

### Azure Lab 03: Role-Based Access Control

#### Lab scenario

I have been asked to create a proof of concept showing how Azure users and groups are created. Also, how role-based access control is used to assign roles to groups. Specifically, I need to:

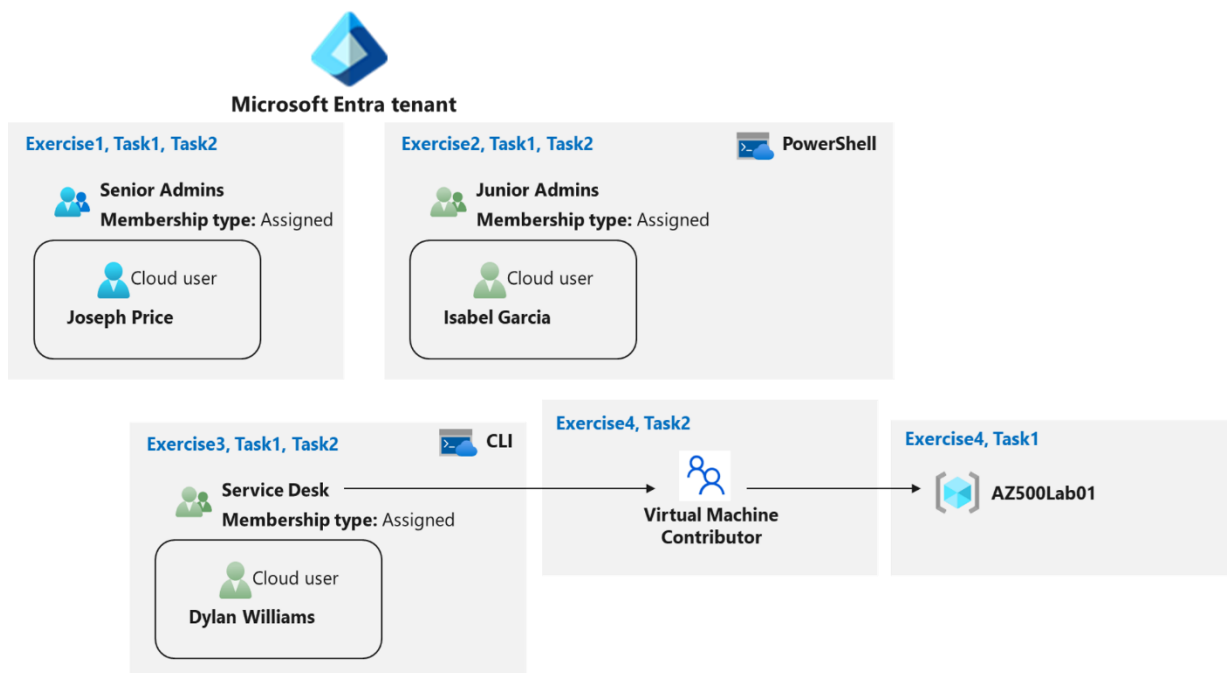
- Create a Senior Admins group containing the user account of Joseph Price as its member.
- Create a Junior Admins group containing the user account of Isabel Garcia as its member.
- Create a Service Desk group containing the user account of Dylan Williams as its member.
- Assign the Virtual Machine Contributor role to the Service Desk group.

#### Lab objectives

In this lab, you will complete the following exercises:

- Exercise 1: Create the Senior Admins group with the user account Joseph Price as its member (the Azure portal).
- Exercise 2: Create the Junior Admins group with the user account Isabel Garcia as its member (PowerShell).
- Exercise 3: Create the Service Desk group with the user Dylan Williams as its member (Azure CLI).
- Exercise 4: Assign the Virtual Machine Contributor role to the Service Desk group.

#### Role-Based Access Control architecture diagram

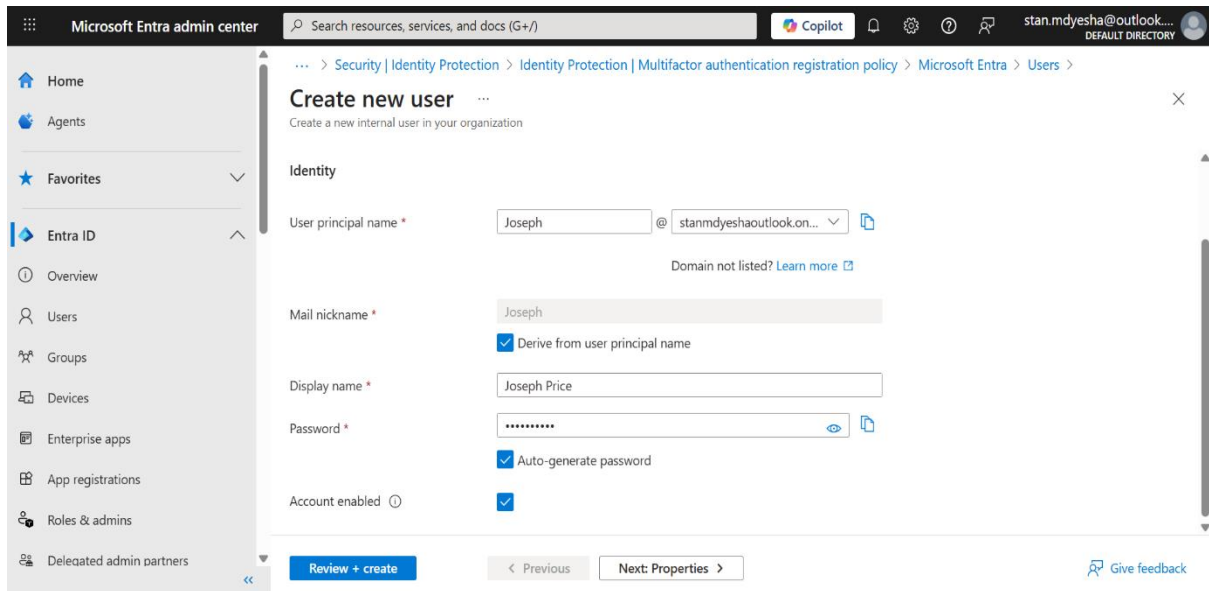


## **Exercise 1: Create the Senior Admins group with the user account Joseph Price as its member.**

In this exercise, I will complete the following tasks:

- Task 1: Use the Azure portal to create a user account for Joseph Price.
- Task 2: Use the Azure portal to create a Senior Admins group and add the user account of Joseph Price to the group.

