

GLADINET, INC

Gladinet Cloud Enterprise

PowerShell Guide

Gladinet, Inc.

7/31/2014

This document discusses the technologies behind Gladinet Cloud Enterprise

Copyright 2014 Gladinet, Inc.

Table of Contents

PowerShell Module Overview	19
Loading the PowerShell Module	19
Get-GceLogin	20
Synopsis.....	20
Syntax	20
Description	20
Parameters.....	20
-Username <String>	20
-Password <String>	21
-ServerUrl <String>	21
<CommonParameters>	22
Inputs	22
Outputs	22
Notes	22
Examples	22
Related links	23
Get-GceTenants.....	23
Synopsis	25
Syntax	25
Description	25
Parameters.....	25
-LoginToken <String>	25
-ServerUrl <String>	25
<CommonParameters>	26
Inputs	26
Outputs	26

Notes	26
Examples	26
Example 1	26
Related links	27
Get-GceTenantUsers	28
Synopsis	29
Syntax	29
Description	29
Parameters.....	29
-TenantGuid <String>	29
-LoginToken <String>	30
-ServerUrl <String>	30
<CommonParameters>	31
Inputs	31
Outputs	31
Notes	31
Examples	31
Example 1	31
Related links	33
Get-GceTenantInfo	33
Synopsis	34
Syntax	34
Description	34
Parameters.....	34
-TenantGuid <String>	34
-LoginToken <String>	35

-ServerUrl <String>	35
<CommonParameters>	36
Inputs	36
Outputs	36
Notes	36
Examples	36
Example 1	36
Related links	37
Get-GceUserInfo.....	37
Synopsis	38
Syntax	38
Description	38
Parameters.....	38
-UserGuid <String>	38
-LoginToken <String>	39
-ServerUrl <String>	39
<CommonParameters>	40
Inputs	40
Outputs	40
Notes	40
Examples	40
Example 1	40
Related links	40
Get-GceUserInfoByEmail.....	41
Synopsis	42
Syntax	42

Description	42
Parameters.....	42
-Email <String>	42
-LoginToken <String>	43
-ServerUrl <String>	43
<CommonParameters>	43
Inputs	44
Outputs	44
Notes	44
Examples	44
Example 1	44
Related links	44
Set-GceUserName	44
Synopsis	45
Syntax	45
Description	45
Parameters.....	45
-UserGuid <String>	45
-FirstName <String>	45
-LastName <String>	46
-LoginToken <String>	46
-ServerUrl <String>	46
<CommonParameters>	47
Inputs	47
Outputs	47
Notes	47

Examples	47
Example 1	47
Related links	48
Set-GceTenantUserPlan	48
Synopsis	48
Syntax	48
Description	48
Parameters.....	49
-TenantGuid < <i>String</i> >	49
-UserPlan < <i>Int32</i> >	49
-LoginToken < <i>String</i> >	49
-ServerUrl < <i>String</i> >	50
<CommonParameters>	50
Inputs	50
Outputs	50
Notes	51
Examples	51
Example 1	51
Related links	51
Set-GceTenantStoragePlan	51
Synopsis	52
Syntax	52
Description	52
Parameters.....	52
-TenantGuid < <i>String</i> >	52
-StoragePlanInGB < <i>Int32</i> >	52

-LoginToken < <i>String</i> >	53
-ServerUrl < <i>String</i> >	53
<CommonParameters>	53
Inputs	54
Outputs	54
Notes	54
Examples	54
Example 1	54
Related links	54
Set-GceTenantAdminEmail	54
Synopsis	55
Syntax	55
Description	55
Parameters.....	55
-TenantGuid < <i>String</i> >	55
-AdminEmail < <i>String</i> >	56
>LoginToken < <i>String</i> >	56
-ServerUrl < <i>String</i> >	56
<CommonParameters>	57
Inputs	57
Outputs	57
Notes	57
Examples	57
Example 1	57
Related links	57
New-GceUser.....	58

Synopsis	59
Syntax	59
Description	59
Parameters.....	59
-TenantGuid <String>	59
-FirstName <String>	59
-LastName <String>	60
-Email <String>	60
-Password <String>	61
-StorageQuotaInGB <Int64>	61
-CreateDefaultDirectory <Boolean>	61
-ForceVersionedRootFolder <Boolean>	62
-EnableOfflineFromRoot <Boolean>	62
-SendWelcomeEmail <Boolean>	62
-LoginToken <String>	63
-ServerUrl <String>	63
<CommonParameters>	64
Inputs	64
Outputs	64
Notes	64
Examples	64
Example 1	64
Related links	65
Reset-GceUserPassword	65
Synopsis	65
Syntax	65

Description	66
Parameters.....	66
-UserEmail <String>	66
-Password <String>	66
-LoginToken <String>	66
-ServerUrl <String>	67
<CommonParameters>	67
Inputs	67
Outputs	67
Notes	68
Examples	68
Example 1	68
Related links	68
Remove-GceUser	68
Synopsis	69
Syntax	69
Description	69
Parameters.....	69
-UserGuid <String>	69
-PurgeStorage <Boolean>	69
-LoginToken <String>	70
-ServerUrl <String>	70
<CommonParameters>	70
Inputs	71
Outputs	71
Notes	71

Examples	71
Example 1	71
Related links	71
Remove-GceTenant	71
Synopsis	72
Syntax	72
Description	72
Parameters.....	72
-TenantGuid < <i>String</i> >	72
-LoginToken < <i>String</i> >	73
-ServerUrl < <i>String</i> >	73
<CommonParameters>	73
Inputs	74
Outputs	74
Notes	74
Examples	74
Example 1	74
Related links	74
Set-GceTenantAdminRights	74
Set-GceTenantAdminRightsEx	75
Set-GceTenantExpirationTime	76
Set-GceTenantStorageConfig	76
New-GceTenant.....	77
Synopsis	78
Syntax	78
Description	78

Parameters.....	79
-OrgName <String>	79
-FirstName <String>	79
-LastName <String>	79
-AdminEmail <String>	80
-AdminPassword <String>	80
-UserPlan <Int32>	80
-StoragePlan <Int32>	81
-ExpirationTime <String>	81
-StorageConfigure <String>	82
-LoginToken <String>	82
-ServerUrl <String>	83
<CommonParameters>	83
Inputs	83
Outputs	83
Notes	83
Examples	83
Example 1	83
Related links	84
Add-GceAdUserByUPN	84
Add-GceAdUserByUPNEx	85
Synopsis	85
Syntax	85
Description	86
Parameters.....	86
-UserUPN <String>	86

-StoragePlan <Int32>	86
-ForceVersioned <Boolean>	87
-EnableOffline <Boolean>	87
-StorageConfigure <String>	87
-LoginToken <String>	88
-ServerUrl <String>	88
<CommonParameters>	88
Inputs	89
Outputs	89
Notes	89
Examples	89
Example 1	89
Related links	90
Set-GceUserStoragePlanGB.....	90
Synopsis	91
Syntax	91
Description	91
Parameters.....	91
-UserID <String>	91
-GBCount <Int64>	91
-LoginToken <String>	92
-ServerUrl <String>	92
<CommonParameters>	92
Inputs	93
Outputs	93
Notes	93

Examples	93
Example 1	93
Related links	93
Set-GceUserStorageConfig	93
Synopsis	94
Syntax	94
Description	94
Parameters.....	94
-UserID <String>	94
-StorageConfigure <String>	95
-LoginToken <String>	95
-ServerUrl <String>	95
<CommonParameters>	96
Inputs	96
Outputs	96
Notes	96
Examples	96
Related links	96
Set-GceEnableDARE	96
Synopsis	97
Syntax	97
Description	97
Parameters.....	97
-Password <String>	97
-LoginToken <String>	97
-ServerUrl <String>	98

<CommonParameters>	98
Inputs	98
Outputs	98
Notes	99
Examples	99
Example 1	99
Related links	99
Set-GceDisableDARE.....	99
Synopsis	99
Syntax.....	100
Description	100
Parameters.....	100
-LoginToken <String>	100
-ServerUrl <String>	100
<CommonParameters>	101
Inputs	101
Outputs	101
Notes	101
Examples	101
Example 1	101
Related links	101
Set-GceEnableTenant.....	101
Synopsis	102
Syntax.....	102
Description	102
Parameters.....	102

-TenantId <String>	102
-LoginToken <String>	103
-ServerUrl <String>	103
<CommonParameters>	103
Inputs	104
Outputs	104
Notes	104
Examples	104
Example 1	104
Related links	104
Set-GceDisableTenant	104
Synopsis	105
Syntax	105
Description	105
Parameters.....	105
-TenantId <String>	105
-LoginToken <String>	105
-ServerUrl <String>	106
<CommonParameters>	106
Inputs	106
Outputs	106
Notes	107
Examples	107
Example 1	107
Related links	107
Set-GceUserAssignedZone	107

Set-GceEnableUser	107
Synopsis	108
Syntax	108
Description	108
Parameters.....	108
-UserID <String>	108
-LoginToken <String>	109
-ServerUrl <String>	109
<CommonParameters>	109
Inputs	110
Outputs	110
Notes	110
Examples	110
Example 1	110
Related links	110
Set-GceDisableUser	110
Synopsis	111
Syntax	111
Description	111
Parameters.....	111
-UserID <String>	111
-LoginToken <String>	111
-ServerUrl <String>	112
<CommonParameters>	112
Inputs	112
Outputs	112

Notes	113
Examples	113
Example 1	113
Related links	113
Get-GceActiveUser	113
Synopsis	113
Syntax	114
Description	114
Parameters.....	114
-DomainID < <i>String</i> >	114
-LoginToken < <i>String</i> >	114
-ServerUrl < <i>String</i> >	115
<CommonParameters>	115
Inputs	115
Outputs	115
Notes	115
Examples	115
Example 1	115
Related links	116
Get-GceNodePerformance	116
Synopsis	116
Syntax	116
Description	117
Parameters.....	117
-LoginToken < <i>String</i> >	117
-ServerUrl < <i>String</i> >	117

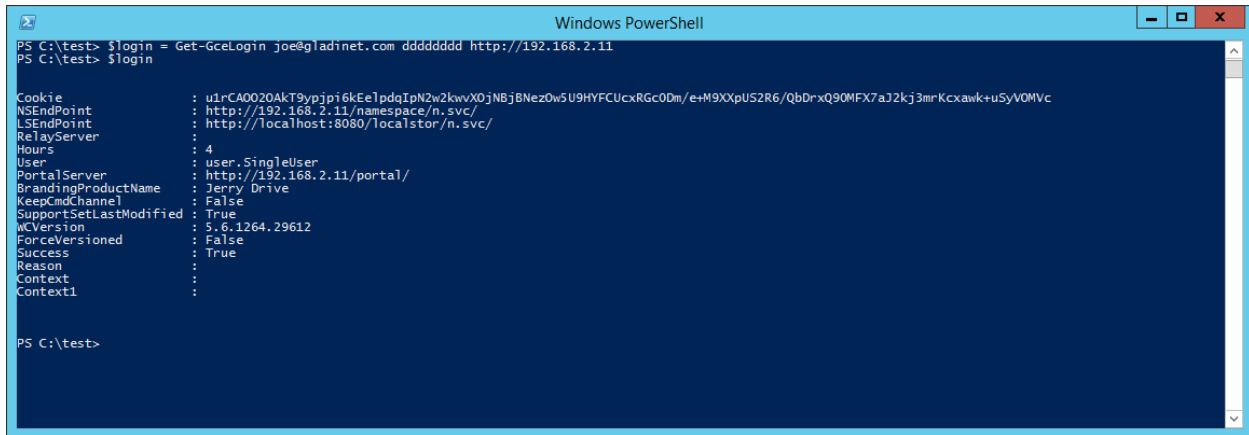
<CommonParameters>	117
Inputs	118
Outputs	118
Notes	118
Examples	118
Example 1	118
Related links	118
Add-GceWorkerNode	119
Get-GceLocalStorageConfig	119
Get-GceAmazonStorageConfig.....	119
Get-GceOpenStackStorageConfig	120

Get-GceLogin

It takes 3 parameters.

1. First parameter is the username,
2. Second parameter is the password
3. Third parameter is the Server Url.

When the login returns, you can inspect the result. When the Success field is True, you can save away the result and the Cookie field will be used later.



```
Windows PowerShell
PS C:\test> $login = Get-GceLogin joe@gladinet.com dddddd http://192.168.2.11
PS C:\test> $login

Cookie           : u1rCA0020AkT9ypj6kEe1pdaIpN2w2kwvX0jNBj8Nez0w5U9HYFCUcxRgc0Dm/eeM9XXpU52R6/QbDrxQ90MF7aJ2kj3mrKcxawk+uSyV0MvC
NSEndPoint       : http://192.168.2.11/namespace/n.svc/
LSEndPoint       : http://localhost:8080/localstor/n.svc/
RelayServer      :
Hours            : 4
User             : user.SingleUser
PortalServer     : http://192.168.2.11/portal/
BrandingProductName : Jerry Drive
KeepCmdChannel   : False
SupportSetLastModified : True
WCVersion        : 5.6.1264.29612
ForceVersioned   : False
Success          : True
Reason           :
Context         :
Context1        :
```

Synopsis

Login to the system and get back security token.

