

# Onboard Automator: Streamlining Employee Onboarding with Azure Logic Apps

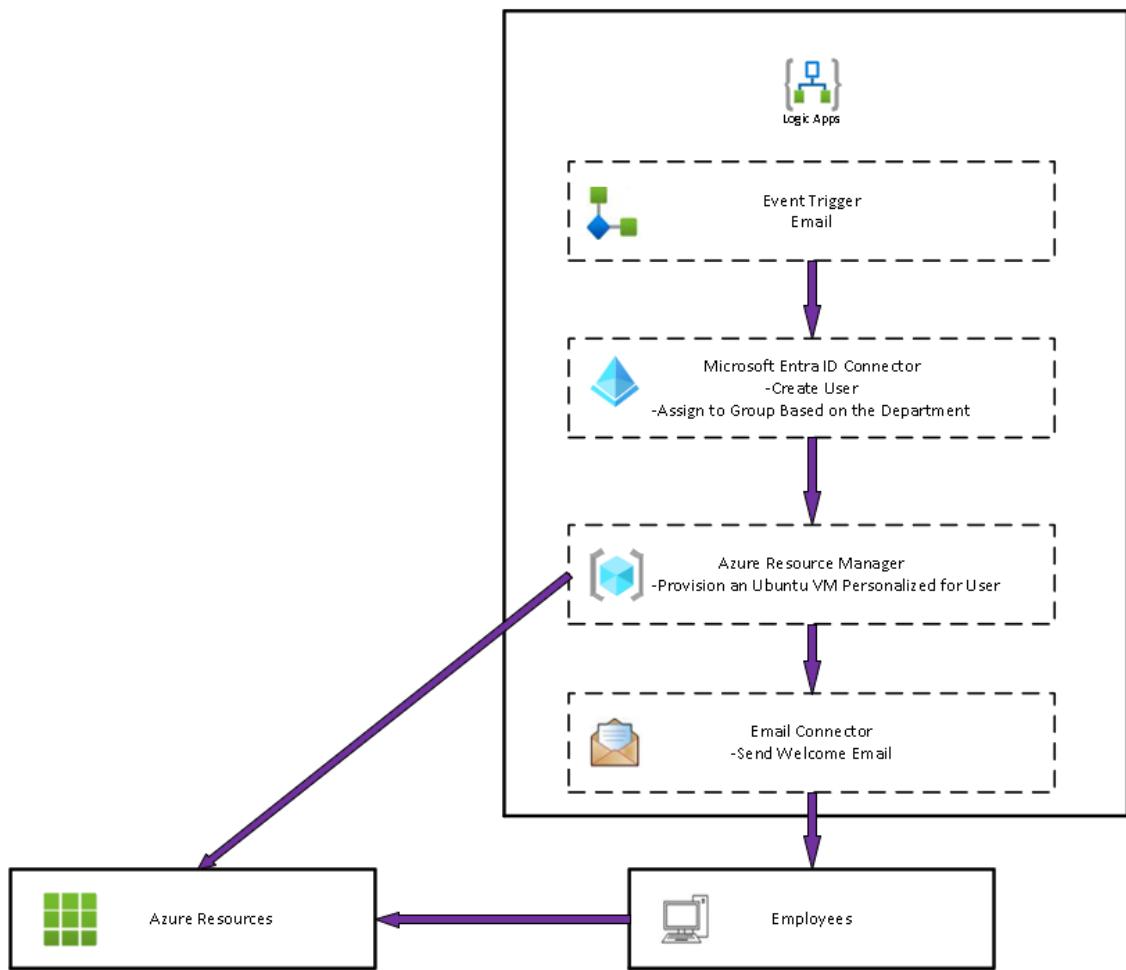
## Detailed Documentation with Screenshots

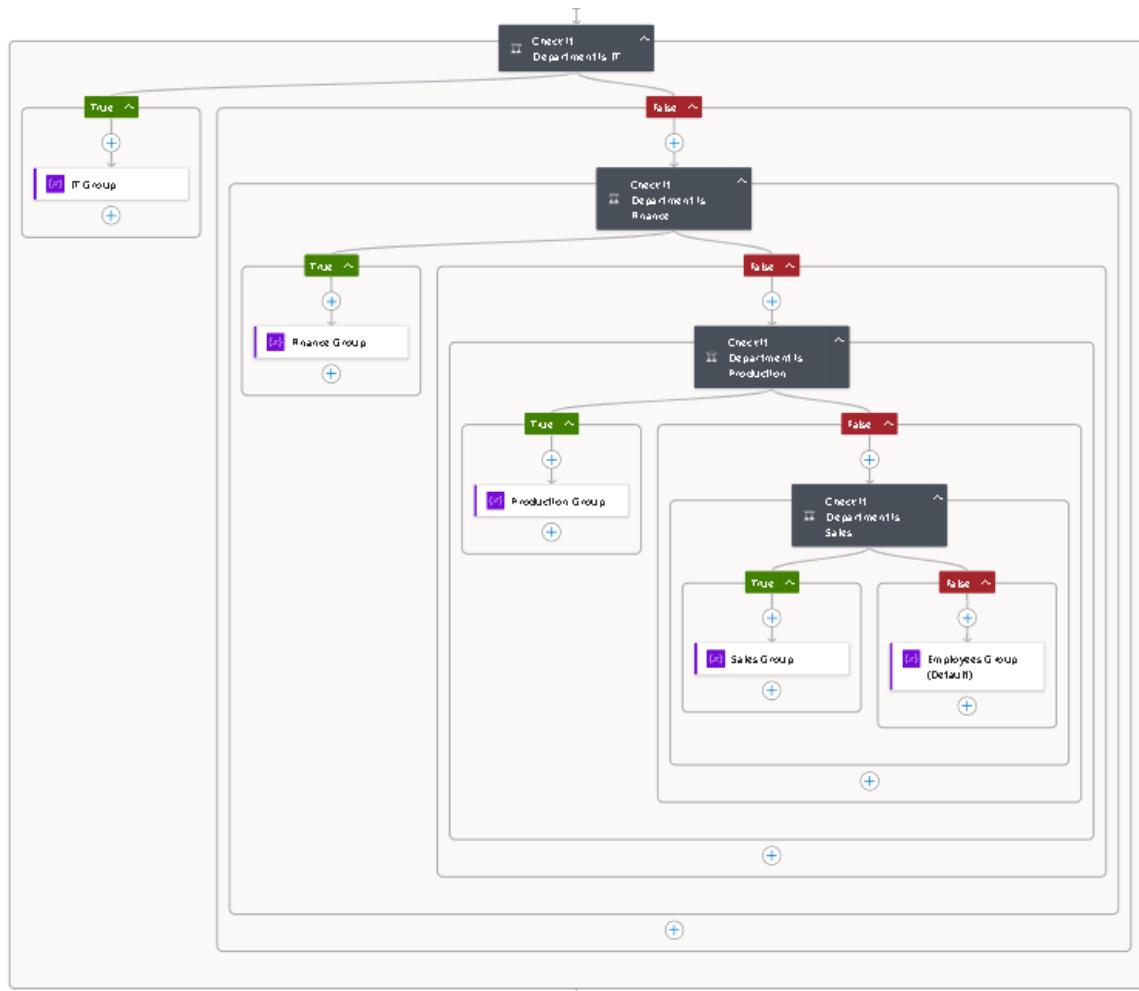
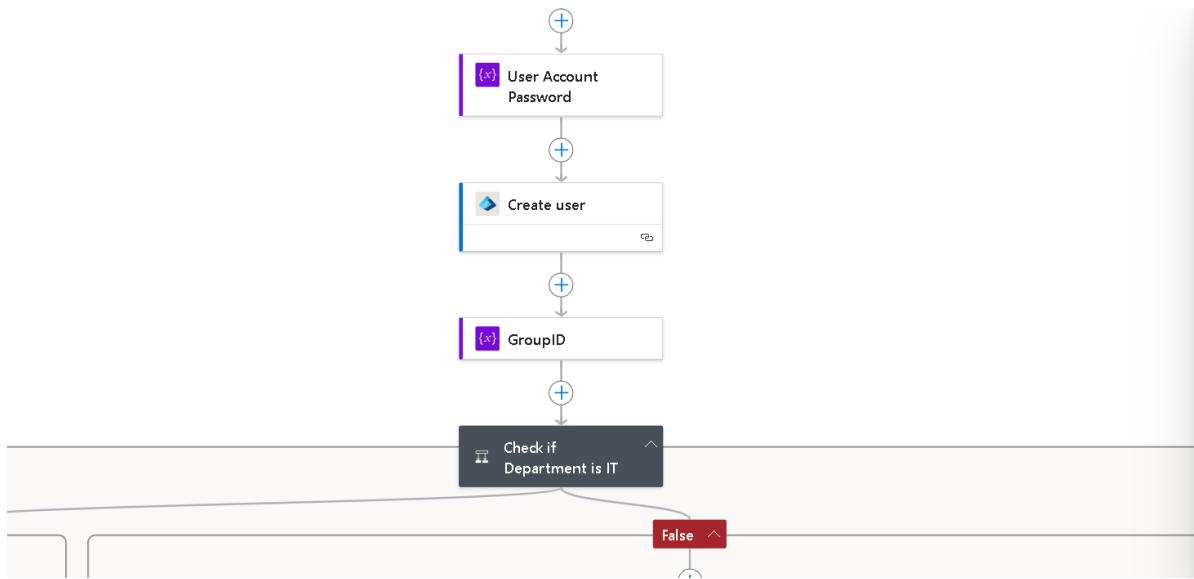
---