### **Task 1: Use the Azure portal to create a user account for Joseph Price**



Microsoft Entra admin center

Search resources, services, and docs (G+/)

Copilot

stan.mdyesha@outlook...  
DEFAULT DIRECTORY

... > Security | Identity Protection > Identity Protection | Multifactor authentication registration policy > Microsoft Entra > Users >

### Create new user

Create a new internal user in your organization

Identity

User principal name \* Joseph @ stanmdyeshaoutlook.on...  
Domain not listed? [Learn more](#)

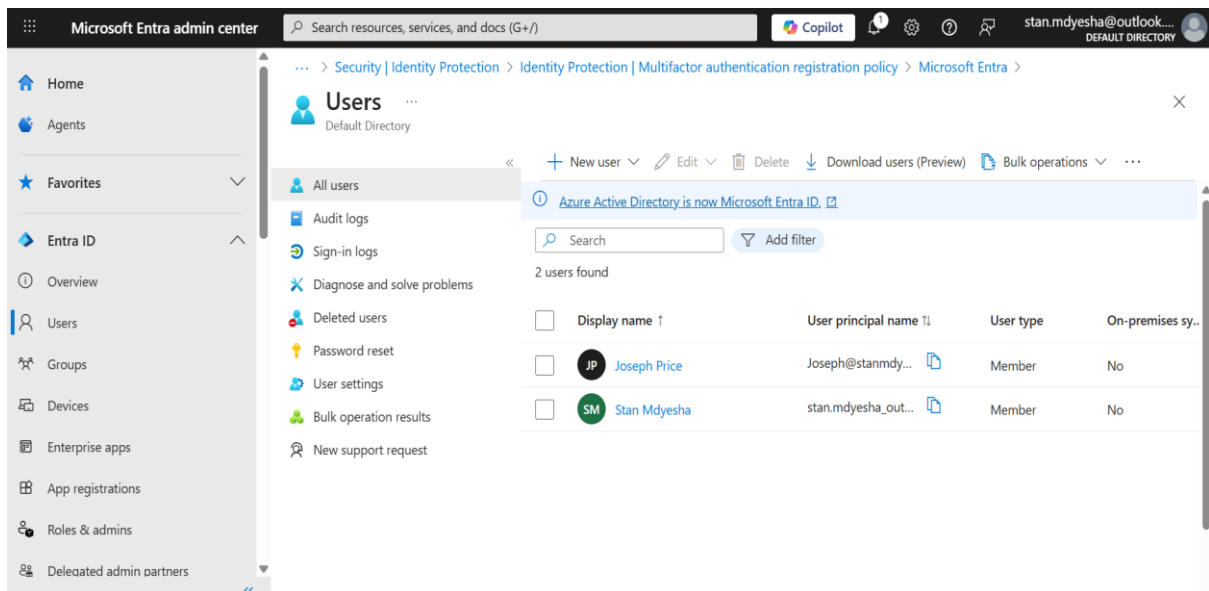
Mail nickname \* Joseph  
☒ Derive from user principal name

Display name \* Joseph Price

Password \*   
☒ Auto-generate password

Account enabled ☒

[Review + create](#) [Previous](#) [Next: Properties](#) [Give feedback](#)



Microsoft Entra admin center

Search resources, services, and docs (G+/)

Copilot

stan.mdyesha@outlook...  
DEFAULT DIRECTORY

... > Security | Identity Protection > Identity Protection | Multifactor authentication registration policy > Microsoft Entra >

### Users

Default Directory

+ New user Edit Delete Download users (Preview) Bulk operations

[Azure Active Directory is now Microsoft Entra ID.](#)

Search Add filter

2 users found

| <input type="checkbox"/> | Display name ↑  | User principal name ↕ | User type | On-premises sy.. |
|--------------------------|-----------------|-----------------------|-----------|------------------|
| <input type="checkbox"/> | JP Joseph Price | Joseph@stanmdy...     | Member    | No               |
| <input type="checkbox"/> | SM Stan Mdyesha | stan.mdyesha_out...   | Member    | No               |

Navigation links: All users, Audit logs, Sign-in logs, Diagnose and solve problems, Deleted users, Password reset, User settings, Bulk operation results, New support request

## Task2: Use the Azure portal to create a Senior Admins group and add the user account of Joseph Price to the group.

Microsoft Entra admin center

Search resources, services, and docs (G+/)

Copilot

stan.mdyesha@outlook...  
DEFAULT DIRECTORY

... > Identity Protection | Multifactor authentication registration policy > Microsoft Entra > Users > Groups | Overview >

### New Group

Got feedback?

Group type \* ⓘ  
Security

Group name \* ⓘ  
Senior Admins ✓

Group description ⓘ  
Enter a description for the group

Membership type ⓘ  
Assigned

Owners  
No owners selected

Members

Create

Microsoft Entra admin center

Search resources, services, and docs (G+/)

Copilot

stan.mdyesha@outlook...  
DEFAULT DIRECTORY

### Add members

Try changing or adding filters if you don't see what you're looking for.

Search ⓘ

36 results found

All Users Groups Devices Enterprise applications

|                                     | Name                                       | Type             | Details                                   |
|-------------------------------------|--------------------------------------------|------------------|-------------------------------------------|
| <input type="checkbox"/>            | AA AAD App Management                      | Enterprise ap... | f0ae4899-d877-4d3c-ae25-679e38eea492      |
| <input checked="" type="checkbox"/> | JP Joseph Price                            | User             | Joseph@stanmdyeshaoutlook.onmicrosoft.com |
| <input type="checkbox"/>            | AR AAD Request Verification Service - P... | Enterprise ap... | c728155f-7b2a-4502-a08b-b8af9b269319      |

Select

Microsoft Entra admin center

Search resources, services, and docs (G+/)

Copilot

stan.mdyesha@outlook...  
DEFAULT DIRECTORY

... > Security | Identity Protection > Identity Protection | Multifactor authentication registration policy > Microsoft Entra > Users > Groups

