PaaS for web applications with OpenShift Origin

To cite this article: A Lossent et al 2017 J. Phys.: Conf. Ser. 898 082037

View the article online for updates and enhancements.

Related content
- Development of a web application for estimate the total charge in an electric discharge
  F H Vera-Rivera, V D Dugar-Zhabon, H J Dulce-Moreno et al.
- Web life: SDSS SkyServer
- Web life: lightsources.org
PaaS for web applications with OpenShift Origin

A Lossent, A Rodriguez Peon and A Wagner
IT Department, CERN, 1211 Geneva 23, Switzerland
E-mail: andreas.wagner@cern.ch

Abstract. The CERN Web Frameworks team has deployed OpenShift Origin to facilitate deployment of web applications and to improving efficiency in terms of computing resource usage. OpenShift leverages Docker containers and Kubernetes orchestration to provide a Platform-as-a-service solution oriented for web applications. We will review use cases and how OpenShift was integrated with other services such as source control, web site management and authentication services.

1. Introduction
The CERN Web Frameworks team is responsible for providing developer-oriented applications (including source control, documentation, issue tracking and continuous integration) as well as central hosting services for CERN web sites and web applications.

When central hosting of Web applications is not appropriate, web application authors at CERN also have the option to set up and maintain their own dedicated web server, at a cost for the organization, namely increased workload for the application authors and increased attack surface from the computer security point of view. Consequently a key challenge for the Web Frameworks team is to maximize the functionality and attractiveness of the central hosting features offered, while keeping the computing resources required to operate the web hosting infrastructure as low as possible.

The Web Frameworks team adopted a strategy consisting in using container technology to run the services it provides, using OpenShift Origin [1] for all container orchestration. In order to address the aforementioned challenge, part of this new infrastructure based on OpenShift is exposed to CERN web application authors following the Platform-as-a-Service (PaaS) paradigm, as the new “PaaS for Web Applications” service.

2. Motivation
The aim of the PaaS for Web Applications service is to modernize CERN’s web central hosting offerings and make it easier to deploy web applications.

The service provides support for modern development frameworks and technologies and is significantly more flexibility compared to previously existing web hosting services at CERN. It thus increases the range of applications that can be hosted on CERN’s web central hosting infrastructure and reduces the need for web application developers to set up their own dedicated web servers.

In addition, the service provides a high level of automation. This makes it easier to get started and create application prototypes and also considerably reduces the effort needed to build, deploy and re-deploy web applications. Web applications may be built from source code which is typically the case for in-house developments, but Web applications may also be deployed from existing binaries or...
container images, making it possible to deploy arbitrary third-party applications to be run on CERN’s central web hosting infrastructure.

In order to provide a complete development pipeline, the PaaS for Web Applications service integrates with CERN’s source code hosting based on GitLab [2] and Continuous Integration/Deployment (CI/CD) processes via GitLab-CI and Jenkins [3].

The PaaS for Web applications service management tools offered for Web application authors include both a web user interface and a command-line interface which facilitates the maintenance and monitoring of the deployed web applications. Furthermore the PaaS for Web Applications service facilitates provisioning multiple instances of the same web application using so called web application templates. The typical use case for such templates is a web application that does not support multi-tenancy: if multiple independent teams wish to use such an application, each team will need their own instance. Application templates enable the manager of a service based on such an application to describe how to automatically provision an instance in CERN’s PaaS for Web applications service that is based on OpenShift. CERN Web application authors may then use the OpenShift web interface to provision instances of the application in a few clicks, in a completely self-service and automated way. Web application templates thus save the service manager from provisioning and maintaining dedicated servers for each application instance.

3. The OpenShift Platform

OpenShift is a container orchestration platform optimized for web applications. It enables building, testing, and deploying web applications without provisioning and maintaining dedicated servers for each application.

OpenShift runs applications in Docker [4] containers. The official description of OpenShift summarises that “Docker containers wrap up a piece of software in a complete filesystem that contains everything it needs to run: code, runtime, system tools, and system libraries - anything you can install on a server. This guarantees that it will always run the same, regardless of the environment it is running in.” [5] OpenShift is built on top of Kubernetes [6]. Kubernetes provides the core container orchestration functionality, including features such as component failure resiliency, monitoring and automatic rescheduling of containers and horizontal scaling. Additions of OpenShift on top of Kubernetes include multi-tenancy (the ability to run multiple independent projects managed by different groups of users), augmented deployment functionality, web routing and load balancing, an extensive web user interface, and application templates.

4. System Architecture of OpenShift

The OpenShift system architecture is illustrated in Figure 1. Application owners manage applications hosted in OpenShift via administration servers. All application management interactions can be done via a CLI (command line interface) or a web interface. The desired configuration and state for each application are stored in the master nodes.