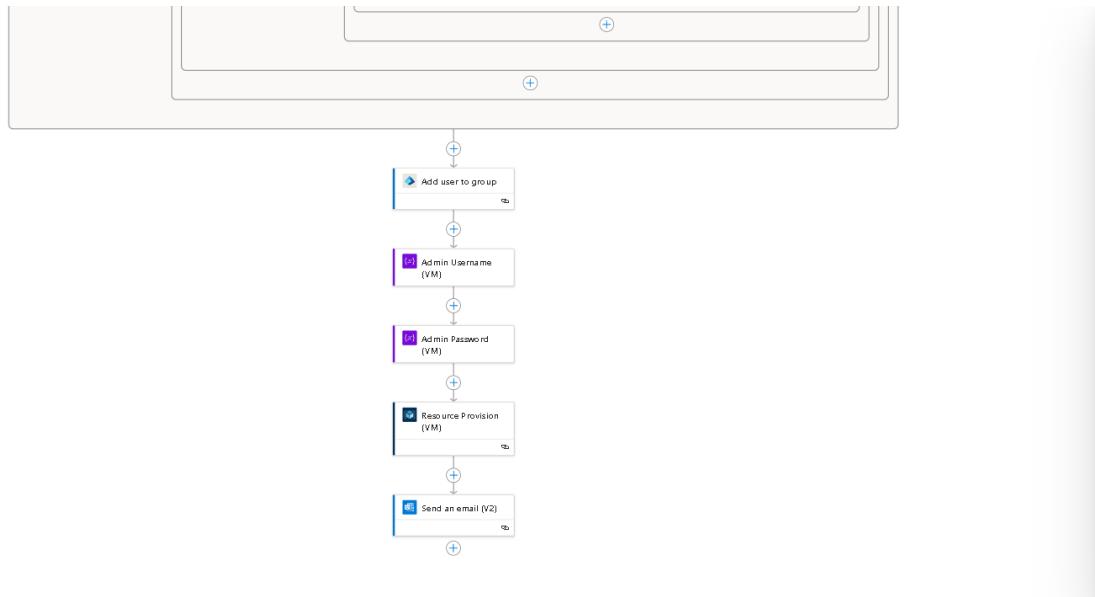
### Table of Contents

1. Introduction
2. Azure AD Setup
3. Logic App Workflow Design
4. Variables for User Credentials
5. Azure AD User Creation
6. GroupID Variable
7. Conditional Logic for Group Assignment
8. Group Assignment
9. Variables for Resource Provisioning (Ubuntu VM) Attributes
10. Resource Provisioning
11. Welcome Email
12. Monitoring and Review
13. Testing the Project
14. Conclusion

## Project Workflow







## Introduction

The Onboard Automator project aims to streamline and automate the process of onboarding new employees into Azure Active Directory (Azure AD) and assigning necessary Azure resources. This project uses Azure Logic Apps, Azure AD, Azure Resource Manager, and Azure Email Service.

### Microsoft Entra ID (Azure AD) Setup

#### Description:

Set up a new Azure AD instance using the Azure portal.

(I already had one Configured)

#### Screenshot:

Name	Value	Users	Groups	Applications	Devices
Tenant ID	[REDACTED]	2	5	0	0
Primary domain	vivekvash1507@gmail.onmicrosoft.com				
License	Microsoft Entra ID Free				

**Alerts:**

- Azure AD is now Microsoft Entra ID: Microsoft Entra ID is the new name for Azure Active Directory. No action is required from you.
- Service Change to Microsoft Entra Connect: We are making security-related service changes to Microsoft Entra Connect Sync and Connect Health. Upgrade to the latest version to avoid any feature disruptions.
- Upcoming MFA Server deprecation: Please migrate from MFA Server to Microsoft Entra Multi-Factor Authentication by September 2024 to avoid any service impact.

## Logic App Workflow Design

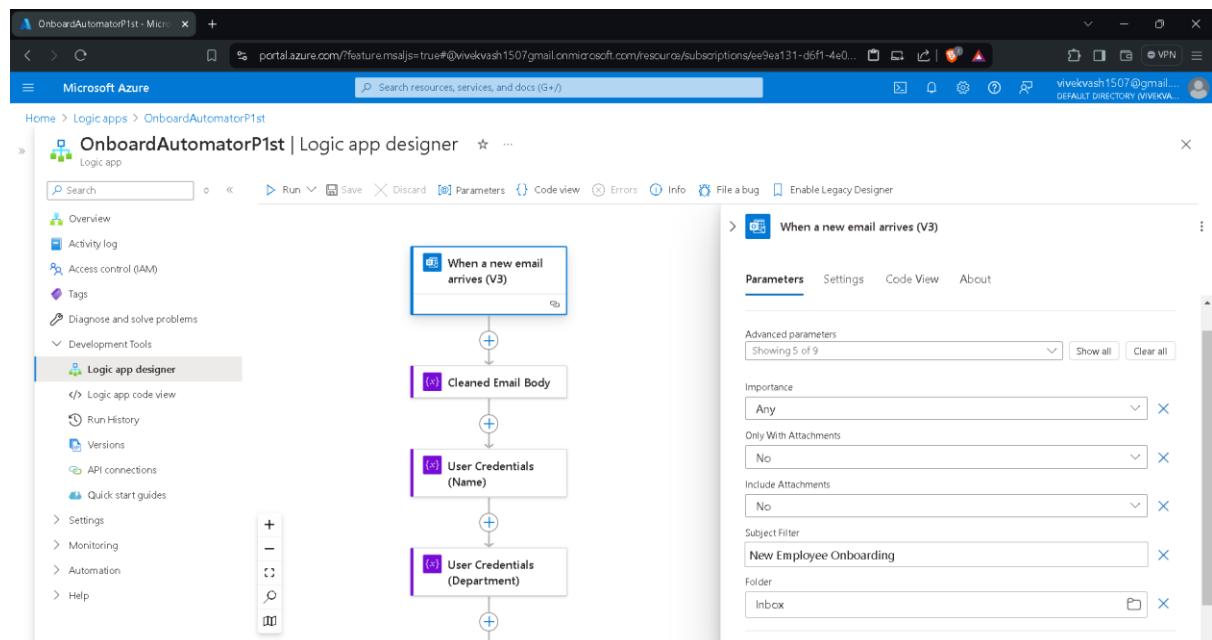
### **Trigger:**

Designed a Logic App workflow triggered by "When a new email arrives (V3)" to a specific mailbox indicating a new employee hire.