### Groups | All groups

Default Directory

New group Download groups Refresh Manage view Delete

Search

Add filter

Search mode Contains

1 group found

|                          | Name ↑           | Object Id                           |
|--------------------------|------------------|-------------------------------------|
| <input type="checkbox"/> | SA Senior Admins | adcf755-d04a-4f61-92b2-937eb9e483c8 |

Overview

All groups

Deleted groups

Diagnose and solve problems

Settings

General

Expiration

Naming policy

Activity

Privileged Identity Management

Access reviews

Audit logs

Bulk operation results

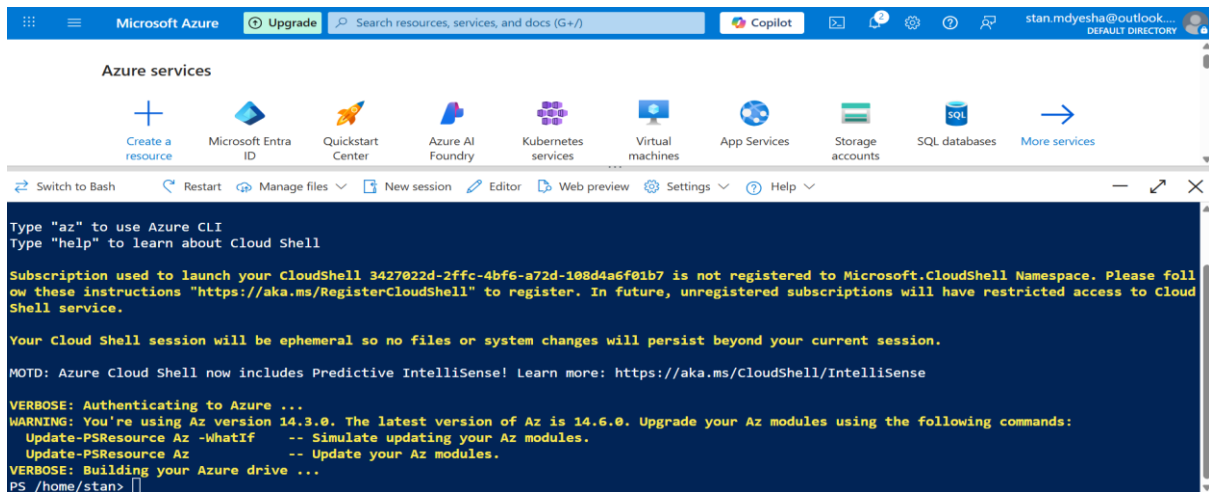
## Exercise 2: Create a Junior Admins group containing the user account of Isabel Garcia as its member.

In this exercise, you will complete the following tasks:

- Task 1: Use PowerShell to create a user account for Isabel Garcia.
- Task 2: Use PowerShell to create the Junior Admins group and add the user account of Isabel Garcia to the group.

### Task 1: Use PowerShell to create a user account for Isabel Garcia.

I set up and connected to PowerShell via the Azure platform.



```

Microsoft Azure | Upgrade | Search resources, services, and docs (G+/) | Copilot | stan.mdyesha@outlook... | DEFAULT DIRECTORY

Azure services
+ Create a resource | Microsoft Entra ID | Quickstart Center | Azure AI Foundry | Kubernetes services | Virtual machines | App Services | Storage accounts | SQL databases | More services

Switch to Bash | Restart | Manage files | New session | Editor | Web preview | Settings | Help

Type "az" to use Azure CLI
Type "help" to learn about Cloud Shell

Subscription used to launch your CloudShell 3427022d-2ffc-4bf6-a72d-108d4a6f01b7 is not registered to Microsoft.CloudShell Namespace. Please follow these instructions "https://aka.ms/RegisterCloudShell" to register. In future, unregistered subscriptions will have restricted access to Cloud Shell service.

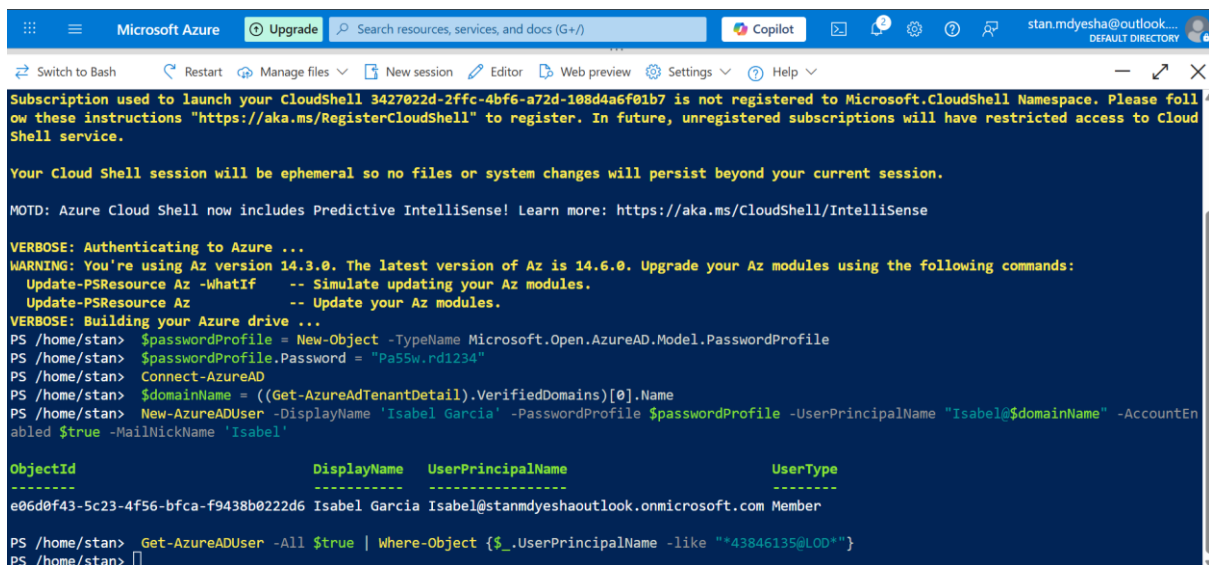
Your Cloud Shell session will be ephemeral so no files or system changes will persist beyond your current session.

MOTD: Azure Cloud Shell now includes Predictive IntelliSense! Learn more: https://aka.ms/CloudShell/IntelliSense

VERBOSE: Authenticating to Azure ...
WARNING: You're using Az version 14.3.0. The latest version of Az is 14.6.0. Upgrade your Az modules using the following commands:
  Update-PSResource Az -WhatIf -- Simulate updating your Az modules.
  Update-PSResource Az -- Update your Az modules.
VERBOSE: Building your Azure drive ...
PS /home/stan>
  
