

Payara Cloud - Click Deploy and Run on the Cloud.

Datasheet

Payara Cloud

Bringing your application to a cloud environment requires more work during the initial set up and more work to maintain than running your application in an on-premises environment.

Let us have a look at the main tasks of bringing your application into a cloud environment: Most cloud environments are based on a Kubernetes cluster to run the various parts of the application. Using a containerized process offers an improvement of the hardware resources in respect to Virtual Machines. The downside of using a Kubernetes cluster is learning Kubernetes itself! Learning how to use the Kubernetes commands and the best practices around containerized images is challenging for most developers. You provide the application and the configuration to Kubernetes through YAML files, and they have challenges due to the indentation requirements. And, especially when you are running your application in a public cloud, you must take into consideration all the security-related aspects of creating container images, as other containers can access and manipulate your container if you don't carefully construct the container image.

Networking is another challenging aspect of cloud environments. Each container gets an IP address assigned so communication to and from the application is possible. This is not different from a baremetal solution. The challenges come with the fact that you have various levels and different visibility.

A Kubernetes Pod can have multiple containers that may lead to port conflicts at that level. On the other side, you need to configure proper routing from the 'outside world' to your process. This also involves defining the domain name for your application and supplying a custom SSL certificate for the secure connection. Of all the infrastructure areas, networking is the least known by developers and often gives them difficulty. As a work-around solution, some less strict rules are implemented as that is the easiest solution. Opening as much as possible makes it work – but this is the worst solution when it comes to maintaining security.

Application Requirements

Payara Cloud will run any app following the Jakarta EE Web Profile specification.

What People Are Saying About Payara Cloud

- [Adam Bien](#)

“Payara Cloud follows opposite design decisions to the "next-generation standards." The application server cluster becomes a Kubernetes Operator and manages the nodes for you. You don't care about the server and only have to push a ThinWAR to the cloud. Payara Cloud hides the Kubernetes complexity and dramatically simplifies the infrastructure. The infrastructure is strictly separated from the business code. Payara Cloud might become the most productive way to run "boring" applications everywhere.”

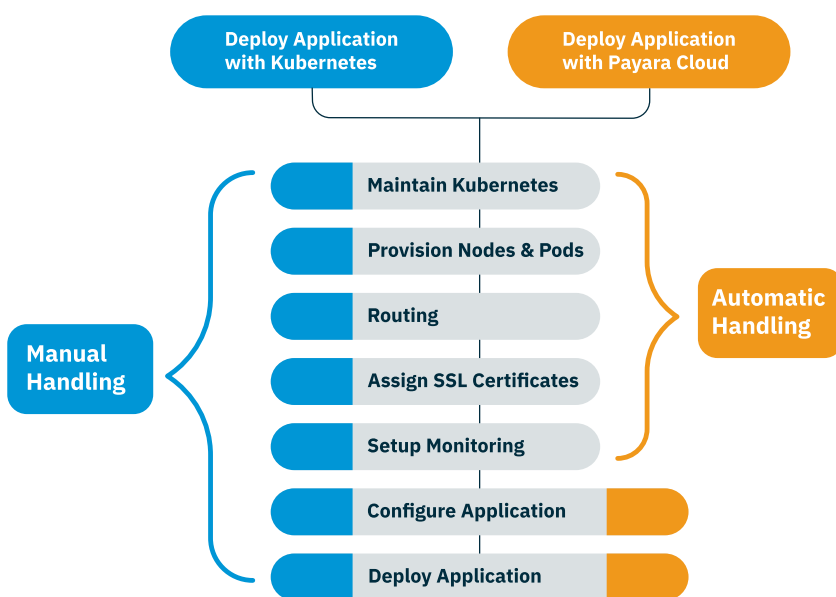
- [Steve Millidge](#)

“We are reimagining what a Jakarta EE runtime is. You bring the war we bring the cloud. Let Payara Cloud deal with infrastructure while you build the code.”

Another major aspect of going to the cloud is the monitoring of the entire system. Most of the time, when companies make the move to the cloud, they also start splitting up their application into several parts. Turning it into a microservices architecture makes sense as you can leverage the benefits of Kubernetes even more. You can scale the various parts of your application individually and upgrade and install them separately. But besides the pitfall that you can end up with a distributed monolith, you also have the issue of monitoring. You need more tools to monitor all those microservices, but you need a centralized place where you can check logging and resource usage. There are several standards around monitoring at the Cloud Native Computing Foundation (CNCF) and there are several applications following these standards, but cloud providers each still have their own custom solutions. Integrating your applications with them can be a challenge and, once you integrate with a specific cloud provider monitoring solution, it ties you to that provider (vendor lock-in) and becomes difficult to move to another provider.