Syntax

```
Get-GceLogin [-UserName] <String> [-Password] <String> [[-ServerUrl]
<String>] [<CommonParameters>]
```

Description

Most of the other functions in the PowerShell CmdLet require a security token from the user. This function login to the system and returns the user's security token.

Parameters

-UserName <String>

Tenant user name for login. Most of the time it is email address.

Required?	True
Position?	0
Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

-Password <String>

Tenant user password

Required?	True
Position?	1
Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

-ServerUrl <String>

Remote server URL. Such as "http://192.168.0.1"

Required?	False
Position?	2
Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

<CommonParameters>

This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see about_CommonParameters (<http://go.microsoft.com/fwlink/?LinkID=113216>).

Inputs

Outputs

LoginResult

The result contains user cookie which can be used for other functions later.

Notes

Examples

Example 1 - Login to localhost

```
PS C:\> $login=Get-GceLogin joe@gladinet.com dd http://localhost
```

```
PS D:\> $login
```

```
Cookie                :  
kMJQR9/NrW+Ujff70i1YG+fzUZJNsmh9JlrMgHt2G1fdrGUX7I4P/NPbH/0cJ0xzz9OhQKfr8  
QfQZRUGFgQypr/x8Mm+HcsXhMd67VPi16fFj70TQ  
NSEndPoint           : http://jerryhp9/namespace/n.svc/  
LSEndPoint           : http://localhost:8080/localstor/n.svc/  
RelayServer          : jerryhp9:8888  
Hours                : 4  
User                 : user.SingleUser  
PortalServer         : http://jerryhp9/portal/  
BrandingProductName  : Jerry Drive  
KeepCmdChannel       : False  
SupportSetLastModified : True  
WCVersion            :
```

```
ForceVersioned      : False
Success             : False
Reason              :
Context             :
Context1            :
```

Related links

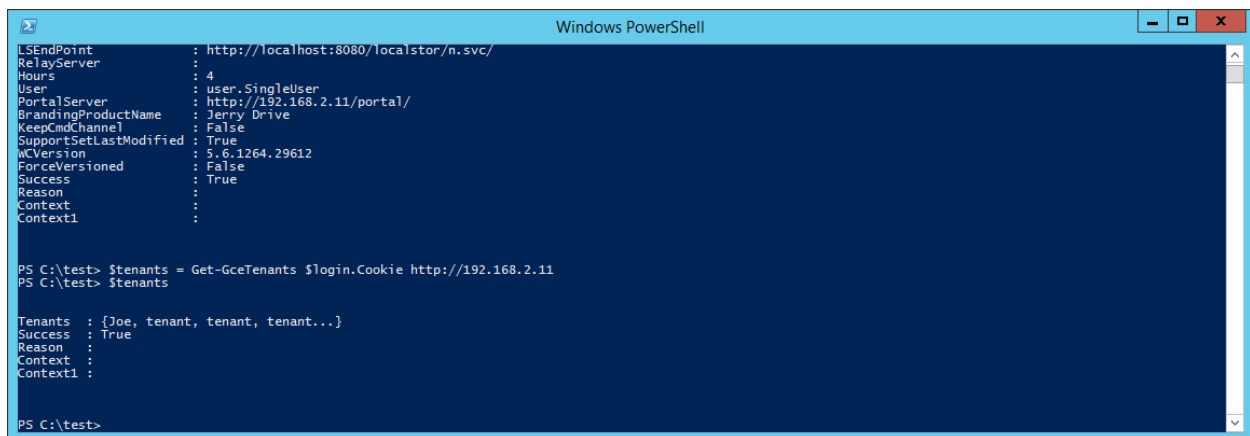
Get-GceTenants

Gets a list of tenants in the system.

It takes two parameters.

1. First Parameter is the LoginToken.
2. Second Parameter is the ServerUrl

When it is successful, the Success field is True. The Tenants field is a list of Tenants.



```
Windows PowerShell
LSEndPoint          : http://localhost:8080/localstor/n.svc/
RelayServer         :
Hours              : 4
User               : user.SingleUser
PortalServer        : http://192.168.2.11/portal/
BrandingProductName : Jerry Drive
KeepCmdChannel      : False
SupportSetLastModified : True
WCVersion           : 5.6.1264.29612
ForceVersioned      : False
Success            : True
Reason             :
Context            :
Context1           :

PS C:\test> $tenants = Get-GceTenants $login.Cookie http://192.168.2.11
PS C:\test> $tenants

Tenants : {Joe, tenant, tenant, tenant...}
Success : True
Reason  :
Context :
Context1 :
```

You can further inspect the tenants by looking into the Tenants field.

```
Windows PowerShell
Success : True
Reason :
Context :
Context1 :

PS C:\test> $tenants = Get-GceTenants $login.Cookie http://192.168.2.11
PS C:\test> $tenants

Tenants : {Joe, tenant, tenant, tenant...}
Success : True
Reason :
Context :
Context1 :

PS C:\test> $tenants.Tenants | Format-List

DomainId      : 2a3f49b8-a91e-4e66-a165-f5db1ca56cfe
AdminId       : 0961d650-d280-49f5-8822-d933b3f63801
ResellerId    :
FirstName     : Joe
LastName      : Admin
Email         : joe@gladinet.com
IsActivated   : True
IsDeleted     : False
```

You can also sent the Tenants to a GridView.

The screenshot shows a GridView window titled '\$tenants.Tenants | Out-GridView'. The table contains the following data:

DomainId	AdminId	ResellerId	FirstName	LastName	Email	IsActivated	IsDeleted	Quota	QuotaUsed
2a3f49b8-a91e-4e66-a165-f5db1ca56cfe	0961d650-d280-49f5-8822-d933b3f63801		Joe	Admin	joe@gladinet.com	True	False	0	66,737,520
efbe612c-fde0-43fc-8c7d-7c56ce2a73da	bca7c86d-5571-42e4-97db-1a5ed0482f56		tenant	1	tenant1@gladinet.com	True	False	1,024	221,379
4351d253-0ad8-4b56-8850-3387b3c8127a	29e46765-9927-414c-9ca3-365e028cd220		tenant	2	tenant2@gladinet.com	True	False	1,024	0
45913cde-2396-4fb1-a28c-1be9df33ae61	30bb0e50-7a76-442d-8ca6-fac5475b51d4		tenant	3	tenant3@gladinet.com	True	False	1,024	0
78ef6a36-e0e8-4c05-a672-9b030fc32f56	fed61cee-0cf3-40ea-a224-7f07fc6f24f5		tenant	4	tenant4@gladinet.com	True	False	500	0
4248e07f-600a-429c-bfc2-d6cf7a2ab4e2	a6307946-934a-4c8c-a20b-9a4a52ed0ec8		amazon	user	amazon1@gladinet.com	True	False	100	0
e4f6053a-bca6-4611-91a3-3293b7ac1eeb	0a3e61c7-ff0c-43db-96f6-b663c261f537		rackspace	1	rack1@gladinet.com	True	False	100	0
93f08cc0-9b66-4267-9b4d-3097a8406b98	3d20ad50-e253-4b3b-9f59-b06a6a838e7d		D1	D1	d1@gladinet.com	True	False	10	82,175

Below the GridView, a PowerShell window shows the detailed properties for the selected tenant (DomainId: 4248e07f-600a-429c-bfc2-d6cf7a2ab4e2):

```
ResellerId      :
FirstName       : D1
LastName        : D1
Email           : d1@gladinet.com
IsActivated     : True
IsDeleted       : False
Quota           : 10
QuotaUsed       : 82175
UserPlan        : 10
UserPlanUsed    : 0
PaypalProfileId :
StorageProviderId : Default
Edition         :
Expire          : 01/01/2015 01:00:00 AM
ResellerName    :
PromoCode       :
LastCheck       :
TeamStatus      :
LastQuotaScan   :
AccountBi       :
Organization     :
IsBiz           : False
IsPersonal      : False
IsCloudStorage  : True
ForceScan       : False

PS C:\test> $tenants.Tenants | Out-GridView
PS C:\test>
```

When you get the Tenants back, pay special attention to the DomainId field:

DomainId : 4248e07f-600a-429c-bfc2-d6cf7a2ab4e2

This will be the TenantID for each tenant and it is different from tenant to tenant.

Synopsis

Gets a list of all tenants in the system.

Syntax

```
Get-GceTenants [-LoginToken] <String> [[-ServerUrl] <String>]  
[<CommonParameters>]
```

Description

Gets a list of all tenants in the system. The caller shall be the master admin who has enough privilege to query the list of tenants.

Parameters

-LoginToken <String>

Logged in user's security token, which is returned as the Cookie field from Get-GceLogin call.

Required?	True
Position?	0
Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

-ServerUrl <String>

The remote service URL.

Required?	False
Position?	1

Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

<CommonParameters>

This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see about_CommonParameters (<http://go.microsoft.com/fwlink/?LinkID=113216>).

Inputs

Outputs

EnumTenantsResult

When it is successful, the Success field is true. The tenants field contains a list of all tenants in system.

Notes

The caller's security token will need to have privilege to list tenants.

Examples

Example 1

```
PS C:\> PS D:\> $tenants = Get-GceTenants $login.Cookie http://localhost
```

```
PS D:\> $tenants
```

```
Tenants : {joe'cc, Jim, adam, c...}  
Success : True  
Reason  :  
Context :  
Context1:
```

```
PS D:\> $tenants.Tenants[0]
```

```
DomainId           : b633d35a-e8d2-4106-88eb-cef6a67216d6
AdminId            : df2383d0-ee8f-4cf4-8575-ab7330bd73b1
ResellerId         :
FirstName          : joe'cc
LastName           : admin'dd
Email              : joe@gladinet.com
IsActivated        : True
IsDeleted          : False
Quota              : 0
QuotaUsed          : 908226177
UserPlan           : 33
UserPlanUsed       : 0
PaypalProfileId    :
StorageProviderId : Default
Edition            :
Expire             :
ResellerName       :
PromoCode          :
LastCheck          :
TeamStatus         :
LastQuotaScan      :
AccountBi          :
Orgnization        :
IsBiz              : False
IsPersonal         : False
IsCloudStorage     : True
ForceScan          : False
```

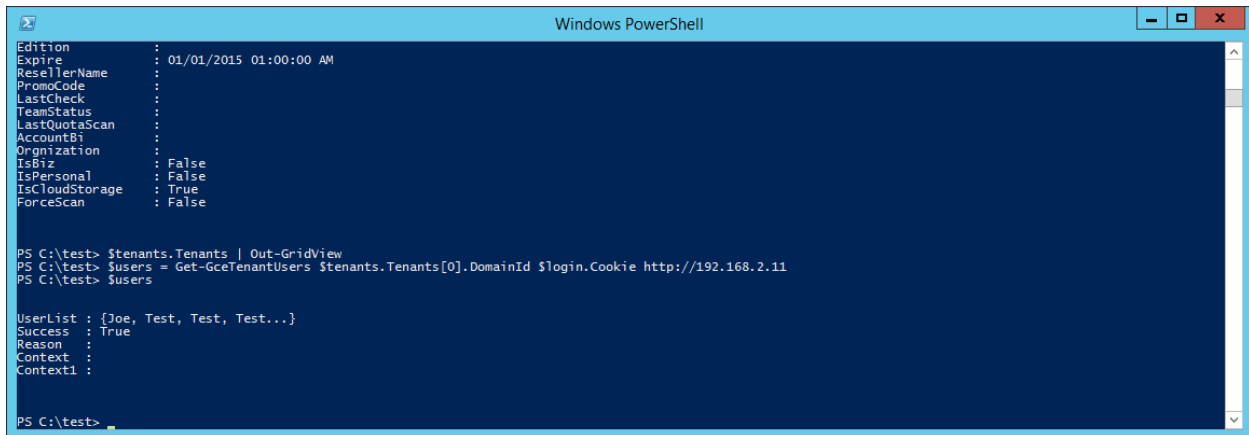
Related links

Get-GceTenantUsers

This function will get all the users in a specific tenant.

It takes 3 parameters.