```

In the below screenshot I ran commands/scripts to achieve the following

- Create a password profile object
- Set the value of the password within the profile object
- Connect to Microsoft Entra ID
- Identify the name of your Microsoft Entra tenant
- Create a user account for Isabel Garcia
- List Microsoft Entra ID users



```

Microsoft Azure | Upgrade | Search resources, services, and docs (G+/) | Copilot | stan.mdyesha@outlook... | DEFAULT DIRECTORY

Switch to Bash | Restart | Manage files | New session | Editor | Web preview | Settings | Help

Subscription used to launch your CloudShell 3427022d-2ffc-4bf6-a72d-108d4a6f01b7 is not registered to Microsoft.CloudShell Namespace. Please follow these instructions "https://aka.ms/RegisterCloudShell" to register. In future, unregistered subscriptions will have restricted access to Cloud Shell service.

Your Cloud Shell session will be ephemeral so no files or system changes will persist beyond your current session.

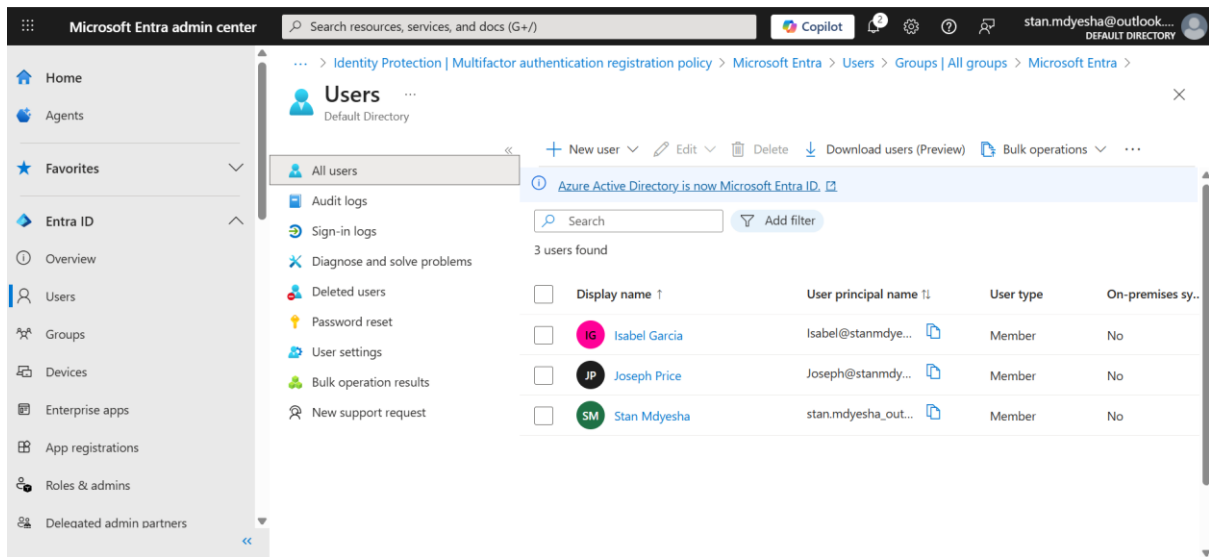
MOTD: Azure Cloud Shell now includes Predictive IntelliSense! Learn more: https://aka.ms/CloudShell/IntelliSense

VERBOSE: Authenticating to Azure ...
WARNING: You're using Az version 14.3.0. The latest version of Az is 14.6.0. Upgrade your Az modules using the following commands:
  Update-PSResource Az -WhatIf -- Simulate updating your Az modules.
  Update-PSResource Az -- Update your Az modules.
VERBOSE: Building your Azure drive ...
PS /home/stan> $passwordProfile = New-Object -TypeName Microsoft.Open.AzureAD.Model.PasswordProfile
PS /home/stan> $passwordProfile.Password = "Pa55w.rd1234"
PS /home/stan> Connect-AzureAD
PS /home/stan> $domainName = ((Get-AzureAdTenantDetail).VerifiedDomains)[0].Name
PS /home/stan> New-AzureADUser -DisplayName 'Isabel Garcia' -PasswordProfile $passwordProfile -UserPrincipalName 'Isabel@$domainName' -AccountEnabled $true -MailNickName 'Isabel'

ObjectId                                DisplayName    UserPrincipalName                                UserType
-----                                -
e06d0f43-5c23-4f56-bfca-f9438b0222d6 Isabel Garcia Isabel@stanmdyeshaoutlook.onmicrosoft.com Member

PS /home/stan> Get-AzureADUser -All $true | Where-Object {$_.UserPrincipalName -like "*43846135@L0D*"}
PS /home/stan>
  
```

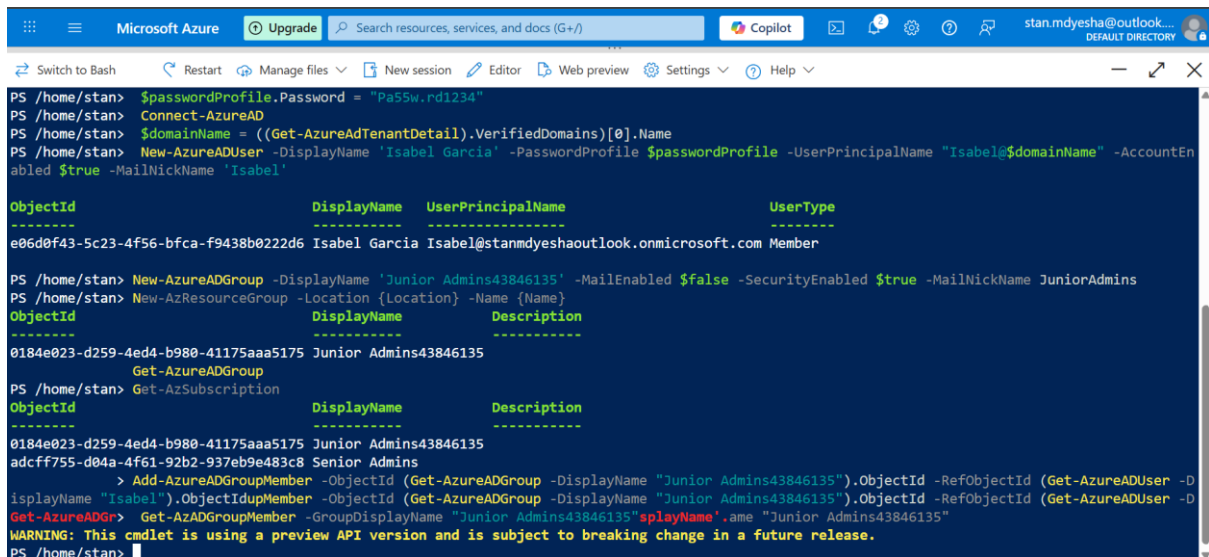
I successfully added **Isabel Garcia** to my list of users through PowerShell.

