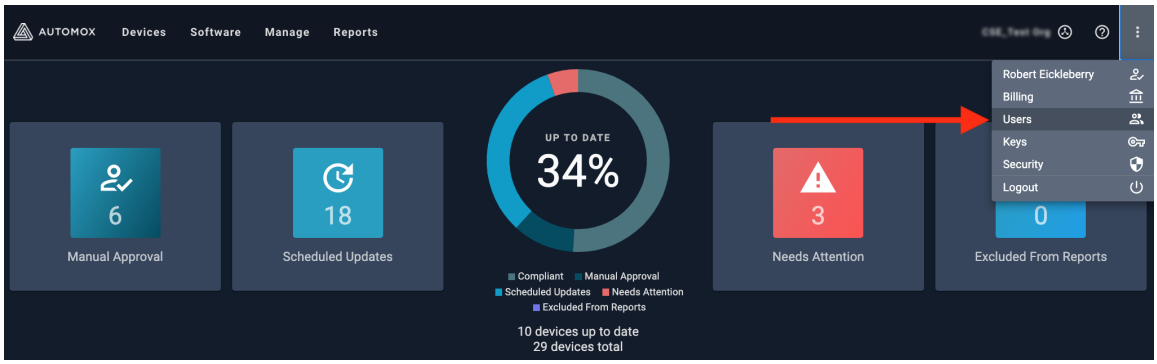
Account Management Using the Automox Console or PowerShell

Follow these best practices to clean up unnecessary user accounts.

Account Management Using the Automox Console

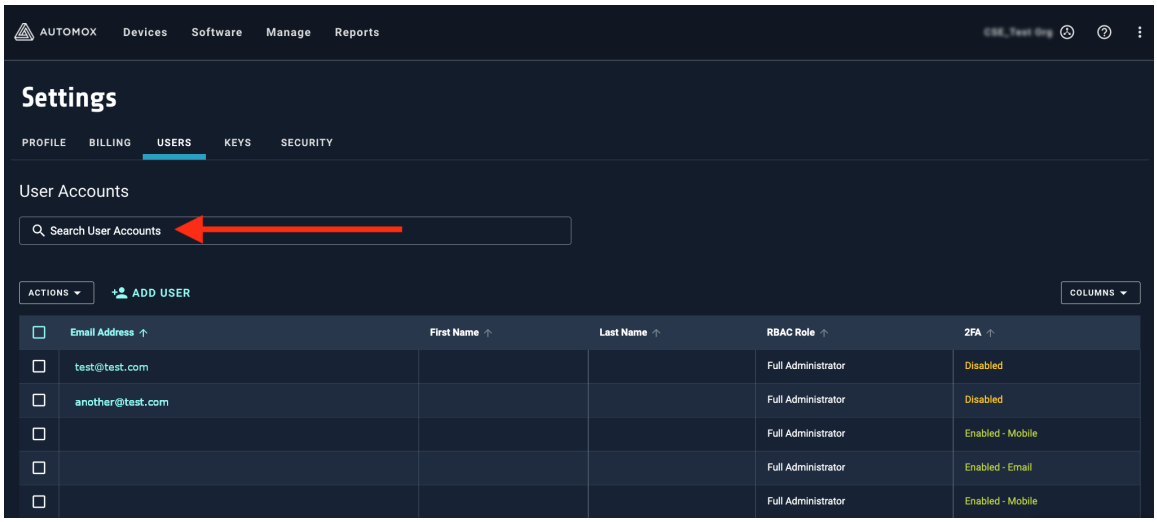
Follow these steps to clean up unnecessary user accounts using the Automox console.

1. Go to **Settings** (⋮) in the console.
2. Select **Users** from the menu.

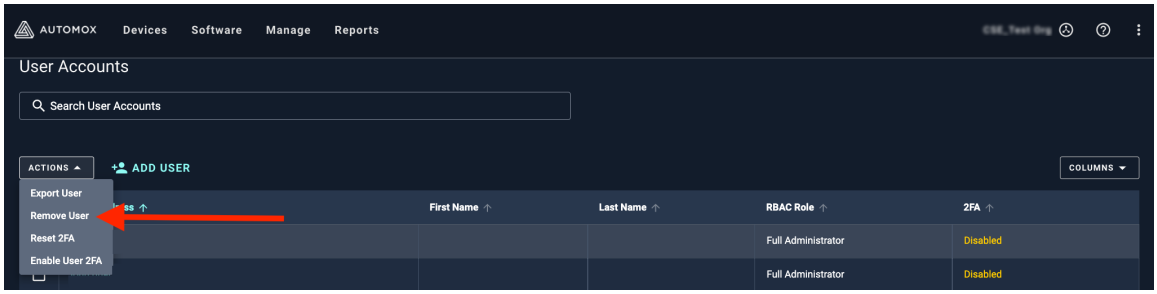


3. Review the **User Accounts** list and select any unnecessary accounts.

   You can also use the search bar to filter the list.