1. First Parameter is TenantGuid (which is the DomainId for each tenant)
2. Second Parameter is LoginToken (represent the user who is issuing the call)
3. Third Parameter is the ServerUrl (which GCE server to call)



```
Windows PowerShell
Edition           : 
Expire            : 01/01/2015 01:00:00 AM
ResellerName     : 
PromoCode        : 
LastCheck        : 
TeamStatus       : 
LastQuotaScan    : 
AccountBi        : 
Organization     : 
IsBiz            : False
IsPersona        : False
IsCloudStorage   : True
ForceScan        : False

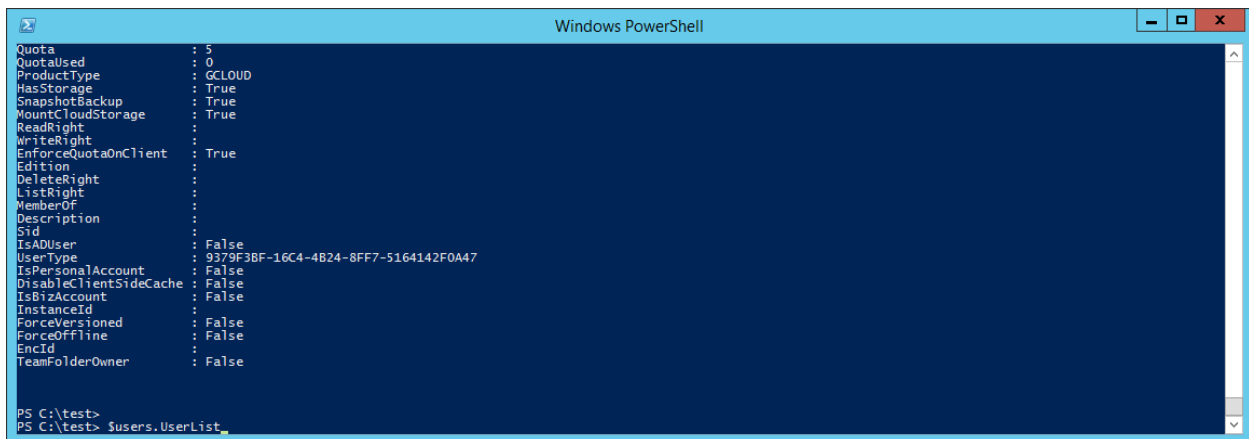
PS C:\test> $tenants.Tenants | Out-GridView
PS C:\test> $users = Get-GceTenantUsers $tenants.Tenants[0].DomainId $login.Cookie http://192.168.2.11
PS C:\test> $users

UserList : {Joe, Test, Test, Test...}
Success  : True
Reason   : 
Context  : 
Context1 : 

PS C:\test>
```

When it is successful, the Success field is True. The UserList contains the list of users.

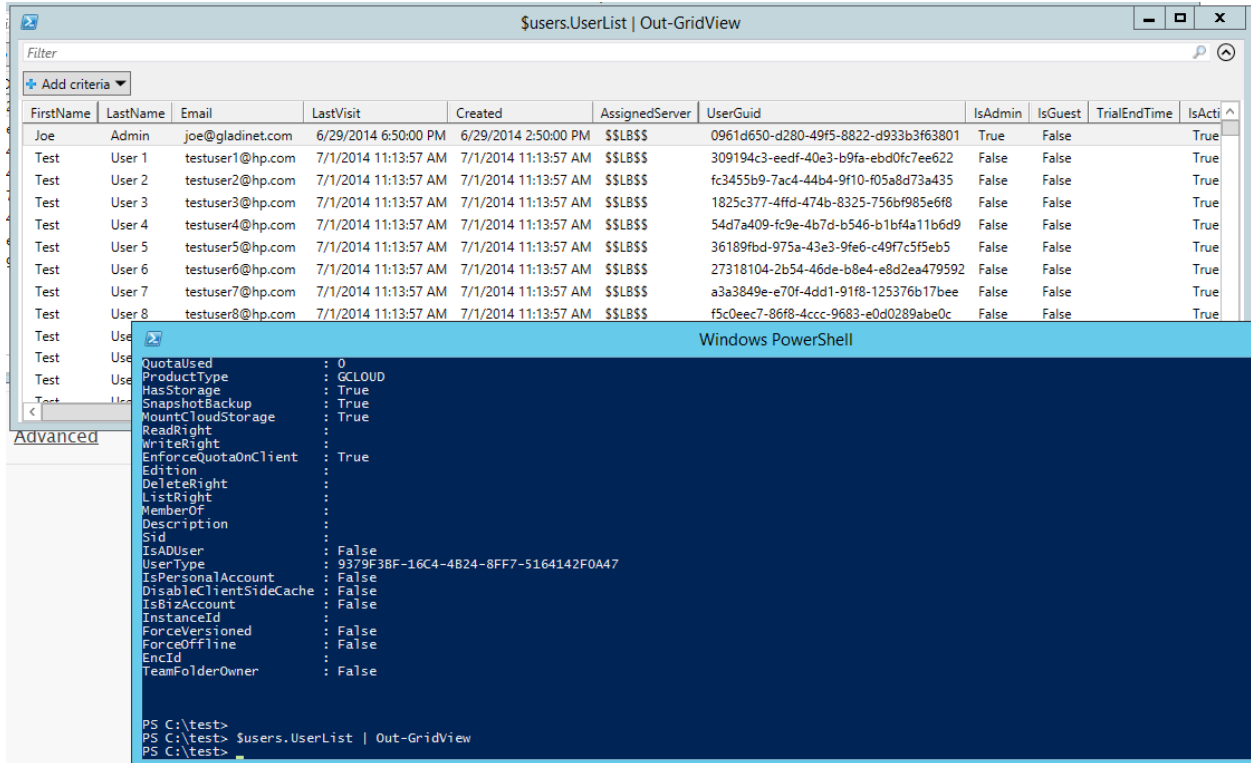
We can inspect the users.



```
Windows PowerShell
Quota            : 5
QuotaUsed        : 0
ProductType     : G-CLOUD
HasStorage       : True
SnapshotBackup   : True
MountCloudStorage : True
ReadRight        : 
WriteRight       : 
EnforceQuotaOnClient : True
Edition          : 
DeleteRight      : 
ListRight        : 
MemberOf         : 
Description      : 
Sid              : 
IsADUser         : False
UserType         : 9379F3BF-16C4-4B24-8FF7-5164142F0A47
IsPersonalAccount : False
DisableClientSideCache : False
IsBizAccount     : False
InstanceId      : 
ForceVersioned   : False
ForceOffline     : False
EncId           : 
TeamFolderOwner : False

PS C:\test>
PS C:\test> $users.UserList
```

We can also sent the list to a GridView



Synopsis

Return a list of users for a specific tenant.

Syntax

```
Get-GceTenantUsers [-TenantGuid] <String> [-LoginToken] <String> [[-
ServerUrl] <String>] [<CommonParameters>]
```

Description

Return a list of all users for a specific tenant. The tenant is specified as the GUID for the tenant. The GUID is the DomainId field for each tenant when you do a Get-GceTenants

Parameters

-TenantGuid <String>

Tenant GUID (DomainId)

Required?	True
Position?	0
Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

-LoginToken <String>

Logged in user's security token, which is returned as the Cookie field from Get-GceLogin call.

Required?	True
Position?	1
Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

-ServerUrl <String>

The remote service URL.

Required?	False
Position?	2
Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

<CommonParameters>

This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see about_CommonParameters (<http://go.microsoft.com/fwlink/?LinkID=113216>).

Inputs

Outputs

EnumUserResult

When it is successful, the Success field is true. The UserList field contains a list of all users in the tenant.

Notes

Examples

Example 1

```
PS C:\> PS D:\> $users = Get-GceTenantUsers "b633d35a-e8d2-4106-88eb-
cef6a67216d6" $login.Cookie http://localhost
PS D:\> $users
```

```
UserList : {joe'cc, joel, TestUser, TestUser...}
Success  : True
Reason   :
Context  :
Context1 :
```

```
PS D:\> $users.UserList[0]
```

```
FirstName      : joe'cc
LastName       : admin'dd
Email          : joe@gladinet.com
```

LastVisit : 11/10/2013 10:49:25 PM
Created : 11/10/2013 5:49:25 PM
AssignedServer : \$\$LB\$\$
UserGuid : df2383d0-ee8f-4cf4-8575-ab7330bd73b1
IsAdmin : True
IsGuest : False
TrialEndTime :
IsActivated : True
IsDeleted : False
DomainId : b633d35a-e8d2-4106-88eb-cef6a67216d6
IsRootAdmin : True
Quota : 0
QuotaUsed : 908226177
ProductType : GLOUD
HasStorage : True
SnapshotBackup : True
MountCloudStorage : True
ReadRight :
WriteRight :
EnforceQuotaOnClient : True
Edition :
DeleteRight :
ListRight :
MemberOf :
Description :
Sid :
IsADUser : False
UserType : 9379F3BF-16C4-4B24-8FF7-5164142F0A47
IsPersonalAccount : False
DisableClientSideCache : False
IsBizAccount : False
InstanceId :
ForceVersioned : False
ForceOffline : False
EncId :
TeamFolderOwner : False

Related links

Get-GceTenantInfo

This function will return a specific tenant information.

It takes 3 parameters.

1. First Parameter is TenantGuid (This is the DomainId for the Tenant)
2. Second Parameter is LoginToken (This is the Login Token represent the user making the call)
3. Third Parameter is ServerUrl (This is the GCE server)

In the following example, we saved away the \$login.Cookie to \$cookie and saved GCE server url to \$url

```
PS C:\test> $tenant = Get-GceTenantInfo $tenantID $cookie $url
PS C:\test> $tenant

Tenants : {Mike}
Success : True
Reason  :
Context :
Context1 :

PS C:\test> |
```

When it is successful, the Success field is True. The Tenants contains the only tenant in the list.

We can inspect the tenant:

```
PS C:\test> $tenant.Tenants

DomainId       : 5863df25-bad9-489d-bfaa-475707c09b15
AdminId        : 47801cf2-ab78-44a1-9add-649c5ef5b33e
ResellerId     :
FirstName      : Mike
LastName       : Lee
Email          : joe@gladinet.com
IsActivated    : True
IsDeleted      : False
Quota          : 0
QuotaUsed      : 201380
UserPlan       : 5
UserPlanUsed   : 0
PaypalProfileId :
StorageProviderId : Default
Edition        :
Expire         : 07/07/2016 09:00:00 PM
ResellerName   :
```

We can also sent the list to a GridView

The screenshot shows a PowerShell console on the left and an Out-GridView window on the right. The PowerShell console displays the output of the command `$tenant.Tenants`, which is a list of tenant objects. The Out-GridView window, titled "\$tenant.Tenants | Out-GridView", displays this list in a table format. The table has columns for DomainId, AdminId, ResellerId, FirstName, LastName, Email, IsActivated, and IsDeleted. The data row shows a tenant with DomainId "5863df25-bad9-489d-bfaa-475707c09b15", AdminId "47801cf2-ab78-44a1-9add-649c5ef5b33e", FirstName "Mike", LastName "Lee", Email "joe@gladinet.com", IsActivated "True", and IsDeleted "False".

Synopsis

Return tenant information based on TenantID.

Syntax

```
Get-GceTenantInfo [-TenantGuid] <String> [-LoginToken] <String> [[-
ServerUrl] <String>] [<CommonParameters>]
```

Description

Return tenant information based on given TenantID. Each tenant is represented by a unique identifier in the GUID format (DomainId).

Parameters

-TenantGuid <String>

Tenant GUID.

Required?	True
-----------	------

Position?	0
Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

-LoginToken <String>

User's security token for the caller of the function.

Required?	True
Position?	1
Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

-ServerUrl <String>

The remote service URL.

Required?	False
Position?	2
Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

<CommonParameters>

This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see about_CommonParameters (<http://go.microsoft.com/fwlink/?LinkID=113216>).

Inputs

Outputs

EnumTenantResults

The field Tenants contains a list with a single entry. The Single Entry is the tenant returned. Before checking the tenant information, make sure the success property is true.

Notes

Examples

Example 1

```
PS C:\> PS D:\> $tenant = Get-GceTenantInfo "b633d35a-e8d2-4106-88eb-
cef6a67216d6" $login.Cookie http://localhost
PS D:\> $tenant
```

```
Tenants : {joe'cc}
Success : True
Reason  :
Context :
Context1 :
```

```
PS D:\> $tenant.Tenants[0]
```

```
DomainId      : b633d35a-e8d2-4106-88eb-cef6a67216d6
AdminId       : df2383d0-ee8f-4cf4-8575-ab7330bd73b1
```

```
ResellerId      :  
FirstName      : joe'cc  
LastName       : admin'dd  
Email          : joe@gladinet.com  
IsActivated    : True  
IsDeleted      : False  
Quota          : 0  
QuotaUsed      : 908226177  
UserPlan       : 33  
UserPlanUsed   : 0  
PaypalProfileId :  
StorageProviderId : Default  
Edition        :  
Expire         :  
ResellerName   :  
PromoCode      :  
LastCheck      :  
TeamStatus     :  
LastQuotaScan  :  
AccountBi      :  
Organization   :  
IsBiz          : False  
IsPersonal     : False  
IsCloudStorage : True  
ForceScan      : False
```

Related links

Get-GceUserInfo

This function will return a specific user information by User ID.

