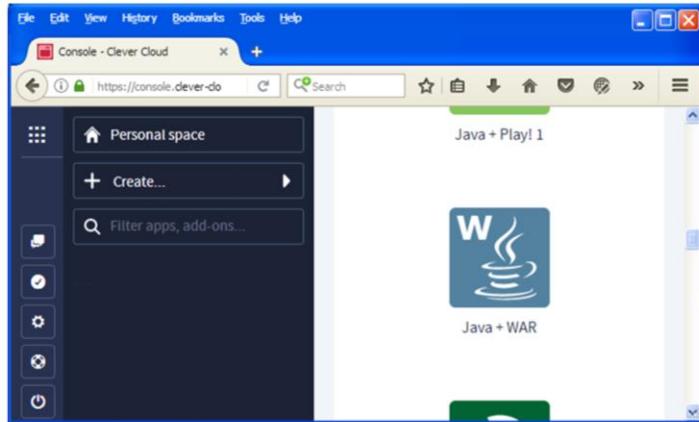
It's now time to create your system.

## Create an application



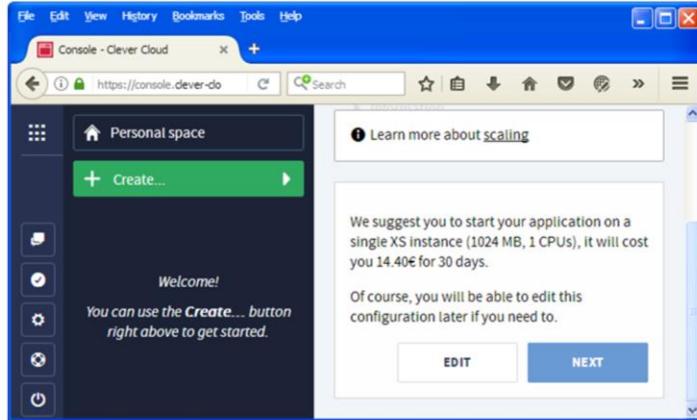
First, we need to create the application. Click on “Personal Space”, then on “+ Create...”, and then on “Create a brand new app”.

## Select the Java + War template



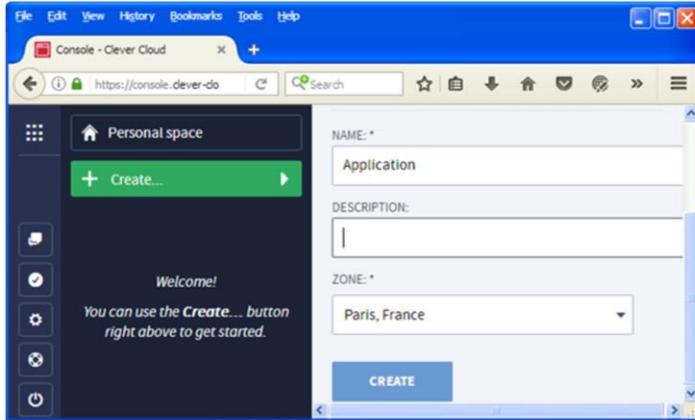
First, you have to select the type of application that you wish to deploy. There are a variety of choices amongst which we have to select “Java + WAR”.

## Keep the default instance



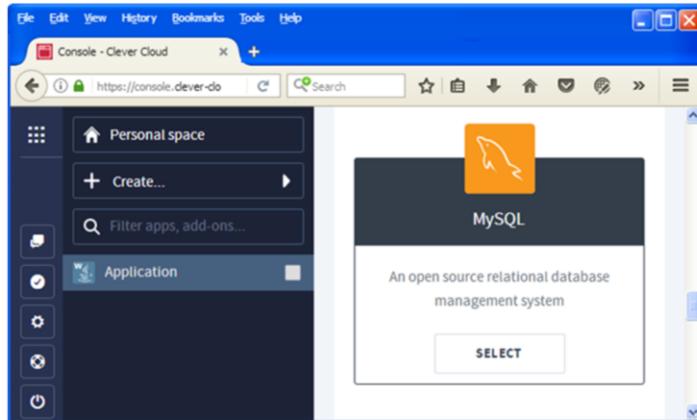
Clever Cloud offers several kinds of machines to deploy your systems. By default, they offer a so-called XS instance which is a virtual machine with 1024 MiB of RAM memory and 1 CPU. This is enough to start working, so hit the “Next” button and let’s go ahead!

## Do basic configuration



Now you have to perform some basic configuration: provide a mandatory name (we suggest that it should be “Application”), an optional description, and a zone. The zone indicates where the physical servers are located; as of the time of writing these lecture notes, the closest zone is “Paris, France”. Push the “Create” button and go ahead.

## Select the MySQL add-on



It's now time to create your database, which requires you to select the MySQL add-on.