### **Configuration:**

- **Importance:** Any
- **Only with Attachments:** No
- **Include Attachments:** No
- **Subject Filter:** "New Employee Onboarding"

### **Screenshot:**

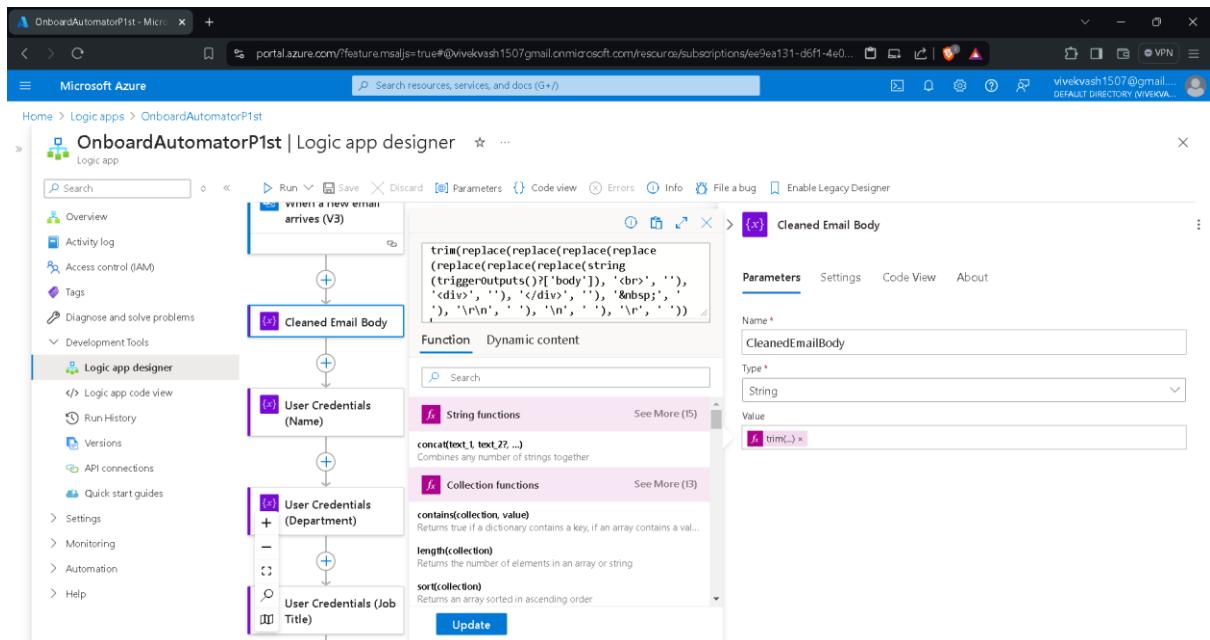


## Variables for User Credentials

### Cleaned Email Body:

- **Variable Name:** CleanedEmailBody
- **Type:** String
- **Details:** Uses "replace" functions to strip out HTML tags and normalize line breaks and "trim" to remove any leading or trailing whitespace of the email body.

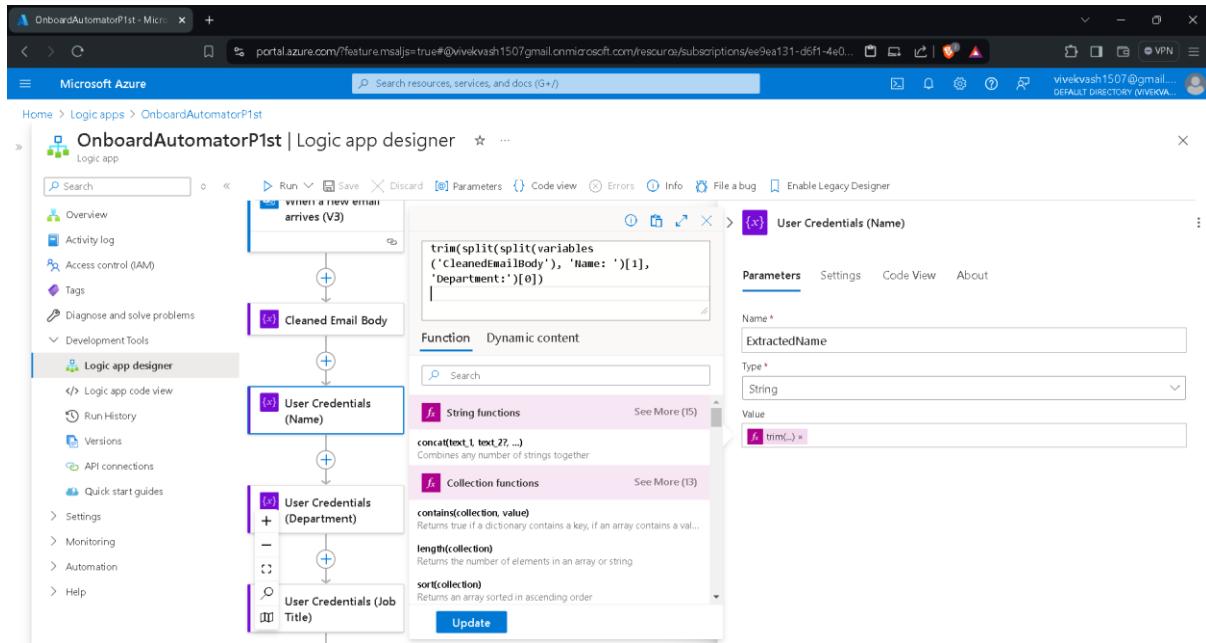
### Screenshot:



## User Credentials (Name):

- **Variable Name:** ExtractedName
- **Type:** String
- **Details:** Uses "split" function to extract the Name based on delimiters (Name:, Department:) and "trim" to remove whitespace.

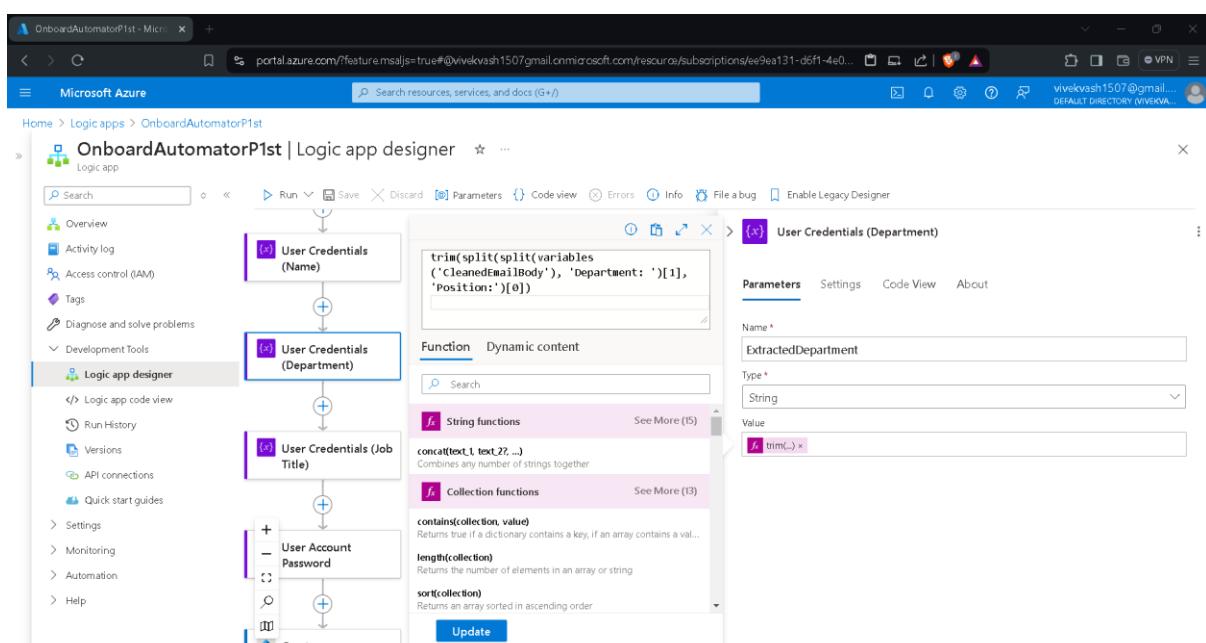
## Screenshot:



## User Credentials (Department):

- **Variable Name:** ExtractedDepartment
- **Type:** String
- **Details:** Uses "split" function to extract the Department based on delimiters (Department:, Position:) and "trim" to remove whitespace.

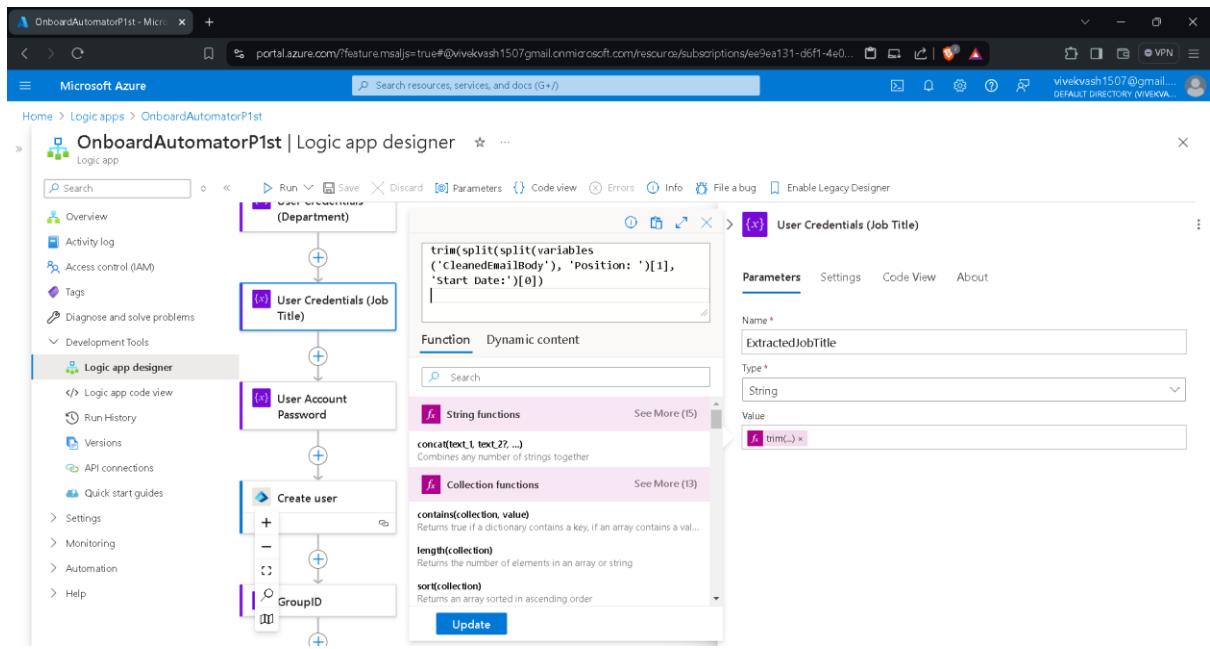
## Screenshot:



