**Five9 Cloud API PowerShell Module Documentation**

**Table of Contents**

1. Authentication & Connection
2. Function Categories
3. Complete Function Reference
4. Common Workflows
5. Best Practices
6. Troubleshooting

**Authentication & Connection**

**Initial Connection**

**Method 1: Interactive Authentication (Recommended for First Use)**

# Interactive mode - prompts for auth method selection

Connect-Five9Cloud -DomainId "12345678"

# You'll be prompted to choose:

# 1. API Control (OAuth2 with CustomerKey/SecretId)

# 2. Cloud Auth (Username/Password)

**Method 2: API Control Authentication (OAuth2)**

# Using CustomerKey and SecretId

Connect-Five9Cloud -DomainId "12345678" `

-CustomerKey "your-customer-key" `

-SecretId "your-secret-id" `

-AuthEndpoint "ApiControl" `

-Region "us"

**Method 3: Cloud Auth (Username/Password)**

# Using username and password

Connect-Five9Cloud -DomainId "12345678" `

-Username "admin@company.com" `

-Password "SecurePassword123!" `

-AuthEndpoint "CloudAuth" `

-Region "us"

**Using Stored Credentials**

# After saving credentials once, reconnect easily

Connect-Five9Cloud -DomainId "12345678" -ExistingAuthorization $true

**Function Categories**

**1. User Management (~25 functions)**

Core user operations including creation, modification, migration, and attribute management.

**2. Roles & Permissions (~30 functions)**

Role-based access control, permission assignments, and security management.

**3. Applications (~10 functions)**

Application access management and user application assignments.

**4. Identity Provider (IDP) (~15 functions)**

SSO configuration, IDP policies, and certificate management.

**5. Migration (~15 functions)**

User migration tools, migration groups, and fitness checks.

**6. Tags & Profiles (~10 functions)**

User categorization, profile management, and metadata.

**7. Security & Policies (~10 functions)**

Password policies, MFA configuration, and security settings.

**8. System Configuration (~5 functions)**

Domain settings, CORS, trusted IP ranges, and speed dials.

**Complete Function Reference**

**USER MANAGEMENT**

**Get-Five9CloudUser**

Retrieves user information with multiple query options.

# Get single user by UID

Get-Five9CloudUser -UserUID "usr\_123456"

# Get all users in domain

Get-Five9CloudUser

# Get users by application

Get-Five9CloudUser -AppId "app\_789"

# Get users with filtering and pagination

Get-Five9CloudUser -Filter "status eq 'Active'" -PageLimit 50

# Get user migration details

Get-Five9CloudUser -UserUID "usr\_123456" -Migration

# Get email verification status

Get-Five9CloudUser -EmailVerification

**Get-Five9CloudUserAttribute**

Retrieves specific user attributes and related data.

# Get user permissions

Get-Five9CloudUserAttribute -UserUID "usr\_123456" -Permissions

# Get user roles

Get-Five9CloudUserAttribute -UserUID "usr\_123456" -Roles

# Get user applications

Get-Five9CloudUserAttribute -UserUID "usr\_123456" -Applications

# Get user tags

Get-Five9CloudUserAttribute -UserUID "usr\_123456" -Tags

# Get user phone numbers

Get-Five9CloudUserAttribute -UserUID "usr\_123456" -Numbers

# Get user MFA factors

Get-Five9CloudUserAttribute -UserUID "usr\_123456" -MFAFactors

**Add-Five9CloudUser**

Creates a new user in the domain.

# Create standard user

Add-Five9CloudUser -Username "john.doe@company.com" `

-FirstName "John" `

-LastName "Doe" `

-Email "john.doe@company.com" `

-Status "Active" `

-Timezone "America/New\_York" `

-Locale "en-US"

# Create API-only user

Add-Five9CloudApiUser -FirstName "API" `

-LastName "Service" `

-Email "api@company.com" `

-Status "Active"

**Update-Five9CloudUser**

Modifies existing user information.