The web applications run in containers in worker nodes. On each worker node runs a Kubernetes “node agent” component called “kubelet” that is fetching the desired container state from the master nodes, and applying the necessary changes to the container to achieve this state.

An application may consist of multiple containers, connected together by a software-defined virtual network (SDN). Communication between containers of the same application is independent of whether containers run in the same worker node or not: this is abstracted by the SDN.

Containers cannot be directly accessed from outside the OpenShift SDN. Applications are instead exposed via router nodes running HAProxy [7]. Application users actually connect to these router nodes, which use virtual hosting to route requests to the appropriate containers via the SDN, whereby HTTP Host header fields are used for clear-text HTTP connections, and Server Name Indication [8] for secure HTTPS connections.
5. Use cases for CERN’s PaaS for Web Applications service
CERN’s PaaS for Web Applications service addresses three main use cases. The first use case are application templates, with Jenkins as the primary example. The second is simplifying the deployment of arbitrary third-party web applications compared to the deployment on dedicated servers. And finally we will see how the PaaS for Web Applications service is used to build, deploy and host custom web applications from source code.

5.1. Web Application templates
The PaaS for Web Applications service enables service managers to define Web application templates. Using a declarative language [9], templates describe how to deploy a given web application as containers on OpenShift. Templates may accept parameters to customize the resulting application instance. Application templates are typically associated with a container image [10] containing the application binaries and all their dependencies.

A primary example is a popular Continuous Integration application for developer teams: Jenkins. Each developer team typically needs a separate instance of Jenkins. Instead of manually provisioning a new dedicated server for each new Jenkins instance, a time-consuming process, application template allow to completely automate instance provisioning and make it entirely self-service for tentative Jenkins users. The time it takes for Jenkins users to obtain their instance is reduced from possibly days to minutes.

Additional benefits of application templates compared to dedicated servers include the more efficient use of computing resource: by leveraging the isolation provided by containers, many instances can be hosted on a single OpenShift worker node, compared to one instance per dedicated server. OpenShift also takes care of monitoring and restarting instances automatically and moves instances to other nodes in case of infrastructure failures. OpenShift provides primitives to reconfigure and re-deploy all Jenkins instances when configuration changes or new versions need to be deployed: this process is entirely under control of the service manager for Jenkins, who maintains the application template.
Another example of application template being actively used at CERN is Grafana which is an open source metric analytics and visualization suite [11]. Like Jenkins, each Web application developer team typically needs their own Grafana instance and OpenShift makes it much simpler and faster to provision Grafana instances as compared to installing dedicated servers.

5.2. Deployment of arbitrary third-party applications
CERN IT services deploy a large number of applications, many of them not written at CERN but provided by private companies or open-source projects. OpenShift offers very simple ways to deploy such third-party applications, especially when they are readily available in the form of Docker container images.

OpenShift runs all applications as containers and supports to run containers using off-the-shelf, third-party Docker images. OpenShift enables configuring the application by providing ways to inject configuration files and environments variables [12]. Such Web applications often need persistent storage and/or databases to store the application state. Persistent storage is provided via integration with CERN network filer services. Databases can be obtained using CERN's “Database-on-Demand service” [13].

Ultimately, applications that are available as pre-built Docker images and whose requirements are compatible with OpenShift can often be deployed and configured within minutes. However, not all applications are compatible, as OpenShift's multi-tenancy implies a number of security constraints such as limits on the Linux UID that can be used by the container. Sometimes small changes in the image are sufficient to make it compatible with OpenShift, and a Docker image build process is provided by OpenShift to implement such modifications [14] [15].

Beyond deployment simplicity, a significant benefit of deploying third-party applications in OpenShift is the automatic rescheduling of containers in case of infrastructure issues (such as a worker node crash), providing a better availability potential than a dedicated standalone server.

An example of such a third-party application is Mattermost (“Workplace messaging for web, PCs and phones”) [16] that has been deployed as CERN PaaS Web Application using directly the container image provided by the Mattermost software authors.

5.3. Custom web application hosting
OpenShift provides a source-to-image build system to automatically create container images from source code [17]. A number of development frameworks are supported natively by the source-to-image process, including Ruby, Python, Django, Node.js, Perl, Java and PHP.

This enables the deployment of custom web applications developed in-house at CERN, with significantly more flexibility than traditional web hosting services could offer: thanks to the isolation provided by containers and because each container image contains the application and all dependencies, applications with conflicting dependencies (e.g. different library or runtime versions) may now run concurrently in the same OpenShift infrastructure. These version conflicts were a major cause for CERN's web application authors setting up dedicated web servers for web applications, rather than using the central web hosting services.

The same benefits, storage and database functionality outlined in 3.2 are available to web applications built from source code. The OpenShift deployment at CERN is fully integrated with CERN's source code hosting platform GitLab.