## User Credentials (Job Title):

- **Variable Name:** ExtractedJobTitle
- **Type:** String
- **Details:** Uses "split" function to extract the Position (Job Title) based on delimiters (Position:, Start Date:) and "trim" to remove whitespace.

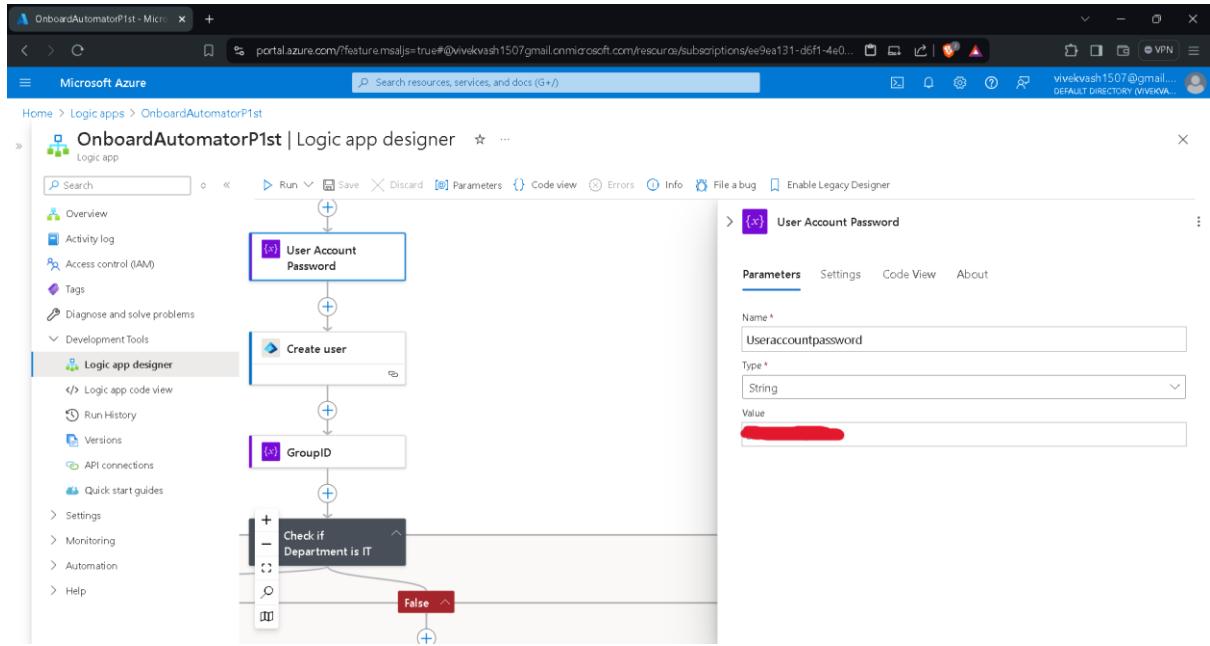
### Screenshot:



## User Account Password:

- **Variable Name:** Useraccountpassword
- **Type:** String
- **Details:** Stores the password for the user to be created.

### Screenshot:

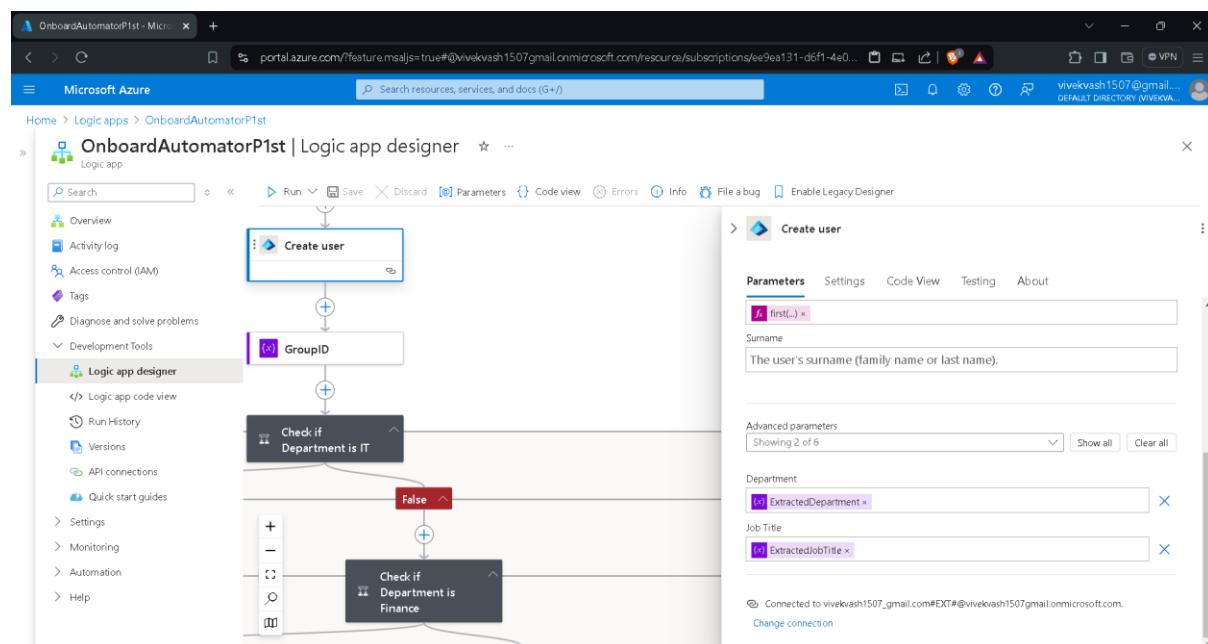
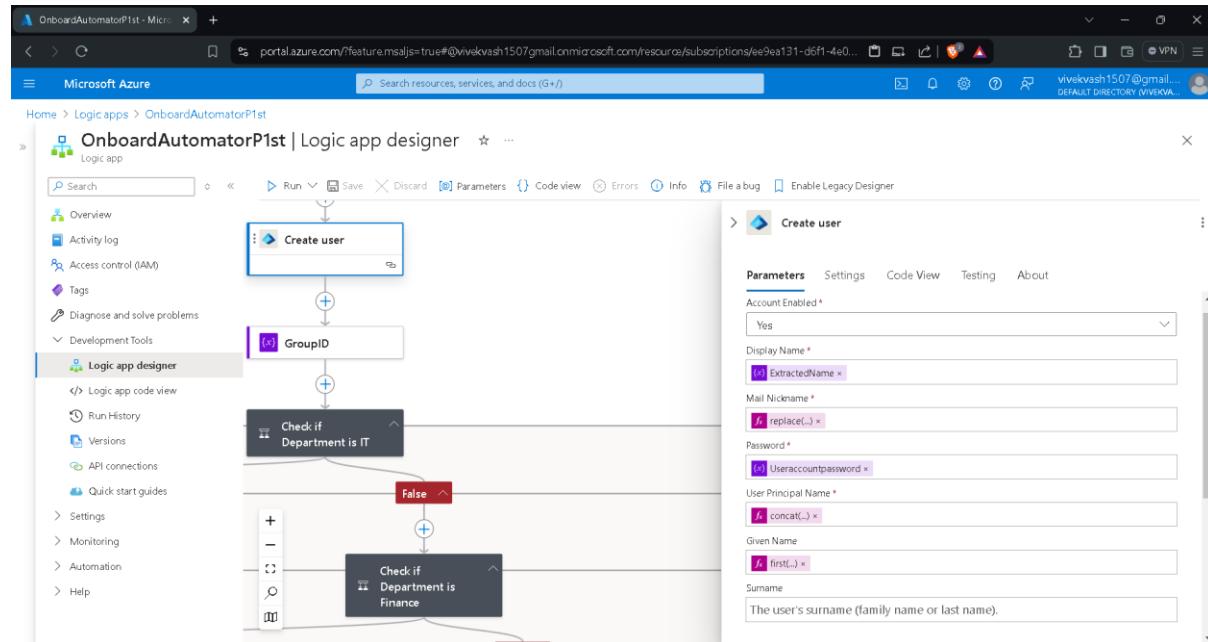


# Azure AD User Creation

## Description:

Automatically create a new user in Azure AD using the details extracted from the email.

