Datasheet

Payara® Platform and Microsoft Azure®

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. Microsoft Azure® is Microsoft's public cloud computing platform providing cloud services for analytics, compute, networking, and storage. Users of Microsoft Azure can pick and choose from the available services to develop or scale new applications or run existing applications in the cloud.

Running Jakarta EE and Eclipse MicroProfile Applications on Azure

Some application servers, including the Payara Platform, are very well suited for cloud deployments, including Docker and Kubernetes containers. Since the platform is cloud-native and optimized for production environments, it can be deployed in any environment: on cloud, on-premises, or hybrid. 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 on Azure, and our commitment to supporting major cloud platforms ensures the Payara Platform is compatible with the services you're already using, including Microsoft Azure, Amazon AWS, and the Google Cloud Platform.

Payara's initial role as Strategic Members of Jakarta EE and MicroProfile, and our current position as a Solutions Member of the Eclipse Foundation, positions the Payara Platform for future compliance with Jakarta EE and ensures that industry standards are evolved to meet Payara Platform users' needs.

Server Requirements

Derived from GlassFish Server Open Source Edition, Payara Server uses the same basic system requirements:

- JDK8u163 or above
- 512MB RAM

Supported Java Virtual Machines

- Oracle JDK 8
- Azul Zulu 8
- OpenJDK 8
- IBM J9 8

Supported Operating Systems

- Windows
  - Windows 7+
  - Windows Server 2008+
- Linux
  - Ubuntu 14.04 (Trusty Tahr) LTS+
  - Debian 7 (Wheezy)+
  - CentOS 6+
  - RHEL 5+
  - OpenSUSE 42.2+
  - SUSE 11.4 & 12.2+
- Other
  - AIX 7.3 TL3+
  - MacOS(OSX) 10.10.5 (Yosemite)+

As you're considering a migration to Microsoft Azure, take a closer look at the capabilities of running the Payara Platform on Azure. Not only is the Payara Platform designed for the cloud, but we can help you complete your migration project on time and under budget with migration assistance or a done-for-you migration solution with our consultancy service, Payara Accelerator.

Strategies for Using the Payara Platform on Microsoft Azure

There are two main approaches to using the Payara Platform on Microsoft Azure and running your Jakarta EE and MicroProfile applications in Azure.

The first approach is adopting a traditional application server architecture and building a traditional Payara Server domain deployment on Azure's strong 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 on Azure.

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

Related Products & Services

- Payara Enterprise Support
- Migration & Project Support
- Payara Accelerator Consultancy

Payara Enterprise Support

Payara Enterprise Production Support options include:

- 24x7 – best for customers running mission critical environments that need rapid response times to critical incidents
- 10x5 – best for customers in need of business hours support

Payara Enterprise Support ensures service level agreement (SLA) operation of your application server with:

- Unlimited tickets
- Customer Knowledge Base
- On-boarding support
- 10-year support lifecycle
- Fully supported production binaries
- Fully supported ecosystem components
- Access to Zulu Enterprise-fully-supported builds of OpenJDK

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.

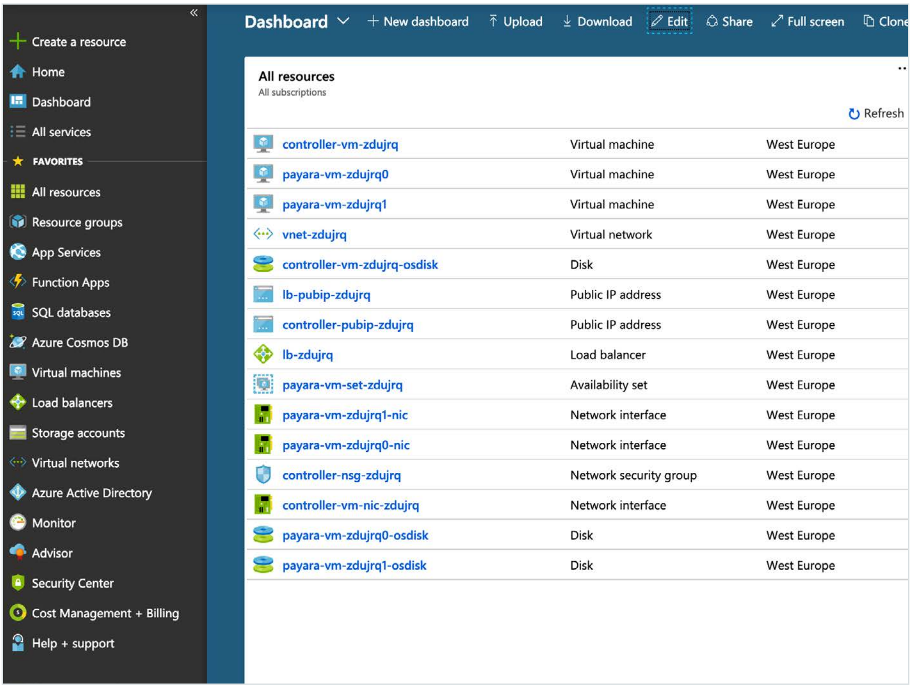
Microsoft Azure Infrastructure as a Service