# Update user details

Update-Five9CloudUser -UserUID "usr\_123456" `

-Email "newemail@company.com" `

-MobileNumber "+1234567890" `

-Timezone "America/Los\_Angeles"

# Update user IDP policy

Update-Five9CloudUserIdpPolicy -UserUID "usr\_123456" `

-IdpPolicyId "idp\_policy\_789" `

-IdpFederationId "fed\_123"

**User Status Management**

# Activate user

Enable-Five9CloudUser -UserUID "usr\_123456"

# Deactivate user

Disable-Five9CloudUser -UserUID "usr\_123456"

# Force logout user

Disconnect-Five9CloudUser -UserUID "usr\_123456"

# Unlock user account

Unlock-Five9CloudUser -UserUID "usr\_123456"

**Password Management**

# Reset user password (sends email)

Reset-Five9CloudUserPassword -UserUID "usr\_123456" -SendEmail $true

# Expire user password

Expire-Five9CloudUserPassword -UserUID "usr\_123456" -ProvideTempPassword $true

# Reset MFA factors

Reset-Five9CloudUserMfaFactors -UserUID "usr\_123456"

**User Operations**

# Copy user (duplicate with new credentials)

Copy-Five9CloudUser -UserUID "usr\_template" -UserCreationData @{

username = "new.user@company.com"

firstName = "New"

lastName = "User"

}

# Make user API-only

Set-Five9CloudUserApiOnly -UserUID "usr\_123456"

# Make user application-only

Set-Five9CloudUserApplicationOnly -UserUID "usr\_123456"

# Remove user

Remove-Five9CloudUser -UserUID "usr\_123456"

**ROLES & PERMISSIONS**

**Get-Five9CloudRole**

Retrieves role information with various filters.

# Get all roles (custom and system)

Get-Five9CloudRole

# Get specific custom role

Get-Five9CloudRole -Role "CustomAgentRole" -Custom

# Get specific system role

Get-Five9CloudRole -Role "Administrator" -System

# Get all system roles only

Get-Five9CloudRole -SystemOnly

# Get role for user profile

Get-Five9CloudRole -UserProfileId "profile\_123"

**Get-Five9CloudPermission**

Retrieves permission information.

# Get all permissions

Get-Five9CloudPermission

# Get specific permission

Get-Five9CloudPermission -Permission "users.read"

# Get current user permissions

Get-Five9CloudPermission -CurrentUser

# Get permissions for a role

Get-Five9CloudPermission -Role "CustomRole"

# Get internal permissions

Get-Five9CloudPermission -Internal

**Role Management**

# Create new role

New-Five9CloudRole -Name "CustomSupervisor" `

-Description "Custom supervisor role with limited admin" `

-Permissions @("users.read", "reports.write", "recordings.access")

# Update role

Update-Five9CloudRole -Role "CustomSupervisor" `

-Description "Updated description" `

-Status "Active"

# Add permission to role

Add-Five9CloudPermissionToRole -Role "CustomSupervisor" `

-Permission "campaigns.manage"

# Remove permission from role

Remove-Five9CloudPermissionFromRole -Role "CustomSupervisor" `

-Permission "campaigns.manage"

# Delete role

Remove-Five9CloudRole -Role "CustomSupervisor"

**User Role Assignment**

# Assign role to user

Add-Five9CloudRoleToUser -UserUID "usr\_123456" -Role "Supervisor"

# Remove role from user

Remove-Five9CloudRoleFromUser -UserUID "usr\_123456" -Role "Agent"

# Copy roles between users

Copy-Five9CloudUserRoles -UserUID "usr\_new" -FromUser "usr\_template"

**User Permission Management**

# Add permission directly to user

Add-Five9CloudPermissionToUser -UserUID "usr\_123456" `

-Permission "reports.advanced"

# Remove permission from user

Remove-Five9CloudPermissionFromUser -UserUID "usr\_123456" `

-Permission "reports.advanced"

# Copy permissions between users

Copy-Five9CloudUserPermissions -UserUID "usr\_new" -FromUser "usr\_template"

**APPLICATIONS**

**Get-Five9CloudApplication**

Retrieves application information.

# Get all applications

Get-Five9CloudApplication

# Get specific application

Get-Five9CloudApplication -AppId "app\_123"

# Get domain-specific applications

Get-Five9CloudApplication -DomainSpecific

# Get current user's applications

Get-Five9CloudApplication -CurrentUser

# Get applications with filters

Get-Five9CloudApplication -Status "Active" -ApiOnly $true

**Application User Management**

# Add application to user

Add-Five9CloudApplicationToUser -UserUID "usr\_123456" -AppId "app\_789"

# Remove application from user

Remove-Five9CloudApplicationFromUser -UserUID "usr\_123456" -AppId "app\_789"

# Copy applications between users

Copy-Five9CloudUserApplications -UserUID "usr\_new" -FromUser "usr\_template"

**IDENTITY PROVIDER (IDP)**

**Get-Five9CloudIdpPolicy**

Manages IDP policies for SSO.

