Guide

How to Run a Payara Server Deployment Group on Microsoft Azure

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

Contents

Microsoft Azure Platform **1**
Azure Portal 1
Understanding the Concepts of Microsoft Azure **3**
Resource Group 3
Subscription 3
Resource Template 3
Installing Azure CLI **3**
Connecting to Subscription **4**
Running Payara Deployment Group on Azure Platform **5**
Create Resource Group 6
Create an SSH key pair 6
List of All Parameters 7
Access the Administration Console 9
Access Application 10
Note on the Load Balancer 11
Access Virtual Machines 11
Payara Platform Running on Azure **11**

Moving data, applications, or business processes to a cloud computing environment offers measurable benefits for 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. Azure is ideal for creating traditional Payara Platform domain deployments to lift and shift existing applications to the cloud.

Microsoft Azure Platform

Microsoft Azure Platform is a very versatile environment. There are several options to run your application on the platform, including:

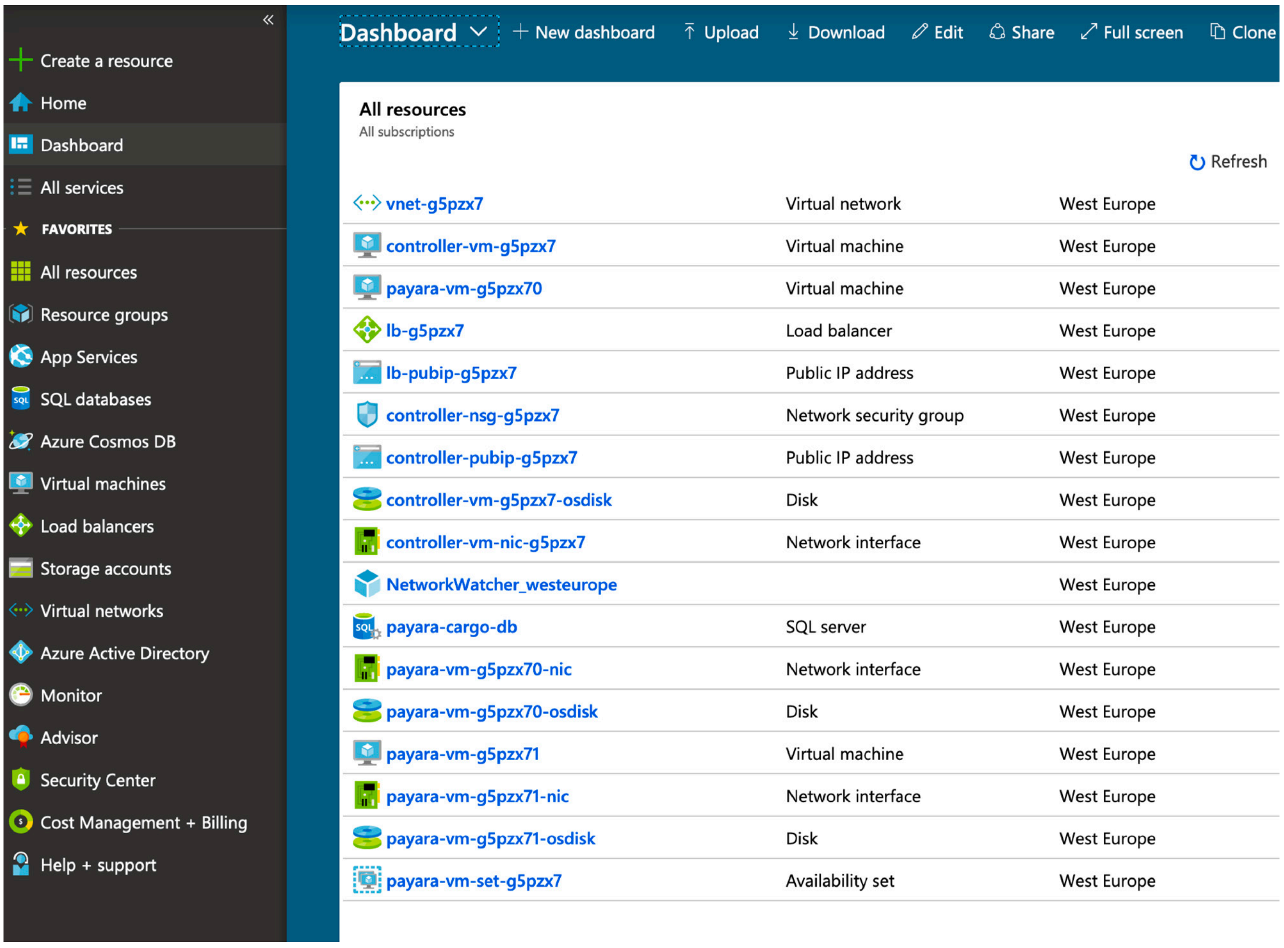
- IaaS: Infrastructure as a Service: It provides only a base infrastructure (Virtual machine, Software Define Network, Storage attached). The end-user has to configure and manage the platform and environment, deploy applications on it.
- PaaS: Platform as a Service: It provides a platform allowing end-user to develop, run, and manage applications without the complexity of building and maintaining the infrastructure.
- CaaS: Container as a Service: Is a form of container-based virtualization in which container engines, orchestration and the underlying compute resources are delivered to users as a service from a cloud provider.

