Devops and Support

A ‘SaaS in a Box’ Tutorial

# Purpose of this tutorial

This tutorial explores one way in which the catalog pattern can assist in devops and support scenarios. In the WTP app, the catalog is extended to store tenant names for each tenant. This allows the catalog to be searched by venue name to determine the server and database name for each tenant. This is the basis of the functionality exposed in the operations experience in the WTP app. In this tutorial, you will see how this can be used in PowerShell to first find a tenant, and then open that tenant’s database in the Azure portal for troubleshooting or management. From the database blade, it’s then easy to step to the pool or server blades and to use the monitoring and other tools provided to identify and mitigate performance or other issues.

# Using the catalog to support devops and customer support activities

Devops and support operations in a SaaS application need to span the fleet of tenant databases distributed across many pools and servers. The catalog, as it contains information about every tenant and database, can be used as the basis of tenant management solutions or integrated with other existing tenant management or CRM systems. To support additional processes centered on the catalog it can be extended to hold additional meta data and to manage the state of database administration tasks.

# Setup

Download and extract **WTPLearningModules.zip** to a convenient folder.

Deploy the **WTP Application.** Ensurethe catalog is initialized using the Demo Assistant app. See the Introduction to the WTP SaaS Application tutorial for deployment instructions.

**SSMS** can be used to explore database schema and execute SQL queries directly.

**PowerShell ISE** is recommended to execute scripts and follow their execution in debug mode.

**PowerShell Tips**

* Open and configure demo- scripts in the PowerShell ISE.
* Use F5 to run the script (using F8 is not advised as the $PSScriptRoot variable is not evaluated when running snippets of a script).
* Use F9 to set a breakpoint to let you trace the script in debug mode to see how it works
* Use F10 to step through the script, F11 to step into a function, and Shift-F11 to step out.

# Walkthrough

## Getting Started

This tutorial can be explored at any time, although its most interesting if the tenant batch has been deployed and there is a load running.

1. Update the user configuration file used by all tutorial scripts. Update again if you redeploy the app.
   1. Open ...\Learning Modules\UserConfig.psm1 in **PowerShell ISE**
   2. Modify **$userConfig.ResourceGroupName** to the resource group used for the deployed app.
   3. Modify **$userConfig.Name** to the User name used for the deployed app.
2. Open …\Learning Modules\Devops and Support\**Demo-DevopsAndSupport**.ps1
3. Modify **$DemoScenario to 1** to start a load running and execute using F5

## Exercise 1: Find a tenant by name and open its associated tenant database in the portal

This exercise highlights how easy it is to open the portal on a specific resource from an application using a deep link. The script uses a simple query to retrieve names of tenants in the catalog that match a search string. A selected tenant name is then used to search the catalog and retrieve that tenant’s database location information, which is then used to format the URL to open the portal.

1. **Execute using** **F5**
2. **Enter “jazz” as the search string**
3. **Select “Fabrikam Jazz Club”** to open the tenant’s database blade in the portal. The public events page for the tenant is also opened.

In a support scenario this would let you see if this tenant’s data is available and their page is responsive.

1. On the database blade in the portal, use the links on the Essentials drop down at the top of the blade to **navigate to the pool blade** for the database so you can see the effect of this database on others in the pool.