



IBM's Platform as a Service: Bluemix

Blueprint

Michael P. Wagner

Under the name Bluemix, IBM offers a platform based on the open-source Cloud Foundry and designed for the further development of web, cloud and mobile applications. It is said to allow preconfigured services to be combined to form modular application environments with just a few clicks of the mouse.



- With Bluemix, IBM has created an environment in which developers can get their software ready for the cloud.
- The Bluemix includes the freely available browser-assisted development platform JazzHub.
- Programmers can freely choose which tools to use.
- Other development environments such as Eclipse can be integrated via plug-ins.

The starting point for Bluemix are “born-in-the-cloud” applications, and developers can use it to expand them further. The Platform as a Service (PaaS) provides the necessary infrastructure, various runtime environments and a number of supporting services. The many examples it includes provide easy orientation for those wanting to press on with the expansion of applications in the environment with the help of any development tool they wish to use. The system mechanizes the resulting code and transfers it to the Bluemix cloud. There the developer can automatically transform it into an executable application using the preconfigured services.

IBM advertises with the fact that Bluemix allows developers to create cloud applications in an extremely short time. In a practical test, it really did only take a few minutes to prepare a simple web application (see box text “Instant apps”) for the cloud. The complete cycle, from changing the application code and assembling the execution components to installing the finished application, was almost equally fast, and it only took a few minutes longer to create a first self-written application. Further development is just as simple as the conventional creation of web applications. The greatest difficulty is finding the suitable combination of services and options among the wealth of examples provided, and the amount of offers can be quite confusing at times.

The use of the Bluemix cloud does not require a special tool for developing an application and can be set up using just a few command line commands. All of the work can be carried out completely in the web, for example using the purely browser-based “JazzHub” platform from IBM, which is easy to combine with the Bluemix cloud environment. Alternatively, developers can use integrated frameworks such as Eclipse by installing a suitable plug-in.

Do-it-yourself cloud applications

The direct route to a Bluemix application always follows the same pattern. At the beginning you have to select an execution environment in which the program logic keeps the various strands of the components together. Services expand the range of functions and provide aids for administration.

All components are available and executable within the cloud in a coordinated

configuration at a click of the mouse. What is missing is the application code, which you can prepare and upload using any desired editor. Tools from the Cloud Foundry carry out build commands, connect the application components and start the resulting application.

The status of the Bluemix applications installed is displayed in a straightforward way on the so-called dashboard using tiles or lists (see Fig. 2). At the very top, the dashboard displays the current memory consumption as well as status and number of installed services. If an application or a service cannot be executed, the status is red. A click on the relevant icon in each case provides further status information, for example in the relevant logs.

Beginners can use up to 20 instances and 2 GByte of main memory free of charge in the first month. After that, the company charges for the use of resources in GByte-hours (GBh), i.e. the quantity of memory used multiplied with the number of operating hours. After the one-off test phase, 1125 GBh are free per month, and each further unit costs slightly more than € 0.05. If you use 1 GByte for one month, you have to pay approx. € 38. The services are charged for according to free contingents and payment models which you can select during setup.

Everything under control

The user interface also manages the structure of the Bluemix cloud. Organizations, spaces and accounts serve as structural elements here. When you create a test account under bluemix.net, the system automatically generates an organization and a space for saving the applications and services.

Users can create further spaces, for example when they want to separate test environments from production environments. The distribution of the software over the environments takes place during uploading via the specification of the target space.

From the top: the “dashboard” of the Bluemix user interface provides an overview of the status of your own applications and the services they use (Fig. 2).

Instant apps

Surprisingly, the fastest route to an autonomous Bluemix application managed independently of the IBM universe goes via the Command Line Tool (CLI) of the Cloud Foundry organization. It is to be found under <https://github.com/cloudfoundry/cli>. All you have to do there is select the platform of the development environment under “Downloads”. It is sufficient to transfer the source code to the Bluemix cloud.

If you set up Bluemix as a deployment target:

```
cf api https://api.ng.bluemix.net
```

and log in via *cf login*, all that is missing is the appropriate source code.