It takes 3 parameters.

1. First Parameter is UserGuid (This is the Unique Id for the user)
2. Second Parameter is LoginToken (This is the Login Token represent the user making the call)
3. Third Parameter is ServerUrl (This is the GCE server)

```
PS C:\test> $user = Get-GceUserInfo $userid $cookie $url
PS C:\test> $user
FirstName : Mike
LastName  : Lee
Email     : joe@gladinet.com
Success   : True
Reason    :
Context   :
Context1  :

PS C:\test>
```

When it is successful, the Success field is True. If it is success it will return user's first name, last name and email.

Synopsis

Return a specific user's information by User's GUID.

Syntax

```
Get-GceUserInfo [-UserGuid] <String> [-LoginToken] <String> [[-
ServerUrl] <String>] [<CommonParameters>]
```

Description

This function returns user's information by User's GUID. The user's GUID is the UserGuid field for each user when it is returned from the Get-GceTenantUsers.

Parameters

-UserGuid <String>

User GUID

Required?	True
-----------	------

Position?	0
Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

-LoginToken <String>

Logged in user's security token, which is returned as the Cookie field from Get-GceLogin call.

Required?	True
Position?	1
Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

-ServerUrl <String>

The remote service URL.

Required?	False
Position?	2
Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

<CommonParameters>

This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see about_CommonParameters (<http://go.microsoft.com/fwlink/?LinkID=113216>).

Inputs

Outputs

QueryUserResult

When it is successful, the Success field is true. The result also return FirstName, LastName and Email. Basically this is a UserGuid to User name and email look up.

Notes

Examples

Example 1

```
PS C:\> PS D:\> $user = Get-GceUserInfo "df2383d0-ee8f-4cf4-8575-ab7330bd73b1" $login.Cookie http://localhost
PS D:\> $user
```

```
FirstName : joe'cc
LastName  : admin'dd
Email     : joe@gladinet.com
Success   : True
Reason    :
Context   :
Context1  :
```

Related links

Get-GceUserInfoByEmail

This function will return a specific user information by user email.

It takes 3 parameters.

1. First Parameter is Email (This is the user's Email)
2. Second Parameter is LoginToken (This is the Login Token represent the user making the call)
3. Third Parameter is ServerUrl (This is the GCE server)

```
PS C:\test> $user = Get-GceUserInfoByEmail "joe@gladinet.com" $cookie $url
PS C:\test> $user

User      : user.SingleUser
Success   : True
Reason    :
Context   :
Context1  :

PS C:\test>
```

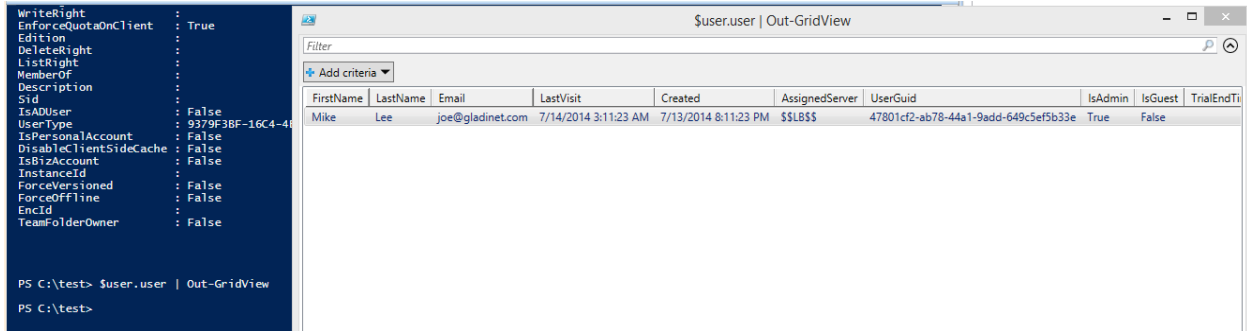
When it is successful, the Success field is True. And also contains a field User which is user information.

We can inspect the user:

```
PS C:\test> $user.user

FirstName      : Mike
LastName       : Lee
Email          : joe@gladinet.com
LastVisit      : 7/14/2014 3:11:23 AM
Created        : 7/13/2014 8:11:23 PM
AssignedServer : $$LB$$
UserGuid       : 47801cf2-ab78-44a1-9add-649c5ef5b33e
IsAdmin        : True
IsGuest        : False
TrialEndTime   :
IsActivated    : True
IsDeleted      : False
DomainId       : 5863df25-bad9-489d-bfaa-475707c09b15
IsRootAdmin    : True
Quota          : 0
QuotaUsed      : 0
ProductType    : CLOUD
HasStorage     : True
SnapshotBackup : True
MountCloudStorage : True
ReadRight      :
WriteRight     :
EnforceQuotaOnClient : True
Edition        :
DeleteRight    :
ListRight      :
```

We can also sent the list to a GridView



Synopsis

Return user information based on email address.

Syntax

```
Get-GceUserInfoByEmail [-Email] <String> [-LoginToken] <String> [[-ServerUrl] <String>] [<CommonParameters>]
```

Description

Return user information based on email address. It will return detail of user information in a SingleUser data structure.

Parameters

-Email <String>

User email address

Required?	True
Position?	0
Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

-LoginToken <String>

Logged in user's security token, which is returned as the Cookie field from Get-GceLogin call.

Required?	True
Position?	1
Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

-ServerUrl <String>

The remote service address.

Required?	False
Position?	2
Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

<CommonParameters>

This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see about_CommonParameters (<http://go.microsoft.com/fwlink/?LinkID=113216>).

Inputs

Outputs

SingleUserResult

When it is successful, the Success field is true. The result also return an user object contains detail of user information.

Notes

Examples

Example 1

Related links

Set-GceUserName

This function will update the user's firstname and lastname based on UserGuid.

It takes 5 parameters.

1. First Parameter is UserGuid (This is the Unique Id for the user)
2. Second Parameter is FirstName (This is user firstname)
3. Third Parameter is LastName (This is user lastname)
4. Fourth Parameter is LoginToken (This is the Login Token represent the user making the call)
5. Fifth Parameter is ServerUrl (This is the GCE server)

```
PS C:\test> $result = Set-GceUserName $userGuid "Mike" "Lee" $cookie $url
PS C:\test> $result

```

Success	Reason	Context	Context1
True			

```
PS C:\test> |
```

When it is successful, the Success field is True.

Synopsis

This function will update user firstname and lastname based on UserGuid.

Syntax

```
Set-GceUserName [-UserGuid] <String> [-FirstName] <String> [-LastName]  
<String> [-LoginToken] <String> [[-ServerUrl] <String>]  
[<CommonParameters>]
```

Description

This function will update user firstname and lastname based on UserGuid. You can get the UserGuid by Get-GceUserInfoByEmail.

Parameters

-UserGuid <String>

User GUID

Required?	True
Position?	0
Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

-FirstName <String>

User first name.

Required?	True
Position?	1

Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

-LastName <String>

User last name.

Required?	True
Position?	2
Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

-LoginToken <String>

Logged in user's cookie.

Required?	True
Position?	3
Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

-ServerUrl <String>

The remote service address.

Required?	False
Position?	4
Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

<CommonParameters>

This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see about_CommonParameters (<http://go.microsoft.com/fwlink/?LinkID=113216>).

Inputs

Outputs

Result

When it is successful, the Success field is true.

Notes

Examples

Example 1

```
PS C:\> PS D:\> $result = Set-GceUserName "ab2990b1-601f-407a-8ae1-515e5d9a3047" "Bill" "Smith" $login.Cookie http://localhost
PS D:\> $result
```

```
Context                                     Success Reason
-----
-----
-----                                     Context1
-----                                     -----
True
```

Related links

Set-GceTenantUserPlan

This function will set the user plan for a specific tenant.

It takes 4 parameters.

1. First Parameter is TenantGuid (This is the DomainId for the Tenant)
2. Second Parameter is UserPlan (This is user plan, an integer)
3. Third Parameter is LoginToken (This is the Login Token represent the user making the call)
4. Fourth Parameter is ServerUrl (This is the GCE server)

```
PS C:\test> Set-GceTenantUserPlan $tenantID 30 $cookie $url
Success Reason          Context          Context1
-----
True
```

When it is successful, the Success field is True.

Synopsis

This function will user quota for a specific tenant

Syntax

```
Set-GceTenantUserPlan [-TenantGuid] <String> [-UserPlan] <Int32> [-LoginToken] <String> [[-ServerUrl] <String>] [<CommonParameters>]
```

Description

This function will user quota for a specific tenant. The tenant is represented by the passed in TenantId.

Parameters

-TenantGuid <String>

Tenant GUID

Required?	True
Position?	0
Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

-UserPlan <Int32>

How many users can be created by current tenant.

Required?	True
Position?	1
Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

-LoginToken <String>

Logged in user's cookie.

Required?	True
Position?	2

Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

-ServerUrl <String>

The remote service address.

Required?	False
Position?	3
Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

<CommonParameters>

This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see about_CommonParameters (<http://go.microsoft.com/fwlink/?LinkID=113216>).

Inputs

Outputs

Result

When it is successful, the Success field is true.

Notes

Examples

Example 1

```
PS C:\> PS D:\> $result = Set-GceTenantUserPlan "d57eac12-8485-4a59-
b74c-dea6d037ba83" 200 $login.Cookie http://localhost
PS D:\> $result
```

```
Context                                     Success Reason
-----
-----                                     -----
True
```

Related links

Set-GceTenantStoragePlan

This function will set size (GB) of storage plan for a special tenant.

It takes 4 parameters.

1. First Parameter is TenantGuid (This is the DomainId for the Tenant)
2. Second Parameter is StoragePlanInGB (This is the size of Storage Plan)
3. Third Parameter is LoginToken (This is the Login Token represent the user making the call)
4. Fourth Parameter is ServerUrl (This is the GCE server)

```
PS C:\test> Set-GceTenantStoragePlan $tenantID 100 $cookie $url
Success Reason Context Context1
-----
True
```

When it is successful, the Success field is true.

Synopsis

This function will update storage quota for a special tenant.

Syntax

```
Set-GceTenantStoragePlan [-TenantGuid] <String> [-StoragePlanInGB]  
<Int32> [-LoginToken] <String> [[-ServerUrl] <String>]  
[<CommonParameters>]
```

Description

This function will update storage quota for a special tenant. The tenant is represented by the Tenant ID (DomainId)

Parameters

-TenantGuid <String>

Tenant GUID

Required?	True
Position?	0
Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

-StoragePlanInGB <Int32>

Size of storage space.

Required?	True
Position?	1

Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

-LoginToken <String>

Logged in user's cookie.

Required?	True
Position?	2
Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

-ServerUrl <String>

The remote service address.

Required?	False
Position?	3
Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

<CommonParameters>

This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable,

OutBuffer, PipelineVariable, and OutVariable. For more information, see about_CommonParameters (<http://go.microsoft.com/fwlink/?LinkID=113216>).

Inputs

Outputs

Result

When it is successful, the Success field is true.

Notes

Examples

Example 1

```
PS C:\> PS D:\> $result = Set-GceTenantStoragePlan "d57eac12-8485-4a59-  
b74c-dea6d037ba83" 200 $login.Cookie http://localhost  
PS D:\> $result
```

	Success	Reason
Context		Context1
-----	-----	-----
	True	

Related links

Set-GceTenantAdminEmail

This function will update the admin user's email address for a specific tenant.

It takes 4 parameters.

1. First Parameter is TenantGuid (This is the DomainId for the Tenant)
2. Second Parameter is AdminEmail (This is tenant email address)
3. Third Parameter is LoginToken (This is the Login Token represent the user making the call)
4. Fourth Parameter is ServerUrl (This is the GCE server)

```
PS C:\test> Set-GceTenantAdminEmail $tenantID "joe@gladinet.com" $cookie $url
          Success Reason          Context          Context1
          -----
          True
PS C:\test>
```

When it is successful, the Success field is true.

Synopsis

Update email address for a specific tenant administrator

Syntax

```
Set-GceTenantAdminEmail [-TenantGuid] <String> [-AdminEmail] <String> [-LoginToken] <String> [[-ServerUrl] <String>] [<CommonParameters>]
```

Description

Update email address for a specific tenant administrator. When success, the tenant administrator's email address will be changed to the new email address.

Parameters

-TenantGuid <String>

Tenant GUID

Required?	True
Position?	0
Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

-AdminEmail <String>

The new email address for the tenant. When success, the tenant administrator's email address will be changed to the new email address.

