



How to Manage and Operate the Payara Platform on Cloud

The Payara® Platform - Production-Ready,
Cloud Native and Aggressively Compatible.

Executive Summary

Moving data, applications, or business processes to a cloud computing environment offers measurable benefits to many businesses. From reduced expenses, improved agility, and better utilization of resources - cloud computing benefits have stimulated initial adoption and continue to drive new businesses into the cloud. Results from the [2019 Jakarta EE Developer survey](#) sponsored by the Eclipse Foundation indicate that 1/3 of Java developers are currently building cloud-native architectures, 30% are planning cloud-native architectures, and 32% expect that 2/3 of their Java applications will run in the cloud within two years' time. The Payara Platform is ideal for Jakarta (Java EE) developers looking to run applications in the cloud.

Running Applications on the Payara Platform in the Cloud

Since the Payara Platform is cloud-native and optimized for production environments, it can be deployed in any environment: on cloud, on-premises, or hybrid. It's designed with support for major cloud platforms, ensuring the Payara Platform is compatible with the services you're already using, including Microsoft Azure, Amazon AWS, and the Google Cloud Platform.

The Payara Platform works with a variety of architectural styles including serverless systems, reactive systems, and microservices. It's an ideal platform for running Jakarta EE and MicroProfile applications in the cloud.



Server Requirements

As a direct replacement for GlassFish Server Open Source Edition, Payara Server uses the same basic system requirements:

- JDK8u163 or above
- 512MB RAM

Support for Any Operating System Running One of the Following Java Virtual Machines

- Oracle JDK8 (u162+)
- Azul Zulu JDK8 (u162+)
- OpenJDK JDK8 (u162+)
- Oracle JDK 11 (11.0.4+)
- Azul Zulu JDK11 (11.0.4+)
- OpenJDK JDK11 (11.0.4+)

Principals of Running Payara Platform in the Cloud

The first approach is adopting a traditional application server architecture and building a traditional Payara Server domain deployment on your cloud platform's Infrastructure as a Service (IAAS) capabilities. This approach uses the IAAS capabilities to configure a number of virtual machines and network elements to build a similar topology as would be built in an on-premise data centre using virtualization technology or physical hardware.

The second approach is to adopt containerization technology like Docker to create a Docker image of your application leveraging Payara Micro's strong container support to build an elastic scalable deployment without worrying about Virtual Machines, network elements and other IAAS concerns.

The first approach has the advantage that it feels familiar to developers and administrators using cloud for the first time or lifting and shifting existing Java EE applications to the cloud to gain the cost and flexibility advantages of on-demand cloud infrastructure. This approach is also useful for additional test infrastructure for applications that may eventually be deployed on-premise in a local data centre. With the traditional domain approach, developers and administrators gain full access to the administration and monitoring capabilities provided by the Domain Administration Server for managing, monitoring, and controlling the Payara instances.

The second approach is more suited to microservices, although it doesn't rule out deployment of small applications. The container approach is more aligned to modern architectures and continuous delivery pipelines and provides the advantage that rapid creation and destruction of containers is possible for elastic workloads or rapid testing.

Payara Enterprise Subscription Include:

Choice of support:

- Migration & Project Support
- 24x7 – for mission critical environments
- 10x5 – business hours support

Ensures service level agreement (SLA) operation of your application server with:

- Unlimited tickets
- Customer Knowledge Base
- On-boarding support
- 10-year software lifecycle
- Fully supported production binaries
- Fully supported ecosystem components
- OpenJDK support



Payara Server on IAAS

Using a cloud platform's Infrastructure as a Service is ideal for creating traditional Payara Server domain deployments where existing applications are lifted and shifted to the cloud. New applications can also utilize this deployment model where rapid elasticity is not required and a normal application server topology is preferred.

Building a Payara domain on virtual machines in the cloud gives access to the full administration console for your Payara instances. Enabling management, monitoring and configuration of multiple Payara Servers across many virtual machines. The Payara administration console gives the operations team a single view of the Payara configuration and Payara Servers.