The Microsoft Azure Platform provides a lot of capabilities for building a traditional compute environment on the cloud. Azure provides compute capacity, scale on demand, and a pricing model whereby you only pay for what you utilize. Azure is ideal for creating traditional Payara Platform 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.

> Azure with traditional Payara Platform deployments is ideal for lift & shift of existing applications to the cloud.

With the Azure platform, you can maintain your IAAS with a graphic UI, the Azure Portal, or using a command line tool.

It’s quick and easy to get started on the Azure Platform through the Azure Portal. To deploy the Payara Platform using the IAAS, you create compute resources like Virtual Machines, Network interfaces, Network Routes, Availability Sets and Load Balancers with a set of easy-to-follow screens and wizards. You don’t need to know all the commands to set up a complete environment. Also, it gives you an overview of what is available in terms of virtual machine configurations, for example, or features.



All resources

All subscriptions

controller-vm-zdujrq	Virtual machine	West Europe
payara-vm-zdujrq0	Virtual machine	West Europe
payara-vm-zdujrq1	Virtual machine	West Europe
vnet-zdujrq	Virtual network	West Europe
controller-vm-zdujrq-osdisk	Disk	West Europe
lb-pubip-zdujrq	Public IP address	West Europe
controller-pubip-zdujrq	Public IP address	West Europe
lb-zdujrq	Load balancer	West Europe
payara-vm-set-zdujrq	Availability set	West Europe
payara-vm-zdujrq1-nic	Network interface	West Europe
payara-vm-zdujrq0-nic	Network interface	West Europe
controller-nsg-zdujrq	Network security group	West Europe
controller-vm-nic-zdujrq	Network interface	West Europe
payara-vm-zdujrq0-osdisk	Disk	West Europe
payara-vm-zdujrq1-osdisk	Disk	West Europe

Once you have created the resources, there is little to no difference with the on-premise situation. To install the Payara Server, the Remote Shell functionality can be used to access the Virtual Machine. Or you can perform any other task you used to do on the on-premise servers - but you have the availability of additional tools. The Azure platform also includes some monitoring and health checks which allows you to instantly see the CPU or disk usage of the server running your Payara Server, for example.

Working with a UI can be error prone, if you need to repeat your work when setting up a production environment after configuration of a test environment. A scripted way of working is preferred to install your application environment using the Payara Platform. With the Azure Command Line tool, you can perform any action from the command prompt. Since you are in a scripting environment, it can be automated. You can perform the Payara Server installation, run the configuration commands through the asadmin Command Line Tool (such as setting up the database connection, among others). You can end up with a completely installed, clustered environment running your application as shown in the image above.

But all your work using the graphical environment of the Portal is not lost. You can export your configuration as a template which also allows the creation of a scripted version. In that way, you can reuse your effort in a maximum way and customize and automate the scripts.

The Payara Platform supports high availability in Azure's Infrastructure as a Service. Utilising the Domain Data Grid, Payara Server can take advantage of availability sets 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 and the Availability Set and utilising the Payara Server administration console, an additional Payara instance can be added to the domain.

Azure App Service and the Payara Platform

Microsoft Azure App Service is a way to quickly build, deploy, and scale enterprise grade Jakarta EE and MicroProfile applications on a fully managed platform. Microsoft Azure App Service provides built in autoscaling and load balancing. Payara Server and Payara Micro can be used to deploy Jakarta EE and MicroProfile applications into Azure App Service through its support for containerized applications. Payara Platform applications can then be deployed to Azure in seconds and App Services can be linked to CI and CD pipelines to automate updates when code changes in repositories.