## Screenshot:



## GroupID Variable

### Description:

Created and configured a variable to store Group ID

- **Manually created five groups (IT, Finance, Production, Sales, Employees) in Azure AD.**
- **Variable Name:** GroupID
- **Type:** String
- **Value:** Left empty initially to be set based on conditional logic.

### Screenshot:

The screenshot shows the 'Groups | All groups' page in the Azure portal. It lists five groups under 'Default Directory':

Name	Object ID	Group type	Membership type	Email	Source
Employees	9f4212d3-8893-4765-8	Microsoft 365	Assigned	Employee@vivekvash1507@gmail.onmicrosoft.com	Cloud
Finance	a0a1e4a6-f463-44cf-8e	Microsoft 365	Assigned	Finance@vivekvash1507@gmail.onmicrosoft.com	Cloud
IT	d6b4f52a-92cf-4632-95	Microsoft 365	Assigned	IT@vivekvash1507@gmail.onmicrosoft.com	Cloud
Production	d339b22f-aac5-44da-91	Microsoft 365	Assigned	Production@vivekvash1507@gmail.onmicrosoft.com	Cloud
Sales	4cb5ff07-1612-4ed6-b	Microsoft 365	Assigned	Sales@vivekvash1507@gmail.onmicrosoft.com	Cloud

The screenshot shows the Logic app designer for the 'OnboardAutomatorP1st' logic app. The workflow starts with a 'Create user' action, followed by a 'GroupID' action which sets the 'GroupID' variable. Then, it branches into two parallel 'Check if Department is' actions: one for 'IT' and one for 'Finance'. The 'IT' check has a 'False' path leading to a second 'Check if Department is Finance' action.

**Logic App Flow:**

```
graph TD; Start(( )) --> CreateUser[Create user]; CreateUser --> GroupID[GroupID]; GroupID --> IT[Check if Department is IT]; GroupID --> Finance[Check if Department is Finance]; IT -- False --> Finance;
```

**GroupID Variable Configuration:**

Parameter	Value
Name *	GroupID
Type *	String
Value	Enter initial value

## Conditional Logic for Group Assignment

### Description:

Check if the Department is IT and set the GroupID variable accordingly.

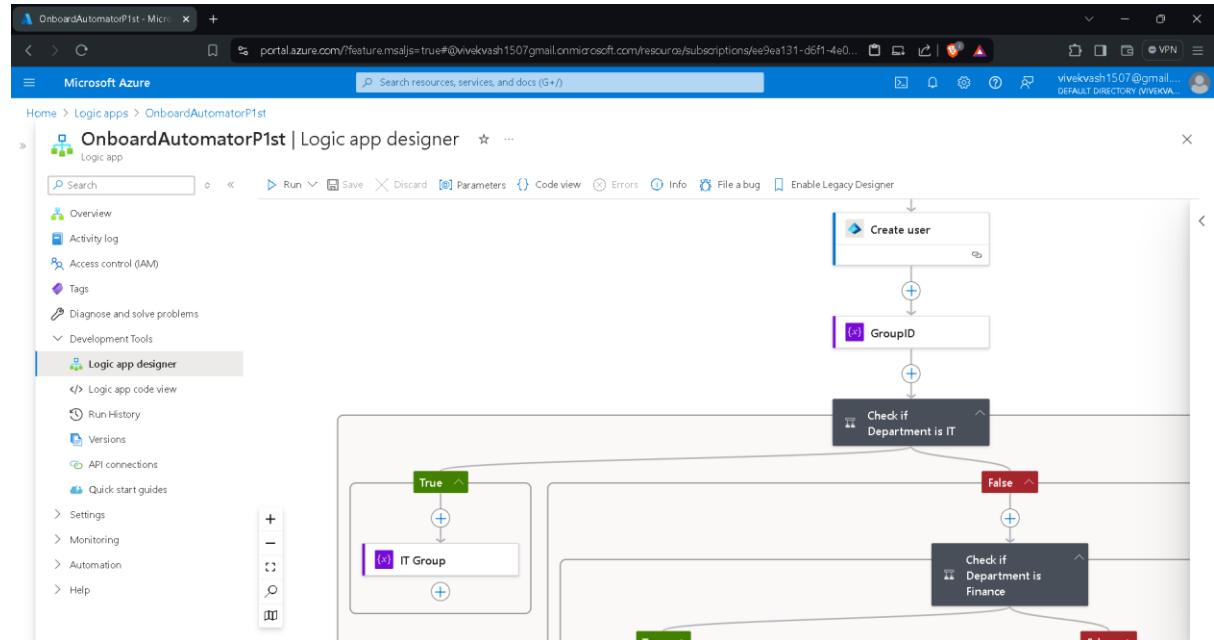
### Top-Level Condition: Check if Department is IT