Payara configuration is fully scriptable using the `asadmin` command line interface and configuration scripts can be combined with the scripting capabilities of your preferred cloud to create fully reproducible builds of both the cloud infrastructure and the installation and configuration of the Payara instances. This enables creation of script repositories enabling configuration as code to reproducibly build multiple environments from test, pre-production, production and disaster recovery.

Payara Server supports high availability in Infrastructure as a Service deployments. Utilising the Domain Data Grid, Payara Server can take advantage of availability sets and zones and load balancing to ensure high availability for user session state without the need for complex "sticky session" style routing. Payara Server supports built-in health checks which can be used by the load balancer for liveness probes for failover. When more scale is needed, it is simple to add another virtual machine to the Virtual Network or VPC and the Availability Set or Zone and utilising the Payara Server administration console, an additional Payara instance can be added to the domain. Creation of additional resources can be scripted and triggered by monitoring.

Integrating the Payara Platform into Containers

The Payara Platform comes with a set of standard Docker images to build container images or they can be used as examples to create customized images that meet your needs.

The Payara standard Docker images can be used as a base with an additional configuration layer containing your application. These base images take advantage of the "thin war" capabilities of Payara Micro to create a single small layer that can be changed rapidly as your application evolves. Once your image is in your cloud provider's container registry instances of your container can be started with a single command line. When Payara Micro is running, it is possible to configure liveness probes to the `/health` endpoint and the container can be configured to restart if there is a failure in either the container or liveness probe. It is also possible to monitor CPU and memory usage of the

containers through the `/metrics` endpoint along with retrieving the log files and running diagnostic commands on the containers.

If you have a more complex Microservices architecture consisting of multiple interacting microservices it is possible to create container groups which starts a number of different Payara microservices together on the same compute resource.

Kubernetes and the Payara Platform

The Payara Platform has extensive Kubernetes support built-in.

To adapt your application to different environments, the Payara Platform can read container environment variables and Kubernetes secrets these can be used to configure database connection URLs, user names, passwords, or any environment-specific application configuration. These environment variables can also be read from the MicroProfile configuration API.

The Payara Platform also supports full clustering and High Availability on Kubernetes using Kubernetes service definitions to discover other data grid members. The Payara Data Grid provides elastic scaling for in-memory data on Kubernetes containers can be horizontally scaled in and out automatically via Kubernetes while retaining in-memory data including user session and cache data.

The Payara Platform exposes liveness and readiness endpoints that can be used by Kubernetes to understand the health of a container. Liveness probes can be used by Kubernetes to decide on whether containers should be restarted to ensure system availability and continuity.

Running Payara Platform applications using Containers or on a Kubernetes cluster in a V-Net or VPC enables the creation of heterogeneous deployment architectures. Combining both traditional Payara Server deployments on virtual machines with Payara Micro based microservices in container instances enables migration from application monoliths or interoperability between new microservice applications and standard Jakarta EE applications.

Payara Server's administration console also integrates with Docker and can control and monitor Payara Server instances running in a container on a Docker node.