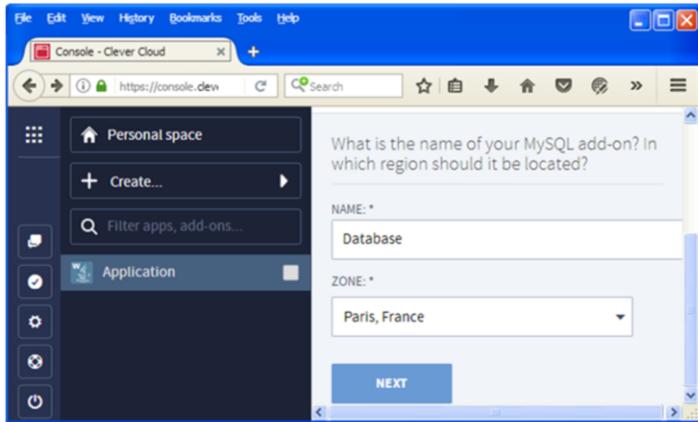
## Select the DEV plan

The screenshot shows a web browser window titled "Console - Clever Cloud" with the URL "https://console.clever-cloud.com". On the left, there's a sidebar with icons for Personal space, Create..., Filter apps, add-ons..., Application (which is selected), and other options. The main content area is a table titled "What kind of MySQL do you need?". The table has columns for PLAN NAME, MAX CONNECTION LIMIT, and MAX DB. It lists four plans: DEV (5 connections, 10 MB storage), S (10 connections, 256 MB storage), M (75 connections, 10 GB storage), and LM (140 connections, 50 GB storage). A red arrow points to the "S" plan row.

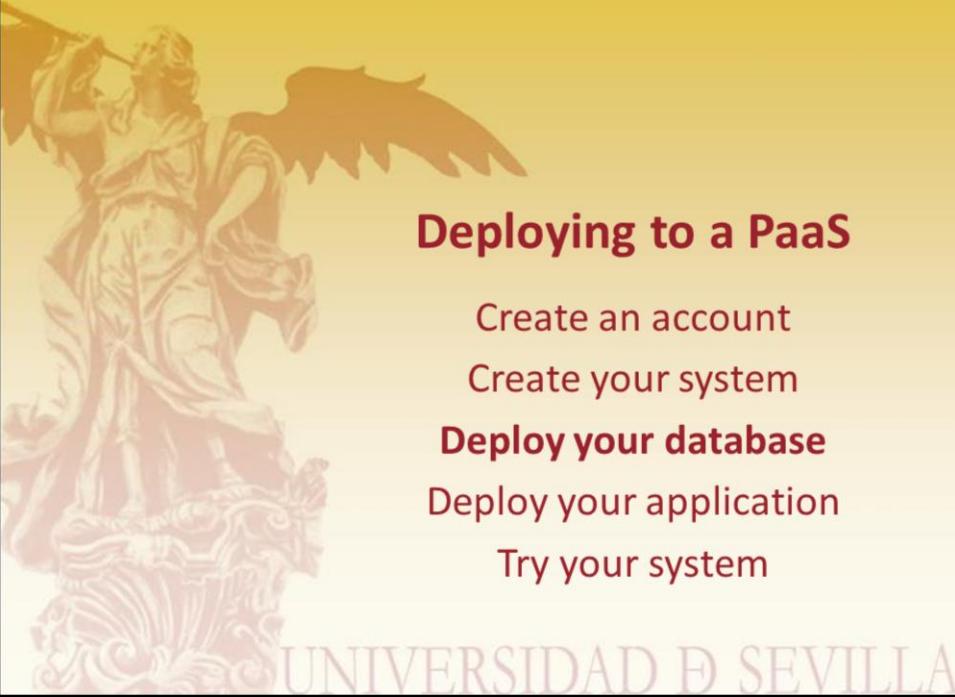
PLAN NAME	MAX CONNECTION LIMIT	MAX DB
DEV	5	10 MB
S	10	256 MB
M	75	10 GB
LM	140	50 GB

Now you have to select a plan for the MySQL add-on. The one called “DEV” is the right for us since it allows up to five concurrent connections to the database (believe us, that’s enough to handle many, many concurrent users), and 10 MiB storage (that’s not a lot... but enough for the purposes of this subject). The good piece of news is that this plan costs 0.00 €... so it’s actually our only choice.

## Perform basic configuration



And perform some basic configuration: provide a mandatory name (we suggest that it should be “Database”), and decide on the zone (“Paris, France” is the closest to us as of the time of writing these lecture notes).



## Deploying to a PaaS

Create an account

Create your system

**Deploy your database**

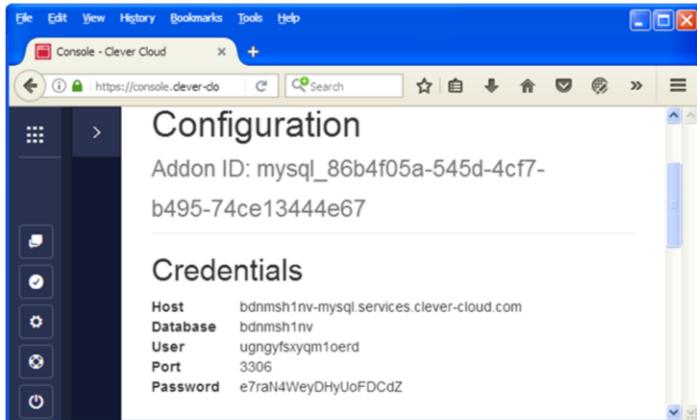
Deploy your application

Try your system

UNIVERSIDAD DE SEVILLA

Creating the system was pretty simple, right? Let's now cast a glance at how to deploy your database.