**Condition:** User Credentials (Department) is equal to IT

**True:** Set variable GroupID to Group ID for IT

**False:** Check if Department is Finance

### Screenshot:



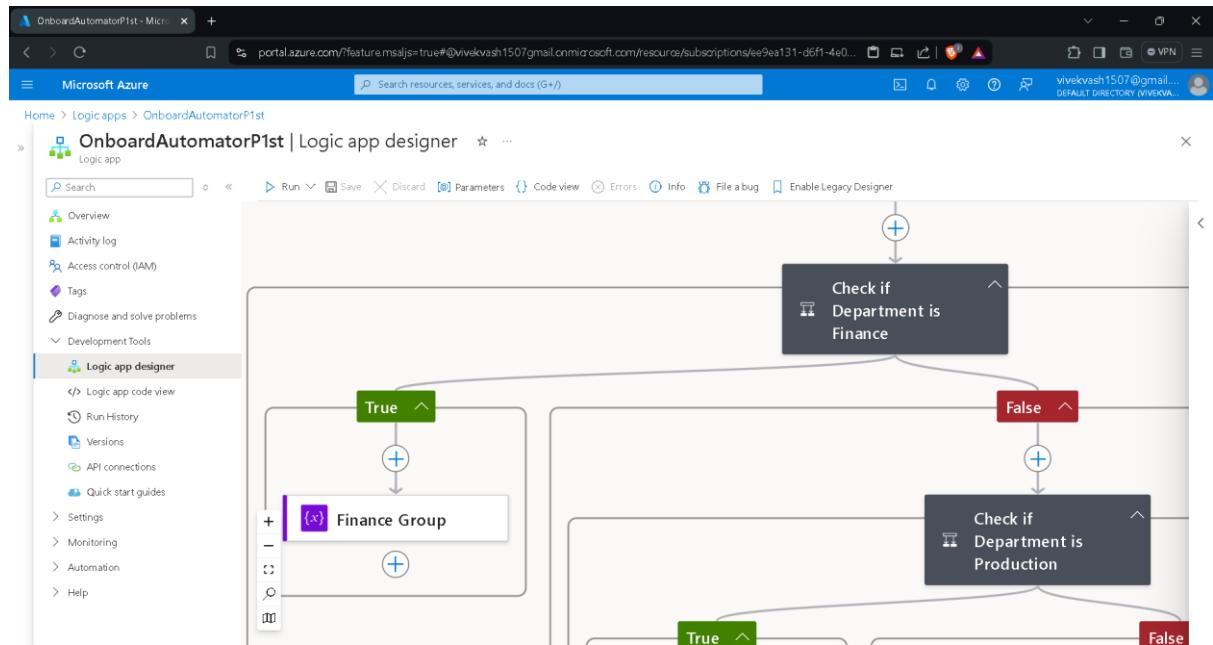
## Nested Conditions

**Condition:** User Credentials (Department) is equal to Finance

**True:** Set variable GroupID to Group ID for Finance

**False:** Check if Department is Production

### Screenshot:

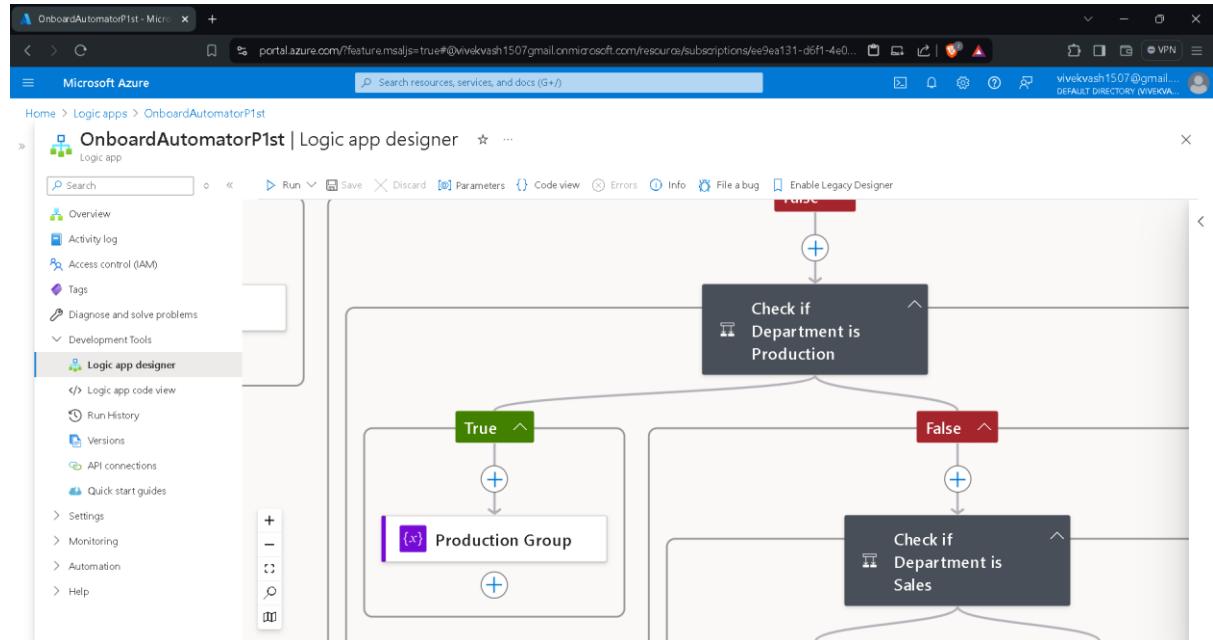


**Condition:** User Credentials (Department) is equal to Production

**True:** Set variable GroupID to Group ID for Production

**False:** Check if Department is Sales

### Screenshot:

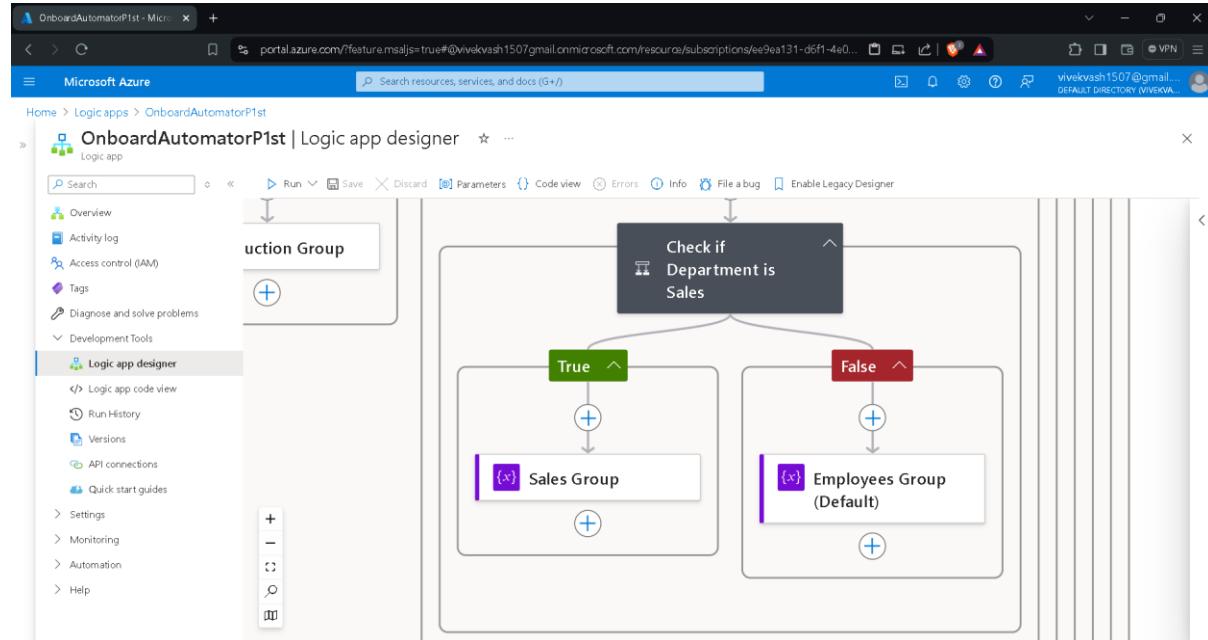


**Condition:** User Credentials (Department) is equal to Sales

**True:** Set variable GroupID to Group ID for Sales

**False:** Set variable GroupID to Group ID for Employees

#### Screenshot:



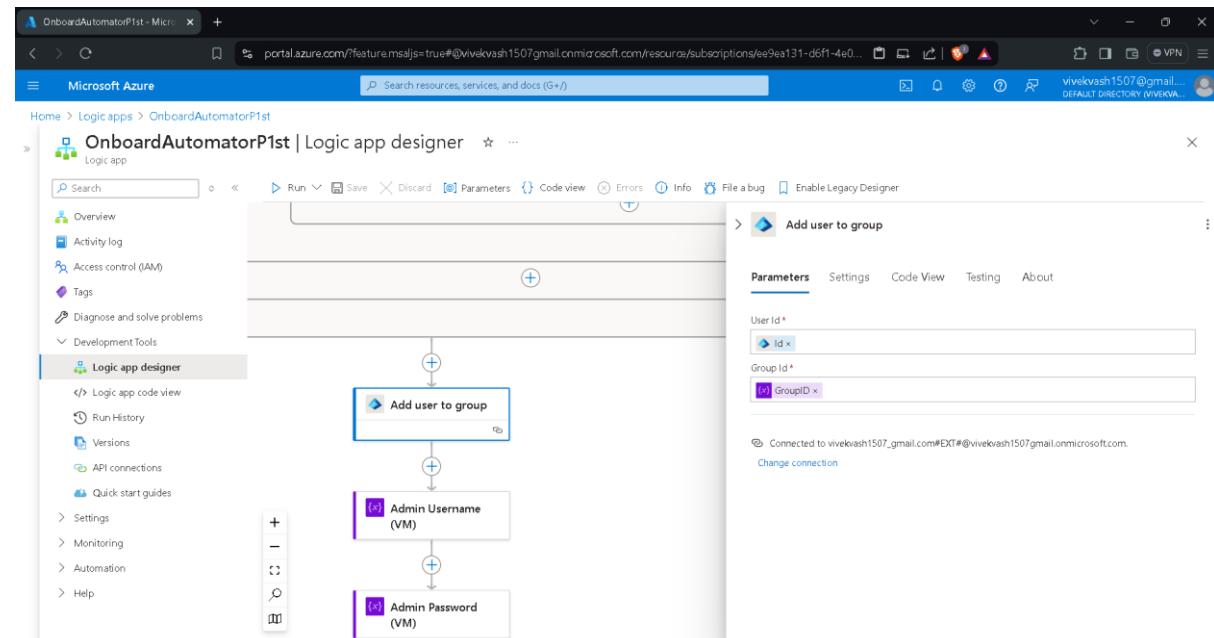
## Group Assignment

#### Description:

Assign the newly created user to the appropriate group based on the department.

- Use the Microsoft Entra ID connector to add the user to the group using the GroupID variable.