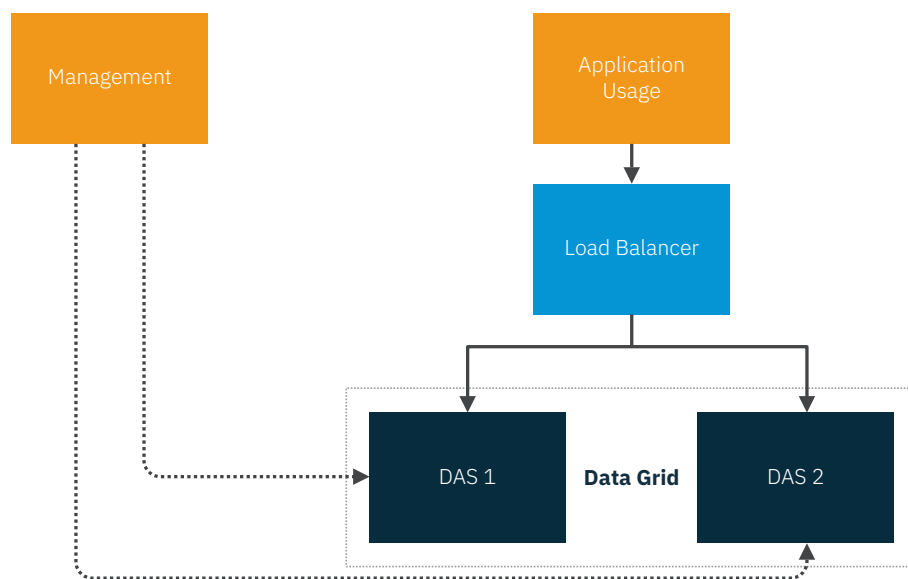
Payara Server Data Grid in the Cloud

The Payara Domain Data Grid brings the advantages of an in-memory computing platform to the applications deployed to Payara. The Payara Domain Data Grid provides high availability, scalability and performance for user session or cached data. While enabling elastic scalability by adding and removing instances of Payara Server or Payara Micro in containers.

The Payara Domain Data Grid works out of the box in traditional domain deployments on cloud IAAS platforms. Utilising knowledge of the topology of the virtual machines hosting the Payara domain, each Payara Server connects to other servers in the domain to form a robust, highly available cluster.

On container platforms and Kubernetes, the Payara Domain Data Grid can utilise Kubernetes service names to discover other grid nodes to form a highly available cluster. Alternatively, in pure Docker environments, the Domain Data Grid can use DNS names to form the grid.

It is possible to join different Domain Servers through the Domain Data Grid of Payara Platform. You need to set up the different domains separately but data, like session information or other data, can be shared between the application(s) running on the 2 (or more) Domain Servers:



Once formed, the Data Grid creates a robust, highly available in-memory data store where data, including user sessions, is available to all Payara instances in the grid.

Cloud Databases and the Payara Platform

The Payara Platform provides database access through both the Java Persistence API that provides object relational mapping capabilities and through standard JDBC (Java Database Connectivity) API. With the Payara Platform you can configure a connection pool to any relational database that comes with a standard JDBC driver including Amazon RedShift, and Amazon RDS instances. On Azure the

Payara platform supports Azure SQL database and standard SQL Server, Postgres and MySQL/Maria DB Azure offerings. For Google Cloud the Payara platform supports Cloud SQL and Cloud Spanner.

For applications to be migrated between different environments like Development, Test, Pre-production and Production database connectivity configuration needs to be externalized. The Payara platform supports full external configuration of key database connectivity attributes including JDBC URLs, usernames and passwords. For IAAS database connectivity configuration can be provided through asadmin scripting or via the administration console. For container environments database configuration can be provided via Kubernetes secrets or Docker environment variables. Integration with cloud specific security stores and Key Vaults is also possible.

Monitoring of the Payara Platform in the Cloud

The Payara Platform has extensive monitoring capabilities built into the runtime. The Payara Platform can monitor both system and JVM metrics as well as application level metrics. For system level metrics CPU, JVM heaps, Garbage Collection and others are available. Server level metrics are also available for example HTTP requests, Session counts, thread pools and connection pools. Application level monitoring is also available through bean pool, servlet and other application component metrics.

Payara exposes these metrics in many ways depending on the monitoring requirement. For traditional IAAS deployments metrics can be viewed in the administration console for any Payara Servers in the domain. For traditional monitoring tools that monitor Java runtimes through JMX all metrics are also available through a JMX interface. The Payara Platform can also expose metrics through a REST interface in a Payara specific format or through Prometheus format which is common in Kubernetes environments. All metrics can also be periodically written to the system log where they can be picked up by log analytics tools including Azure Monitor, ELK stack

In addition to metrics the Payara Platform provides diagnostic data including Healthchecks which periodically monitor critical system resources including heap size, thread pools, stuck threads and others to determine the overall health of the service. These checks are exposed through the log files and can be used in combination with a log analytics service to provide alerting.