To use Azure App Service with the Payara Platform, first you need to create a container registry to store a Docker image containing your application and the Payara Platform configuration. The Payara team provide standard Docker containers which can be used to build and host your application. Once you have a container image in the registry it is a simple matter to create an app service plan and create and start your application. The application starts in seconds and is immediately accessible from a public URL within Azure. The application can easily be deployed across Azure regions for high availability and can be easily scaled in a single command to add additional instances of your container image.

To adapt your application to different environments, the Payara Platform can read Appservice configuration settings as environment variables. The environment variables can be used to configure database connection URLs, user names, passwords, or any environment-specific application configuration. These app service settings can also be read from the MicroProfile configuration API.

Azure App Service provides very simple, rapid deployment of Jakarta EE and MicroProfile applications which can be auto-scaled across multiple availability zones. It is particularly suited to the deployment of RESTful services for microservice architecture and rapid deployment of test instances for web applications. Azure App Service deployed applications are always accessible from a public URL and communication between different App Service applications is potentially across the public Internet. If more complex microservice architectures are required then Azure Container Instances are more suitable.

Azure Container Instances and the Payara Platform

Microsoft Azure Container instances offer a fast way to run containers on the Azure cloud platform without having to worry about the creation of virtual machines or configuring Docker or Kubernetes. Running Jakarta EE and MicroProfile applications on Azure using Container Instances is similar in approach as used for Azure App Service. However, container instances have the advantage that they can be deployed into a previously configured Azure Virtual Network (V-Net). That means the application can communicate securely with other Container Instance applications or with traditionally deployed applications in a standard domain in the same V-Net. This ensures all communication is behind the V-Net firewall and provides greater opportunity to configure the network firewall rules and routing. With Azure Container Instances your Payara application can be easily scaled by adding more instances.

Combining both traditional Payara Server deployments on virtual machines with Payara Micro based microservices in container instances enables migration from application monoliths or interoperation between new microservice applications and standard Jakarta EE applications.

To use Azure Container Instances a similar process is followed as Azure App Service. First, you need a container registry for hosting your Docker containers. The Payara standard Docker images can be used as a base with an additional configuration layer containing your application. Once your image is in the container registry instances of your container can be started with a single command line. When Payara is running, it is possible to configure liveness probes 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 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.

To adapt your application to different environments, the Payara Platform can read container instance environment variables and 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.

Payara Platform applications or microservices deployed as Container Instances are rapid to deploy and scale and provide more configurability than for Azure App Service. Running Payara Platform applications using Container Instances in a V-Net enables the creation of heterogenous 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 interoperation between new microservice applications and standard Jakarta EE applications.

Azure Kubernetes Service and the Payara Platform

Azure Kubernetes Service simplifies the creation and management of Kubernetes and enables more complex container orchestration and management. Similar to Azure Container Instances and Azure App Services, Jakarta EE and MicroProfile applications can be deployed to Azure Kubernetes Service using Payara Server or Payara Micro Docker container images. Azure Kubernetes infrastructure can be created with a simple command and can be integrated into your V-Net to communicate with other deployments you have in Azure. Complete discussion of Kubernetes is beyond the scope of this guide.

Deploying Payara Micro or Payara Server into Azure Kubernetes Service is similar as for Azure Container Instances and Azure App Service in that a Payara Docker image can be used to layer on your application deployment and then store in a container registry. Kubernetes Pod definitions can then reference this container image and the start, stop and scaling of the pods can be managed through Kubernetes tooling. The Payara Platform has extensive Kubernetes support built into the platform and can utilize Kubernetes Secrets to configure your application to access other resources. The Payara Platform also supports full clustering and High Availability on Kubernetes using Kubernetes service definitions to discover other data grid members.

> The Payara Platform supports full clustering and High Availability on Kubernetes using Kubernetes service definitions to discover other data grid members.