The simplest example is to be found in git. When you install its Command Line Client (<http://git-scm.com/download>), you can – after defining a local directory for the source code – download the simple example via:

```
git clone https://hub.jazz.net/git/7
communitysample/env-javaweb
```

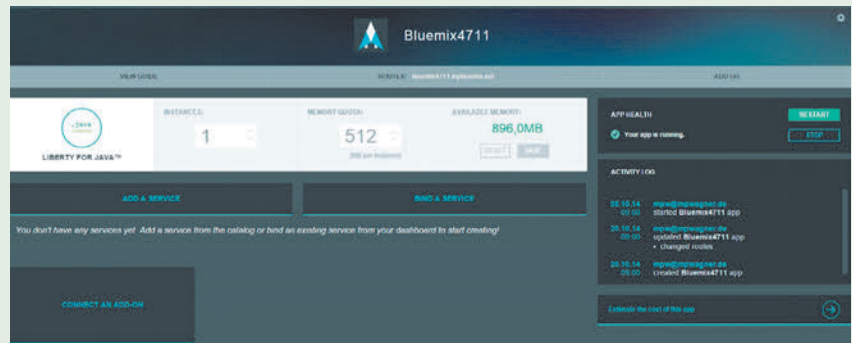
This application uses Java Server Pages and outputs all variables of its runtime environment as an HTML table. The code can be deployed using a command:

```
cf push Bluemix4711 -m 64m
```

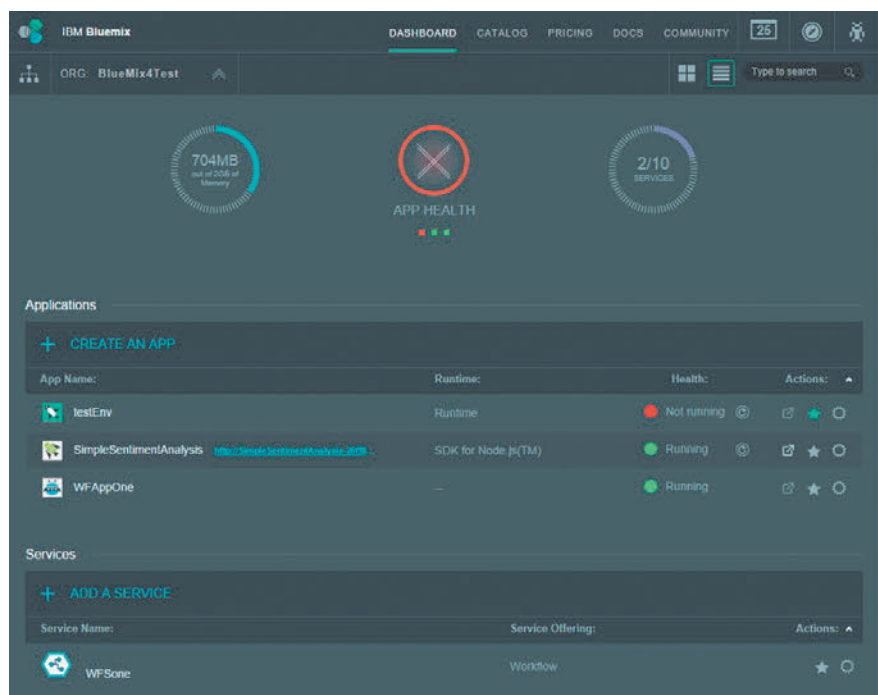
The process requires some time for compiling, installing and starting. Then you can view the output of the first cloud application of your own in a browser under <http://Bluemix4711.mybluemix.net> (see Fig. 1).

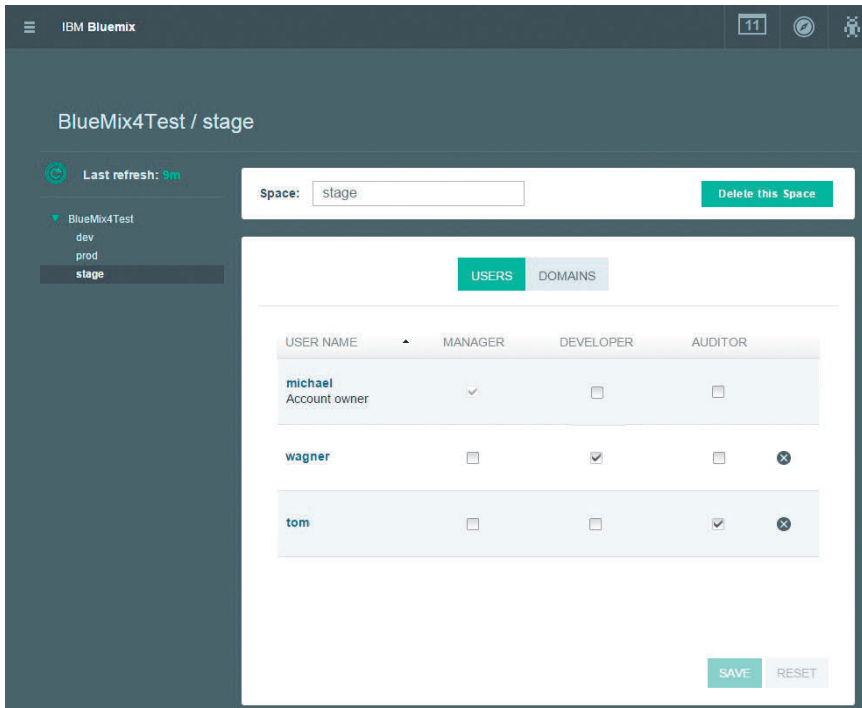
If you change the source code of the application, for example by inserting HTML formatting into *index.jsp*, a *cf push* is sufficient. Bluemix automatically regenerates the cloud application and performs a restart.