Required?	True
Position?	1
Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

-LoginToken <String>

Logged in user's cookie.

Required?	True
Position?	2
Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

-ServerUrl <String>

The remote service address.

Required?	False
Position?	3

Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

<CommonParameters>

This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see about_CommonParameters (<http://go.microsoft.com/fwlink/?LinkID=113216>).

Inputs

Outputs

Result

When it is successful, the Success field is true.

Notes

Examples

Example 1

```
PS C:\> PS D:\> $result = Set-GceTenantAdminEmail "d57eac12-8485-4a59-  
b74c-dea6d037ba83" test1@gladinet.com $login.Cookie http://localhost  
PS D:\> $result
```

```
Context                                     Success Reason  
-----  
-----  
True
```

Related links

New-GceUser

This function will create a new user under a specific tenant.

It takes 12 parameters.

1. First Parameter is TenantGuid (This is the DomainId for the Tenant)
2. Second Parameter is FirstName (This is the user's firstname)
3. Third Parameter is LastName (This is the user's lastname)
4. Fourth Parameter is Email (This is the user's email)
5. Fifth Parameter is Password (This is the user's Password)
6. Sixth Parameter is StorageQuotaInGB (This is the size of Storage Quota)
7. Seventh Parameter is CreateDefaultDirectory (Boolean, This is the flag to identify if create default folder structure)
8. Eighth Parameter is ForceVersionedRootFolder (Boolean, This is the flag to identify if force use version for folder name)
9. Ninth Parameter is EnableOfflineFromRoot (Boolean, whether to enable Windows client to make the folder offline starting from the root folder)
10. Tenth Parameter is SendWelcomeEmail (Boolean, Identify if send out welcome email)
11. Eleventh Parameter is LoginToken (This is the Login Token represent the user making the call)
12. Twelfth Parameter is ServerUrl (This is the GCE server)

```
PS C:\test> $user = New-GceUser $tenantID "Jeffrey" "Hunk" "Jeffery@gladinet.com" "dd" 5 1 1 1 0 $cookie $url
PS C:\test> $user

Guid           : 18c731cd-f5f1-4663-a367-f82a4efda823
LapseSeconds  : 0
Success        : True
Reason         :
Context        :
Context1       :

PS C:\test> |
```

When it is successful, the Success field is true. It also returns the new user's GUID.

Synopsis

Create a new user under a specific tenant.

Syntax

```
New-GceUser [-TenantGuid] <String> [-FirstName] <String> [-LastName]
<String> [-Email] <String> [-Password] <String> [-StorageQuotaInGB]
<Int64> [-CreateDefaultDirectory] <Boolean> [-ForceVersionedRootFolder]
<Boolean> [-EnableOfflineFromRoot] <Boolean> [-SendWelcomeEmail]
<Boolean> [-LoginToken] <String> [[-ServerUrl] <String>]
[<CommonParameters>]
```

Description

Create a new user. The new user will be under the specified tenant (one of the input parameter). The user is a normal Gladinet user, not an imported Active Directory user. If you want to import active directory user, please use Add-GceAdUserByUPNEx instead.

Parameters

-TenantGuid <String>

Tenant Guid, the new user will be under this tenant.

Required?	True
Position?	0
Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

-FirstName <String>

User's first name.

Required?	True
Position?	1
Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

-LastName <String>

User's last name.

Required?	True
Position?	2
Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

-Email <String>

User's email address.

Required?	True
Position?	3
Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

-Password <String>

User's password.

Required?	True
Position?	4
Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

-StorageQuotaInGB <Int64>

The storage quota (in GB) for the new user.

Required?	True
Position?	5
Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

-CreateDefaultDirectory <Boolean>

If it is true , system will create default folder structure.

Required?	True
Position?	6
Default value	

Accept pipeline input?	false
Accept wildcard characters?	False

-ForceVersionedRootFolder <Boolean>

This is the flag to identify if every files will be versioned.

Required?	True
Position?	7
Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

-EnableOfflineFromRoot <Boolean>

This flag identifies whether to allow user to fetch all the files for offline access from the root folder.

Required?	True
Position?	8
Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

-SendWelcomeEmail <Boolean>

If set to true, system will send out a welcome email to user's email address.

Required?	True
-----------	------

Position?	9
Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

-LoginToken <String>

Logged in user's security token, which is returned as the Cookie field from Get-GceLogin call.

Required?	True
Position?	10
Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

-ServerUrl <String>

The remote service URL.

Required?	False
Position?	11
Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

<CommonParameters>

This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see about_CommonParameters (<http://go.microsoft.com/fwlink/?LinkID=113216>).

Inputs

Outputs

GuidResult

When it is successful, the Success field is true. The GUID field contains new User GUID (UserGuid).

Notes

Examples

Example 1

```
PS C:\> PS D:\> $tenantid
```

```
Guid           : d1f45759-2386-429e-aa9f-560307354d1f
LapseSeconds  : 0
Success       : True
Reason        :
Context       :
Context1     :
```

```
PS D:\> $newuserid = New-GceUser "d1f45759-2386-429e-aa9f-560307354d1f"
"Acme" "User1" "acmeuser1@gladinet.com" "password1" 10 0 0 0 0
$login.Cookie http://loca
lhost
PS D:\> $newuserid
```



```
Guid          : 59434cc3-ee3a-4b19-a1a8-9b99fb374c4c
LapseSeconds  : 0
Success       : True
Reason        :
Context       :
Context1      :
```

Related links

Reset-GceUserPassword

This function will reset the user's password based on email address.

It takes 4 parameters.

1. First Parameter is userEmail (This is user's email address)
2. Second Parameter is Password (This is the new password)
3. Third Parameter is LoginToken (This is the Login Token represent the user making the call)
4. Fourth Parameter is ServerUrl (This is the GCE server)

```
PS C:\test> Reset-GceUserPassword "jeffery@gladinet.com" "dddd" $cookie $url
Success Reason          Context          Context1
-----
True
```

When it is successful, the Success field is true.

Synopsis

Reset a specific user's password

Syntax

```
Reset-GceUserPassword [-UserEmail] <String> [-Password] <String> [-
LoginToken] <String> [[-ServerUrl] <String>] [<CommonParameters>]
```

Description

Reset a specific user's password. The input will be the user's email and the new password to be used.

Parameters

- userEmail <String>

User email address

Required?	True
Position?	0
Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

- Password <String>

User's new password

Required?	True
Position?	1
Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

- LoginToken <String>

Logged in user's cookie.

Required?	True
Position?	2
Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

-ServerUrl <String>

The remote service URL.

Required?	False
Position?	3
Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

<CommonParameters>

This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see about_CommonParameters (<http://go.microsoft.com/fwlink/?LinkID=113216>).

Inputs

Outputs

Result

When it is successful, the Success field is true.

Notes

Examples

Example 1

```
PS C:\> PS D:\> $result = Reset-GceUserPassword joe@gladinet.com
password1 $login.Cookie http://localhost
PS D:\> $result
```

```
Context                                     Success Reason
-----                                     -
-----                                     -----
                                         Context1
                                         -----
                                         True
```

Related links

Remove-GceUser

This function will delete an existing user based on UserGuid.

It takes 4 parameters.

1. First Parameter is UserGuid (This is user GUID)
2. Second Parameter is PurgeStorage (This is identify if delete user data after delete user)
3. Third Parameter is LoginToken (This is the Login Token represent the user making the call)
4. Fourth Parameter is ServerUrl (This is the GCE server)

```
PS C:\test> Remove-GceUser $user.Guid 1 $cookie $url
Success Reason      Context      Context1
-----
True
```

```
PS C:\test>
```

When it is successful, the Success field is true.

Synopsis

Remove an existing user.

Syntax

```
Remove-GceUser [-UserGuid] <String> [-PurgeStorage] <Boolean> [-LoginToken] <String> [[-ServerUrl] <String>] [<CommonParameters>]
```

Description

Remove an existing user.

Parameters

-UserGuid <String>

User GUID

Required?	True
Position?	0
Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

-PurgeStorage <Boolean>

If it is set to true it will schedule deletion of user's data.

Required?	True
Position?	1
Default value	

Accept pipeline input?	false
Accept wildcard characters?	False

-LoginToken <String>

Logged in user's cookie.

Required?	True
Position?	2
Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

-ServerUrl <String>

The remote service address.

Required?	False
Position?	3
Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

<CommonParameters>

This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable,

OutBuffer, PipelineVariable, and OutVariable. For more information, see about_CommonParameters (<http://go.microsoft.com/fwlink/?LinkID=113216>).

Inputs

Outputs

Result

When it is successful, the Success field is true.

Notes

Examples

Example 1

```
PS C:\> PS D:\> $result = Remove-GceUser "59434cc3-ee3a-4b19-a1a8-9b99fb374c4c" 0 $login.Cookie http://localhost
PS D:\> $result
```

	Success	Reason
Context		Context1
-----	-----	-----
	True	

Related links

Remove-GceTenant

This function will delete a specific tenant based on tenant GUID.

It takes 3 parameters.

1. First Parameter is TenantGuid (This is the DomainId for the Tenant)
2. Second Parameter is LoginToken (This is the Login Token represent the user making the call)
3. Third Parameter is ServerUrl (This is the GCE server)

```
PS C:\test> Remove-GceTenant $tenant.Guid $cookie $url
Success Reason Context Context1
-----
True

PS C:\test> |
```

When it is successful, the Success field is true.

Synopsis

Remove an existing tenant.

Syntax

```
Remove-GceTenant [-TenantGuid] <String> [-LoginToken] <String> [[-
ServerUrl] <String>] [<CommonParameters>]
```

Description

Remove an existing tenant. The tenant is specified by the TenantID (DomainId), a GUID represent the tenant.

Parameters

-TenantGuid <String>

Tenant GUID

Required?	True
Position?	0
Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

-LoginToken <String>

Logged in user's cookie.

Required?	True
Position?	1
Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

-ServerUrl <String>

The remote service address.

Required?	False
Position?	2
Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

<CommonParameters>

This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see about_CommonParameters (<http://go.microsoft.com/fwlink/?LinkID=113216>).

Inputs

Outputs

Result

When it is successful, the Success field is true.

Notes

Examples

Example 1

```
PS C:\> PS D:\> $result = Remove-GceTenant "d1f45759-2386-429e-aa9f-560307354d1f" $login.Cookie http://localhost
PS D:\> $result
```

	Success	Reason
Context		Context1
-----	-----	-----
	True	

Related links

Set-GceTenantAdminRights

This function will set the rights for a specific tenant.

It takes 8 parameters.

1. First Parameter is TenantGuid (This is the DomainId for the Tenant)
2. Second Parameter is TeamAttachExternalStorage (Boolean, allow current tenant to attach additional storage)
3. Third Parameter is TeamEditLDAPSetting (Boolean, allow current tenant edit LDAP setting)
4. Fourth Parameter is TeamViewGroupPolicy (Boolean, whether the tenant can view Group Policy page)

5. Fifth Parameter is TeamEditAdminInfo (Boolean, whether the tenant can edit tenant Administrator's information)
6. Sixth Parameter is TeamEditBranding (Boolean, whether the tenant can edit branding information)
7. Seventh Parameter is is LoginToken (This is the Login Token represent the user making the call)
8. Eighth Parameter is ServerUrl (This is the GCE server)

```
PS C:\test> Set-GceTenantAdminRights $tenant.Guid 1 0 0 1 1 $cookie $url
User      :
Success   : True
Reason    :
Context   :
Context1  :

PS C:\test>
```

When it is successful, the Success field is true.

Set-GceTenantAdminRightsEx

This function will set the rights for a specific tenant.

It takes 10 parameters.

1. First Parameter is TenantGuid (This is the DomainId for the Tenant)
2. Second Parameter is TeamAttachExternalStorage (Whether the tenant can attach additional storage)
3. Third Parameter is TeamEditLDAPSetting (whether the current tenant can edit LDAP setting)
4. Fourth Parameter is TeamViewGroupPolicy (whether the tenant can view group policy)
5. Fifth Parameter is TeamEditAdminInfo (whether the tenant can edit tenant Administrator's information)
6. Sixth Parameter is TeamEditBranding (whether allow edit branding information)
7. Seventh Parameter is ForceVersioned (whether to force tenant's root folder to be versioned folder)
8. Eighth Parameter is AllowGuest (whether allow tenant to add guest user account)

9. Ninth Parameter is LoginToken (This is the Login Token represent the user making the call)
10. Tenth Parameter is ServerUrl (This is the GCE server)

```
PS C:\test> Set-GceTenantAdminRightsEx $tenant.Guid 1 0 0 1 1 1 1 $cookie $url
User      :
Success   : True
Reason    :
Context   :
Context1  :
```

When it is successful, the Success field is true.