Database Access on Azure with the Payara Platform

Microsoft Azure provides fully-managed Cloud SQL databases for use by your Azure-hosted cloud services. The Payara Platform can utilize these databases using standard JDBC drivers and configuration of data sources in the usual way for Java EE and Jakarta EE applications. All the usual Java EE capabilities like JPA can be used to access Azure cloud databases, in addition to the usual configuration of data sources through Payara Services administration console or through the deployment of datasource configuration embedded within your web application. The Payara Platform supports automatic configuration of environment-specific datasource properties through property replacement. For example, database usernames, passwords, and connection URLs typically change as you move from development through test environments and into production. Within Payara these properties can be defined as variables;

The Payara Platform provides easy integration to cloud environments without the need to update the Java EE EAR or WAR file to configure for different databases.

```
<data-source>
      <name>java:global/AzureSQLDB</name>
      <class-name>com.microsoft.sqlserver.jdbc.SQLServerXADataSource</class-
name>
      <server-name>${ENV=DB_HOST}</server-name> <port-number>1433</port-number>
      <database-name>${ENV=DB_NAME}</database-name>
      <user>${ENV=JDBC_USER}</user>
      <!-- Example of using a Payara password alias in the datasource
definition -->
      <password>${ENV=JDBC_PASSWORD}</password>
</data-source>
```

These variables can then be provided as environment variables through scripting when using virtual machines or via container configuration when using Docker container-based solutions like Azure Container Instances, App Services or Azure Kubernetes Service. In addition, the Payara Platform supports retrieving these properties through Kubernetes Secrets or via Azure Key Vault to provide additional security for passwords. These capabilities provide easy integration to cloud environments without the need to update the Java EE EAR or WAR file to configure for different databases or the need to update the datasource configuration in the Payara domain.

Integration with Other Azure Services

In addition to running Payara Server and Payara Micro on Azure, the Payara Platform can integrate with other Azure services. The Payara Platform is continually evolving and creating deeper integration with Azure services.

Payara Cloud Connectors for Azure

The Payara Cloud Connectors project provides a standard adapter to Azure Service Bus. This enables Payara applications to use standard Message Driven Beans to send and receive messages to Azure Service Bus. Azure Service Bus is reliable Cloud Messaging as a Service. The Cloud Connector for service bus can enable your application to send and receive messages without the need to manage an external JMS broker using Azure Service Bus as a highly reliable and scalable messaging infrastructure. Service Bus combined with the Payara Cloud Connector can be used to create asynchronous message pipelines between applications to microservices or to feed data to JBatch applications to trigger batch jobs. The Payara Cloud Connector for Service Bus is supported for both Payara Server and Payara Micro applications so can be used in both traditional IAAS and Container deployments.

> Payara applications use standard Message Driven Beans to send and receive messages to Azure Service Bus.

Payara Cloud connectors also provides an MQTT connector which can integrate with Azure IoT hub. Azure IoT hub can connect, monitor and manage many IOT devices. Devices can embed Payara Micro and use Payara’s MQTT cloud connector to send messages to the Azure IoT Hub. Alternatively, Payara applications deployed within Azure can use the Payara MQTT Cloud Connector to retrieve messages from devices and provide back end services to devices.

Payara Platform is Ideal for Running Existing or Newly Developed Java EE Applications on Azure

If you are considering lifting and shifting existing Java EE applications to Azure, the Payara Platform is the perfect platform choice. With extensive support for Azure and out of the box clustering on Azure 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 Azure's 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 Microsoft Azure for you. Your Payara Enterprise Support contract provides support for the Payara Platform on Azure and direct access to Payara Engineers to ask questions and assist you if you have problems in moving your application to Microsoft Azure.

Payara Server Resources

Try Payara Server. Experience the benefits of developing Java 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/documentation/getting-started-with-payara

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 Java EE applications.

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

About Payara Services, Ltd

We are a dedicated team of professionals devoted to Open Source, Java, our customers, and the community. We are major contributors to the development and engineering effort of the Payara Server Open Source Project and the Payara Foundation.

Our global team of specialist Payara Server Support Engineers delivers 24/7 production, development & migration support directly to our customers worldwide. When our customers need support – they get it straight from the engineers rather than an outsourced help desk employee.

As Solutions Members of the Eclipse Foundation, we invest our resources and expertise to improve, innovate and develop Open Source technologies. We are involved in shaping the future of the industry via our direct contribution to Eclipse EE4J (Payara's Director, Steve Millidge, is a Project Management Committee member), Eclipse MicroProfile, Eclipse IDE and many other projects.

As founding members and active committers to the Eclipse MicroProfile initiative, we contribute to optimize Enterprise Java for microservices architectures with the aim to drive forward industry innovation, further develop and contribute to the open source community.

We love to share Payara Server with the world, and when our commitment to continuous development and support of the Open Source software is combined with contributions from our community, we can ensure the highest quality solution. Let's continue to learn, create and shape the future of the industry together.

sales@payara.fish **+44 207 754 0481** **www.payara.fish**