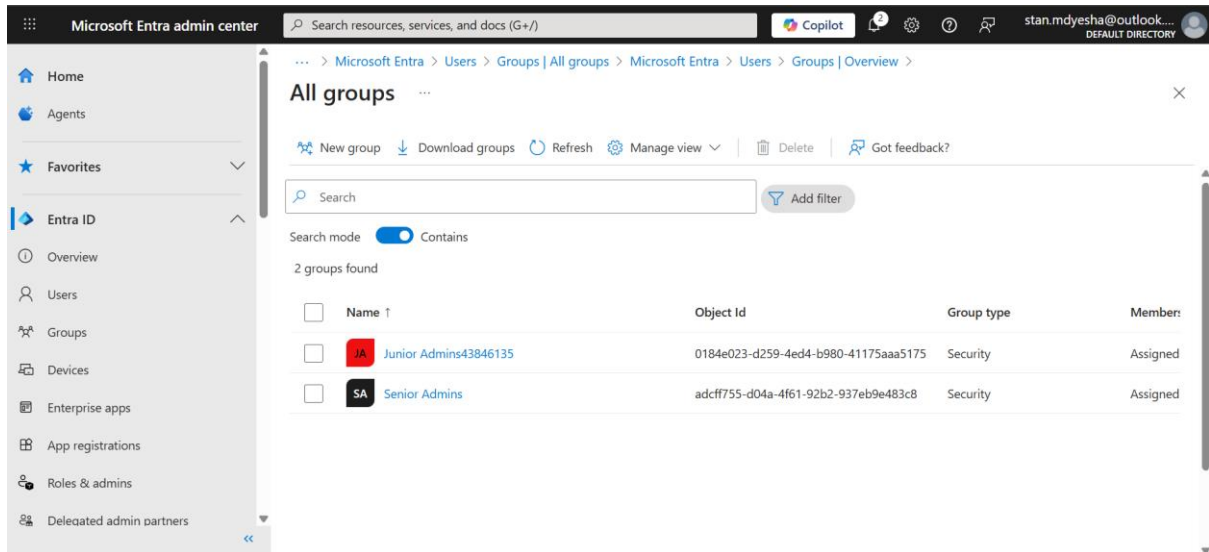


## Task2: Use PowerShell to create the Junior Admins group and add the user account of Isabel Garcia to the group.

In the below screenshot I ran commands/scripts to achieve the following

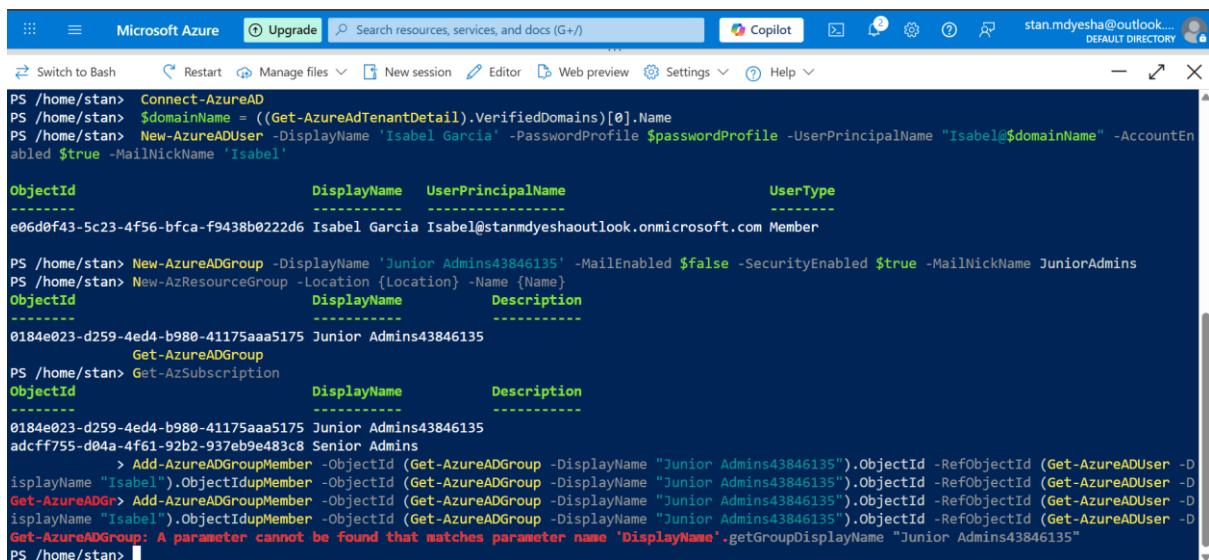
- **Create a new security group** named Junior Admins
- **List groups** in your Microsoft Entra tenant (the list should include the Senior Admins and Junior Admins groups)
- **Obtain a reference** to the user account of Isabel Garcia
- Add the user account of Isabel to the Junior Admins43846135 group
- Verify that the Junior Admins43846135 group contains the user account of Isabel





I noticed that Isabel was not successfully added to the Junior Admin group. I was getting this error:

Get-AzureADGroup: A parameter cannot be found that matches parameter name 'DisplayName'.getGroupDisplayName "Junior Admins43846135"





After researching potential solutions, I ran into this new error:

**Add-AzureADGroupMember: Cannot bind argument to parameter 'RefObjectId' because it is null.**

This error indicates that either the group or user wasn't found, resulting in a \$null value for the RefObjectId.