#### Screenshot:

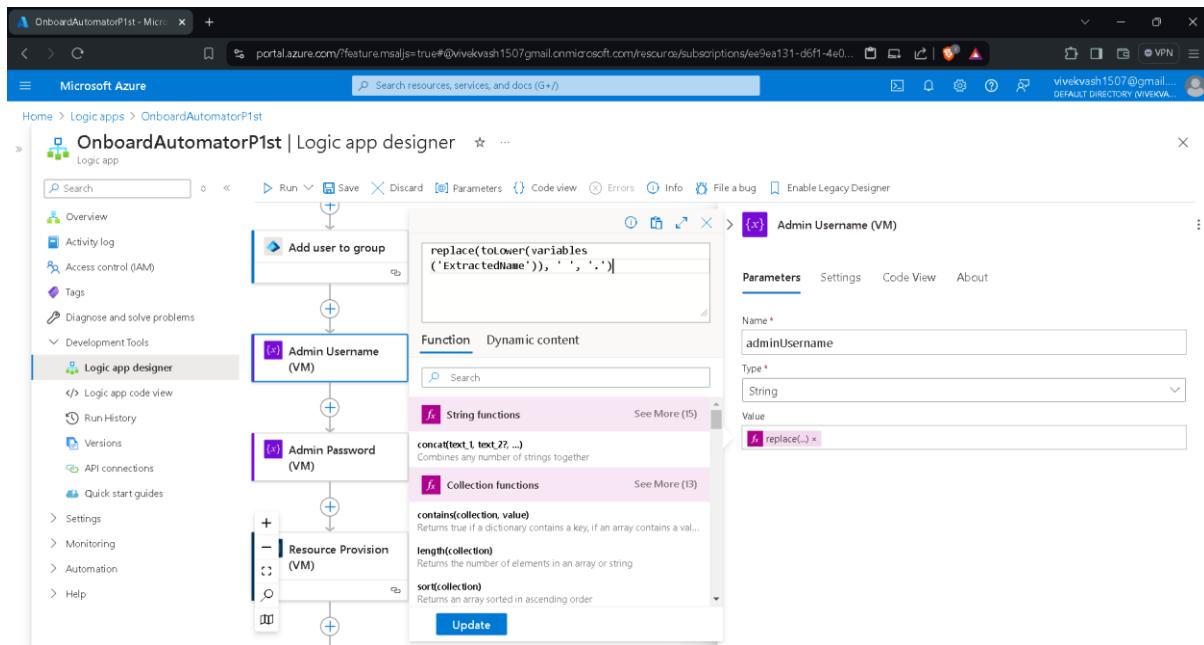


## Variables for Resource Provisioning (Ubuntu VM) Attributes

### Admin Username (VM):

- **Variable Name:** adminUsername
- **Type:** String
- **Details:** Stores the new employee's name in lowercase.

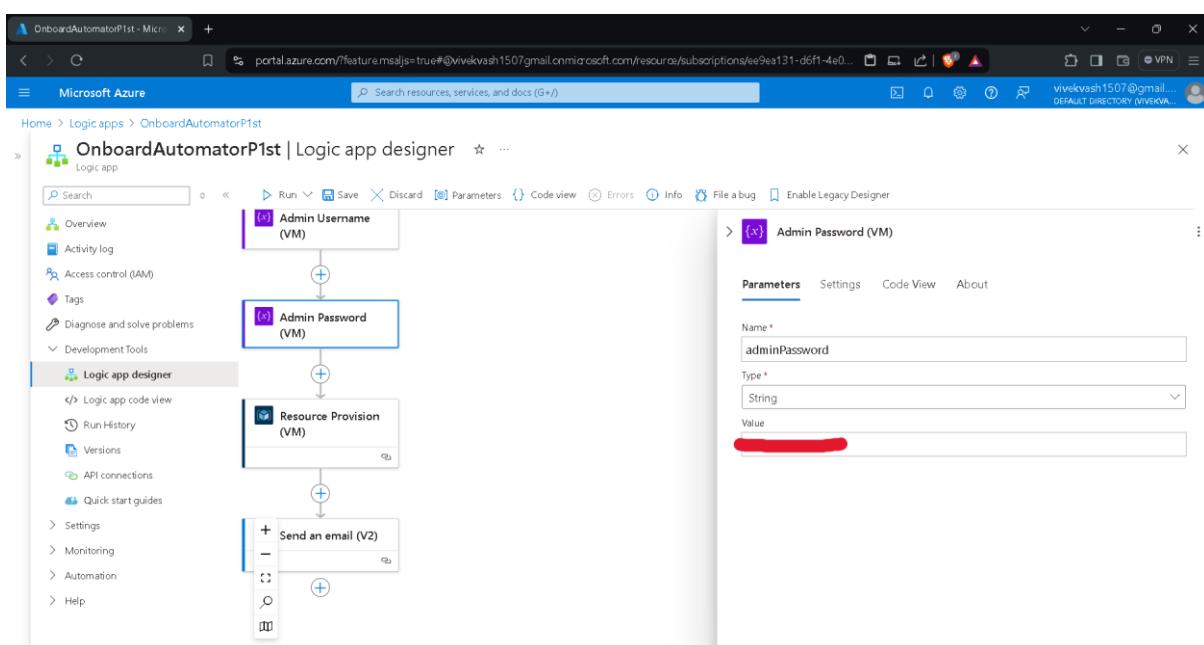
### Screenshot:



### Admin Password (VM):

- **Variable Name:** adminPassword
- **Type:** String
- **Details:** Stores the password to be used for the VM.

### Screenshot:



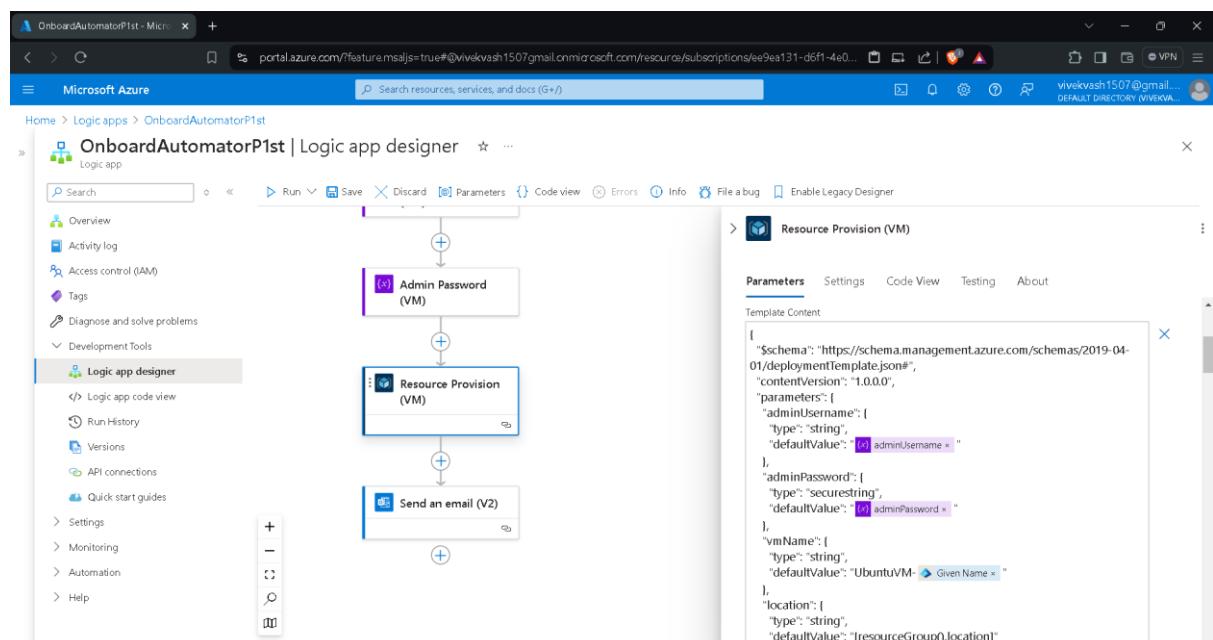
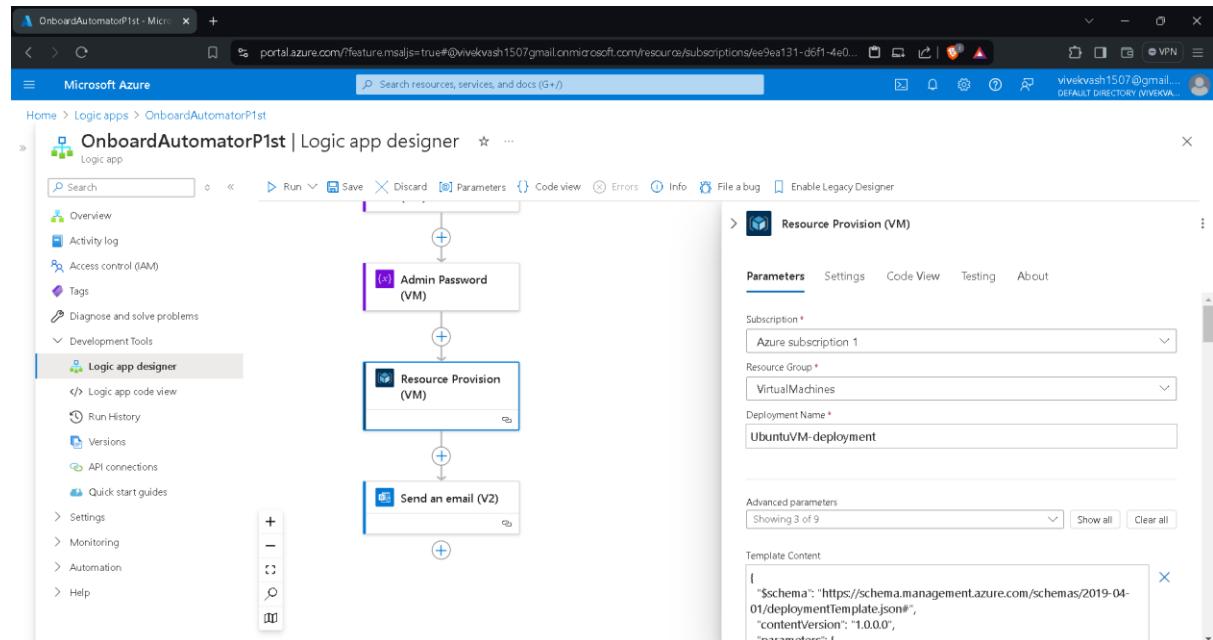
# Resource Provisioning

## Description:

Provision a new Ubuntu VM using Azure Resource Manager.