4. Select the **Actions** drop-down menu and click **Remove User**.



5. Confirm the selected accounts so that only approved user accounts remain in the console.

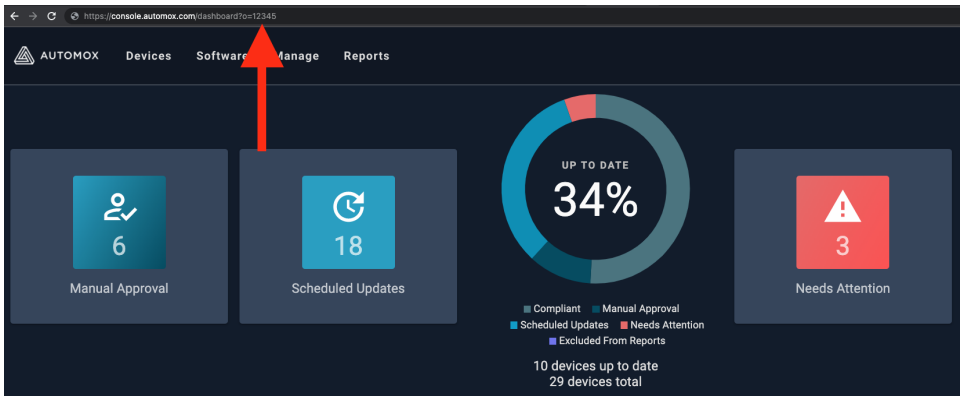
Account Management Using PowerShell

Follow these steps to delete multiple users from an organization based on the user email addresses using a PowerShell script. This logs the successes and failures in a location that you specify.

There are four areas in the code that you'll need to update to get the script to function:

1. `$orgID` = 'YOUR_ORG_ID'

   You will need your Org ID, which can be found by looking at the URL. Select the value after the "?o="



> **Note:** The user must be a Global or Zone administrator to manage API keys for a zone. See Managing Keys for more information.

2. `$apiKey` = 'YOUR_API_KEY'

   In your console, go to **Settings > Keys** and select the API key. Note that the API key is per user, so that another user will have different API keys.

3. `$logPath`

   Set this variable to the folder where you want to save logs

4. `$toDelete`

   Set your list of users you want deleted by their email addresses

Delete Multiple Users Script

```
### Replace the variables in this block with your Org ID & API key ###

$orgID = 'YOUR_ORG_ID'
$apiKey = 'YOUR_API_KEY'


# Modify log path as desired
$logPath = 'C:\Temp\'


# List of users identified by email to delete
$toDelete = @('user@email.com','user2@email.com','user3@email.com')
```

```
#########################################################################


$page = 0
$limit = 500
$headers = @{ "Authorization" = "Bearer $apiKey" }


# Easier to maintain, especially if multiple organizations, or for repurposing for different API Tables
$apiInstance = 'https://console.automox.com/api/'
$apiTable = 'users'


# Initialize empty arrays to store server IDs
$users = @()
$delList = @()
$failDelete = @()
$successDelete = @()


# Generate list of all users
while($true) {


    $orgAndKey = "?o=$orgID&api_key=$apiKey&l=$limit&p=$page"


    #put components together
    $uri = $apiInstance + $apiTable + $orgAndKey


    #Get the json body of the Web Request
    $resp = (Invoke-WebRequest -Method GET -Uri $uri -UseBasicParsing).Content | ConvertFrom-Json


    $users += $resp
    $page += 1


    if($resp.count -lt $limit) {
        break
    }
}


# Narrow down to list of users that need to be deleted
ForEach ($user in $toDelete) {


    $match = $users | Where-Object {$_.email -EQ $user}
    $delList += $match
}


# Delete designated users. Log whether successful or failed to delete.
ForEach ($delUser in $delList) {
    try {
        $id = $delUser.id
        $email = $delUser.email
        $name = $delUser.firstname+' '+$delUser.lastname
        $url = "https://console.automox.com/api/users/$id"+"?o=$orgID"
        Invoke-WebRequest -Method Delete -Uri $url -Headers $headers
        $successDelete += @{"Email"=$email;"ID"=$id;"Name"=$name}
```

```
        Write-Output "Successfully Deleted User: $email"
    }
    catch {
        $failDelete += @{"Email"=$email;"ID"=$id;"Name"=$name}
        Write-Output "Failed to Delete User: $email"
    }
}


# Output logging into json files for later review/manipulation
$successDelete | ConvertTo-Json | Out-File $logPath\Delete_Success.json
$failDelete | ConvertTo-Json | Out-File $logPath\Delete_Failed.json
```

Related Topics:

- Community: Delete Multiple Users From An Organization