# Suscripción Nueva en Azure

Cloud Computing

Table of Contents

[Suscripción Nueva en Azure 1](#_Toc85097351)

[Revisiones 3](#_Toc85097352)

[Pasos 4](#_Toc85097353)

[Grupo Arquitectura (tareas) 4](#_Toc85097354)

[Seguridad (tareas) 5](#_Toc85097355)

Revisiones

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| Sept/20/2021 | 1.0 | Initial version | Enterprise Architecture |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Pasos

## Grupo Arquitectura (tareas)

1. **Usar un Private Session del browser** y entrar a suscripción en Partner Portal como usuario llc
2. Ir a Cost and Managements, escoger opción de Cost Management a la izq.
3. **Verificar que el scope diga customer for commerce root y termina en 4bbd-8d4f-883718c2bf9b para Evertec, si no es así cambia el scope a Evertec Inc., no Evertec LLC, este debe ser el scope:**
4. **Graphical user interface, text, application

   Description automatically generated**
5. Buscar Azure subscriptions en la izq. Y dar clic en +Add
6. Crear suscripción bajo EvertecLLC. Si esto no te trae la siguiente pantalla entonces debes ir a través de un Search en Subscription y de allí darle +Add
7. Application, table

   Description automatically generated
8. Crear Automation Account aut-[account name]-runbook1 en un RG nuevo rg-[account name-autoacautomation[##], es necesario hacerlo con la cuenta llc para crear el Run as Account
9. Dar logout y volver a entrar como llc, esta vez escoger la subs nueva
10. Crear usuario (tu usuario como Owner, crear a Lila y Melvin) con el llc
11. En el Azure Portal ir a la suscripción y subir el Azure PowerShell, crear el cloudshell RG y StorageAccount si pide su creación
12. Desde VSCode correr el script de registrar providers a nivel de la suscripción, verificar que el de Blueprints este registrado
13. Instalar los módulos de Azure en Automation Account (az.accounts, az.compute, az.automation, az.storage, az.resources, az.profile)
14. NEW - Correr el script SetTagsToAllCurrentResources desde VSCode para crear los runbooks, publicar los runbooks, crear los schedules y atachar los schedules a los runbooks
15. Pegar el código de los runbooks (PowerShellScript, ver [anejos](#_Anejos)). **Editar los valores de las variables desde el editor del portal de Azure no desde este documento**
16. FIN
17. Crear los Runbooks (actualizar los parámetros de los Tags y el Subscription ID de los PowerShells)
18. Publicar los runbooks llamados CreateTagsToAllResourceGroups y CreateBackupTagtoAllVm
19. Crearle un Schedule a los runbooks llamados CreateTagsToAllResourceGroups y CreateBackupTagtoAllVm, usar como nombre para el schedule:
20. sch-[account name] -[RG/BKP]tag1, recurrente, corre cada día, de madrugada, horario de PR
21. Logout del portal y entrar con tu usuario
22. Modificar el PowerShell SetTagtoAllCurrentResources.ps1, cambiar la información de los Tags usando la información del runbook CreateTagtoAllResources y correrlo en Visual Studio Code

Falta solo el crear la cuenta en Cloudcheckr una vez aparezca en el portal

## Seguridad (tareas)

1. Crear Management Groups bajo EvertecLLC y mover la suscripción al Management Groups nuevo
2. Eliminar los Owners que no sean de Seguridad de Información
3. Correr los blueprints CommonPolicies, CommonResourceGroups, CreateEventHubName, CreateLogAnalyticsWorkspace, CreateStorageAccountforEventHub
4. Crear los Custom Roles ASG Team Monitoring [account name], ASGNetworkWatcher [account name], Network Watcher Managements [account name], Stop Restart Start VM [account name], Virtual Machine Update Manager [account name] (en la cuenta plot crt se unificaron en un solo role Stop Restart Start VM+Virtual Machine Update Manager), hay otros customs que fueron creados en varias suscripciones
5. Asignar los permisos a los grupos de cada custom role a nivel de la suscripción

# Anejos

Script de tag de resucursos para Runbook CreateTagtoAllResourceGroups:

############################################################################

# Walter Gonzalez #

# Mayo 2020 #

# #

############################################################################

# Update Tags - Applied to all the resources in a subscription

############################################################################

# CHANGE THIS PER SUBSCRIPTION #

############################################################################

# RunBook for Update Tags - Applied at the RG level

#Parameters -----------------------------------------------------------------------------------------------------

$mySubscription = " TBD "

#Code of the application the resource is associated with in the CMDB.