# Get all IDP policies

Get-Five9CloudIdpPolicy

# Get specific IDP policy

Get-Five9CloudIdpPolicy -IdpPolicyId "idp\_123"

# Get legacy IDP policies

Get-Five9CloudIdpPolicy -Legacy

# Get default policy

Get-Five9CloudIdpPolicy -DefaultPolicy $true

**IDP Policy Management**

# Create SAML IDP policy

Add-Five9CloudIdpPolicy -Name "CompanySSO" `

-Description "Main company SSO" `

-Saml @{

issuer = "https://idp.company.com"

ssoUrl = "https://idp.company.com/sso"

nameIdFormat = "email"

} `

-Certificate "-----BEGIN CERTIFICATE-----..."

# Update IDP policy

Update-Five9CloudIdpPolicy -IdpPolicyId "idp\_123" `

-Name "Updated SSO" `

-Description "Updated description"

# Set default IDP policy

Set-Five9CloudDefaultIdpPolicy -IdpPolicyId "idp\_123"

# Remove default status

Remove-Five9CloudDefaultIdpPolicy -IdpPolicyId "idp\_123"

**Get-Five9CloudIdpCertificate**

Manages certificates for IDP policies.

# List certificates for policy

Get-Five9CloudIdpCertificate -IdpPolicyId "idp\_123"

# Get specific certificate

Get-Five9CloudIdpCertificate -IdpPolicyId "idp\_123" `

-IdpCertificateId "cert\_456"

# Get legacy certificates

Get-Five9CloudIdpCertificate -IdpPolicyId "idp\_123" -Legacy

**Certificate Management**

# Add certificate to IDP policy

Add-Five9CloudIdpPolicyCertificate -IdpPolicyId "idp\_123" `

-Certificate "-----BEGIN CERTIFICATE-----..."

# Activate certificate

Enable-Five9CloudIdpCertificate -IdpPolicyId "idp\_123" `

-IdpCertificateId "cert\_456"

# Remove certificate

Remove-Five9CloudIdpPolicyCertificate -IdpPolicyId "idp\_123" `

-IdpCertificateId "cert\_456"

**MIGRATION**

**Get-Five9CloudMigrationGroup**

Manages user migration groups.

# Get all migration groups

Get-Five9CloudMigrationGroup

# Get specific group

Get-Five9CloudMigrationGroup -GroupId "mig\_123"

# Get users in migration group

Get-Five9CloudMigrationGroup -GroupId "mig\_123" -Users

# Get groups with user statistics

Get-Five9CloudMigrationGroup -UserStats $true

**Migration Group Management**

# Create migration group

Add-Five9CloudMigrationGroup -Name "Phase1Migration" `

-StartDate (Get-Date).AddDays(7) `

-Type "Pilot" `

-IdpPolicyId "idp\_123"

# Update migration group

Update-Five9CloudMigrationGroup -GroupId "mig\_123" `

-Status "Active" `

-StartDate (Get-Date)

# Add user to migration group

Add-Five9CloudUserToMigrationGroup -GroupId "mig\_123" `

-UserUID "usr\_456"

# Remove user from migration group

Remove-Five9CloudUserFromMigrationGroup -GroupId "mig\_123" `

-UserUID "usr\_456"

# Apply filter to group

Set-Five9CloudMigrationGroupFilter -GroupId "mig\_123" `

-Matcher "role eq 'Agent'"

**Domain Migration**

# Enable domain migration

Enable-Five9CloudDomainMigration -UserType "agents"

# Check migration status

Get-Five9CloudDomainMigrationDetails -FitnessStats $true -UserStats $true

# Sync migration

Sync-Five9CloudDomainMigration -UserType "agents"

# Pause migration

Disable-Five9CloudDomainMigration -UserType "agents"

# Complete migration

Complete-Five9CloudDomainMigration -UserType "agents"

# Rollback if needed

Undo-Five9CloudDomainMigration

**User Migration**

# Convert user to new system

Convert-Five9CloudUser -UserUID "usr\_123456"

# Check user migration details

Get-Five9CloudUserMigrationDetails -UserUID "usr\_123456"

# Resolve user fitness issues

Resolve-Five9CloudUserFitness -UserUID "usr\_123456" -UserType "agents"

# Rollback user migration

Undo-Five9CloudUserMigration -UserUID "usr\_123456"

**TAGS & PROFILES**

**Get-Five9CloudTag**

Manages user tags for categorization.