## Get the connection details



Select the “Database” tab and then the “Add-on Dashboard”. This is a simple information panel from which you can copy the information that you need to configure the data source of your project. Realise that this panel shows the host name, the port number, the database name, the user name, and its password. Unfortunately, you cannot change these data with the free plan; they are not intuitive, but functional, so let’s go ahead!

## Reconfigure your data source

```
<bean id="dataSource"
      class="com.mchange.v2.c3p0.ComboPooledDataSource"
      destroy-method="close">

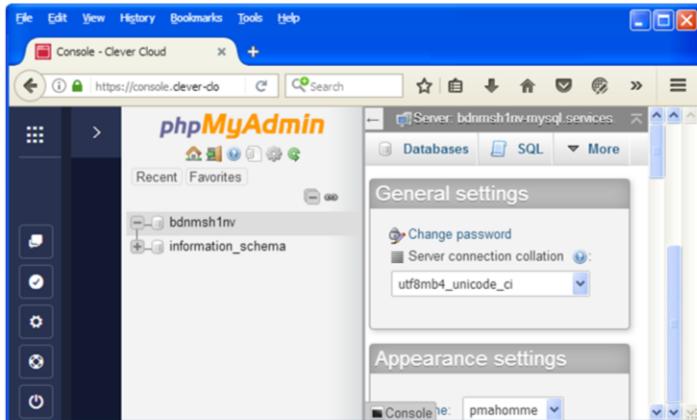
    <property name="driverClass" value="com.mysql.jdbc.Driver" />

    <property name="jdbcUrl" value="jdbc:mysql://bdn...oud.com/bdn...1nv" />
    <property name="user" value="ugngyfsxyqm1oerd" />
    <property name="password" value="e7raN4WeyDHyUoFDCdZ" />

    <property name="minPoolSize" value="1" />
    <property name="maxPoolSize" value="5" />
</bean>
```

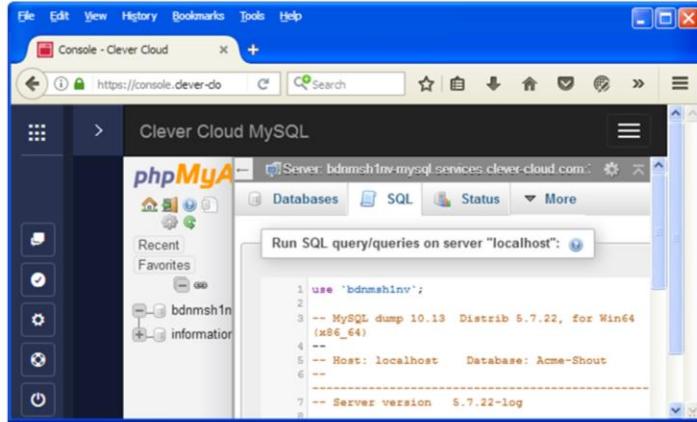
You have to re-configure the data source of your project in file “data.xml”. Note that you only have to change properties “jdbcUrl” (which is not fully shown on the slide), “user”, and “password”. That’s all.

## Enter the database console



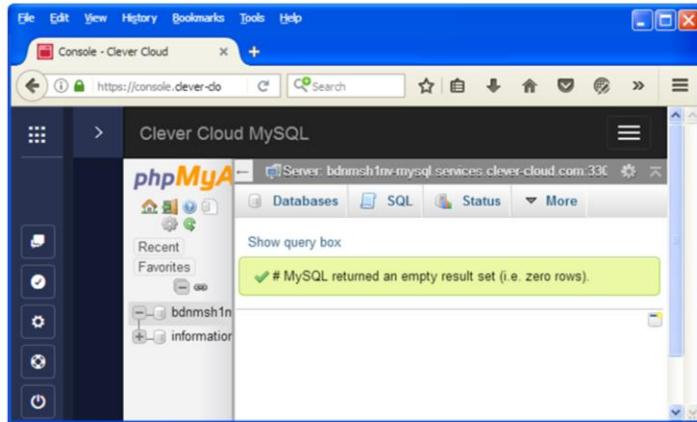
Now, enter the database console, which is available by clicking on “Database” > “Add-on Dashboard” > “Database manager”. Login with the previous credentials if prompted.

## Enter the database artefact



Play a bit around. Unfortunately, the free plan doesn't allow you to change a lot of configuration parameters, but it allows you to paste the database artefact and execute it. Note that you have to change the references to your project name to match the name of your database. For instance, take a look at the "use" sentence in the first line; originally, it read "use 'Acme-Certifications'", but we had to change it to match the name of the database that is provided by Clever Cloud.

## And execute it



There's a “Go” button to run the script; press it, wait for a few moments and everything should work like a charm. If not, please, review the error messages in order to diagnose what the problem is.



## Deploying to a PaaS

Create an account

Create your system

Deploy your database

**Deploy your application**

Try your system

UNIVERSIDAD DE SEVILLA

Let's go on. It's time to learn how to deploy your application, which is a bit more involved.