In this user guide, we focus a bit more on the Infrastructure as a Service where 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.

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

Azure Portal

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

vnet-g5pzx7	Virtual network	West Europe
controller-vm-g5pzx7	Virtual machine	West Europe
payara-vm-g5pzx70	Virtual machine	West Europe
lb-g5pzx7	Load balancer	West Europe
lb-pubip-g5pzx7	Public IP address	West Europe
controller-nsg-g5pzx7	Network security group	West Europe
controller-pubip-g5pzx7	Public IP address	West Europe
controller-vm-g5pzx7-osdisk	Disk	West Europe
controller-vm-nic-g5pzx7	Network interface	West Europe
NetworkWatcher_westeurope		West Europe
payara-cargo-db	SQL server	West Europe
payara-vm-g5pzx70-nic	Network interface	West Europe
payara-vm-g5pzx70-osdisk	Disk	West Europe
payara-vm-g5pzx71	Virtual machine	West Europe
payara-vm-g5pzx71-nic	Network interface	West Europe
payara-vm-g5pzx71-osdisk	Disk	West Europe
payara-vm-set-g5pzx7	Availability set	West Europe

Once you have created the resources, there is little to no difference with the on-premise situation. To configure Payara Server, the Remote Shell functionality can be used to access the Virtual Machine. Or you can perform any other task that you would perform 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 the 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, for example).

You 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.

Understanding the Concepts of Microsoft Azure

Let us first explore a few of the concepts first:

Resource Group

A container that holds related resources for an Azure solution like IaaS or Azure Containers. The resource group includes those resources that you want to manage as a group. These resources are very versatile and include Virtual Machines, Networks, and IP address.

Subscription

You can create several subscriptions within your Azure account, each containing multiple Resource Groups. You mainly define some Subscription to have a separation in the billing and management of the resources.

Resource Template

With a Resource Template, multiple resources can be created at once. With the command-line tool, we can create multiple Virtual Machines, Networks, etc. But when our ‘deployment’ needs multiple resources, like the Payara Domain Server, Payara instances within a Deployment Group, etc.. it is easier to create them in one go.

Creating a resource from scratch is a rather tedious job, but you can download the template file for a certain resource you have created with the Azure Portal.

Installing Azure CLI

The installation of the Azure CLI is very straight forward. You can install it on Windows through an installer, or on Mac you can use the package manager Homebrew. On Linux, there is a shell script which performs the task. More info can be found on the page https://docs.microsoft.com/en-us/cli/azure/install-azure-cli?view=azure-cli-latest.