# Get all tags

Get-Five9CloudTag

# Get specific tag

Get-Five9CloudTag -TagId "tag\_123"

# Get current user's tags

Get-Five9CloudTag -CurrentUser

**Tag Management**

# Create new tag

New-Five9CloudTag -Name "VIPSupport" `

-Description "Users handling VIP customers"

# Update tag

Set-Five9CloudTag -TagId "tag\_123" `

-Name "PremiumSupport" `

-Description "Updated description"

# Add tag to user

Add-Five9CloudTagToUser -UserUID "usr\_123456" -TagId "tag\_789"

# Remove tag from user

Remove-Five9CloudTagFromUser -UserUID "usr\_123456" -TagId "tag\_789"

# Delete tag

Remove-Five9CloudTag -TagId "tag\_123"

**Get-Five9CloudUserProfile**

Manages user profiles for bulk settings.

# Get all user profiles

Get-Five9CloudUserProfile

# Get users with specific profile

Get-Five9CloudUserProfile -UserProfileId "prof\_123" -Users

# Get profiles with filters

Get-Five9CloudUserProfile -Name "AgentProfile" -Filter "status eq 'Active'"

**Profile Management**

# Create user profile

Add-Five9CloudProfile -Name "RemoteAgents" `

-Description "Profile for remote agents" `

-Timezone "America/New\_York" `

-Locale "en-US"

# Update profile

Update-Five9CloudProfile -UserProfileId "prof\_123" `

-Name "UpdatedProfile" `

-Timezone "America/Los\_Angeles"

# Associate user with profile

Add-Five9CloudUserProfile -UserProfileId "prof\_123" `

-UserUID "usr\_456"

# Remove user from profile

Remove-Five9CloudUserProfile -UserProfileId "prof\_123" `

-UserUID "usr\_456"

# Add permission to profile

Add-Five9CloudPermissionToUserProfile -UserProfileId "prof\_123" `

-Permission "reports.view"

# Add application to profile

Add-Five9CloudApplicationToUserProfile -UserProfileId "prof\_123" `

-AppId "app\_789"

**SECURITY & POLICIES**

**Get-Five9CloudPasswordPolicy**

Manages password policies.

# Get domain password policy

Get-Five9CloudPasswordPolicy

# Get current user's password policy

Get-Five9CloudPasswordPolicy -CurrentUser

**Password Policy Management**

# Update password policy

Update-Five9CloudPasswordPolicy `

-Complexity @{

minLength = 12

requireUppercase = $true

requireLowercase = $true

requireNumbers = $true

requireSpecialChars = $true

} `

-Lockout @{

maxAttempts = 5

lockoutDuration = 30

} `

-Age @{

maxAgeDays = 90

minAgeDays = 1

historyCount = 5

}

# Reset to defaults

Reset-Five9CloudPasswordPolicy

**Get-Five9CloudMFAFactor / Get-Five9CloudMFAPolicy**

Manages multi-factor authentication.

# Get MFA policy

Get-Five9CloudMFAPolicy

# Get available MFA factors

Get-Five9CloudMFAFactor

# Get specific factor

Get-Five9CloudMFAFactor -FactorId "totp"

**MFA Management**

# Create MFA policy

Add-Five9CloudMFAPolicy

# Set factor in MFA policy

Set-Five9CloudFactorInMFAPolicy -MfaPolicyId "mfa\_123" `

-FactorId "totp" `

-Enrollment "Required"

# User MFA enrollment

Register-Five9CloudMobileFactor

Register-Five9CloudEmailFactor -Email "user@company.com"

Register-Five9CloudSmsFactor -PhoneNumber "+1234567890"

# Activate MFA factors

Enable-Five9CloudMobileFactor -EnrolledFactorId "enr\_123" -PassCode "123456"

Enable-Five9CloudEmailFactor -EnrolledFactorId "enr\_456" -PassCode "654321"

# Unenroll MFA

Unregister-Five9CloudMobileFactor -EnrolledFactorId "enr\_123"

**Security Settings**

# Get security events

Get-Five9CloudSecurityEvents -UserUID "usr\_123456" `

-Filter "eventType eq 'LOGIN\_FAILED'" `

-Sort @("-timestamp")

# Change user password (self-service)

Set-Five9CloudMyPassword -OldPassword "OldPass123!" `

-NewPassword "NewPass456!" `

-RevokeSessions $true

**SYSTEM CONFIGURATION**

**Get-Five9CloudScope**

Manages API scopes and permissions.