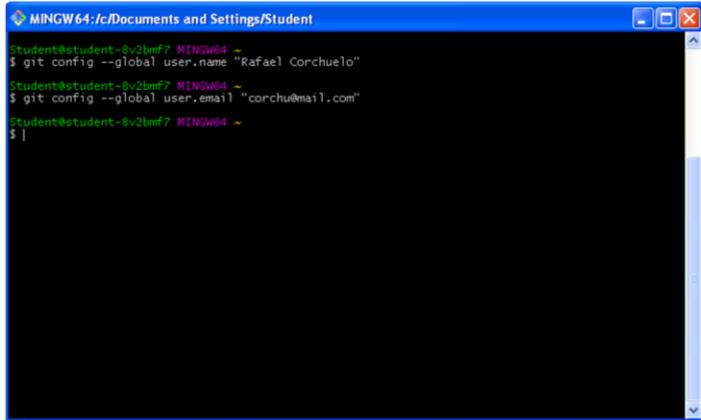
## Install git!

---



Clever Cloud relies on Git to upload your project. We can't assume that you're familiar with Git because there are a variety of artefact management systems available and you can decide on the one that you're going to use in this subject. So, we'll let you know about the minimum steps that you have to perform. The first step is, obviously, to install git, which should be like ringing a bell. If you're using the virtual machine that we provide in this subject, you can safely omit this step because it's pre-installed in that machine.

## Set your user name and email



A screenshot of a Windows-style terminal window titled "MINGW64:/c/Documents and Settings/Student". The window contains the following text:

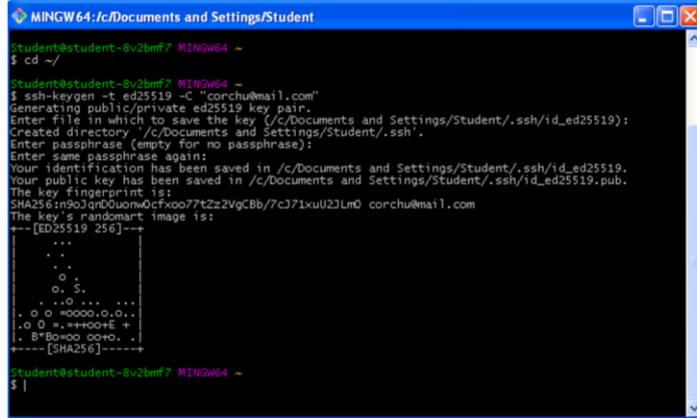
```
student@student-8v2bmf7 MINGW64 ~
$ git config --global user.name "Rafael Corchuelo"
student@student-8v2bmf7 MINGW64 ~
$ git config --global user.email "corchu@mail.com"
student@student-8v2bmf7 MINGW64 ~
$ |
```

If it's the first time that you use git, you have to provide your user name and email address. Open a Git Bash shell and execute the following commands:

```
> git config --global user.name "<your-name>"
> git config --global user.email "<your-email>"
```

Pretty easy so far!

## Generate your git SSH key (if required)



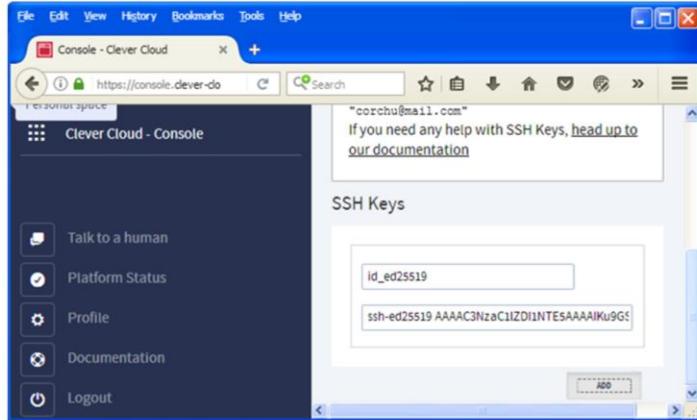
```
MINGW64:/c/Documents and Settings/Student
$ cd ~
$ ssh-keygen -t ed25519 -C "corchu@mail.com"
Generating public/private ed25519 key pair.
Enter file in which to save the key (/c/Documents and Settings/Student/.ssh/id_ed25519):
Created directory '/c/Documents and Settings/Student/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Documents and Settings/Student/.ssh/id_ed25519.
Your public key has been saved in /c/Documents and Settings/Student/.ssh/id_ed25519.pub.
The key fingerprint is:
SHA256:rn9oJqnDQuonwOcfxoo77zZ2VgCBb/7c71xuU2JLmQ corchu@mail.com
The key's randomart image is:
----[ED25519 256]----
... .
. .
o .
o. S.
..o ...
., o o =0000.0...
., o o ==H00HE +
| . B=+0000000000. |
+---[SHA256]-----+
$ |
```

Now you have to generate a new SSH key, which is not difficult at all. Execute the following instructions:

```
> cd ~/
> ssh-keygen -t ed25519 -C "<your-email>"
```

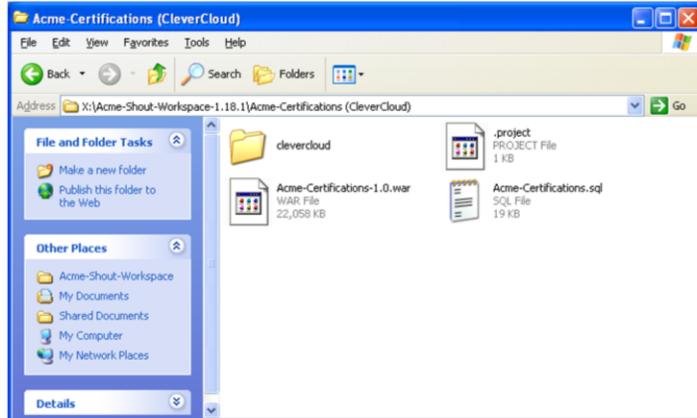
The first instruction changes the current folder to your home folder. The second one generates the key; during the process, it requests you to provide the name of the output file and a password to encode it (note that the password is referred to as passphrase since it's recommended that it should be a long piece of text). This will generate files “id\_ed25519” and “id\_ed25519.pub” in your “~/.ssh” folder.