Connecting to Subscription

After the command line tool is installed, you have to connect it to your Azure Subscription. Once you are registered on the Azure Platform, you have a Subscription available that you can use to create the resource in. When you have defined this Subscription within the Azure CLI, you are ready to send commands to the Azure Platform.

Shell command

```
az account set -s <subscription-id>
```

If you have access to multiple Subscriptions or you are not sure which one is active, you can execute the show command to get information about it.

Shell command

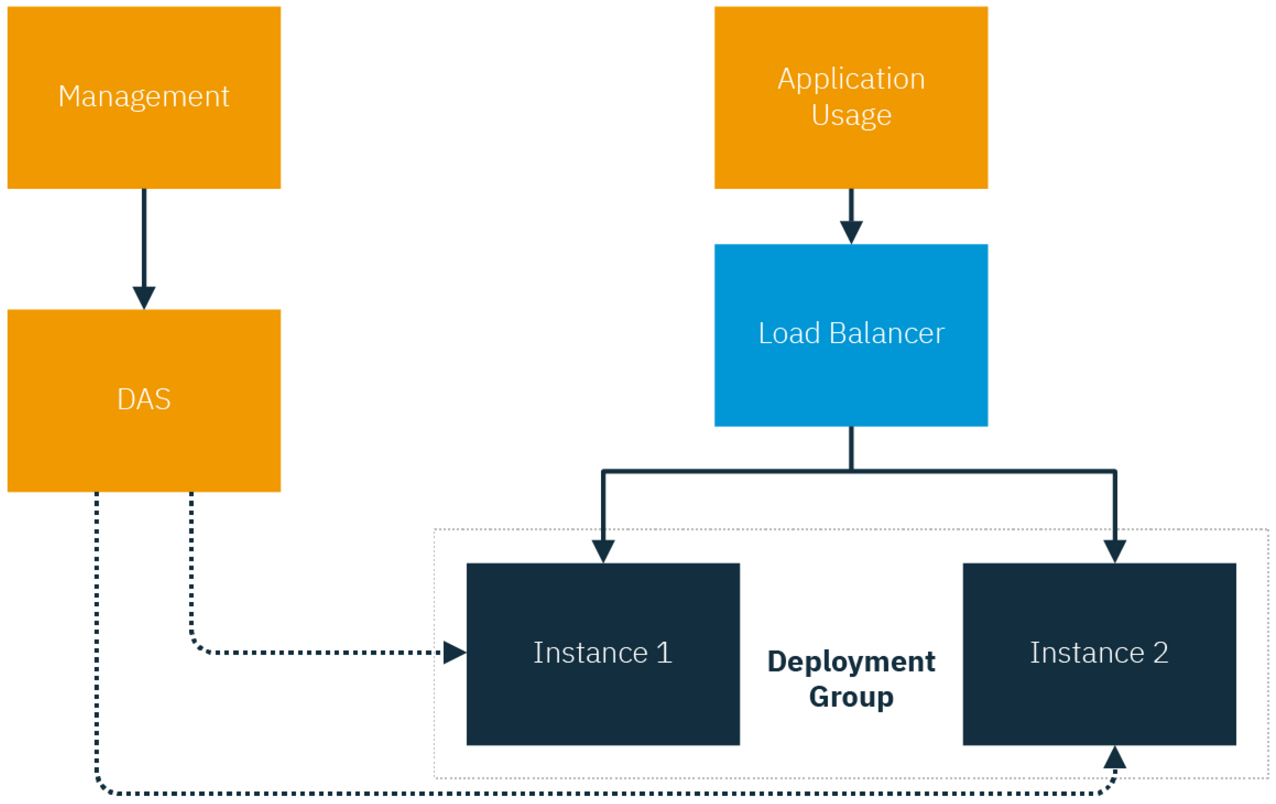
```
az account show
```

Output

```
{
    "environmentName": "AzureCloud",
    "id": "<subscription-id>",
    "isDefault": true,
    "name": "Payara MPN Subscription",
    "state": "Enabled",
    "tenantId": "<tenant-id>",
    "user": {
        "name": "<user-email>",
        "type": "user"
    }
}
```

