resource group: RG-ManualDeploy-os08

Security group: GDBC2018\_os08\_sg

user:g-objectsharp-user037@gbcd2018.onmicrosoft.com

#2

**Achievement**

In the Azure portal you will create a new Azure Web App with a simple App Service Plan. You will also create an Azure SQL Server that can later be used to contain the data of your Web application.

Using the Azure Portal you can connect to the SQL Database to create a table and add some data. Create a table called TestArtist containing an ID and names of artists. Attached you will find a script called create-artists.sql.

ACCEPTANCE CRITERIA

* When you browse to the Azure website, you will see the welcome screen of web apps where they explain how to deploy applications
* A SQL Database has been created containing 1 table TestArtist

Webapps: os08-web-asp

SQL DB: os08-web-sqldb

SQL server: os08-web-sqlserver.database.windows.net

Admin: admin08

Pswd: Icecream08

CREATE TABLE TestArtist ( Id int primary key, Name varchar(255) not null )

#3

The development team experimented with the manual creation of Resource Groups, but soon found that the manual steps are not the way to go. Everybody needs to be able to create a resource group and this need to be a predictable and reliable process. With the true DevOps Mindset they want to start automating everything.

### Achievement

In this achievement you will create an automation script to automatically create Resource Groups. You will also grant your security group the needed permissions to these groups. This should be done in an automated fashion (Azure CLI / Powershell or...) To get you familiar with Azure and the concept of Resource Groups / Security and Access you need to do the following things.

* Create an automation script to create a resource group for your team
  + RG-Playground-Teamname
* Create an automation script to add your AAD Security group to the resource groups
  + Security group (Teamname) – Contributor

**ACCEPTANCE CRITERIA**

* 1 resource groups is created in an automated fashion
* Security Group (Teamname) has Contributor access on the Resource Group. This has been granted in an automated fashion
* Users in the security group should be able to access the Resource Group and be able to create resources (e.g. a Web App)

<http://hadzimahmutovic.com/azure/azure-cli/cheatsheet/2018/03/14/azure-cli-cheat-sheet.html>

az group create --name RG-cli-os08 --location eastus

az ad group list --display-name GDBC2018\_os08\_sg <get object id of security group)

az role assignment create --role Contributor --assignee-object-id 4bb5175d-dfc0-407c-9ee8-e3c65c8481bc --resource-group RG-cli-os08

#4

The team created the required Azure resources to publish their website. But they want to create more environments that are exactly the same so they can use this for testing purposes. One of the benefits of the cloud is they can create new resources fast and discard them when not needed anymore. The team wants to be able to run an automation script that can create the previously created resources (resource group, Web application, SQL server)

### Achievement

In this achievement you will create a Web App and SQL Database in a fully automated fashion. The deployment of the web application is still done from Visual Studio, but at least the creation of the required resources are done automatically. Create a new resource group or use the existing resource groups from previous challenges.

* A script that can be run from the commandline that creates a Web Application and SQL Server **Note:** you should choose a complex password for the SQL Server admin password otherwise it fails setting the password without specific reason
* The Resources are named uniquely based on a provided parameter in the script "Environment\_name". For example when running the script with the parameter "Dev", the Web App is called WebApp-TeamName-Dev.
* After running the script the resources are available in Azure
* The SQL Server firewall settings allows access to Azure services

az appservice plan create -g "RG-cli-os08" -n web-asp-os08

az webapp create -g RG-cli-os08 -p web-asp-os08 -n webapp-os08-dev

az sql server create --admin-password Icecream08 --admin-user admin08 --location EastUS --name sql-server-os08-dev --resource-group RG-cli-os08

az sql server firewall-rule create --start-ip-address "0.0.0.0" --end-ip-address "0.0.0.0" --name sql-server-os08-dev-firewall --resource-group RG-cli-os08 --server sql-server-os08-dev

az sql db create -g RG-cli-os08 -s sql-server-os08-dev -n sql-db-os08-dev --service-objective S0

#5

Now that automation scripts are created, the move to DevOps has really started. The automation scripts are available to be used, but we want to keep our sources safe, track the changes we made and be able to use the scripts in our build and release pipeline. This way the team can ensure a reliable and stable way of rolling out new environments on demand. The team wants to use Git as the version control repository and they want to keep the automation separate from the application sources. They create a new Git Repo in the same team project, so the sources are kept separate.

### Achievement

In this achievement you will create a new Git Repository in your VSTS Team project that holds all the automation scripts and Infrastructure as Code files. Call this Git Repo Team-IaC.

**ACCEPTANCE CRITERIA**

* A new Git Repository that only contains the automation scripts to create the required Azure resources
* Git Repository contains the automation scripts for Resource Group creation and Resources creation
* Team members are able to clone the Git Repo
* Team members can make changes to the Git Repo

Git-repo-os08