## Add your SSH Git key to Clever Cloud



Now you have to inform Clever Cloud about your key. Don't worry, it's a public key; it's intended to be distributed openly. Click on the "Profile" button and then on the "SSH Keys" tab; then enter "id\_ed25519" as the name of the key and the contents of file "id\_ed25519.pub" as the value of the key.

# Create a folder for your deployment



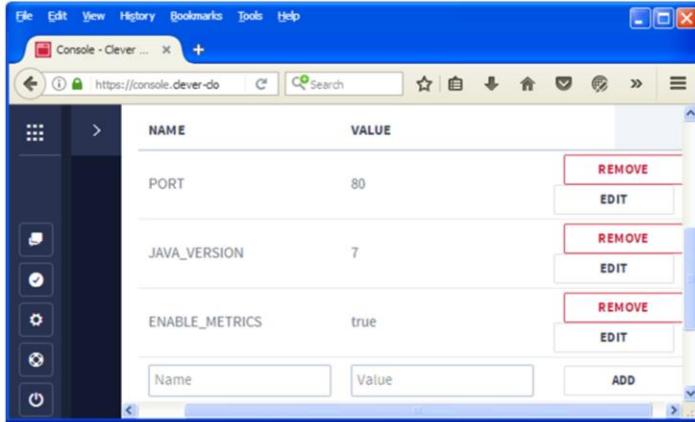
It's strongly recommended that you should create a new folder for your deployment, like in this slide. Inside the folder, put the database artefact, the war package, and create a folder called "clevercloud". (It's mandatory to use this name; you can't use "CleverCloud" or "Clever-Cloud" since they won't work; the ".project" file is created automatically by Eclipse.)

## Add file clevercloud/war.json

```
{  
  "deploy": {  
    "container": "TOMCAT7",  
    "war": [{  
      "file": "Acme-Certifications-1.0.war",  
      "context": "/"  
    }]  
  }  
}
```

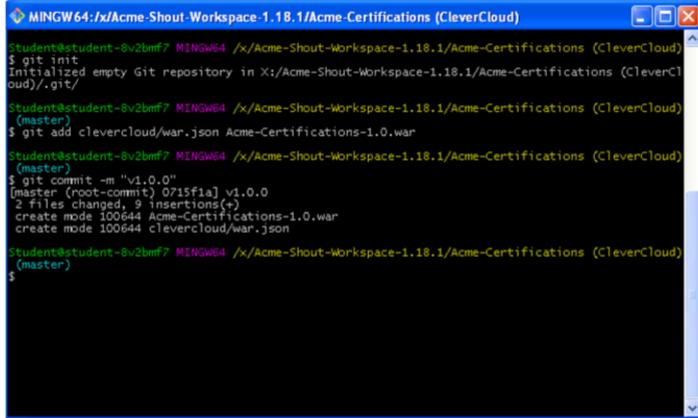
Now add a configuration file called “war.json” to folder “clevercloud”. This configuration file instructs Clever Cloud on how to run your system, namely: key “container” provides information about the version of Tomcat to be used (Note that it’s written “TOMCAT7” and not “Tomcat7” or “Tomcat-7.0”), key “file” provides the name of the file that contains your war package, and key “context” provides the context path to which it must be deployed.

## Set some variables



Now, we have to set some environment variables. Click on “Application” > “Environment Variables” to display a page that should look more or less like in this slide. Change the port to “80”, then set the Java version to “7”, add variable “ENABLE\_METRICS” and set it to “true” (this will help us monitor the performance of our system at runtime).

## Commit your changes locally



A screenshot of a Windows Command Prompt window titled "MINGW64:/x/Acme-Shout-Workspace-1.18.1/Acme-Certifications (CleverCloud)". The window shows the following terminal session:

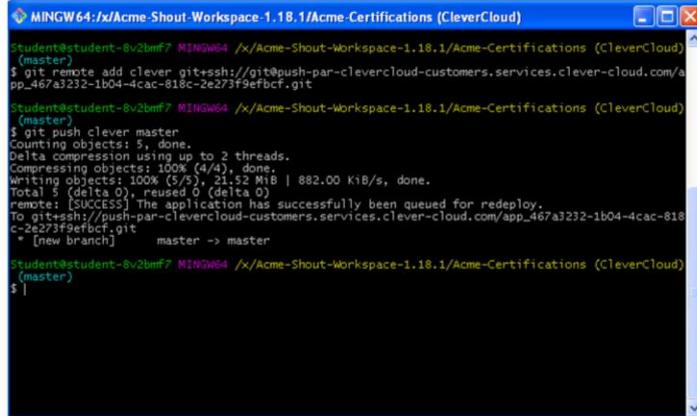
```
Student@student-8v2bmf7 MINGW64 /x/Acme-Shout-Workspace-1.18.1/Acme-Certifications (CleverCloud)
$ git init
Initialized empty Git repository in /x/Acme-Shout-Workspace-1.18.1/Acme-Certifications (CleverCloud)/.git/
Student@student-8v2bmf7 MINGW64 /x/Acme-Shout-Workspace-1.18.1/Acme-Certifications (CleverCloud)
(master)
$ git add clevercloud/war.json Acme-Certifications-1.0.war
Student@student-8v2bmf7 MINGW64 /x/Acme-Shout-Workspace-1.18.1/Acme-Certifications (CleverCloud)
(master)
$ git commit -m "v1.0.0"
[master (root-commit) 0715f1a] v1.0.0
 2 files changed, 9 insertions(+)
 create mode 100644 Acme-Certifications-1.0.war
 create mode 100644 clevercloud/war.json
Student@student-8v2bmf7 MINGW64 /x/Acme-Shout-Workspace-1.18.1/Acme-Certifications (CleverCloud)
(master)
$
```