Running Payara Deployment Group on Azure Platform

As an example and a starting point, we have prepared a Resource Template which allows you to create the following scenario:



The resources we need for this scenario are:

- A Virtual Machine for the Payara Domain Server
- A Virtual Machine for each of the Payara instances
- A network resource to connect all the different resources
- A load balancer to route the user request to the application deployed on the instances

Next we'll cover the steps for setting up this environment.

Create Resource Group

All resources are grouped within a Resource Group on the Azure Platform for easier management. The Resource Group also indicates the location of the data center of the resources.

Shell command

```
az group create -g payara_rg -l WestEurope
```

The above command creates a Resource Group with the name *payara_rg* and the location is in a data center in West Europe.

Create an SSH key pair

We need an SSH key pair to configure deployed VMs for remote SSH access. You can create a key pair using the following command (On a Windows environment, you can use the Putty program to generate the key):

Shell script

```
ssh-keygen -t rsa -N "" -f </path/to/my_payara_id_rsa>
```

This will generate a 2048 bit key without a passphrase (password) and we need to pass the public and private key to the Resource Template so that we can access the created Virtual Machines later on through SSH.

The Resource Template is located in our Payara Example Github Repository, have a look at https://github.com/payara/Payara-Examples/tree/master/cloud-providers/azure/template. The Template comprises of multiple files, but only the ***azuredeploy.json*** file one is required to be on your local machine. The other files are resolved by the Azure platform through the _artifactsLocation parameter defined in the ***azuredeploy.json*** file.

When you have downloaded the entire repo or the ***azuredeploy.json*** file, you can build the environment with the following command

shell script

```
az group deployment create -g payara_rg -n payara_demo --template-file
azuredeploy.json --parameters sshPublicKey="$(cat /path/to/my_payara_id_
rsa.pub)" sshPrivateKey="$(cat /path/to/my_payara_id_rsa | base64 -w 0)"
payaraAdminServerPassword=<your_desired_payara_admin_password>
```

On MacOSX, the command ‘base64’ doesn’t need any parameters.

This command can take a few minutes to execute since it has to provision also some virtual machines for example. Once the environment is up and running, you should treat it the same as you would with any on-premise server.

The meaning of the parameters are:

- -g: The resource group where the resources will be placed
- -n: The name of the deployment. Optional. If not specified the name (without extension) of the template file is taken
- --template-file: The template file which is used for the creation of this deployment.
- --parameters: The required parameters we are passing to the Azure Resource Manager

Another option is to define the parameters in a JSON file and reference this file on the command line like:

```
--parameters @azuredeploy.parameters.json
```

The parameters JSON file needs a specific structure. An example of it is placed in the GitHub repository : https://github.com/payara/Payara-Examples/blob/master/cloud-providers/azure/template/azuredeploy.parameters.json

List of All Parameters

Parameter Name	Description	Default value
_artifactsLocation	The base URI where artifacts required by this template are located and this location should be accessible by the Azure Platform. De default value points to the Payara Github repository containing the additional artifacts.	The GitHub Payara Example repository location
_artifactsLocationSasToken	The sasToken required to access _artifactsLocation. When the template is deployed using the accompanying scripts, a sasToken will be automatically generated.	empty
location	Azure Location for all resources	Location of the Resource Group

