



The Retail Innovators

SAP Dynamic Pricing by GK

# Installation Guide

Version: v4.0.0



## COPYRIGHT

© 2022 SAP SE or an SAP affiliate company. All rights reserved. No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP SE or an SAP affiliate company.

1. You may not use the SAP Material for a purpose competitive with SAP or its products unless otherwise clearly permitted by applicable law.
2. You may not use the SAP corporate logo.
3. No use of other SAP trademarks is granted under this section. For information regarding use of SAP trademarks, see <http://www.sap.com/corporate/en/legal/trademark.html>.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. All other product and service names mentioned are the trademarks of their respective companies.

Internal Document Information: 1223593586 | 2022-03-02

## TABLE OF CONTENTS

1	Introduction	4
2	Cloud: Microsoft Azure	4
2.1	Prerequisites .....	4
2.2	Basic Setup .....	4
2.2.1	Install CLI.....	4
2.2.2	Login and technical user .....	4
2.2.2.1	Portal .....	4
2.2.2.2	CLI .....	4
2.3	Application setup .....	5
2.3.1	Step 1: Resource group .....	5
2.3.2	Step 2: Create database .....	5
2.3.3	Step 3: Create the network.....	6
2.3.4	Step 4: Network security group.....	7
2.3.5	Step 5: Storage.....	8
2.3.5.1	Create the file storage.....	8
2.3.5.2	Accessing the storage .....	8
2.3.6	Step 5: Create the logging environment .....	9
2.3.7	Step 6: Running the container .....	10
2.3.8	Additional components .....	10
2.3.9	Resource group and docker container registry.....	11
2.4	Required Cloud Services.....	11

# 1 Introduction

This guide describes the installation of the software for these scenarios:

- Cloud: Microsoft Azure

Please contact your consultant for an on-premises Installation Guide.

## 2 Cloud: Microsoft Azure

### 2.1 Prerequisites

- Get a user account for Azure
- Access to AIR docker image file in docker registry (Azure account)

### 2.2 Basic Setup

#### 2.2.1 Install CLI

There is a CLI tool to access the Azure environment via terminal.

```
sudo apt-get update
sudo apt-get install -y libssl-dev libffi-dev python-dev build-essential
curl -L https://aka.ms/InstallAzureCli | sudo bash
```

After that, the **az** CLI tool can be used.

#### 2.2.2 Login and technical user

To start the user access has to be set up before going into the container deployment.

There are two ways to create the technical user:

- via portal or
- with the CLI.

##### 2.2.2.1 Portal

Go to Microsoft Azure login page: <https://portal.azure.com>

Login and create an app registration in active directory.

##### 2.2.2.2 CLI

At first send login command to open a new login session.

```
az login # Option: Provide credentials directly with the command
```

After executing this command, a browser window with a login to a Microsoft account should open. Otherwise is possible to provide the correct credentials at the login command also.

Now a technical user has to be created for further API access.

```
az ad sp create-for-rbac --name prdTechnicalAppId --password "PASSWORD"

# Response
{
  "appId": "APP-ID",
  "displayName": "prdTechnicalAppId",
  "name": "http://prdTechnicalAppId",
  "password": "PASSWORD",
  "tenant": "TENANT-ID"
}
```

After that, a login with this technical user is possible.

```
az login --service-principal --username APP-ID --password PASSWORD --tenant TENANT-ID

az acr login --name prdContainerRegistry
```

## 2.3 Application setup

In the Microsoft Azure Cloud are many small components combined together to build an application environment.

In our case the following resources are used:

### Required Cloud Services

Cloud Service	Description
Container Instance	CPU, RAM
Application Gateway	Public IP and private IP (for on Premises access)
Azure Database for PostgreSQL Server	Managed Database
Virtual network	Virtual network for container, database and gateways
Virtual Network Gateway	VPN gateway for accessing on Premises systems
Storage, Managed Disk HDD	Persistency for temporary files that exceed the container file system limit of 2GB
Azure Monitor - Log Analytics	Storing and evaluating log files. At time of writing, 5 GB per month are for free.

With those components the application could be built with the following steps.

### 2.3.1 Step 1: Resource group

A resource group is defined by Azure as a life cycle group. That means it contains all elements which have the same life cycle.

They are created together and should be destroyed together.

```
az group create --name "${resourcegroup}" \
  --location "${location}" \
  --subscription "${subscription}" \
  --tags TYPE=APPLICATION-GROUP
```

### 2.3.2 Step 2: Create database

After creating the resource group the real components could be created.