Now we have to commit our changes to a Git repository. In this guideline, we assume that you've created a new copy of your project for deployment purposes only. So, we're going to work on the master branch of a new repository. Open a new Git Bash shell and issue the following commands:

```
> git init
> git add clevercloud/war.json Acme-Certifications-1.0.war
> git commit -m "v1.0.0"
```

The first command initialises the repository; the second command adds “war.json” configuration file and the “Acme-Certification-1.0.war” war package to the repository; and the third command commits the changes to the repository.

# Push your system to Clever Cloud



The screenshot shows a terminal window titled "MINGW64: /x/Acme-Shout-Workspace-1.18.1/Acme-Certifications (CleverCloud) (master)". The user has run the command "git remote add clever git+ssh://git@push-par-clevercloud-customers.services.clever-cloud.com/app\_467a3232-1b04-4cac-818c-2e273f9efbcf.git". Then, they run "git push clever master", which compresses objects, writes them to disk, and pushes them to the remote repository. The output shows the progress: "Counting objects: 5, done.", "Delta compression using up to 2 threads.", "Compressing objects: 100% (4/4), done.", "Writing objects: 100% (5/5), 21.52 MiB | 882.00 KiB/s, done.", "Total 5 (delta 0), reused 0 (delta 0)", "remote: [SUCCESS] The application has successfully been queued for redeploy.", "To git+ssh://push-par-clevercloud-customers.services.clever-cloud.com/app\_467a3232-1b04-4cac-818c-2e273f9efbcf.git", "\* [new branch] master -> master". Finally, the user types a closing brace "}" at the prompt.

Now we have to push our local repository to Clever Cloud. To do so, we have to execute the following commands:

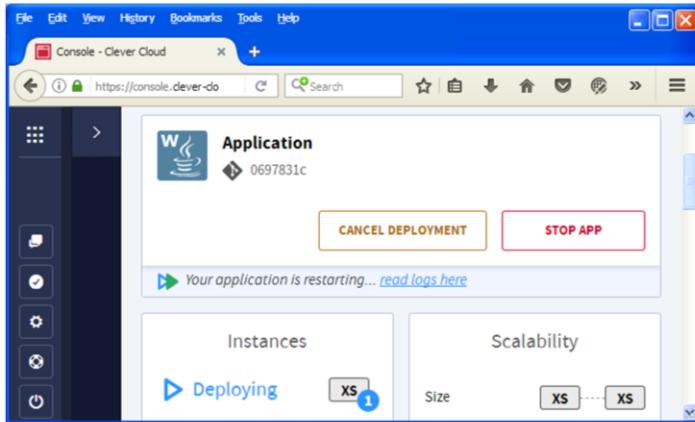
```
> git remote add clever git+ssh://git@push-par-clevercloud-
customers.services.clever-cloud.com/app_5140da12-289f-45c0-a19d-
5f7ed73e86f1.git
> git push clever master
```

The first command registers the deployment URL of your project at Clever Cloud; the second command pushes your local repository to Clever Cloud.

**NOTE:** you can find the deployment URL by clicking on “Application” and then on “Information”.

**NOTE:** if you make a mistake and have to start the process over, then use the following command: “git push –f clever master”. The “–f” parameter forces git to push the project in your local repository even in cases in which it’s not been properly synchronised with the remote repository.