6. Timeline of the CERN PaaS for Web Applications service
The PaaS for Web Applications service was initiated with the goal of automating the provisioning of Jenkins instances. The scope was progressively extended to the hosting of general web applications.

- Early 2014: A new "Continuous Integration with Jenkins" service is offered to CERN users. It enables CERN users to request instances of the Jenkins Continuous Integration software, provisioned for them as virtual machines from the CERN Private Cloud. Despite significant investment in automating the processes, manual steps are still required. Jenkins lacking multi-
tenancy, dedicated Virtual Machines are assigned to each instance resulting in poor efficiency in terms of computing resource usage: memory and CPU are statically allocated to virtual machines, while they are only really used when Continuous Integration jobs actually run; and most instances have only moderate activity.

- End 2014: First evaluation of a Platform-as-a-Service solution to further automate deployment and management of these Jenkins instances, using OpenShift v2. While promising, the effort of deploying Jenkins in OpenShift v2 is found to be too important for the expected gains and the model of using virtual machines for Jenkins are retained.
- Mid 2015: Red Hat releases OpenShift v3, based on Docker and Kubernetes. The use of these streamlined technologies significantly reduces the effort required to deploy Jenkins in OpenShift (as compared to OpenShift v2) and opens the possibility to completely automate instance provisioning. At the same time, containers facilitate the co-hosting of many Jenkins instances on a small number of virtual machines, significantly improving the efficiency in terms of computing resource usage.
- Autumn 2015: OpenShift is evaluated for hosting other web applications than Jenkins, both for internally developed applications and for the deployment of 3rd party applications. The effort necessary to build, deploy and maintain small to medium-sized web applications as containers in OpenShift is found to be significantly reduced compared to deploying the same applications as Virtual Machines.
- Beginning of 2016: OpenShift is identified as a strategic platform for the future evolution of the CERN Web Services that is providing web hosting services for about 14000 web sites, applications and content management systems.
- Q1 2016: integration of OpenShift with other CERN computing services. This includes the integration with network storage services for application data persistence, integration with authentication services to provide single-sign-on and integration with the existing CERN Web Services portal as “PaaS for Web Applications” so that the lifecycle of web applications hosted in OpenShift is managed consistently with existing web sites.
- May 2016: Mattermost as first production application in OpenShift made available.
- June 2016: production use of OpenShift for Jenkins and migration of existing Jenkins instances from Virtual Machines to the new infrastructure. Jenkins is now provided as an application template in OpenShift. Thanks to containers enabling co-hosting of multiple instances in OpenShift, the number of worker Virtual Machines required to provide the "Continuous Integration with Jenkins" service drops from 28 “large” virtual machines to only 4 “xlarge” virtual machines (“xlarge” virtual machine providing twice the amount of CPU and RAM than “large”).
- Q3 2016: Production use of the “PaaS for Web Applications” service based on OpenShift for all three identified use cases (deployment of self-service application templates like Jenkins, deployment of internally developed applications, and deployment of third-party web applications like the Mattermost software) with more than 100 web applications.

7. Implementation and Deployment of CERN’s PaaS for Web Applications service
As of October 2016, CERN’s PaaS for Web Applications service is deployed in two OpenShift clusters (one for development and application testing, one for production applications) using a total of 27 virtual machines. About half are larger virtual machines serving as worker nodes, the other half consisting in smaller machines providing redundant infrastructure services such as masters and routers. The virtual machines are provisioned in the CERN Private Cloud based on OpenStack and configured with the CERN Configuration Management System (primarily Puppet).

Persistent storage for OpenShift applications is mainly provisioned from NFS filers. NFS filers are themselves virtual machines running in the CERN Private Cloud and actual storage is provided by a Ceph cluster via OpenStack Cinder volumes. Applications also have access to other shared file systems.
that are widely used at CERN, namely the “CernVM File System” CVMFS [18] and CERN’s “EOS Large Disk Storage” [19].

The CERN Web Services [20] portal is used by CERN users to create OpenShift applications in a self-service manner. While most application management tasks are performed from the built-in OpenShift CLI or web interface, the Web Services portal complements the OpenShift interface by adding management of ownership, quota, application lifecycle, registration with Single-Sign-On authentication and DNS name allocation in a way consistent with other types of web hosting offered to CERN users.

A quota system limits the amount of resources (CPU, RAM, persistent storage) available to each application.

The OpenShift clusters used by the PaaS for Web Applications service also host a number of applications for the Web Frameworks team’s own internal needs. Other services (or components of services) provided by the Web Frameworks section thus run inside OpenShift alongside user applications and share infrastructure components like master nodes and routers. This simplifies the operation of the Web Frameworks services by running all services as containers on a single infrastructure.