$applicationCode = " TBD "

#Name of the application, service, or workload the resource is associated with in the CMDB

$applicationName = " TBD "

#Name of the module or component associated with the application.

$applicationModule = " TBD "

#Evertec group responsible for the implementation of the project

$group = "Digital Solutions"

#Owner of the application, workload, or service.(email)

$ownerName = " TBD "

#Accounting cost center associated with this resource

$costCenter = " TBD "

#Deployment environment of this application, workload, or service

$env = " TBD "

#Person responsible for approving costs related to this resource.(email)

$approver = " TBD "

#Top-level division of your company that owns the workload the resource belongs to.

#In smaller organizations, this may represent a single corporate or shared top-level organizational element

$businessUnit = " TBD "

#User that requested the creation of this application.(email)

$requestor = " TBD "

#Service Level Agreement level of this application, workload, or service

$serviceClass = " TBD "

#Date when this application, workload, or service was first deployed

$startDate = " TBD "

#End of parameters----------------------------------------------------------------------------------------------------

#Start

# Ensures you do not inherit an AzContext in your runbook

Disable-AzContextAutosave –Scope Process

#Authenticate using the account configured when the automation account was created

$connection = Get-AutomationConnection -Name AzureRunAsConnection

while(!($connectionResult) -And ($logonAttempt -le 10))

{

$LogonAttempt++

# Logging in to Azure...

$connectionResult = Connect-AzAccount `

-ServicePrincipal `

-Tenant $connection.TenantID `

-ApplicationId $connection.ApplicationID `

-CertificateThumbprint $connection.CertificateThumbprint

Start-Sleep -Seconds 30

}

Select-AzSubscription -SubscriptionId $mySubscription

#Update tags for all services

$allResourcesGroup = Get-AzResourceGroup

$tags = @{"applicationCode" = $applicationCode; "applicationName" = $applicationName;"applicationModule" = $applicationModule; "group" = $group; "ownerName" = $ownerName;"costCenter" = $costCenter; "env" = $env; "approver"= $approver; "businessUnit"= $businessUnit; "requestor" = $requestor; "serviceClass" = $serviceClass; "startDate" = $startDate}

foreach($resourceList in $allResourcesGroup){

write-output $resourceList.ResourceId

write-output $resourceList.Location

write-output $resourceList.Tags

if ($resourceList.Tags.applicationCode -eq $null){

$resourceGroup = Get-AzResourceGroup -Name $resourceList.ResourceGroupName

Update-AzTag -ResourceId $resourceGroup.ResourceId -tag $tags -Operation Merge

}

}

Script de etiquetas de recursos para Runbook CreateBackupTagtoAllResources:

############################################################################

# Walter Gonzalez #

# Junio 2020 #

# #

############################################################################

#Just for VMs

# Update Tags - Applied to all the VMs in a subscription

############################################################################

# CHANGE THIS PER SUBSCRIPTION #

############################################################################

$mySubscription = "TBA"

# RunBook for Update Tags - Applied at the Subscription level

#Parameters -----------------------------------------------------------------------------------------------------

#The resource needs a backup plan? The value will be set in the VSP or the Backup Team will set the value in

#accordance with the owner

$backup = ""

#End of parameters----------------------------------------------------------------------------------------------------

function doLogin{

#Start

# Ensures you do not inherit an AzContext in your runbook

Disable-AzContextAutosave –Scope Process

#Authenticate using the account configured when the automation account was created

$connection = Get-AutomationConnection -Name AzureRunAsConnection

while(!($connectionResult) -And ($logonAttempt -le 10))

{

$LogonAttempt++

# Logging in to Azure...

$connectionResult = Connect-AzAccount `

-ServicePrincipal `

-Tenant $connection.TenantID `

-ApplicationId $connection.ApplicationID `

-CertificateThumbprint $connection.CertificateThumbprint

Start-Sleep -Seconds 30

}

}# end login func

function doTagging { param( [string]$myTag )

Select-AzSubscription -SubscriptionId $mySubscription

#Update tags for all services

$allResources = Get-AzVM | Select Id,Name,Tags

$tags = @{"backup" = $myTag}

foreach($resourceList in $allResources){

write-output $resourceList.id

write-output $resourceList.Name

write-output $resourceList.tags

if ($resourceList.Tags.backup -eq $null){

Update-AzTag -ResourceId $resourceList.id -Tag $tags -Operation Merge

write-output $resourceList.tags

}

}

}#end doTagging func

#Func calls

doLogin

doTagging $backup