## Check deployment

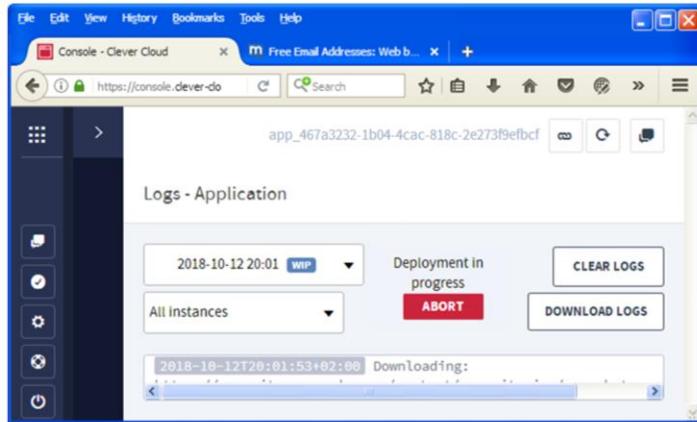


UNIVERSIDAD DE SEVILLA

126

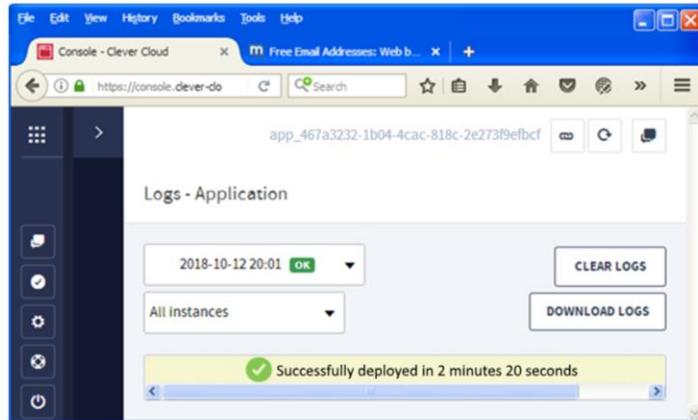
Check your deployment at Clever Cloud. Wait for a few seconds, and the status of your application will change to reflect that it's been deployed.

## Check the log



Check the log to see what's happening.

## Check it was deployed



After a while, the log should show a green tick that indicates that the deployment succeeded. If not, please, go through it and try to diagnose what the problem was.



## Deploying to a PaaS

Create an account

Create your system

Deploy your database

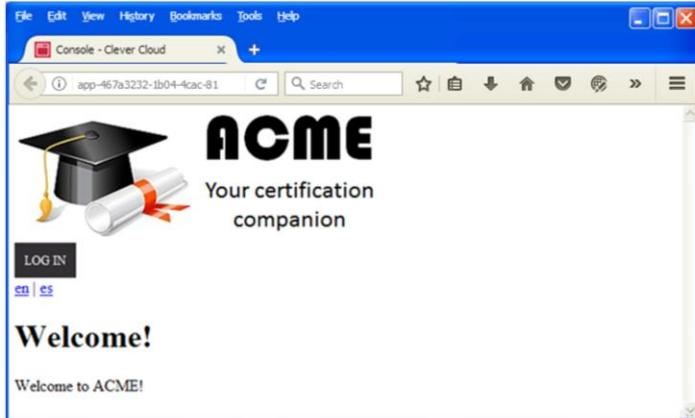
Deploy your application

**Try your system**

UNIVERSIDAD DE SEVILLA

Fine! It's about time to try your system!

## Play with your system

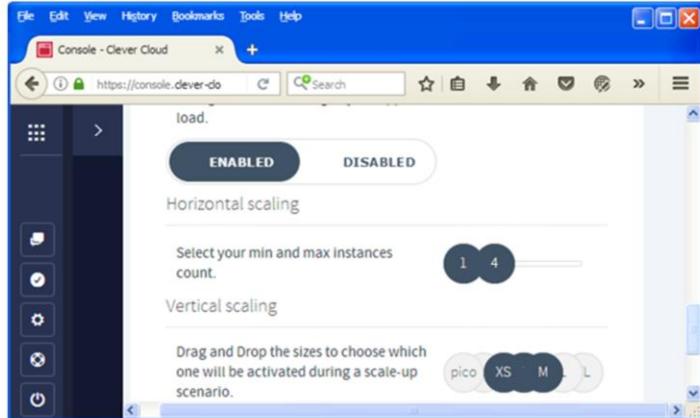


UNIVERSIDAD DE SEVILLA

130

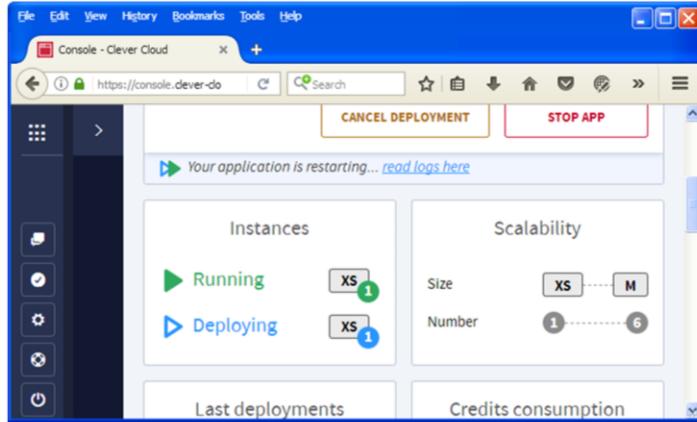
Make your browser for your system. The URL of your system is of the following form: "http://<app-id>.cleverapps.io", where "<app-id>" refers to your application identifier, which you can find by clicking on "Application" and then clicking on "Information".

## Configure scalability



By default, your application is configured to run on a single XS computer (1 MiB of RAM memory and 1 CPU). You can easily configure the scalability of your system by clicking on “Application” > “Scalability”. Enable your application to scale, and then select the number of machines on which it’s going to run (from 1 up to 20) and the type of machine (from Pico to XL machines). It’s Clever Cloud that decides when your application has to scale vertically or horizontally. For the purposes of this subject, we recommend that you should scale from 1 to 4 machines, from XS to M instances. Later, when we study performance tests, you’ll realise that it’s really difficult to exceed the capabilities of a single XS machine.