I ran a script to then determine if the group exists. The group does exist but seemingly not with the exact same name.

```
Microsoft Azure | Upgrade | Search resources, services, and docs (G+) | Copilot | stan.mdyesha@outlook... | DEFAULT DIRECTORY

Switch to Bash | Restart | Manage files | New session | Editor | Web preview | Settings | Help

Add-AzureADGroupMember -ObjectId (Get-AzureADGroup -DisplayName "Junior Admins43846135").ObjectId -RefObjectId (Get-AzureADUser -D
isplayName "Isabel").ObjectId -Action {Action} -IpAddress {IpAddress} -Name {Name} -Priority {Priority} -ResourceGroupName {Re
Get-AzureADGroupConnect-AzureAD -Action {Action} -IpAddress {IpAddress} -Name {Name} -Priority {Priority} -ResourceGroupName {Re
parameter name 'DisplayName'.
PS /home/stan> $User = Get-AzureADUser -Filter "DisplayName eq 'Isabel'"
PS /home/stan> $Group = Get-AzureADGroup -Filter "DisplayName eq 'Junior Admins43846135'"SPRODMCA.onmicrosoft.com"
PS /home/stan> Add-AzureADGroupMember -ObjectId $Group.ObjectId -RefObjectId $User.ObjectId
Add-AzureADGroupMember: Cannot bind argument to parameter 'RefObjectId' because it is null.ame {Name} -Priority {Priority} -ResourceGroupName {Re
PS /home/stan> Connect-MgGraph -Scopes "Group.ReadWrite.All","User.ReadWrite.All"
To sign in, use a web browser to open the page https://microsoft.com/devicelogin and enter the code C5YFXZ9YT to authenticate.
Connect-MgGraph -Scopes "Group.ReadWrite.All","User.ReadWrite.All"ease try again.
To sign in, use a web browser to open the page https://microsoft.com/devicelogin and enter the code CMPXANYQE to authenticate.
Connect-MgGraph -Scopes "Group.ReadWrite.All","User.ReadWrite.All"ease try again.
PS /home/stan> $Group = Get-AzureADGroup -Filter "DisplayName eq 'Junior Admins43846135'"
PS /home/stan> $User = Get-AzureADUser -Filter "DisplayName eq 'Isabel'"
PS /home/stan> Add-AzureADGroupMember -ObjectId $Group.ObjectId -RefObjectId $User.ObjectId
Add-AzureADGroupMember: Cannot bind argument to parameter 'RefObjectId' because it is null.ame {Name} -Priority {Priority} -ResourceGroupName {Re
PS /home/stan> Write-Host "=== Searching for Group ===" -ForegroundColor Yellow
=== Searching for Group ===akeAnalyticsJob -Account "ContosoAdlAccount" -JobId "a0a78d72-3fa8-4564-9b18-6becb3fda48a"
PS /home/stan> if ($Group) {AzureADGroup -Filter "DisplayName eq 'Junior Admins43846135'"
>> Write-Host "Group FOUND: $($Group.DisplayName)" -ForegroundColor Green
>> Write-Host "ObjectId: $($Group.ObjectId)" -ForegroundColor Green
>> } else {
>> Write-Host "Group NOT found with exact name 'Junior Admins43846135'" -ForegroundColor Red
>> } Get-AzureADGroup | Where-Object {$_.DisplayName -like "*Junior*" -or $_.DisplayName -like "*Admin*"} | Select-Object DisplayName, ObjectId
Group FOUND: Junior Admins43846135ject {$_.DisplayName -like "*Junior*" -or $_.DisplayName -like "*Admin*"} | Select-Object DisplayName,
ObjectId: 0184e023-d259-4ed4-b980-41175aa5175
PS /home/stan>
```

I then ran a script to determine if "Isabel" existed. Isabel does exist.

```
Microsoft Azure | Upgrade | Search resources, services, and docs (G+) | Copilot | stan.mdyesha@outlook... | DEFAULT DIRECTORY

Switch to Bash | Restart | Manage files | New session | Editor | Web preview | Settings | Help

To sign in, use a web browser to open the page https://microsoft.com/devicelogin and enter the code C5YFXZ9YT to authenticate.
Connect-MgGraph -Scopes "Group.ReadWrite.All","User.ReadWrite.All"ease try again.
To sign in, use a web browser to open the page https://microsoft.com/devicelogin and enter the code CMPXANYQE to authenticate.
Connect-MgGraph -Scopes "Group.ReadWrite.All","User.ReadWrite.All"ease try again.
PS /home/stan> $Group = Get-AzureADGroup -Filter "DisplayName eq 'Junior Admins43846135'"
PS /home/stan> $User = Get-AzureADUser -Filter "DisplayName eq 'Isabel'"
PS /home/stan> Add-AzureADGroupMember -ObjectId $Group.ObjectId -RefObjectId $User.ObjectId
Add-AzureADGroupMember: Cannot bind argument to parameter 'RefObjectId' because it is null.ame {Name} -Priority {Priority} -ResourceGroupName {Re
PS /home/stan> Write-Host "=== Searching for Group ===" -ForegroundColor Yellow
=== Searching for Group ===akeAnalyticsJob -Account "ContosoAdlAccount" -JobId "a0a78d72-3fa8-4564-9b18-6becb3fda48a"
PS /home/stan> if ($Group) {AzureADGroup -Filter "DisplayName eq 'Junior Admins43846135'"
>> Write-Host "Group FOUND: $($Group.DisplayName)" -ForegroundColor Green
>> Write-Host "ObjectId: $($Group.ObjectId)" -ForegroundColor Green
>> } else {
>> Write-Host "Group NOT found with exact name 'Junior Admins43846135'" -ForegroundColor Red
>> } Get-AzureADGroup | Where-Object {$_.DisplayName -like "*Junior*" -or $_.DisplayName -like "*Admin*"} | Select-Object DisplayName, ObjectId
Group FOUND: Junior Admins43846135ject {$_.DisplayName -like "*Junior*" -or $_.DisplayName -like "*Admin*"} | Select-Object DisplayName,
ObjectId: 0184e023-d259-4ed4-b980-41175aa5175
PS /home/stan> Wait-AzDataLakeAnalyticsJob -Account "ContosoAdlAccount" -JobId "a0a78d72-3fa8-4564-9b18-6becb3fda48a"
=== Searching for User ===
PS /home/stan> if ($User) {AzureADUser -Filter "DisplayName eq 'Isabel'"
>> Write-Host "User FOUND: $($User.DisplayName)" -ForegroundColor Green
>> Write-Host "ObjectId: $($User.ObjectId)" -ForegroundColor Green
>> } else {
>> Write-Host "User NOT found with exact name 'Isabel'" -ForegroundColor Red
>> } Get-AzureADUser | Where-Object {$_.DisplayName -like "*Isabel*"} | Select-Object DisplayName, UserPrincipalName, ObjectId
Get-AzureADUser | Where-Object {$_.DisplayName -like "*Isabel*"} | Select-Object DisplayName, UserPrincipalName,
```