Until now, running your applications in a cloud environment requires the developer to do a lot of infrastructure work to set up and maintain Kubernetes, provision nodes and pods, routing, assign SSL Certificates, set up the monitoring and configure the application.

Today, **you can use Payara Cloud** - a serverless architecture that allows you to build and run Jakarta EE Web apps without dealing with infrastructure management. Payara Cloud scans your application for database usage and configuration parameters defined using the MicroProfile Config specification, and then presents you with a configuration screen to enter these values. That's all you need to do to connect to your database and deploy the application. The provisioning of the Kubernetes resources, setting up the routing, networking aspects, and providing the SSL certificate for your endpoints are all handled for you. Some Platform As a Service solutions come close to this concept, but with Payara Cloud, the entire infrastructure is shielded away from the user.



The diagram on the left shows you the difference in the number of steps for deploying an application with Kubernetes when you handle each step yourself compared to the automated process of deploying an application using Payara Cloud. With Payara Cloud, you only need to configure and deploy the Jakarta EE application. All the other steps including monitoring, assign SSL Certificates, routing, provisioning nodes and pods and even maintaining Kubernetes are handled for you by the Payara Cloud environment.

Payara Cloud: Upload. Deploy. Run.

Payara Cloud is an all-in-one serverless platform and PaaS solution for Jakarta EE (Java EE) and MicroProfile applications that simplifies cloud infrastructure management by eliminating the need for application servers and Kubernetes knowledge.

Key Features of Payara Cloud:

- Automated Cloud Infrastructure. All-in-one serverless platform and PaaS eliminates the need for application servers, configuration of Kubernetes, Docker, routing, and certificates to simplify the process of running Jakarta EE and MicroProfile applications on the cloud.
- Monitoring and Diagnostics Integration. View application status and receive alerts for problems to solve application runtime problems quickly.
- Built-in Security. Protects your applications from the actions of other applications while allowing apps in the same namespace to communicate.
- Innovative 'Serverless Server'. Separates infrastructure from business code. Application cluster becomes a Kubernetes Operator and manages nodes for you.

Payara Cloud Subscriptions Include:

Use of Cloud Resources

- Kubernetes Pods to process the User requests
- Routing component to make application accessible for users
- Log analytics tool to store log entries
- Storage of application, log entries and monitoring artifacts (dumps)

Payara Cloud Software

- Provisions the application on the Azure platform
- Capture the configuration for an application
- Access to the logs and debug tools
- Built-in security
- Commercial support available with Premium level subscriptions

The Steps Payara Cloud Performs Are:

- 1** Prepare a container with your application.
- 2** Define a Kubernetes Deployment to run the container and application.
- 3** Provide a publicly available DNS name to access your application.
- 4** Request an SSL certificate for your application to access it securely.
- 5** Integrate with the Microsoft Azure Monitoring and Analytics system.

For an end-user, the provided functionality of an application is the most important aspect. Traditionally, Jakarta EE developers could focus on the most important aspect of development, which is the business logic that supplies the functionality of the finished application, while keeping the operations and deployments separated from the development task. With the rise of the DevOps movement, there was more and better communications between the developers and the operations department. But it also meant that infrastructure elements were introduced into the development process and that diverts the focus of the developers away from application development.

Cloud environments and the cloud-native approach also require that people learn new frameworks and processes, further distracting developers from coding - as most of the new frameworks and processes that need to be learned are not really focused on the actual goal of giving the end-users the required functionality in the finished application.

But now with Payara Cloud, developers can once again focus on creating the Web application based on the Jakarta Web Profile without worrying about the infrastructure. All of the infrastructure parts are taken care of for you based on the configuration value you provide during the upload of your WAR. All the Kubernetes interactions and Cloud Provider supplied functionality is taken care of for you, too, letting you run your applications in the cloud with ease.

[Subscribe to Payara Cloud Today.](#)

Payara Cloud Resources

Try Payara Cloud. Deploy and manage Jakarta EE (Java EE) applications without the complexity of building and maintaining infrastructure with our all-in-one, automated application server and cloud infrastructure management eliminates the need to learn Kubernetes, ingress, and application servers.

[Learn More.](#)

Just getting started with Payara Cloud? Watch a video tutorial on our [Payara Cloud Playlist](#), and read our step-by-step documentation to get the most out of Payara Cloud.

[Learn More.](#)



sales@payara.fish



+44 207 754 0481



www.payara.fish