The SKU "B\_Gen5\_1" is a naming convention to create a defined database server. (B means it is basic, Gen5 is the processor generation, 1 is the number of CPU cores)

We see 4 steps:

1. Create database server
2. Add a firewall rule to provide access only to some hosts
3. Add a firewall rule to switch on portal setting "Allow access to Azure services"
4. Add a new database in the server

```

az postgres server create --name "${dbservername}" \
  --resource-group "${resourcegroup}" \
  --location "${location}" \
  --admin-user "${dbadmin}" \
  --admin-password "${dbadmin_password}" \
  --ssl-enforcement ENABLED \
  --sku-name B_Gen5_1 \
  --storage-size ${db_storage_size_in_gb} \
  --version 10

az postgres server firewall-rule create --server "${dbservername}" \
  --resource-group "${resourcegroup}" \
  --name AllowPrudsysIP \
  --start-ip-address ${startip} \
  --end-ip-address ${endip}

az postgres server firewall-rule create --server "${dbservername}" \
  --resource-group "${resourcegroup}" \
  --name AllowAllWindowsAzureIps \
  --start-ip-address 0.0.0.0 \
  --end-ip-address 0.0.0.0

az postgres db create --name ${dbname} \
  --resource-group "${resourcegroup}" \
  --server-name "${dbservername}" \
  --charset "UTF8" \
  --collation "de_DE" \
  --subscription "${subscription}"

```

### 2.3.3 Step 3: Create the network

After the database we need a basic network environment.

Only with this basic network it is possible to configure a VPN to access on premises systems either.

There are 5 steps to create such a network:

1. Create the virtual network
2. Create two subnets inside of that network (one as base subnet and one for the runtime container)
3. Create a public IP which is accessible from the internet
4. Create a network security group to allow IP filtering for incoming requests
5. Create an application gateway, which provides routings and http rules for traffic control

```

az network vnet create \
  --name "${vnet_name}" \
  --resource-group "${resourcegroup}" \
  --location "${location}" \
  --address-prefix 10.242.4.0/22 \
  --subnet-name "${subnet_name}" \
  --subnet-prefix 10.242.4.0/24

az network vnet subnet create \
  --name "${subnet_name}" \
  --resource-group "${resourcegroup}" \
  --vnet-name "${vnet_name}" \
  --address-prefix 10.242.5.0/24

az network vnet subnet create \
  --name "${aci_subnet_name}" \
  --resource-group "${resourcegroup}" \
  --vnet-name "${vnet_name}" \
  --address-prefix 10.242.4.0/24

az network public-ip create \
  --resource-group "${resourcegroup}" \
  --name myPublicIPAddress \
  --dns-name "${resourcegroup}"

az network application-gateway create \
  --name "${gateway_name}" \
  --location "${location}" \
  --resource-group "${resourcegroup}" \
  --capacity 2 \
  --sku "Standard_Small" \
  --public-ip-address "myPublicIPAddress" \
  --vnet-name "${vnet_name}" \
  --subnet "${subnet_name}" \
  --http-settings-port 8080 \
  --servers "10.242.4.4" "10.242.4.5"

```

### 2.3.4 Step 4: Network security group

Create a network security group to allow IP filtering of incoming hosts. A network security group is a collection of rules based on IPs, ports or protocols(e.g. TCP, UDP).

The security group must be linked to a subnet to protect it.

```

# Create a network security group (NSG) for the front-end subnet.
az network nsg create \
  --resource-group ${resourcegroup} \
  --name ${networksecuritygroup} \
  --location ${location}

# Associate the back-end NSG to the back-end subnet.
az network vnet subnet update \
  --vnet-name ${vnet_name} \
  --name ${aci_subnet_name} \
  --resource-group ${resourcegroup} \
  --network-security-group ${networksecuritygroup}

# Create NSG rules to allow HTTP & HTTPS traffic inbound.
az network nsg rule create \
  --resource-group ${resourcegroup} \
  --nsg-name ${networksecuritygroup} \
  --name AllowedAccessors \
  --access Allow \
  --protocol Tcp \
  --direction Inbound \
  --priority 100 \
  --source-address-prefix "${allowed_ips}" \
  --source-port-range "*" \
  --destination-address-prefix "*" \
  --destination-port-range "*"

```

### 2.3.5 Step 5: Storage

For file exchange with external systems there is the file storage.