8. Outlook

Alongside the PaaS for Web Applications service, work is in progress to consolidate more services in the existing OpenShift clusters.

For instance, the CERN Web Frameworks team is providing web hosting services for about 14000 web sites and applications, currently on Web servers based on traditional virtual machines. The requirements for most of these web sites are modest and would not justify that each of them is converted into a Platform-as-a-Service application. A new generation of web hosting based on containers is thus being designed to run these web sites inside OpenShift, but keeping the current web site paradigm rather than the PaaS approach. This enables traditional web hosting to benefit from OpenShift features (such as container isolation, re-scheduling in case of incident and horizontal scaling) while not exposing the full complexity of OpenShift for the simple use cases.

Other work in progress includes moving complex services aimed at developers, such as GitLab and GitLab-CI to OpenShift. This is motivated by an expectation of important time savings thanks to the high level of automation and ease of deployment and maintenance of applications compared to dedicated virtual machines, as well as the opportunity to consolidate services on a common infrastructure.

9. Conclusions

The deployment of the new PaaS for Web Applications service based on OpenShift Origin allowed to address a number of important challenges for the CERN Web Frameworks team. First of all it provides a high level of flexibility for central hosting of Web applications in order to minimize the need for dedicated web servers. The PaaS for Web Applications service is also simplifying and fully automating the otherwise time-consuming process of provisioning multiple Web application instances. It allows to integrate web hosting services with development pipelines, hence reducing operational workload by consolidating heterogeneous services on a common infrastructure and it allows taking advantage of the automation capabilities of modern container orchestration solutions.

The use of standard, portable technologies like Docker containers is expected to facilitate the future operation and evolution of the services provided by CERN’s Web Frameworks team. For these reasons, more services are now being moved to the PaaS for Web Applications service, which has become a strategic component for the CERN Web Frameworks team and its parent unit in the IT department of CERN, the IT-CDA (Collaboration, Devices and Applications) group.
References

[1] Red Hat, "OpenShift Origin," [Online]. Available: https://www.openshift.org/.
[2] GitLab, "GitLab," [Online]. Available: https://gitlab.com.
[3] The Jenkins Project, "Jenkins - Build great things at any scale," [Online]. Available: https://jenkins.io/.
[4] Docker Inc, "Docker - Build, Ship, Run," [Online]. Available: https://www.docker.com/.
[5] Docker Inc, "What is Docker?," [Online]. Available: https://www.docker.com/what-docker.
[6] The Kubernetes Authors, "Kubernetes - Production-Grade Container Orchestration," [Online]. Available: https://kubernetes.io/.
[7] HAProxy, "HAProxy - The Reliable, High Performance TCP/HTTP Load Balancer," [Online]. Available: http://www.haproxy.org/.
[8] Internet Engineering Task Force (IETF), "RFC6066 - Transport Layer Security (TLS) Extensions: Extension Definitions," [Online]. Available: https://tools.ietf.org/html/rfc6066.
[9] Red Hat, "OpenShift Origin - Developer Guide - Templates," [Online]. Available: https://docs.openshift.org/latest/dev_guide/templates.html.
[10] Docker Inc., "Docker Documentation - Glossary - Definition of Image," [Online]. Available: https://docs.docker.com/glossary/?term=image.
[11] Grafana Labs, "Grafana Documentation," [Online]. Available: http://docs.grafana.org/.
[12] Red Hat, "OpenShift Origin - Developer Guide - ConfigMaps," [Online]. Available: https://docs.openshift.org/latest/dev_guide/configmaps.html.
[13] CERN, "CERN's Database on Demand Service," [Online]. Available: https://cern.service-now.com/service-portal/service-element.do?name=database-on-demand.
[14] Red Hat, "OpenShift Origin - Guidelines - Creating Images," [Online]. Available: https://docs.openshift.org/latest/creating_images/guidelines.html#openshift-origin-specific-guidelines.
[15] Red Hat, "OpenShift Origin - Builds and Image Streams," [Online]. Available: https://docs.openshift.org/latest/architecture/core_concepts/builds_and_image_streams.html#docker-build.
[16] Mattermost, Inc., "Mattermost - Open source, private cloud Slack-alternative," [Online]. Available: https://about.mattermost.com/.
[17] Red Hat, "OpenShift Origin - Creating Images - Source-to-Image (S2I)," [Online]. Available: https://docs.openshift.org/latest/creating_images/s2i.html.
[18] CERN, "CernVM File System (CernVM-FS)," [Online]. Available: http://cernvm.cern.ch/portal/filesystem.
[19] CERN, "EOS Large Disk Storage at CERN," [Online]. Available: http://eos.web.cern.ch/.
[20] CERN, "CERN Web Services," [Online]. Available: https://cern.ch/web.