For tracing purposes, it is sufficient to change the application name of Bluemix4711 to one of your own in the scripts mentioned. If the desired name is taken, an error message is output.



The open-source framework node RED provides a graphical development environment for script-based applications within Bluemix (Abb. 1).





Space on offer: the Bluemix cloud is divided up into organizations and spaces in which the users can assume different roles (Fig. 3).

Several accounts with the administration roles Manager, Billing Manager and Auditor are assigned to an organization. The user can link them with different domains if he wishes, so he is not forced to use the automatically generated domain <user defined>.bluemix.net. You have to sort out the certifying of the necessary

SSL certificates yourself. An organization determines the quotas for the number of services and the quantity of GBh used. The user can change the name of the organization if necessary.

The “Create a Space” button allows you to create any desired number of spaces in the Manager role. These serve

to delimit the application environments from each other, for example in testing (staging) and in production. It seems natural to use several of them when the configuration of the services changes between releases. In addition, you can distribute the software separately according to target environments. The usual cycle of development, testing, staging and production is covered by automated deployment in the Bluemix cloud. You have to create a suitable automatic execution of the tests yourself; identically configured spaces are necessary for tests generated manually.

One of the strengths of Bluemix is doubtlessly the wide range of software components it provides for application development. To make it easier to get started, so-called “boilerplates” provide preconfigured combinations of components with which programmers can start work immediately. The boilerplates consist of an execution environment and one or several services. If you wish, you can combine the components to form individual environments without using a boilerplate. This is easier than you might think.

Command lines and text editor suffice

An execution environment is instanced and started by for example clicking the Java runtime environment and defining an unambiguous name for the applica-

Comparison offers to IBM’s Bluemix

Besides the cloud environments listed below, the most important offers comparable with IBM’s Bluemix include those of a number of smaller companies, some of them extremely innovative. The majority of other cloud platforms with a broader approach such as SaaS and IaaS are not directly comparable as they do not achieve the same degree of automation for the development of cloud applications.

AWS’ Elastic Beanstalk supports Java, Node.js, PHP, Python, Ruby and .NET environments. It can be used via git or the Java web application archive service. It supports Docker Container for the rapid and flexible configuration of the cloud applications.

Google’s App Engine (GAE) knows Java, Go, PHP, Python and JVM languages such as Groovy, JRuby, Scala and Clojure, but it is basically dependent on the programming

language. Application development takes place within the limits set by the GAE-API. Administration of the cloud applications is automated. A special memory service is available.

Salesforce’s Heroku cloud environment came into being in Ruby, but it currently also supports Java, Node.js, Scala, Clojure, Python and PHP. It cooperates with memory services such as Cloudant, Couchbase, MongoDB, Redis and PostgreSQL. Heroku includes a git service of its own.

Language mix

Microsoft’s Azure Web Sites (MWAS) consists of mixed applications in the languages Java, Node.js, PHP and Python. .NET environments can be used too. Developers can create cloud applications on the basis of a website catalog with over fifty offers. The applications can be expanded step by step, and the code can be

maintained using a number of different web services.

Red Hat’s OpenShift Cloud supports the languages Haskell, Java, JavaScript, Perl, PHP, Python and Ruby, and also .NET. The frameworks it knows include Ruby on Rails and Sinatra as well as the databases Microsoft SQL Server, MongoDB, MySQL and PostgreSQL. OpenShift can use native applications for the underlying Red Hat Enterprise Linux, so developing and compiling don’t have to take place in the cloud.

VMware’s open-source framework Cloud Foundry on which Bluemix is based is being further developed by Pivotal, with the funding coming from VMware and EMC. It provides a special Cloud environment which supports Java, Node.js, PHP, Python, Ruby and Scala, the database services MongoDB, MySQL, Postgres, Redis and the message broker RabbitMQ.

tion. The organization and the space are pre-assigned automatically, but the developer can change them. A text editor and a command line tool belonging to Cloud Foundry are sufficient for adding the source code. When you create a Bluemix environment, you can manage the sources in the freely available development environment JazzHub and operate projects in a completely web-based way.