Attempted various methods to add Isabel to the Junior Admin group including the following:

```

Microsoft Azure | Upgrade | Search resources, services, and docs (G+) | Copilot | stan.mdyesha@outlook... | DEFAULT DIRECTORY

Switch to Bash | Restart | Manage files | New session | Editor | Web preview | Settings | Help

ObjectID: 0184eWrite-Host "n=== Searching for User ===" -ForegroundColor Yellow
PS /home/stan> Wait-AzDataLakeAnalyticsJob -Account "ContosoAdlAccount" -JobId "a0a78d72-3fa8-4564-9b18-6becb3fda48a"
=== Searching for User ===
PS /home/stan> $user = Get-AzureADUser -Filter "DisplayName eq 'Isabel'"
PS /home/stan> Write-Host "n=== Searching for User ===" -ForegroundColor Yellow
PS /home/stan> Wait-AzDataLakeAnalyticsJob -Account "ContosoAdlAccount" -JobId "a0a78d72-3fa8-4564-9b18-6becb3fda48a"
=== Searching for User ===
PS /home/stan> if ($user) {AzureADUser -Filter "DisplayName eq 'Isabel'"
>> Write-Host "User FOUND: $($user.DisplayName)" -ForegroundColor Green
>> Write-Host "ObjectID: $($user.ObjectID)" -ForegroundColor Green
>> } else {
>> Write-Host "User NOT found with exact name 'Isabel'" -ForegroundColor Red
>> } Get-AzureADUser | Where-Object {$_.DisplayName -like "*Isabel*"} | Select-Object DisplayName, UserPrincipalName, ObjectID
User NOT found with exact name 'Isabel'$.DisplayName -like "*Isabel*"} | Select-Object DisplayName, UserPrincipalName,
Searching for users with 'Isabel' in the name...

DisplayName    UserPrincipalName    ObjectID
-----
Isabel Garcia  Isabel.Garcia@stanmdyeshaooutlook.onmicrosoft.com  e06d0f43-5c23-4f56-bfca-f9438b0222d6
>> if ($group -and $user) {
>> Write-Host "n=== Adding User to Group ===" -ForegroundColor Cyanhod {Method} -Path {Path}
>> Add-AzureADGroupMember -ObjectID $group.ObjectID -RefObjectID $user.ObjectID
>> Write-Host "SUCCESS: Added $($user.DisplayName) to $($group.DisplayName)" -ForegroundColor Green
>> } Write-Host "nCannot proceed - either group or user was not found." -ForegroundColor Red
>> Write-Host "nCannot proceed - either group or user was not found." -ForegroundColor Red
Cannot proceed - either group or user was not found.
PS /home/stan>

```

Finally, success... thanks to Stack Overflow, and DeepSeek.

The key issues and solutions were:

1. **Parameter Syntax:** Get-AzureADGroup uses -Filter parameter, not -DisplayName
2. **Exact Names Matter:** Had to use "Isabel Garcia" instead of just "Isabel" – I don't know how I missed this one.
3. **Step-by-step Debugging:** Checking each variable before proceeding

```

Microsoft Azure | Upgrade | Search resources, services, and docs (G+) | Copilot | stan.mdyesha@outlook... | DEFAULT DIRECTORY

Switch to Bash | Restart | Manage files | New session | Editor | Web preview | Settings | Help

>> $group = $allGroups | Where-Object {$_.DisplayName -like "*Junior Admin*"}
>>
>> if ($group) {
>> Write-Host "Group FOUND with similar name: $($group.DisplayName)" -ForegroundColor Green
>> Write-Host "ObjectID: $($group.ObjectID)" -ForegroundColor Green
>> } } $allGroups | Select-Object DisplayName, ObjectID | Sort-Object DisplayName | Format-Table
Group FOUND: Junior Admins43846135ect DisplayName, ObjectID | Sort-Object DisplayName | Format-Tab
ObjectID: 0184eWrite-Host "n=== Getting User ===" -ForegroundColor Yellow
PS /home/stan> Wait-AzDataLakeAnalyticsJob -Account "ContosoAdlAccount" -JobId "a0a78d72-3fa8-4564-9b18-6becb3fda48a"
=== Getting User ===
PS /home/stan> if ($user) {AzureADUser -Filter "DisplayName eq 'Isabel Garcia'"
>> Write-Host "User FOUND: $($user.DisplayName)" -ForegroundColor Green
>> Write-Host "ObjectID: $($user.ObjectID)" -ForegroundColor Green
>> } Write-Host "User NOT found!" -ForegroundColor Red
User FOUND: Isabel GarciaOT found!" -ForegroundColor Red
ObjectID: e06> if ($group -and $user) {
>> Write-Host "n=== Adding User to Group ===" -ForegroundColor Cyanhod {Method} -Path {Path}
>> Write-Host "Adding user $($user.DisplayName) to group $($group.DisplayName)" -ForegroundColor Yellow
>> Add-AzureADGroupMember -ObjectID $group.ObjectID -RefObjectID $user.ObjectID
>> Write-Host "SUCCESS: Added $($user.DisplayName) to $($group.DisplayName)" -ForegroundColor Green
>> } Write-Host "nCannot proceed - either group or user was not found." -ForegroundColor Red
>> Write-Host "nCannot proceed - either group or user was not found." -ForegroundColor Red
=== Adding User to Group ===
Adding user Isabel Garcia to group Junior Admins43846135
SUCCESS: Added Isabel Garcia to Junior Admins43846135
PS /home/stan>

```



Microsoft Entra admin center

Search resources, services, and docs (G+)

Home > All groups > Junior Admins43846135 >

## Members

Junior Admins43846135

+ Add members Bulk operations Refresh Manage view Remove Got feedback?