Set-GceTenantExpirationTime

This function will set the expiration time for a specific tenant.

It takes 4 parameters.

1. First Parameter is TenantGuid (This is the DomainId for the Tenant)
2. Second Parameter is ExpirationTime (This is current tenant expiration time. Format is "MM/DD/YYYY HH:mm:ss")
3. Third Parameter is LoginToken (This is the Login Token represent the user making the call)
4. Fourth Parameter is ServerUrl (This is the GCE server)

```
PS C:\test> Set-GceTenantExpirationTime $tenant.Guid 07/22/2014 $cookie $url
Success Reason          Context          Context1
-----
True
```

When it is successful, the Success field is true.

Set-GceTenantStorageConfig

This function will set the storage configuration for a specific tenant.

It takes 4 parameters.

1. First Parameter is TenantGuid (This is the DomainId for the Tenant)

9. Ninth Parameter is StorageConfigure (This is base64 encoded storage configuration string)
10. Tenth Parameter is LoginToken (This is the Login Token represent the user making the call)
11. Eleventh Parameter is ServerUrl (This is the GCE server)

```
PS C:\test> $localConfig = Get-GceLocalStorageConfig "c:\storage" "test" "test" 1 0

PS C:\test> $localConfig
PD94bWwgdMvyc2lvcj0iMS4wIiB1bmNvZGluZz0idXRmlTgipz48U3RvcmlFZUR1c2NyaXB0b3I+PE5hbWU+SG9tZSBTdG9yYWdlPC90Yw1lPjxHdWlkPjQxZWQ2YzI2LWQ1ZjgtNDI0Zi1hMzExLWJiNGZmODgxND81NTwvR3VpZD48RW5kcG9pbmQ+aHR0cDovL2xvY2FsaG9zZDo4MDgwL2dsYWRmaWxlczZyL2Yuc3ZjLzwwRw5kcG9pbmQ+PFNlcnZpY2VUeXB1Pjk8L1NlcnZpY2VUeXB1PjxRdW90YT4wPC9RdW90YT48RXhwaXJlPjk5OTk6MTI6MTZlcmVjM6NTk6NT1aPC9FahBpcmlU+PE5hbWVhYXN1cyAvPjxVc2VyTmFtZT50ZXRhOj9vcmV0YmFtZT48U2VjcmV0S2V5PnRlc3Q8L1N1Y3JldEt1eT48QnVja2V0IC8+PEFjY2Vzcj1Bvaw50PmM6XHN0b3JhZ2U8OFjY2Vzcj1Bvaw50PjxUcnV1Rw5jcn1wdG1vbj57VHJlZX08L1RydWVhYXN1cyAvPjxU3RvcmlFZUR1c2NyaXB0b3I+

PS C:\test> $tenant = New-GceTenant "Gladinet" "Mark" "Delin" "mark@gladinet.com" "dd" 3 1 "01/01/2015" $localConfig $cookie $url

PS C:\test> $tenant

Guid           : 1fe6a9e4-3c49-4c76-8a47-b823eb4785c5
LapseSeconds  : 0
Success        : True
Reason         :
Context        :
Context1       :
```

When it is successful, the Success field is true. It also return user tenant's GUID.

Synopsis

Create new tenant

Syntax

```
New-GceTenant [[-OrgName] <String>] [-FirstName] <String> [-LastName] <String> [-AdminEmail] <String> [-AdminPassword] <String> [-UserPlan] <Int32> [-StoragePlan] <Int32> [-ExpirationTime] <String> [-StorageConfigure] <String> [-LoginToken] <String> [[-ServerUrl] <String>] [<CommonParameters>]
```

Description

This function will create a new tenant. You can specify the initial configuration parameters for the new tenant.

Parameters

-OrgName <String>

Organization name.

Required?	False
Position?	0
Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

-FirstName <String>

Tenant administrator's first name.

Required?	True
Position?	1
Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

-LastName <String>

Tenant administrator's last name.

Required?	True
Position?	2

Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

-AdminEmail <String>

Email address of tenant administrator. It will be used for login username.

Required?	True
Position?	3
Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

-AdminPassword <String>

Password of tenant administrator

Required?	True
Position?	4
Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

-UserPlan <Int32>

Number of users the tenant can have.

Required?	True
Position?	5
Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

-StoragePlan <Int32>

The Storage Quota (in GB) to be allocated to the tenant

Required?	True
Position?	6
Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

-ExpirationTime <String>

Expiration Time. The format is "MM/DD/YYYY HH:mm:ss". It can also be a normal .NET datetime format such as

```
PS D:\> [System.DateTime]::Now
```

Thursday, July 31, 2014 12:30:55 PM

```
PS D:\> [System.DateTime]::MaxValue
```

Friday, December 31, 9999 11:59:59 PM

Required?	True
Position?	7
Default value	
Accept pipeline input?	false
Accept wildcard characters?	True

-StorageConfigure <String>

A base64 encoded string that contains storage configuration.

Required?	True
Position?	8
Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

-LoginToken <String>

Logged in user's security token, which is returned as the Cookie field from Get-GceLogin call.

Required?	True
Position?	9
Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

-ServerUrl <String>

The remote service URL.

Required?	False
Position?	10
Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

<CommonParameters>

This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see about_CommonParameters (<http://go.microsoft.com/fwlink/?LinkID=113216>).

Inputs

Outputs

GuidResult

When it is successful, the Success field is true. The GUID field contains new tenant ID (DomainId).

Notes

When setting the tenant's storage configuration, need to make sure the folder, or container or bucket exist before the call.

Examples

Example 1

```
PS C:\> PS D:\> $storagecfg = Get-GceLocalStorageConfig "e:\storage"  
"gladuser" "password123" 0 0
```

```
PS D:\> $storagecfg
PD94bWwgdMvYc2lvbj0iMS4wIiBlbmNvZGluZz0idXRmLTgiPz48U3RvcnFnZURlc2NyaXB0
b3I+PE5hbWU+SG9tZSBTdG9yYWdlPC90YW1lPjxHdWlkPjQxZWQ2YzI2LWQ1ZjgtNDI0Zi1h
MzExLWJiNGZmODg
xNDBlNTwvR3VpZD48RW5kcG9pbnQ+aHR0cDovL2xvY2FsaG9zdDo4MDgwL2dsYWRmaWxlc3Z
yL2Yuc3ZjLzZwRW5kcG9pbnQ+PFNlcnZpY2VUeXB1Pjk8L1NlcnZpY2VUeXB1PjxRdW90YT4
wPC9RdW90YT48RX
hwaXJlPjk5OTktMTItMzEgMjM6NTk6NTlaPC9FeHBpcnU+PE5hbWVbBGlhcyAvPjxVc2VyTm
FtZT5nbGFkdXNlcjwvVXNlck5hbWU+PFNlY3JldEt1eT5wYXNzd29yZDEyMzVwU2VjcmV0S2
V5PjxCdWNRZXQgL
z48QWNjZXNzUG9pbnQ+ZTpcc3RvcnFnZTwvQWNjZXNzUG9pbnQ+PERhdGFbdFJlc3RFbmnYe
XB0aW9uPmZhbHNlPC9EYXRhQXRSZXN0RW5jcmlwdGlvbj48L1N0b3JhZ2VEZXNjcmlwdG9yP
g==
PS D:\> $tenantid = New-GceTenant "ACME2 Corp" "Joe" "Johnson"
"joejohn@gladinet.com" "password1" 10 100 "Friday, December 31, 9999
11:59:59 PM" $storagecfg $lo
gin.Cookie http://localhost
PS D:\> $tenantid
```

```
Guid          : d1f45759-2386-429e-aa9f-560307354d1f
LapseSeconds  : 0
Success       : True
Reason        :
Context       :
Context1      :
```

Related links

Add-GceAdUserByUPN

This function will import an AD user under a specific tenant by UPN (User-Principal-Name).

It takes 5 parameters.

1. First Parameter is UserUPN (This is AD UPN for a special user)

2. Second Parameter is StoragePlan (This is the size of storage plan)
3. Third Parameter is StorageConfigure (This is base64 encoded storage configuration string)
4. Fourth Parameter is LoginToken (This is the Login Token represent the user making the call)
5. Fifth Parameter is ServerUrl (This is the GCE server)

Add-GceAdUserByUPNEx

This function will import an AD user under a specific tenant by UPN.

It takes 7 parameters.

1. First Parameter is UserUPN (This is AD UPN for a special user)
2. Second Parameter is StoragePlan (This is the size of storage plan)
3. Third Parameter is ForceVersioned (Force version folder from root)
4. Fourth Parameter is EnableOffline (Enable User's folder offline from root)
5. Fifth Parameter is StorageConfigure (This is base64 encoded storage configuration string)
6. Sixth Parameter is LoginToken (This is the Login Token represent the user making the call)
7. Seventh Parameter is ServerUrl (This is the GCE server)

Synopsis

Add team user from Active Directory by UPN, with two additional flags - ForceVersioned and EnableOffline.

Syntax

```
Add-GceAdUserByUPNEx [-UserUPN] <String> [-StoragePlan] <Int32> [-ForceVersioned] <Boolean> [-EnableOffline] <Boolean> [-StorageConfigure] <String> [-LoginToken] <String> [[-ServerUrl] <String>] [<CommonParameters>]
```

Description

Add an active directory user to a tenant. The active directory user is specified by the User-Principal-Name.

This is an enhanced version of Add-GceAdUserByUPN, with two additional flags, ForceVersioned and EnableOffline.

Parameters

-UserUPN <String>

User's UPN in the Active Directory.

Required?	True
Position?	0
Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

-StoragePlan <Int32>

Storage plan for added user, how many users can be created.

Required?	True
Position?	1
Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

-ForceVersioned <Boolean>

If set ForceVersioned to true, the structure of storage folder will be versioned.

Required?	True
Position?	2
Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

-EnableOffline <Boolean>

Enable user offline.

Required?	True
Position?	3
Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

-StorageConfigure <String>

Storage configuration string, it is a base64 encoded string. You can use storage configuration helper function to get back the string. Such as Get-GceLocalConfiguration. If the storage configuration is empty, the imported user will be allocated a default storage location.

Required?	True
Position?	4

Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

-LoginToken <String>

Logged in user's cookie.

Required?	True
Position?	5
Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

-ServerUrl <String>

The remote service address.

Required?	False
Position?	6
Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

<CommonParameters>

This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable,

OutBuffer, PipelineVariable, and OutVariable. For more information, see about_CommonParameters (<http://go.microsoft.com/fwlink/?LinkID=113216>).

Inputs

Outputs

GuidResult

Return the GuidResult. GuidResult inherits from Result with an additional Guid property. If it is successful, the Guid is new user's unique identifier in a GUID format.

Notes

The tenant must have Active Directory setup and sufficient permission to access Active Directory.

Examples

Example 1

```
PS C:\> PS C:\test> $login = Get-GceLogin tenant1@gladinet.com dddddddd
http://192.168.2.11
PS C:\test> $login
```

```
Cookie                                :
VqcUOOT3pJxlBExF0QgHZvoTYebnTpo/wmDn2djL2cW5ugu85rwtGtEcpzovWby+urqTodqQ
0eupja/Af44dcgf0gRQCqr
                                     gGeXMBmhW45TTQqy6q
NSEndPoint                            : http://192.168.2.11/namespace/n.svc/
LSEndPoint                            : http://localhost:8080/localstor/n.svc/
RelayServer                            :
Hours                                  : 4
User                                    : user.SingleUser
PortalServer                          : http://192.168.2.11/portal/
BrandingProductName                   : Jerry Drive
KeepCmdChannel                        : False
SupportSetLastModified                : True
WCVersion                              : 5.6.1264.29612
```

```
ForceVersioned      : False
Success             : False
Reason              :
Context             :
Context1            :
```

```
PS C:\test> $userid = Add-GceAdUserByUPNEx t1e7@tsys.gladinet.com 10 0 0
"" $login.Cookie http://192.168.2.11
PS C:\test> $userid
```

```
Guid                : 3207e31b-417e-4101-8017-acd82842c516
LapseSeconds        : 0
Success             : True
Reason              :
Context             :
Context1            :
```

Related links

Set-GceUserStoragePlanGB

This function will update the storage plan for specific user.

It takes 4 parameters.

1. First Parameter is UserID (This is user Guid)
2. Second Parameter is GBCount (This is size of storage plan)
3. Third Parameter is LoginToken (This is the Login Token represent the user making the call)
4. Fourth Parameter is ServerUrl (This is the GCE server)

```
PS C:\test> Set-GceUserStoragePlanGB $userGuid 1024 $cookie $url
Success Reason          Context          Context1
-----
True
```

When it is successful, the Success field is true.

Synopsis

This function will update the storage plan for specific user.