# Get all scopes

Get-Five9CloudScope

# Get domain-specific scopes

Get-Five9CloudScope -ForDomain

# Get specific scopes

Get-Five9CloudScope -Scopes "users.read,users.write"

**Get-Five9CloudTrustedRange**

Manages trusted IP ranges.

# Get trusted IP ranges

Get-Five9CloudTrustedRange

**Trusted Range Management**

# Set trusted IP ranges

Set-Five9CloudTrustedRanges -Enabled $true `

-IpRanges @(

@{ start = "192.168.1.1"; end = "192.168.1.255" },

@{ start = "10.0.0.1"; end = "10.0.0.255" }

)

**CORS Configuration**

# Get allowed CORS origins

Get-Five9CloudAllowedCorsOrigins

# Set allowed CORS origins

Set-Five9CloudAllowedCorsOrigins -Items @(

"https://app.company.com",

"https://portal.company.com"

)

**Get-Five9CloudSpeedDial**

Manages speed dial configurations.

# Get all speed dials

Get-Five9CloudSpeedDial

# Get specific speed dial

Get-Five9CloudSpeedDial -SpeedDialId "sd\_123"

**Speed Dial Management**

# Create speed dial

Add-Five9CloudSpeedDial -Code "\*99" `

-DialedNumber "+18001234567" `

-Description "Customer Support Hotline" `

-EmergencyNumber $false

# Update speed dial

Update-Five9CloudSpeedDial -SpeedDialId "sd\_123" `

-Code "\*88" `

-DialedNumber "+18009876543"

# Remove speed dial

Remove-Five9CloudSpeedDial -SpeedDialId "sd\_123"

**VERINT INTEGRATION**

**Get-Five9CloudVerintSetting**

Manages Verint WFO settings.

# Get user's Verint settings

Get-Five9CloudVerintSetting -UserUID "usr\_123456"

**Verint Settings Management**

# Add Verint settings to user

Add-Five9CloudVerintSettings -UserUID "usr\_123456" `

-Packages @(

'call-recording',

'call-recording.quality-monitoring',

'workforce-management'

)

# Update Verint settings

Update-Five9CloudVerintSettings -UserUID "usr\_123456" `

-Packages @(

'call-recording',

'call-recording.quality-monitoring',

'call-recording.speech-analytics',

'workforce-management',

'performance-management'

)

# Remove Verint settings

Remove-Five9CloudVerintSettings -UserUID "usr\_123456"

**EMAIL VERIFICATION**

# Check domain email verification status

Get-Five9CloudDomainEmailVerificationStatus

# Enable email verification

Enable-Five9CloudDomainEmailVerification

# Disable email verification

Disable-Five9CloudDomainEmailVerification

# Initiate user email verification

Start-Five9CloudUserEmailVerification

# Confirm email token

Confirm-Five9CloudEmailToken -Token "verification\_token\_here"

**Common Workflows**

**New Employee Onboarding**

# 1. Create the user

$newUser = Add-Five9CloudUser -Username "jane.smith@company.com" `

-FirstName "Jane" -LastName "Smith" `

-Email "jane.smith@company.com" `

-Status "Active" -Timezone "America/Chicago"

# 2. Assign role

Add-Five9CloudRoleToUser -UserUID $newUser.userUID -Role "Agent"

# 3. Add to appropriate profile

Add-Five9CloudUserProfile -UserProfileId "prof\_agents" `

-UserUID $newUser.userUID

# 4. Add required applications

Add-Five9CloudApplicationToUser -UserUID $newUser.userUID `

-AppId "app\_five9"

# 5. Configure MFA

Update-Five9CloudUser -UserUID $newUser.userUID `

-MfaPolicyId "mfa\_standard"