A file storage could be up to 5120 GB and is accessible from the internet via SMB 3.0 protocol.

#### 2.3.5.1 Create the file storage

Creating a file storage is a two step process:

1. Create a storage account and remember the storage account id
2. Get the key of the storage account id
3. Create a share inside the storage account with the account id and key

```
STORAGEACCT=$(az storage account create \
  --resource-group "${resourcegroup}" \
  --name "${storageaccountname}" \
  --location ${location} \
  --sku Standard_LRS \
  --query "name" | tr -d ' ')

STORAGEKEY=$(az storage account keys list \
  --resource-group "${resourcegroup}" \
  --account-name $STORAGEACCT \
  --query "[0].value" | tr -d ' ')

az storage share create \
  --account-name $STORAGEACCT \
  --account-key $STORAGEKEY \
  --quota ${quota} \
  --name "${sharename}"
```

#### 2.3.5.2 Accessing the storage

Linux

To mount such an Azure share there must be installed the cifs-utils package, which provides the SMB 3.0 access.

If the package is installed, the mount command is:

```
sudo mount -v -t cifs //$${storageaccountname}.file.core.windows.net/${storageaccountname}
/my/local/mountdirectory -o
vers=3.0,username=sanicarepricing,password=${storage_access_key1},dir_mode=0777,file_mode=0777,serverin
o
```

After executing this command any user on the system could read and write inside the storage inside the folder "/my/local/mountdirectory".

Windows

1. Open file explorer with Ctrl + E.
2. Go to this computer and map network drive
3. Use the UNC path "mystorageaccountname.file.core.windows.net/mystorageaccountfileshare" from connect page in the Azure portal.
4. Select a drive and use the UNC path as target.
5. Use the storage account name with the prefix "AZURE\" as user name and one of the keys as password.
6. The storage is now connected and could be used for file operations.



7. If the connection should be released, do that by right click and select Disconnect.

### 2.3.6 Step 5: Create the logging environment

To get access to the container logfiles inside the Azure logging platform, a log analytics workspace must be created.

This workspace must live inside the same subscription as the running container.

In the case the container could be connected with the log workspace.

```
az group deployment create \
  --resource-group ${resourcegroup} \
  --name "LogAnalyticsWorkspace-${resourcegroup}" \
  --template-file scripts/log_workspace_template.json \
  --parameters workspaceName=LogAnalyticsWorkspace-${resourcegroup}
```

Because the log workspace is not implemented in the Azure CLI, creation must be executed via the generic template API of Azure.

The following JSON describes a basic log workspace instance used by the upper command.

```
{
  "$schema": "https://schema.management.azure.com/schemas/2014-04-01-preview/deploymentTemplate.json#",
  "contentVersion": "1.0.0.0",
  "parameters": {
    "workspaceName": {
      "type": "String",
      "metadata": {
        "description": "Specifies the name of the workspace."
      }
    },
    "location": {
      "type": "String",
      "allowedValues": [
        "westeurope"
      ],
      "defaultValue": "westeurope",
      "metadata": {
        "description": "Specifies the location in which to create the workspace."
      }
    },
    "sku": {
      "type": "String",
      "allowedValues": [
        "Standalone",
        "PerNode",
        "PerGB2018"
      ],
      "defaultValue": "PerGB2018",
      "metadata": {
        "description": "Specifies the service tier of the workspace: Standalone, PerNode, Per-GB"
      }
    }
  },
  "resources": [
    {
      "type": "Microsoft.OperationalInsights/workspaces",
      "name": "[parameters('workspaceName')]",
      "apiVersion": "2015-11-01-preview",
      "location": "[parameters('location')]",
      "properties": {
        "sku": {
          "Name": "[parameters('sku')]"
        },
        "features": {
          "searchVersion": 1
        }
      }
    }
  ]
}
```

### 2.3.7 Step 6: Running the container

After creating the whole environment, the application is ready to start.

To start a container Azure provides the following CLI command:

Currently automated connecting of a container with the log analytics workspace is not finished. The log analytics workspace id and key must be set via manifest settings before container startup. (1. Do setup, 2. Get the ids and set them in manifest. 3. Start container.)