Direct members All members

Search Add filter

1 group member found

| Name          | Type | Email |
|---------------|------|-------|
| Isabel Garcia | User |       |

## Task 2: Use Azure CLI to create the Service Desk group

Microsoft Azure Upgrade

Search resources, services, and docs (G+)

Switch to PowerShell Restart Manage files New session Editor Web preview Settings Help

Requesting a Cloud Shell. Succeeded. Connecting terminal...

Subscription used to launch your CloudShell 3427022d-2ff0-4bf6-a72d-108d4a6f01b7 is not registered to Microsoft.CloudShell Namespace. Please follow these instructions "https://aka.ms/RegisterCloudShell" to register. In future, unregistered subscriptions will have restricted access to Cloud Shell service.

Your Cloud Shell session will be ephemeral so no files or system changes will persist beyond your current session.

```
stan [ ~ ]$ az ad group create --display-name "Service Desk" --mail-nickname "ServiceDesk"
{
  "@odata.context": "https://graph.microsoft.com/v1.0/$metadata#groups/$entity",
  "classification": null,
  "createdDateTime": "2025-11-06T11:07:26Z",
  "creationOptions": [],
  "deletedDateTime": null,
  "description": null,
  "displayName": "Service Desk",
  "expirationDateTime": null,
  "groupTypes": [],
  "id": "d52206f2-9ace-43b3-bc7c-7f93ea380c46",
  "isAssignableToRole": null,
  "mail": null,
  "mailEnabled": false,
  "mailNickname": "ServiceDesk",
  "membershipRule": null,
  "membershipRuleProcessingState": null,
  "onPremisesDomainName": null,
  "onPremisesLastSyncDateTime": null,
  "onPremisesNetBiosName": null,
}
```

Microsoft Entra admin center

Search resources, services, and docs (G+)

Home > Groups | Overview > All groups >

## Service Desk

Group

Delete Got feedback?

Overview

Diagnose and solve problems

Manage

- Properties
- Members
- Owners
- Roles and administrators
- Administrative units
- Group memberships
- Applications
- Licenses
- Azure role assignments

Activity

Basic information

SD Service Desk

Membership type Assigned

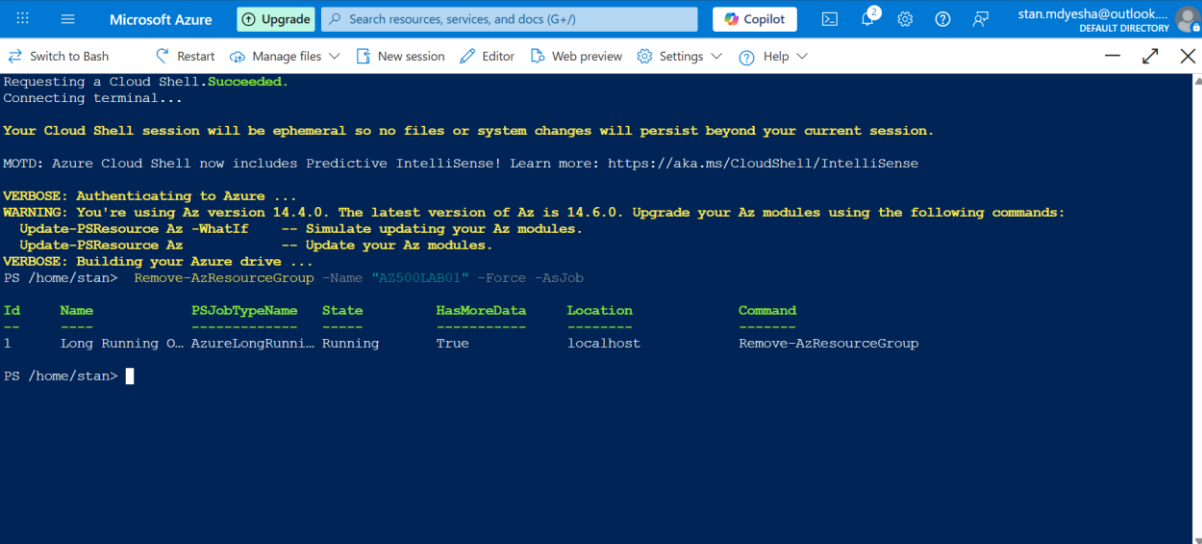
Source Cloud

Type Security

Object ID d52206f2-9ace-43b3-bc7c-7f93ea380c46

Created on 11/6/2025, 1:07 PM

## Clean up resources



The screenshot shows the Microsoft Azure Cloud Shell interface. The top bar includes the Microsoft Azure logo, an 'Upgrade' button, a search bar, and a 'Copilot' button. The user's email 'stan.mdyesha@outlook...' is visible in the top right. The interface has a dark blue background with a terminal window. The terminal shows the following output:

```

Requesting a Cloud Shell. Succeeded.
Connecting terminal...

Your Cloud Shell session will be ephemeral so no files or system changes will persist beyond your current session.

MOTD: Azure Cloud Shell now includes Predictive IntelliSense! Learn more: https://aka.ms/CloudShell/IntelliSense

VERBOSE: Authenticating to Azure ...
WARNING: You're using Az version 14.4.0. The latest version of Az is 14.6.0. Upgrade your Az modules using the following commands:
  Update-PSResource Az -WhatIf -- Simulate updating your Az modules.
  Update-PSResource Az -- Update your Az modules.
VERBOSE: Building your Azure drive ...
PS /home/stan> Remove-AzResourceGroup -Name "AZ500LAB01" -Force -AsJob

```

| Id | Name              | PSJobTypeName     | State   | HasMoreData | Location  | Command                |
|----|-------------------|-------------------|---------|-------------|-----------|------------------------|
| 1  | Long Running O... | AzureLongRunni... | Running | True        | localhost | Remove-AzResourceGroup |

PS /home/stan> █