# 6. Add tags for reporting

Add-Five9CloudTagToUser -UserUID $newUser.userUID -TagId "tag\_newHire"

Add-Five9CloudTagToUser -UserUID $newUser.userUID -TagId "tag\_team\_sales"

# 7. Send password reset email

Reset-Five9CloudUserPassword -UserUID $newUser.userUID -SendEmail $true

**Bulk User Migration to SSO**

# 1. Create migration group

$migGroup = Add-Five9CloudMigrationGroup -Name "SSO\_Migration\_Q1" `

-StartDate (Get-Date).AddDays(7) `

-Type "Production" `

-IdpPolicyId "idp\_companysso"

# 2. Get users to migrate

$users = Get-Five9CloudUser -Filter "status eq 'Active' and role eq 'Agent'"

# 3. Add users to migration group

foreach ($user in $users.items) {

Add-Five9CloudUserToMigrationGroup -GroupId $migGroup.id `

-UserUID $user.userUID

}

# 4. Start migration

Enable-Five9CloudDomainMigration -UserType "agents"

# 5. Monitor progress

do {

$status = Get-Five9CloudDomainMigrationDetails -UserStats $true

Write-Host "Migrated: $($status.migratedCount) of $($status.totalCount)"

Start-Sleep -Seconds 60

} while ($status.status -eq "InProgress")

# 6. Complete migration

Complete-Five9CloudDomainMigration -UserType "agents"

**Security Audit**

# 1. Check password policy

$pwPolicy = Get-Five9CloudPasswordPolicy

Write-Host "Password Policy Review:"

Write-Host "Min Length: $($pwPolicy.complexity.minLength)"

Write-Host "Max Age: $($pwPolicy.age.maxAgeDays) days"

# 2. Review admin users

$admins = Get-Five9CloudUser -Role "Administrator"

Write-Host "`nAdministrators: $($admins.items.Count)"

# 3. Check MFA enrollment

foreach ($admin in $admins.items) {

$mfa = Get-Five9CloudUserAttribute -UserUID $admin.userUID -MFAFactors

if ($mfa.factors.Count -eq 0) {

Write-Warning "Admin $($admin.username) has no MFA enrolled!"

}

}

# 4. Review recent security events

$events = Get-Five9CloudSecurityEvents `

-Filter "timestamp ge '$(Get-Date -Date (Get-Date).AddDays(-7) -Format yyyy-MM-dd)'" `

-Sort @("-timestamp")

$failedLogins = $events.items | Where-Object { $\_.eventType -eq "LOGIN\_FAILED" }

Write-Host "`nFailed logins in past 7 days: $($failedLogins.Count)"

# 5. Check trusted IP ranges

$trustedIPs = Get-Five9CloudTrustedRange

if (-not $trustedIPs.enabled) {

Write-Warning "Trusted IP ranges not enabled!"

}

**Offboarding Employee**

function Remove-Five9CloudEmployee {

param(

[Parameter(Mandatory=$true)]

[string]$UserUID,

[switch]$PreserveForAudit

)

# 1. Force logout

Disconnect-Five9CloudUser -UserUID $UserUID

# 2. Revoke all sessions

Expire-Five9CloudUserPassword -UserUID $UserUID

# 3. Remove from all groups

$groups = Get-Five9CloudMigrationGroup

foreach ($group in $groups.items) {

$groupUsers = Get-Five9CloudMigrationGroup -GroupId $group.id -Users

if ($UserUID -in $groupUsers.items.userUID) {

Remove-Five9CloudUserFromMigrationGroup -GroupId $group.id -UserUID $UserUID

}

}

# 4. Remove applications

$apps = Get-Five9CloudUserAttribute -UserUID $UserUID -Applications

foreach ($app in $apps.items) {

Remove-Five9CloudApplicationFromUser -UserUID $UserUID -AppId $app.id

}

# 5. Remove roles

$roles = Get-Five9CloudUserAttribute -UserUID $UserUID -Roles

foreach ($role in $roles.items) {

Remove-Five9CloudRoleFromUser -UserUID $UserUID -Role $role.name

}

if ($PreserveForAudit) {

# 6a. Deactivate for audit trail

Disable-Five9CloudUser -UserUID $UserUID

# 7a. Add terminated tag

Add-Five9CloudTagToUser -UserUID $UserUID -TagId "tag\_terminated"

Write-Host "User deactivated and tagged for audit"