Parameter Name	Description	Default value
customSubnetId	Azure resource ID of the subnet where the Payara cluster is to be deployed. If this is empty, a new *vnet* and a *subnet* will be created.	empty
sshUsername	The user name used in the SSH communication between the Payara Server Domain Server and the Payara instances.	payaraadmin
sshPublicKey	The Public Key part used for the communication between the Payara Server Domain Server and the Payara instances.	No default, Required
sshPrivateKey	Base64-encoded SSH private key (corresponding to the sshPublicKey param). This is to allow the Payara Domain Server (on the controller VM) to log on to each Payara instance VM and configure it.	No default, Required
payaraAdminServerPassword	Password you want to set for the Payara Domain Server's 'admin' account. Do not put any dollar sign in the password, as the shell script may interpret it unexpectedly.	No default, Required
controllerVmSku	Machine size for the controller VM running the Payara Domain Server. The list of valid values can be found on this page (https://docs.microsoft.com/en-us/azure/virtual-machines/windows/sizes-compute).	Standard_F4s_v2
serverSku	Machine size for each Payara VM in the cluster. The list of valid values can be found on this page (https://docs.microsoft.com/en-us/azure/virtual-machines/windows/sizes-compute).	Standard_F4s_v2
serverCount	The number of Instances in the Deployment Group.	2
loadBalancerSku	The type of load balancer which needs to be used. Valid values are 'Basic' and 'Standard'. For the deifference bewteen both type, have a look at https://docs.microsoft.com/en-us/azure/load-balancer/load-balancer-overview.	Basic (free)

Besides these parameters, other customizations can be performed. This can be achieved by taking the following steps:

- Copy the template files from the GitHub Repository to a location which is accessible by Azure (like your own GitHub Repository)
- Make sure you adjust the ***_artifactsLocation*** parameter value within the ***azuredeploy.json*** file. Otherwise, some of the artifacts will still be created from the definition in the Payara GitHub and not from your copy.
- Make updates to the Resource template as you see fit.

Some of the changes that you can perform:

You can change the Payara version which is installed within the file scripts/setup_payara.sh. However, make sure that you are using a Payara version which is compatible with the Java version which is installed on the Virtual Machines.

Update the names of the resources which are created. These names are defined by the variable section which you can find at the end of the *azuredeploy.json* file.

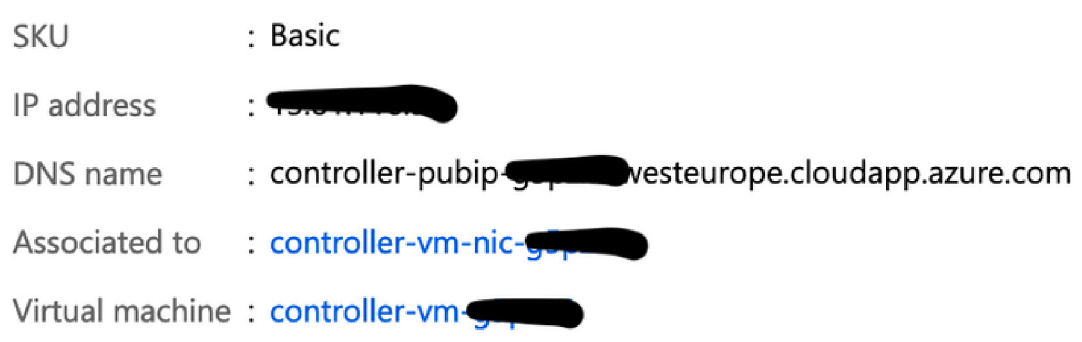
Change the Payara Server configuration. The Payara installation is performed by the scripts/payara_setup.sh script. By changing this script, you change how the Payara Server is installed and configured on the Virtual Machines.

Change the Resource template by adding an SQL server. You can have a look at the blog https://blog.payara.fish/payara-platform-on-microsoft-azure-accessing-sql-databases which describes how you can add and integrate. Once you have created the resource, you can request the template for it through the Azure Portal and integrate it in this Resource Template.

Access the Administration Console

Now that everything is set up, we can access the Web Administration Console of the Domain Server. It allows us to make further configuration changes or access monitoring and logging information for example. The Resource Template has also provisioned a public IP Address and DNS name for the Domain Server.