## Re-deploy your system

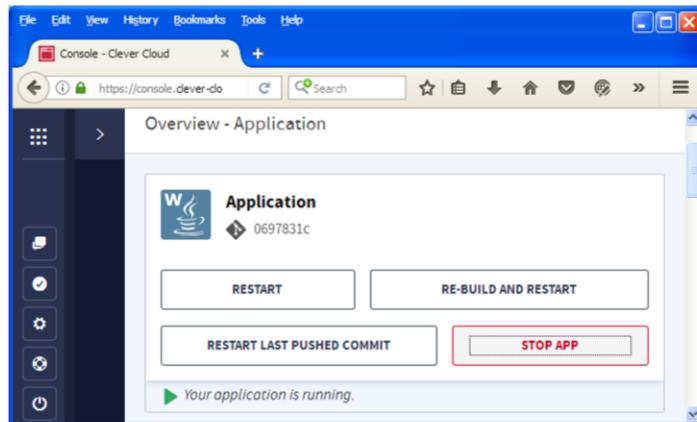


UNIVERSIDAD DE SEVILLA

132

If you change the scalability configuration, you have to redeploy your application. It'll commonly take a bit, but not a lot of time. Try to increase the workload so that Clever Cloud scales your application horizontally or vertically... but it won't be easy to achieve.

## Stop your app when done!



And don't forget to switch the application off when you finish using it. It's important that it doesn't consume any credit from your account.

## This is a good question

---



This is a very good question: what about scaling the database?

## And this is the answer

---

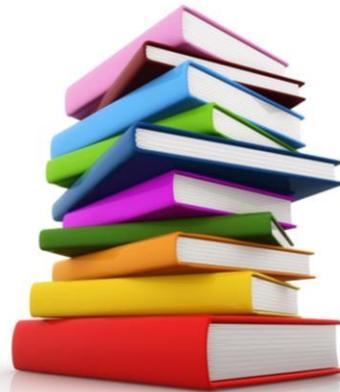


We're sorry, you can't with a free plan... but the server's very powerful, you won't have any problems with it.

And this is the answer: you can't scale the database server with a free plan; don't worry, the default server's very powerful. Please, recall that it's not our goal in this subject to learn to create systems that can be deployed to a multi-computer architecture. So, this is enough.

# Bibliography

---



UNIVERSIDAD DE SEVILLA

136

Our recommendation is that you should take these lecture notes as your primary source of information. If you need more information, you may take a look at the following books, which have chapters that are specifically targeted towards deploying database and application artefacts:

MySQL, 5th edition

P. Dubois

Addison-Wesley, 2013

Apache Tomcat 7 essentials

Tanuj Khare

Packt Publishing, 2012

This bibliography is available in electronic format for our students at the USE's virtual library. If you don't know how to have access to the USE's virtual library, please, ask our librarians for help.

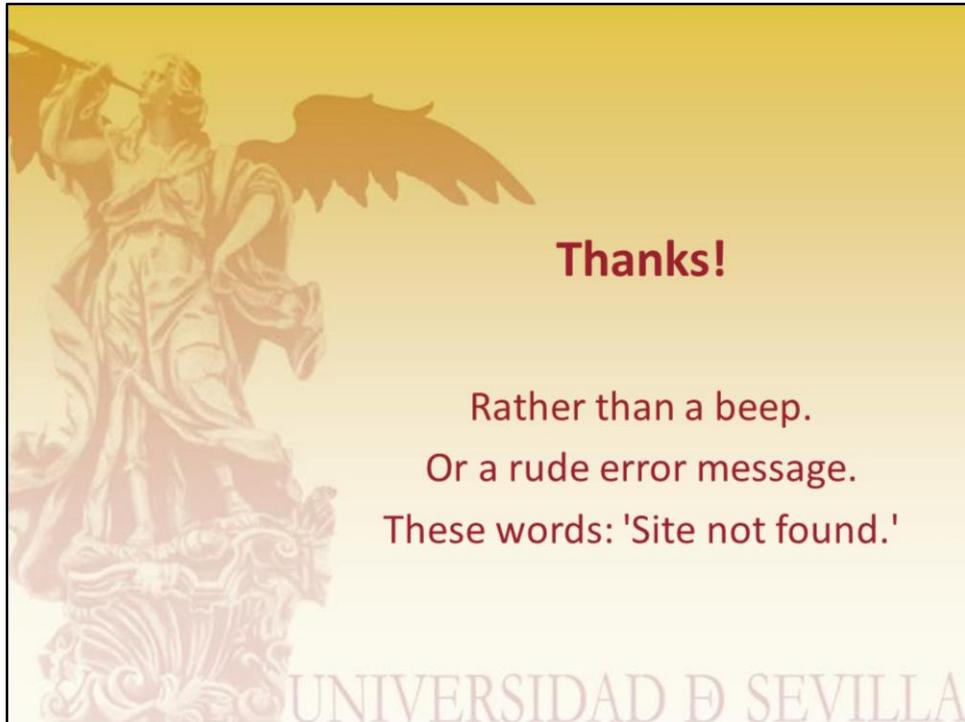
Should you need further information regarding Clever Cloud, please, browse their on-line documentation, which is available at <https://www.clever-cloud.com/doc/>.

Time for questions, please

---



Time for questions, please. Note that we won't update the slides with the questions that are posed in the lectures. Please, attend them and take notes.



Thanks for attending this lecture! See you next day!