The Payara Platform also provides application level diagnostics including a Request Tracing service that detects long running application requests and provides detailed tracing of the steps of the request to indicate where time was taken. The Payara platform also provides notification of Slow SQL calls that exceed configurable thresholds to notify operators of poorly performing queries indicating the query executed and the parameters. Both of these diagnostic services can be written to log files to integrate with a log analytics platform in both containers and IAAS deployments.

Support for Running Applications in the Cloud with the Payara Platform

If you are considering modernizing and moving existing Java EE applications to the cloud, the Payara Platform is the perfect platform choice. With extensive support for the cloud services you're already using, including Microsoft Azure, Amazon AWS, and Google - along with out of the box clustering on IaaS, migration is simple and painless. If you are looking to build new containerized applications or microservice architectures, the Payara Platform's extensive support for container-based products provides the perfect deployment platform for your solutions. If you need assistance, the Payara Accelerator team can provide customized consultancy services to advise on architectures and migration strategies or the Accelerator team can lift and shift your application onto the cloud for you. A Payara Platform Enterprise subscription provides support for the Payara Platform on cloud and direct access to Payara Engineers to ask questions and assist you if you have problems in moving your application to the cloud. [Find our more about Payara Platform Enterprise here.](#)

Payara Platform Cloud Resources

- [Using Containers with the Payara Platform – Data Sheet](#)
- [The Payara Platform and Docker](#)
- [Payara Platform and Docker Blog Posts](#)
- [Payara Platform and Kubernetes Blog Posts](#)
- [Payara Platform Migration Guide](#)

Payara Server Resources

Try Payara Server. Experience the benefits of developing Jakarta EE applications in our Java application server.

Download: <https://www.payara.fish/downloads>

Just getting started with Payara? Watch a video tutorial, read technical overviews and resources to get the most out of Payara Server.

Learn more: <https://www.payara.fish/get-started/>

Get involved. Join the Payara Community and help feed the fish! Payara Server is, and always will be, open source and we want your ideas, feedback and collaboration for ensuring Payara Server is the best option for production Jakarta EE applications.

Learn more: <https://www.payara.fish/community>

Spread the Word. Already using Payara Platform? Want to help us spread the word about our open source software? Join the Reef Community Program so we can support your efforts.

Learn more: <https://www.payara.fish/reef>

About Payara Services, Ltd

Payara Services is a global open source company and a recognized leader in the creation of innovative infrastructure software for today and tomorrow. We are proud to nurture and grow an open and collaborative community that builds on the needs of all to advance our software and services while providing support, stability, and security.

Our engaged team operates with the freedom and support to develop industry-leading products and services that enable our users to create world-class solutions across a diverse range of industries. We help shape the future of the industry through our direct contributions to Jakarta EE and Eclipse MicroProfile® as Eclipse Foundation Solutions Members and members of the Project Management Committee.



Microsoft® and Azure® are registered trademarks of Microsoft Corporation in the United States and/or other countries.

Jakarta EE® and MicroProfile® are registered trademarks of the Eclipse Foundation.

Docker and the Docker logo are trademarks or registered trademarks of Docker, Inc. in the United States and/or other countries. Docker, Inc. and other parties may also have trademark rights in other terms used herein.

Kubernetes is a registered trademark of The Linux Foundation in the United States and/or other countries.



sales@payara.fish



+44 207 754 0481



www.payara.fish

Payara Services Ltd 2020 All Rights Reserved. Registered in England and Wales; Registration Number 09998946
Registered Office: Malvern Hills Science Park, Geraldine Road, Malvern, United Kingdom, WR14 3SZ