Syntax

```
Set-GceUserStoragePlanGB [-UserID] <String> [-GBCount] <Int64> [-LoginToken] <String> [[-ServerUrl] <String>] [<CommonParameters>]
```

Description

This function will update the storage plan for specific user. The storage plan is measured in GB

Parameters

-UserID <String>

User GUID

Required?	True
Position?	0
Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

-GBCount <Int64>

Storage Quota in GB

Required?	True
Position?	1

Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

-LoginToken <String>

Logged in user's cookie.

Required?	True
Position?	2
Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

-ServerUrl <String>

The remote service address.

Required?	False
Position?	3
Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

<CommonParameters>

This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable,

OutBuffer, PipelineVariable, and OutVariable. For more information, see about_CommonParameters (<http://go.microsoft.com/fwlink/?LinkID=113216>).

Inputs

Outputs

Result

When it is successful, the Success field is true.

Notes

Examples

Example 1

```
PS C:\> PS D:\> $result = Set-GceUserStoragePlanGB "ab2990b1-601f-407a-8ae1-515e5d9a3047" 50 $login.Cookie http://localhost
PS D:\> $result
```

	Success	Reason
Context		Context1
-----	-----	-----
	True	

Related links

Set-GceUserStorageConfig

This function will update the storage configuration for a specific user.

It takes 4 parameters.

1. First Parameter is UserID (This is user GUID)
2. Second Parameter is StorageConfigure (This is base64 encoded storage configuration string)
3. Third Parameter is LoginToken (This is the Login Token represent the user making the call)

4. Fourth Parameter is ServerUrl (This is the GCE server)

```
PS C:\test> $localConfig = Get-GceLocalStorageConfig "c:\storage" "test" "test" 1 0
PS C:\test> $localConfig
PD94bWwgdmVyc2lrbj0iMS4wIiB1bmNvZGluz0idXRmlTgiPz48U3RvcnFnZURlc2NyaXB0b3I+PE5hbWU+SG9tZSBTdG9yYwdlPC90Yw1lPjxHdWlkPjQxZWQ2YzI2LWQ
1ZjgtNDI0Zi1hMzExLWJiNGZmODgxND8lNTwvR3VpZD48RW5kcG9pbmQ+aHR0cDovL2xvY2FsaG9zdDo4MDgwL2dsYWRmaWx1c3ZyL2Yuc3ZjLzwwRw5kcG9pbmQ+PFN1cn
ZpY2VUeXB1Pjk8L1N1cnZpY2VUeXB1PjxRdW90YT4wPC9RdW90YT48RXhwaXJlPjk5OTktMTI0MzEgMjM6NTk6NTl1aPC9FeHBpcmlU+PE5hbWVBBG1hcyAvPjxVc2VyTmFtZ
T50ZXR0PC9Vc2VyTmFtZT48U2VjcmV052V5PnRlc3Q8L1N1Y3JldEt1eT48QnVja2V0IC8+PEFjY2Vzc1BvaW50PmM6XHNob3JhZ2U8L0FjY2Vzc1BvaW50PjxUcnV1RW5j
cn1wdGlvbj57VHJ1ZX08L1RydwVfYmNyeXB0aW9uPjwvU3RvcnFnZURlc2NyaXB0b3I+
```

```
PS C:\test> Set-GceUserStorageConfig $userGuid $localConfig $cookie $url
Success Reason          Context          Context1
-----
True
```

When it is successful, the Success field is true.

Synopsis

This function will update the storage configuration for a specific user.

Syntax

```
Set-GceUserStorageConfig [-UserID] <String> [-StorageConfigure] <String>
[-LoginToken] <String> [[-ServerUrl] <String>] [<CommonParameters>]
```

Description

This function will update the storage configuration for a specific user. The storage configuration can be constructed by helper function such as Get-GceLocalStorageConfig. This way you can specify storage location for each individual users.

Parameters

-UserID <String>

User GUID

Required?	True
Position?	0
Default value	
Accept pipeline input?	false

Accept wildcard characters?	False
-----------------------------	-------

-StorageConfigure <String>

A base64 encoded string which can get from the helper function, such as Get-GceLocalStorageConfig.

Required?	True
Position?	1
Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

-LoginToken <String>

Logged in user's cookie.

Required?	True
Position?	2
Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

-ServerUrl <String>

The remote service address.

Required?	False
Position?	3

Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

<CommonParameters>

This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see about_CommonParameters (<http://go.microsoft.com/fwlink/?LinkID=113216>).

Inputs

Outputs

Result

When it is successful, the Success field is true.

Notes

Examples

Related links

Set-GceEnableDARE

This function will enable DARE (Date-At-Rest-Encryption) function for a specific tenant.

It takes 3 parameters.

1. First Parameter is Password (This is the password used to generate encryption key)
2. Second Parameter is LoginToken (This is the Login Token represent the user making the call)
3. Third Parameter is ServerUrl (This is the GCE server)


```
PS C:\test> Set-GceEnableDARE gg $cookie $url
Success Reason          Context          Context1
-----
True
```

When it is successful, the Success field is true.

Synopsis

Enable Data-At-Rest Encryption (DARE) for a specific tenant

Syntax

```
Set-GceEnableDARE [-Password] <String> [-LoginToken] <String> [[-
ServerUrl] <String>] [<CommonParameters>]
```

Description

Enable Data-At-Rest Encryption (DARE) for a specific tenant.

Parameters

-Password <String>

Password used to encrypt folder. This password will be used to generate the key for AES-256.

Required?	True
Position?	0
Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

-LoginToken <String>

Logged in user's cookie. It is very important that the security token represent the tenant admin because the encryption will be enabled for the tenant, according to the security token.

Required?	True
Position?	1
Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

-ServerUrl <String>

The remote service address.

Required?	False
Position?	2
Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

<CommonParameters>

This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see about_CommonParameters (<http://go.microsoft.com/fwlink/?LinkID=113216>).

Inputs

Outputs

Result

When it is successful, the Success field is true.

Notes

Examples

Example 1

```
PS C:\> PS D:\> $result = Set-GceEnableDARE pass123 $login1.Cookie
http://localhost
PS D:\> $result
```

Context	Success Reason
-----	-----
	Context1

	True

```
PS D:\>
```

Related links

Set-GceDisableDARE

This function will disable DARE (Data-At-Rest-Encryption) function for a specific tenant.

It takes 2 parameters.

1. First Parameter is LoginToken (This is the Login Token represent the user making the call)
2. Second Parameter is ServerUrl (This is the GCE server)

```
PS C:\test> Set-GceDisableDARE $cookie $url
Success Reason          Context          Context1
-----
True
```

When it is successful, the Success field is true.

Synopsis

Disable Data-At-Rest Encryption for a specific tenant

Syntax

```
Set-GceDisableDARE [-LoginToken] <String> [[-ServerUrl] <String>]  
[<CommonParameters>]
```

Description

Disable Data-At-Rest Encryption for a specific tenant. The tenant is specified by the security token. So the disable DARE will apply to the caller's tenant.

Parameters

-LoginToken <String>

Logged in user's cookie. It is very important that the security token represent the tenant admin because the encryption will be enabled for the tenant, according to the security token.

Required?	True
Position?	0
Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

-ServerUrl <String>

The remote service URL.

Required?	False
Position?	1
Default value	
Accept pipeline input?	false

Accept wildcard characters?	False
-----------------------------	-------

<CommonParameters>

This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see about_CommonParameters (<http://go.microsoft.com/fwlink/?LinkID=113216>).

Inputs

Outputs

Result

When it is successful, the Success field is true.

Notes

Examples

Example 1

```
PS C:\> PS D:\> $result = Set-GceDisableDARE $login1.Cookie  
http://localhost  
PS D:\> $result
```

	Success	Reason
Context		Context1
-----	-----	-----
	True	

Related links

Set-GceEnableTenant

This function will enable a specific tenant.

It takes 3 parameters.

1. First Parameter is TenantId (This is domain ID of tenant)
2. Second Parameter is LoginToken (This is the Login Token represent the user making the call)
3. Third Parameter is ServerUrl (This is the GCE server)

```
PS C:\test> Set-GceEnableTenant $tenant.Guid $cookie $url  
  
Success Reason          Context          Context1  
-----  
True
```

When it is successful, the Success field is true.

Synopsis

Enable an existing tenant

Syntax

```
Set-GceEnableTenant [-TenantId] <String> [-LoginToken] <String> [[-  
ServerUrl] <String>] [<CommonParameters>]
```

Description

Enable an existing tenant based on tenant GUID.

Parameters

-TenantId <String>

Tenant GUID

Required?	True
Position?	0
Default value	
Accept pipeline input?	false

Accept wildcard characters?	False
-----------------------------	-------

-LoginToken <String>

Logged in user's cookie.

Required?	True
Position?	1
Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

-ServerUrl <String>

The remote service address.

Required?	False
Position?	2
Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

<CommonParameters>

This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see about_CommonParameters (<http://go.microsoft.com/fwlink/?LinkID=113216>).

Inputs

Outputs

Result

When it is successful, the Success field is true.

Notes

Examples

Example 1

```
PS C:\> PS D:\> $result = Set-GceEnableTenant "d57eac12-8485-4a59-b74c-  
dea6d037ba83" $login.Cookie http://localhost  
PS D:\> $result
```

	Success	Reason
Context		Context1
-----	-----	-----
	True	

Related links

Set-GceDisableTenant

This function will disable a specific tenant.

It takes 3 parameters.

1. First Parameter is TenantId (This is domain ID of tenant)
2. Second Parameter is LoginToken (This is the Login Token represent the user making the call)
3. Third Parameter is ServerUrl (This is the GCE server)

```
PS C:\test> Set-GceDisableTenant $tenant.Guid $cookie $url  
  
Success Reason          Context          Context1  
-----  
True
```


When it is successful, the Success field is true.

Synopsis

Disable an existing tenant.

Syntax

```
Set-GceDisableTenant [-TenantId] <String> [-LoginToken] <String> [[-  
ServerUrl] <String>] [<CommonParameters>]
```

Description

Disable an existing tenant based on the passed in tenant GUID.

Parameters

-TenantId <String>

Tenant GUID

Required?	True
Position?	0
Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

-LoginToken <String>

Logged in user's cookie.

Required?	True
Position?	1

Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

-ServerUrl <String>

The remote service address.

Required?	False
Position?	2
Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

<CommonParameters>

This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see about_CommonParameters (<http://go.microsoft.com/fwlink/?LinkID=113216>).

Inputs

Outputs

Result

When it is successful, the Success field is true.

Notes

Examples

Example 1

```
PS C:\> PS D:\> $result = Set-GceDisableTenant "d57eac12-8485-4a59-b74c-  
dea6d037ba83" $login.Cookie http://localhost  
PS D:\> $result
```

```
Context                                     Success Reason  
-----  
-----  
True
```

Related links

Set-GceUserAssignedZone

This function will update assigned zone for a specific user.

It takes 4 parameters.

1. First Parameter is UserID (This is user GUID)
2. Second Parameter is ZoneName (This is zone name)
3. Third Parameter is LoginToken (This is the Login Token represent the user making the call)
4. Fourth Parameter is ServerUrl (This is the GCE server)

```
PS C:\test> Set-GceUserAssignedZone $userGuid zonename $cookie $url  
  
Success Reason      Context      Context1  
-----  
True
```

When it is successful, the Success field is true.

Set-GceEnableUser

This function will return a specific tenant information.

It takes 3 parameters.

1. First Parameter is UserID (This is user GUID)
2. Second Parameter is LoginToken (This is the Login Token represent the user making the call)
3. Third Parameter is ServerUrl (This is the GCE server)

```
PS C:\test> Set-GceEnableUser $userGuid $cookie $url  
  
Success Reason          Context          Context1  
-----  
True
```

When it is successful, the Success field is true.

Synopsis

Enable an existing user

Syntax

```
Set-GceEnableUser [-UserID] <String> [-LoginToken] <String> [[-  
ServerUrl] <String>] [<CommonParameters>]
```

Description

Enable an existing user based on user's GUID.

Parameters

-UserID <String>

User GUID

Required?	True
Position?	0
Default value	
Accept pipeline input?	false

Accept wildcard characters?	False
-----------------------------	-------

-LoginToken <String>

Logged in user's cookie.

Required?	True
Position?	1
Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

-ServerUrl <String>

The remote service URL.

Required?	False
Position?	2
Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

<CommonParameters>

This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see about_CommonParameters (<http://go.microsoft.com/fwlink/?LinkID=113216>).

Inputs

Outputs

Result

When it is successful, the Success field is true.

Notes

Examples

Example 1

```
PS C:\> PS D:\> $result = Set-GceEnableUser "59434cc3-ee3a-4b19-a1a8-9b99fb374c4c" $login.Cookie http://localhost
PS D:\> $result
```

	Success	Reason
Context		Context1
-----	-----	-----
	True	

Related links

Set-GceDisableUser

This function will disable a specific user.