Jazz Hub provides convenient environment

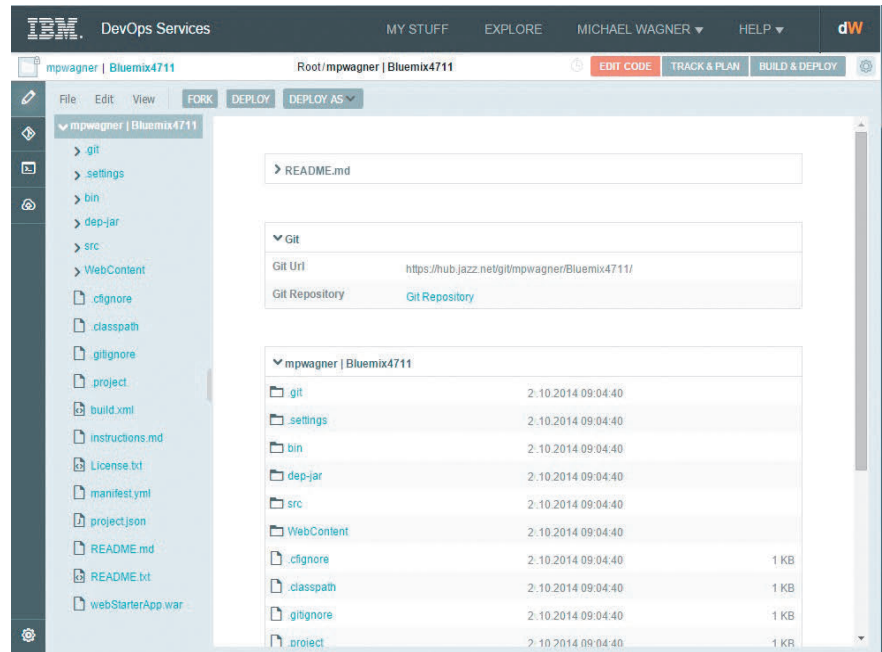
If you activate the “Add GIT” button in a freshly created environment, the system copies the code of the Bluemix application to a git repository and links it with your own account in JazzHub. With “EDIT CODE”, JazzHub opens and the code can be changed. Using the “BUILD & DEPLOY” button in JazzHub, you can copy changes to Bluemix and trigger compilation, deployment and restart of the application (see Fig. 3). As the framework supports the coordinating of several teams of developers, a distributed environment for developing applications for the cloud on the web level alone is quickly created using a combination of JazzHub and Bluemix.

Although the documentation says that Bluemix supports all browsers, it functions most smoothly under Google Chrome. It expressly only works with the Internet Explorer as of version 8, and in the popular Firefox, many an interaction terminates for unknown reasons. It is apparent that not all interface elements are working without errors yet. Some of the services cannot be installed with all browsers because the set-up dialogs do not work.

Back to earth as friends

There is also trouble with single sign-on between Bluemix and the DevOps services in JazzHub. Occasionally, interactions that change between the services cannot be completed because the authorized user is not recognized. The integrated match of the IBM and the Jazz ID doesn't help, and neither does creating identical user IDs and passwords. It is annoying when a rejection of this kind occurs after a long period of interaction.

IBM's offer of using the Bluemix cloud free of charge for one month sounds generous at first. However, with 2GByte of memory and a maximum of 10 services, it soon reaches its limits. For



Integrated: Bluemix smoothly integrates IBM's framework JazzHub and allows the purely browser-based development of cloud applications (Fig. 4).

this review, the authors had to discard applications they had just created to make space for testing other services. You can't really test more complex setups – the memory is exhausted sooner than the 10 services are.

Also, it can get really costly once the test month is over. The free 1125 GBh mentioned above correspond with a consumption of around 1.5 GByte per month. As each service is charged separately, you can soon lose track (see table). Not for nothing does IBM provide a special service helping you to predict the prices of the applications in the form of the “Estimate the cost of this app” button. You should precisely calculate what you put together so easily. Unfortunately, there is no tool for registering the current use of resources. With the exception of the instruments of application development,

you don't have much chance of optimizing the use of resources and thus reducing costs.

Conclusion

Bluemix is a convincing solution due to the far-reaching integration of the cloud environment it offers, the easy-to-operate user interface and the smooth interaction of development tools. In development, the high automation capability during the creating and installing of applications and the wide range of services offered shorten the route to an executable cloud application.

IBM is rapidly improving Bluemix at present. During the tests, other services were added and some were combined, modified or renamed or left the beta stage. The user interface is also affected. It will remain interesting to see in which direction the Bluemix cloud will develop against the backdrop of growing competition. (rh)

assessment

- ⊕ Purely browser-based development
- ⊕ Rapid configuration
- ⊕ Complete integration
- ⊖ Low memory volume in the test phase
- ⊖ High costs for complex applications

Dr. Michael P. Wagner

works in Munich as a company consultant and communication scientist.