You can find these values on the Azure Portal. Look for the 'Public IP address' with the name 'controller-pubip-xxxxxx'. The details of this resource will give you the information you need to access the Domain Server.



SKU	: Basic
IP address	: [redacted]
DNS name	: controller-pubip-[redacted]westeurope.cloudapp.azure.com
Associated to	: controller-vm-nic-[redacted]
Virtual machine	: controller-vm-[redacted]

You can also retrieve this information through the Azure CLI tool, by issuing the following command:

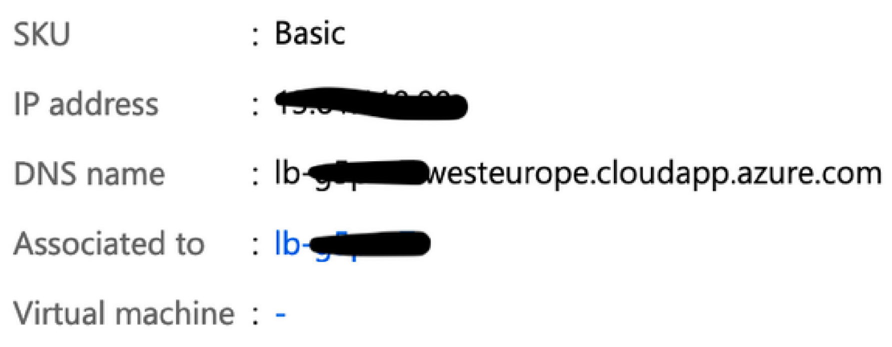
shell command

```
az group deployment show -g payara_rg -n payara_azure --query properties.
outputs.controllerDNS.value
```

The access to the Administration console is secured since it is accessible through a public IP. So you need to use SSL (https) and the user name and password you have specified during the creation of the environment. The user name is by default *payaraadmin* (but can be changed by a parameter) and the password is a required parameter you need to specify during creation.

Access Application

In a similar way, you can found out the DNS name and the public IP Address of the load balancer which routes the user requests to our Payara instances. In the Azure Portal, you need to look for the 'Public IP address' resource with the name 'lb-pubip-xxxxxx'.



SKU	: Basic
IP address	: [redacted]
DNS name	: lb-[redacted]westeurope.cloudapp.azure.com
Associated to	: lb-[redacted]
Virtual machine	: -

Or the equivalent azure CLI command is:

shell command

```
az group deployment show -g my_payara_rg -n payara_azure --query properties.
outputs.loadBalancerDNS.value
```

Note on the Load Balancer

A note on the operation on the load balancer is required here. By default, requests coming from a certain Client IP are always directed to the same node. So requests from your browser should end up always on the same Payara Instance. But requests from another user will end up then on another instance. For more information on the load balancer internals can be found at https://docs.microsoft.com/en-us/azure/load-balancer/load-balancer-overview.

Access Virtual Machines

During the creation of the environment, we have supplied the public and private key so that we can access the Virtual machines using SSH for example. You also need to know the DNS name or IP address associated with the VM running the Payara Domain Server.

shell command

```
ssh -i </path/to/my_payara_id_rsa> payaraadmin@controller-pubip-xxxxxx.
westeurope.cloudapp.azure.com
```

You can always configure your environment to link a certain host with the private key so that the -i parameter isn't required.

Payara Platform Running on Azure

With the help of the Resource Template, you can create an environment in a consistent way. Instead of issuing different commands with a set of parameters which need to correspond, this is a better way to set up your environment in a more consistent and stress-free way.

Together with the infrastructure, like Virtual Machines, it also installs the Payara Server and configures it at the same time.

Once the script has run, you have some servers you can access in the same way you would access servers on-premise. You can access them through Secure Shell making configuration changes as needed, through the web-based Admin Console of Payara Server to perform any Payara related task you need and adjust your environment easily by adding another Virtual Machine for an additional instance within the Deployment Group. And, just as with physical servers, changes are persistent when you stop and start the Virtual Machine.