It takes 3 parameters.

1. First Parameter is UserID (This is user Guid)
2. Second Parameter is LoginToken (This is the Login Token represent the user making the call)
3. Third Parameter is ServerUrl (This is the GCE server)

```
PS C:\test> Set-GceDisableUser $userGuid $cookie $url
Success Reason          Context          Context1
-----
True
```

When it is successful, the Success field is true.

Synopsis

Disable an existing user.

Syntax

```
Set-GceDisableUser [-UserID] <String> [-LoginToken] <String> [[-
ServerUrl] <String>] [<CommonParameters>]
```

Description

Disable an existing user based on user's GUID (UserGuid).

Parameters

-UserID <String>

User GUID

Required?	True
Position?	0
Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

-LoginToken <String>

Logged in user's cookie.

Required?	True
Position?	1
Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

-ServerUrl <String>

The remote service address.

Required?	False
Position?	2
Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

<CommonParameters>

This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see about_CommonParameters (<http://go.microsoft.com/fwlink/?LinkID=113216>).

Inputs

Outputs

Result

When it is successful, the Success field is true.

Notes

Examples

Example 1

```
PS C:\> PS D:\> Set-GceDisableUser "59434cc3-ee3a-4b19-a1a8-9b99fb374c4c" $login.Cookie http://localhost
```

```
Context                                     Success Reason
-----                                     -
-----                                     -
True
```

Related links

Get-GceActiveUser

This function will return all active user for a specific tenant.

It takes 3 parameters.

1. First Parameter is DomainID (This is domain ID of tenant)
2. Second Parameter is LoginToken (This is the Login Token represent the user making the call)
3. Third Parameter is ServerUrl (This is the GCE server)

```
PS C:\test> Get-GceActiveUser $tenant.Guid $cookie $url
```

```
UserList : {}
Success  : True
Reason   :
Context  :
Context1 :
```

When it is successful, the Success field is true.

Synopsis

Get active user for specific tenant.

Syntax

```
Get-GceActiveUser [-DomainID] <String> [-LoginToken] <String> [[-  
ServerUrl] <String>] [<CommonParameters>]
```

Description

Get back the online users (web portal users) based on DomainID (TenantID). It will return a list of users for a specific tenant, when these users have a web browser open to the portal.

Parameters

-DomainID <String>

Tenant Guid

Required?	True
Position?	0
Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

-LoginToken <String>

Logged in user's security token, which is returned as the Cookie field from Get-GceLogin call.

Required?	True
Position?	1
Default value	
Accept pipeline input?	false

Accept wildcard characters?	False
-----------------------------	-------

-ServerUrl <String>

The remote service URL.

Required?	False
Position?	2
Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

<CommonParameters>

This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see about_CommonParameters (<http://go.microsoft.com/fwlink/?LinkID=113216>).

Inputs

Outputs

EnumUserResult

When it is successful, the Success field is true. Also the UserList field is a list of online users under the specific tenant.

Notes

Examples

Example 1

```
PS C:\> PS D:\> $users1 = Get-GceActiveUser "b633d35a-e8d2-4106-88eb-  
cef6a67216d6" $login.Cookie http://localhost
```

```
PS D:\> $users1
```

```
UserList : {joe'cc admin'dd}
Success  : True
Reason   :
Context  :
Context1 :
```

Related links

Get-GceNodePerformance

This function will return each cluster node performance.

It takes 2 parameters.

1. First Parameter is LoginToken (This is the Login Token represent the user making the call)
2. Second Parameter is ServerUrl (This is the GCE server)

```
PS C:\test> Get-GceNodePerformance $cookie $url
RequestCount           : 0
TotalRequestCount     : 0
FireChangesQueueLength : 0
PendingCompleteNodeRequests : 0
PendingNodeRequests   : 0
PollStatusQueueLength : 0
RegisteredClientCount  : 0
RequestTime           : 0
Success               : True
Reason                :
Context               :
Context1              :
```

When it is successful, the Success field is true.

Synopsis

Query for the server (worker node)'s performance data.

Syntax

```
Get-GceNodePerformance [-LoginToken] <String> [[-ServerUrl] <String>]
[<CommonParameters>]
```

Description

The call will return basic performance information related to a worker node (url parameter).

Parameters

-LoginToken <String>

User cookie.

Required?	True
Position?	0
Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

-ServerUrl <String>

The remote service URL.

Required?	False
Position?	1
Default value	
Accept pipeline input?	false
Accept wildcard characters?	False

<CommonParameters>

This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable,

OutBuffer, PipelineVariable, and OutVariable. For more information, see about_CommonParameters (<http://go.microsoft.com/fwlink/?LinkID=113216>).

Inputs

Outputs

NodePerformanceResult

It contains

TotalRequestCount, PendingCompleteNodeRequests, PollStatusQueueLength, RegisteredClientCount and RequestTime.

Notes

Examples

Example 1

```
PS C:\> PS D:\> $perf = Get-GceNodePerformance $login.Cookie
PS D:\> $perf
```

```
RequestCount           : 0
TotalRequestCount      : 4
FireChangesQueueLength : 0
PendingCompleteNodeRequests : 0
PendingNodeRequests    : 0
PollStatusQueueLength  : 0
RegisteredClientCount  : 0
RequestTime            : 129
Success                : True
Reason                 :
Context                :
Context1               :
```

Related links

Add-GceWorkerNode

This function will add a new cluster work node to the cluster.

It takes 6 parameters.

1. First Parameter is NodeName (This is name of new node)
2. Second Parameter is ExternalUrl (This is external url of new node)
3. Third Parameter is InternalUrl (This is internal url of new node)
4. Fourth Parameter is AllowMMC (This is flag to identity allow MMC)
5. Fifth Parameter is LoginToken (This is the Login Token represent the user making the call)
6. Sixth Parameter is ServerUrl (This is the GCE server)

Get-GceLocalStorageConfig

This function will return a base64 encoded storage configuration for local storage.

It takes 5 parameters.

1. First Parameter is StorageLocation (This is the path where to save files to local)
2. Second Parameter is Username (This is username who has permissions to access the path)
3. Third Parameter is Password (This is password for the user who has permissions to access the path)
4. Fourth Parameter is DataAtRest (This is flag to allow DARE)
5. Fifth Parameter is IsTeamUser (This is flag to identify current configuration is for tenant or for teamuser, it is not required, default is false which means tenant)

```
PS C:\test> $localConfig = Get-GceLocalStorageConfig "c:\storage" "test" "test" 1 0
PS C:\test> $localConfig
PD94bWwgdMvYc21vbj01M54wLi81bmNvZGluZz0idXRmLTg1Pz48U3RvcMFnZUR1c2NyaXB0b3I+PE5hbWU+SG9tZSBTdG9yYXd1PC90Yw11PjxHdw1kPjQxZWQ2YzI2LWQ1ZjgtNDI0ZT1hMzExLWJ1NGZmODgxND81NTwvR3VpZD48RW5kcG9pbmQ+aHR0cDovL2xvY2FsaG9zdDo4MDgwL2dsYWwRmaWx1c3ZyL2Yuc3ZjLzwvRw5kcG9pbmQ+PFN1cnZpY2VUeXB1Pjk8L1N1cnZpY2VUeXB1PjxRdW90YT4wPC9RdW90YT48R3hwax1Pjks0TKtMT1tMzEgMjM6NTk6NT1aPC9FeHBpcmlU+PE5hbWVBBG1hcyAvPjxvc2VyTmFtZT50ZXRnO0PC9vc2VyTmFtZT48U2VjcmV0S2V5PnRlc3Q8L1N1Y3J1dEt1eT48QnVja2V0IC8+PEFjY2Vzc1BvaW50PmM6XHN0b3JhZ2U8OFjY2Vzc1BvaW50PjxUcnV1Rw5jcn1wdG1vbj57VHJlZXR08L1RydWVfYmNyeXB0aw9uPjwvU3RvcMFnZUR1c2NyaXB0b3I+
```

It will return a base64 encoded string.

Get-GceAmazonStorageConfig

This function will return a base64 encoded storage configuration for Amazon cloud storage.

It takes 5 parameters.

1. First Parameter is AccessKey (This is AccessKey of Amazon account)
2. Second Parameter is SecretKey (This is SecretKey of Amazon account)
3. Third Parameter is Bucket (This is Bucket of Amazon account)
4. Fourth Parameter is DataAtRest (This is flag to allow DARE)
5. Fifth Parameter is IsTeamUser (This is flag to identify current configuration is for tenant or for teamuser, it is not required, default is false which means tenant)

```
PS C:\test> Get-GceAmazonStorageConfig "XjeiuDfhseyAxz" "sepyneFgshsPQ" "GladinetBucket" 1 1
PD94bWwgdmVyc2lvdj0iMS4wIi8lbnVzG1uz0idXRmlTgiPz48U3RvcmlFnZUR1c2NyaXB0b3I+PE5hbWU+R2xhZG1uZXQGU3RvcmlFnZTwvTmFtZT48R3VpZD45YjE1Yjh
jZj0zmkjk5LTQwZjItOTJkMi05ZjRiNjA5MzVhYzE8L0d1awQ+PEVuzHBvaw5OPmh0dHA6Ly9sb2NhbGhvc3Q6ODAw4MC9nbGFkb3B1bnMvZy5zdG90b3R5bWVudD48U2
J2awN1VHlwZT45PC9TZjJ2awN1VHlwZT48U3RvcmlFnZUR1c2NyaXB0b3I+PE5hbWU+R2xhZG1uZXQGU3RvcmlFnZTwvTmFtZT48R3VpZD45YjE1YjhjZj0zmkjk5LTQwZjItOTJkMi05ZjRiNjA5MzVhYzE8L0d1awQ+PEVuzHBvaw5OPmh0dHA6Ly9sb2NhbGhvc3Q6ODAw4MC9nbGFkb3B1bnMvZy5zdG90b3R5bWVudD48U2
2V5SwQ+WGp1aXVEZmhzZlBHo8L0FjY2Vzc0t1eU1kPjxTZWVzXRLZk+e1RydWV9PC9ucnV1Rw5jcnlwdG1vdj48L1N0b3JhZ2VEZXRjcm1wdG9yPg==
```

It will return a base64 encoded string.

Get-GceOpenStackStorageConfig

This function will return a base64 encoded storage configuration for OpenStack cloud storage.

It takes 6 parameters.

1. First Parameter is UserName (This is user name of OpenStack account)
2. Second Parameter is SecretKey (This is SecretKey of OpenStack account)
3. Third Parameter is Container (This is Container of OpenStack account)
4. Fourth Parameter is AccessPoint (This is access point of OpenStack account)
5. Fifth Parameter is DataAtRest (This is flag to allow DARE)
6. Sixth Parameter is IsTeamUser (This is flag to identify current configuration is for tenant or for teamuser, it is not required, default is false which means tenant)

```
PS C:\test> Get-GceOpenStackStorageConfig "XjeiuDfhseyAxz" "sepyneFgshsPQ" "Gladinet" "GladinetBucket" 1 1
PD94bWwgdmVyc2lvdj0iMS4wIi8lbnVzG1uz0idXRmlTgiPz48U3RvcmlFnZUR1c2NyaXB0b3I+PE5hbWU+R2xhZG1uZXQGU3RvcmlFnZTwvTmFtZT48R3VpZD43ZDA2Mzk
z0S0Z2DRhLTQwZjAtYTg0ZC0wODM1ZjE0NWEwZWE8L0d1awQ+PEVuzHBvaw5OPmh0dHA6Ly9sb2NhbGhvc3Q6ODAw4MC9nbGFkb3B1bnMvZy5zdG90b3R5bWVudD48U2
Yydm1jZVR5cGU+OTVwU2VydmljZVR5cGU+PFF1b3RHPjA8L1F1b3RHPjxFeHBpcmlU+OTk5OS0xMz0zMSA5Mzo1OT01OV08L0V4cG1yZT48TmFtZUFsawFzIC8+PEFjY2Vzc
0t1eU1kP1hqZW11RGZoc2V5QXh6PC9BY2N1c3NLZX1JZD48U2VjcmV052V5PnN1cHluzUZnc2hzUFE8L1N1Y3J1dEt1eT48QnVja2V0PkdsYWRpbmV0PC9cdWNRZXQ+PEZv
bGR1ciAvPjxBY2N1c3N0b21ud05HbGFkaW51dE1Y2t1dDwvQWVzXNzUG9bnQ+PE1zRG1yPmZhbHN1PC9jc0Rpcj48VHJ1ZUVuY3J5cHRpb24+e1RydWV9PC9ucnV1Rw5
jcnlwdG1vdj48L1N0b3JhZ2VEZXRjcm1wdG9yPg==
```

It will return a base64 encoded string.