} else {

# 6b. Permanently delete

Remove-Five9CloudUser -UserUID $UserUID

Write-Host "User permanently removed"

}

}

# Usage

Remove-Five9CloudEmployee -UserUID "usr\_123456" -PreserveForAudit

**Best Practices**

1. **Connection Management**

# Always test connection before operations

if (Test-Five9CloudConnection) {

# Perform operations

} else {

Connect-Five9Cloud -DomainId "12345678" -ExistingAuthorization $true

}

# Implement token refresh in long-running scripts

function Ensure-Five9CloudConnection {

if (-not (Test-Five9CloudConnection)) {

Write-Host "Token expired, reconnecting..."

Connect-Five9Cloud -DomainId $global:Five9CloudToken.DomainId -ExistingAuthorization $true

}

}

1. **Error Handling**

# Wrap operations in try-catch blocks

try {

$user = Get-Five9CloudUser -UserUID "usr\_123456"

if ($user) {

Update-Five9CloudUser -UserUID $user.userUID -Status "Active"

}

} catch {

Write-Error "Failed to update user: $\_"

# Log error or take corrective action

}

1. **Bulk Operations**

# Use pagination for large datasets

$allUsers = @()

$pageCursor = $null

do {

$params = @{ PageLimit = 100 }

if ($pageCursor) { $params['PageCursor'] = $pageCursor }

$response = Get-Five9CloudUser @params

$allUsers += $response.items

$pageCursor = $response.nextPageCursor

} while ($pageCursor)

1. **Filtering and Sorting**

# Use OData filters for efficient queries

Get-Five9CloudUser -Filter "status eq 'Active' and role eq 'Agent'"

# Sort results

Get-Five9CloudUser -Sort @("lastName", "-createdDate") # - prefix for descending

**Troubleshooting**

**Common Issues and Solutions**

**Authentication Failures**

# Issue: "Failed to authenticate"

# Solution: Verify credentials and domain ID

Connect-Five9Cloud -DomainId "12345678" -CustomerKey "key" -SecretId "secret" -Verbose

# Check for detailed error messages

$Error[0].Exception.Message

**Token Expiration**

# Issue: "Token expired"

# Solution: Implement automatic reconnection

function Invoke-Five9CloudOperation {

param([scriptblock]$Operation)

$maxRetries = 2

$retryCount = 0

while ($retryCount -lt $maxRetries) {

try {

if (-not (Test-Five9CloudConnection)) {

Connect-Five9Cloud -DomainId $global:Five9CloudToken.DomainId -ExistingAuthorization $true

}

return & $Operation

} catch {

$retryCount++

if ($retryCount -ge $maxRetries) { throw }

}

}

}

**Rate Limiting**

# Issue: "429 Too Many Requests"

# Solution: Implement throttling

function Invoke-ThrottledRequest {

param(

[scriptblock]$Request,

[int]$DelayMs = 100

)

Start-Sleep -Milliseconds $DelayMs

try {

& $Request

} catch {

if ($\_.Exception.Response.StatusCode -eq 429) {

Write-Warning "Rate limited, waiting..."

Start-Sleep -Seconds 5

& $Request

} else {

throw

}

}

}

**Missing Permissions**

# Check current user permissions

$myPerms = Get-Five9CloudPermission -CurrentUser

$requiredPerm = "users.write"

if ($requiredPerm -notin $myPerms.permissions) {

Write-Error "Missing required permission: $requiredPerm"

Write-Host "Contact your administrator to grant this permission"

}

**Debug Mode**

# Enable verbose output for troubleshooting

$VerbosePreference = "Continue"

Get-Five9CloudUser -UserUID "usr\_123456" -Verbose

# Capture full error details

$ErrorActionPreference = "Stop"

try {

# Your operation here

} catch {

$\_ | Format-List -Force

$\_.Exception | Format-List -Force

}

**Appendix**

**Module Information**

# Get module version

Get-Module PSFive9CloudAPI | Select-Object Version

# List all available functions

Get-Command -Module PSFive9CloudAPI | Sort-Object Name

**Support Resources**

* PowerShell Module Repository: https://github.com/justingray82/PSFive9CloudAPI
* Issue Tracker: https://github.com/justingray82/PSFive9CloudAPI/issues

*Last Updated:* *Module Version: 1.2* *Total Functions: ~120*