With the container startup following parameters are important:

- Used docker registry, path to image with username and password
- Resources: Number of cpus, amount of memory
- Used network
- Environment variables used by the application inside the running container
- Link to the file storage
- Link to the log analytics workspace

```
# Get starteg key for mounting into container instance
STORAGE_KEY=$(az storage account keys list --resource-group ${servicename} --account-name
${servicename} --query "[0].value" --output tsv)

az container create \
  --resource-group "${servicename}" \
  --location westeurope \
  --image devregistryprudsys.azurecr.io/product/prudsys/pricing:${version} \
  --name ${servicename} \
  --cpu ${number_of_cpus} \
  --memory ${memory_in_gb} \
  --registry-login-server devregistryprudsys.azurecr.io \
  --registry-username ${registry-username} \
  --registry-password ${registry-password} \
  --vnet vnet-pricing \
  --subnet subnet-pricing-aci \
  --subnet-address-prefix 10.242.4.0/24 \
  --ports 8080 \
  --environment-variables \
    'DBHOST'="prudsys-${servicename}-db.postgres.database.azure.com" \
    'DBUSER'="${db_user}@prudsys-${servicename}-db" \
    'DBPASS'=${db_password} \
    'DBNAME'='prudsys' \
    'OPERATION_MODE'='CLOUD' \
    'OPERATION_MODE_CLOUD_INSTANCE'=${instanceid} \
    'OPERATION_MODE_CLOUD_ADMIN_PASSWORD'='${air-admin-password}' \
    'OPERATION_MODE_CLOUD_LOG_FORMAT'='CONSOLE_JSON' \
    'OPERATION_MODE_CLOUD_PLUGIN_REPOSITORY'='/opt/prudsys/plugins' \
    'OPERATION_MODE_CLOUD_USE_DB_PLATFORMSTORAGE'='true' \
    'MEM_JAVA_PERCENT'=${java_memory_percent} \
    'MEM_TOTAL_KB'=${java_heap_in_kb} \
  --log-analytics-workspace ${log_analytics_workspace_id} \
  --log-analytics-workspace-key ${log_analytics_workspace_primary_key} \
  --azure-file-volume-account-name ${servicename} \
  --azure-file-volume-account-key ${STORAGE_KEY} \
  --azure-file-volume-share-name ${servicename} \
  --azure-file-volume-mount-path /mnt/azurestorage/
```

### 2.3.8 Additional components

The following components are used as Microsoft Azure services, but they are not part of a standard customer installation.

### 2.3.9 Resource group and docker container registry

As a next step, a resource group has to be created. This is a logical container to deploy and manage Azure resources in dedicated locations.

```
az group create --name prdGroup --location "westeurope"

# Response of az:
{
  "id": "/subscriptions/33e8d9b6-eb80-4a28-8eb3-5b80ee61cc9d/resourceGroups/prdGroup",
  "location": "westeurope",
  "managedBy": null,
  "name": "prdGroup",
  "properties": {
    "provisioningState": "Succeeded"
  },
  "tags": null
}
```

After the resource group, the container registry can be created.

```
az acr create --resource-group prdGroup --name prdContainerRegistry --sku Basic

# Response of az:
{
  "adminUserEnabled": false,
  "creationDate": "2018-12-03T08:55:20.954151+00:00",
  "id": "/subscriptions/33e8d9b6-eb80-4a28-8eb3-5b80ee61cc9d/resourceGroups/prdGroup/providers/Microsoft.ContainerRegistry/registries/prdContainerRegistry",
  "location": "westeurope",
  "loginServer": "prdcontainerregistry.azurecr.io",
  "name": "prdContainerRegistry",
  "provisioningState": "Succeeded",
  "resourceGroup": "prdGroup",
  "sku": {
    "name": "Basic",
    "tier": "Basic"
  },
  "status": null,
  "storageAccount": null,
  "tags": {},
  "type": "Microsoft.ContainerRegistry/registries"
}
```

## 2.4 Required Cloud Services

Cloud Service	Description
Container Instance	CPU, RAM
Application Gateway	Public IP and private IP (for on Premises access)
Azure Database for PostgreSQL Server	Managed Database
Virtual network	Virtual network for container, database and gateways
Virtual Network Gateway	VPN gateway for accessing on Premises systems
Storage, Managed Disk HDD	Persistency for temporary files that exceed the container file system limit of 2GB
Azure Monitor - Log Analytics	Storing and evaluating log files. At time of writing, 5 GB per month are for free.

## CONTACT

GK Software SE  
Waldstraße 7  
08261 Schöneck  
Germany

T +49 (0) 3 74 64 84 – 0

F +49 (0) 3 74 64 84 – 15

[documentation@gk-software.com](mailto:documentation@gk-software.com)

[www.gk-software.com](http://www.gk-software.com)