- **Template Content:** Detailed ARM template provided in the GitHub Repo to create and configure the VM.
- **Configuration:** Selected subscription, resource group, deployment name, and used parameters for VM configuration (adminUsername, adminPassword, vmName, location).

## Screenshot:



OnboardAutomatorP1st - Microsoft Azure

portal.azure.com/?feature.msals=true#vivekvash1507@gmail.com/microsoft.com/resource/subscriptions/ee9ea131-d6f1-4e0...

Microsoft Azure

Home > Logic apps > OnboardAutomatorP1st

OnboardAutomatorP1st | Logic app designer

Logic app

Search

Run Save Discard Parameters Code view Errors Info File a bug Enable Legacy Designer

Overview Activity log Access control (IAM) Tags Diagnose and solve problems Development Tools Logic app designer (selected) Logic app code view Run History Versions API connections Quick start guides Settings Monitoring Automation Help

+

Resource Provision (VM)

Admin Password (VM)

Send an email (V2)

Parameters Settings Code View Testing About

```

    "variables": [
        "networkInterfaceName": "[concat(parameters('vmName'), '-nic')]",
        "networkSecurityGroupName": "[concat(parameters('vmName'), '-nsq')]",
        "virtualNetworkName": "[concat(parameters('vmName'), '-vnet')]",
        "subnetName": "default",
        "publicIpAddressName": "[concat(parameters('vmName'), '-pip')]",
        "addressPrefix": "10.0.0.0/16",
        "subnetPrefix": "10.0.0.0/24",
        "virtualNetworkID": "[resourceId('Microsoft.Network/virtualNetworks', variables('virtualNetworkName'))]",
        "subnetRef": "[concat(variables('virtualNetworkID'), '/subnets/', variables('subnetName'))]"
    ],
    "resources": [
        {
            "type": "Microsoft.Network/publicIPAddresses",
            "apiVersion": "2020-08-01",
            "name": "variables('publicIpAddressName')",
            "location": "[parameters('location')]",
            "properties": {
                "publicIPAllocationMethod": "Dynamic"
            }
        },
        {
            "type": "Microsoft.Network/virtualNetworks",
            "apiVersion": "2020-06-01",
            "name": "variables('virtualNetworkName')",
            "location": "[parameters('location')]",
            "properties": {
                "addressSpace": {
                    "addressPrefixes": [
                        "variables('addressPrefix')"
                    ]
                },
                "subnets": [
                    {
                        "name": "variables('subnetName')",
                        "properties": {
                            "addressPrefix": "variables('subnetPrefix')"
                        }
                    }
                ]
            }
        }
    ]
}

```

OnboardAutomatorP1st - Microsoft Azure

portal.azure.com/?feature.msals=true#vivekvash1507@gmail.com/microsoft.com/resource/subscriptions/ee9ea131-d6f1-4e0...

Microsoft Azure

Home > Logic apps > OnboardAutomatorP1st

OnboardAutomatorP1st | Logic app designer

Logic app

Search

Run Save Discard Parameters Code view Errors Info File a bug Enable Legacy Designer

Overview Activity log Access control (IAM) Tags Diagnose and solve problems Development Tools Logic app designer (selected) Logic app code view Run History Versions API connections Quick start guides Settings Monitoring Automation Help

+

Resource Provision (VM)

Admin Password (VM)

Send an email (V2)

Parameters Settings Code View Testing About

```

    "variables": [
        "networkInterfaceName": "[concat(parameters('vmName'), '-nic')]",
        "networkSecurityGroupName": "[concat(parameters('vmName'), '-nsq')]",
        "virtualNetworkName": "[concat(parameters('vmName'), '-vnet')]",
        "subnetName": "default",
        "publicIpAddressName": "[concat(parameters('vmName'), '-pip')]",
        "addressPrefix": "10.0.0.0/16",
        "subnetPrefix": "10.0.0.0/24",
        "virtualNetworkID": "[resourceId('Microsoft.Network/virtualNetworks', variables('virtualNetworkName'))]",
        "subnetRef": "[concat(variables('virtualNetworkID'), '/subnets/', variables('subnetName'))]"
    ],
    "resources": [
        {
            "type": "Microsoft.Network/publicIPAddresses",
            "apiVersion": "2020-08-01",
            "name": "variables('publicIpAddressName')",
            "location": "[parameters('location')]",
            "properties": {
                "publicIPAllocationMethod": "Dynamic"
            }
        },
        {
            "type": "Microsoft.Network/virtualNetworks",
            "apiVersion": "2020-06-01",
            "name": "variables('virtualNetworkName')",
            "location": "[parameters('location')]",
            "properties": {
                "addressSpace": {
                    "addressPrefixes": [
                        "variables('addressPrefix')"
                    ]
                },
                "subnets": [
                    {
                        "name": "variables('subnetName')",
                        "properties": {
                            "addressPrefix": "variables('subnetPrefix')"
                        }
                    }
                ]
            }
        }
    ]
}

```

OnboardAutomatorP1st - Microsoft Azure

Home > Logic apps > OnboardAutomatorP1st

### OnboardAutomatorP1st | Logic app designer

```

graph TD
    Start(( )) --> Admin[Admin Password (VM)]
    Admin --> Provision[Resource Provision (VM)]
    Provision --> Email[Send an email (V2)]
  
```

**Resource Provision (VM)**

**Parameters**

```

{
  "type": "Microsoft.Network/networkSecurityGroups",
  "apiVersion": "2020-06-01",
  "name": "[variables('networkSecurityGroupName')]",
  "location": "[parameters('location')]",
  "properties": {
    "securityRules": [
      {
        "name": "default-allow-ssh",
        "properties": {
          "priority": 1000,
          "protocol": "Tcp",
          "access": "Allow",
          "direction": "Inbound",
          "sourceAddressPrefix": "*",
          "sourcePortRange": "22",
          "destinationAddressPrefix": "*",
          "destinationPortRange": "22"
        }
      }
    ]
  }
}
  
```

OnboardAutomatorP1st - Microsoft Azure

Home > Logic apps > OnboardAutomatorP1st

### OnboardAutomatorP1st | Logic app designer

```

graph TD
    Start(( )) --> Admin[Admin Password (VM)]
    Admin --> Provision[Resource Provision (VM)]
    Provision --> Email[Send an email (V2)]
  
```

**Resource Provision (VM)**

**Parameters**

```

{
  "type": "Microsoft.Network/networkInterfaces",
  "apiVersion": "2020-11-01",
  "name": "[variables('networkInterfaceName')]",
  "location": "[parameters('location')]",
  "dependsOn": [
    "[resourceId('Microsoft.Network/virtualNetworks', variables('virtualNetworkName'))]"
  ],
  "properties": {
    "ipConfigurations": [
      {
        "name": "ipconfig1",
        "properties": {
          "publicIpAddressName": "[variables('publicIpAddressName')]"
        }
      }
    ]
  }
}
  
```

The screenshot shows the Microsoft Azure Logic App designer interface for a logic app named "OnboardAutomatorP1st". The workflow consists of three main steps:

- A trigger step: "Admin Password (VM)".
- An action step: "Resource Provision (VM)".
- A final action step: "Send an email (V2)".

The "Resource Provision (VM)" step is currently selected, and its configuration details are displayed in the "Parameters" tab of the right-hand pane. The JSON configuration is as follows:

```
name: "ipconfig1",
properties: {
    "privateIPAllocationMethod": "Dynamic",
    "publicIPAddress": {
        "id": "[resourceId('Microsoft.Network/publicIPAddresses', variables('publicIpAddressName'))]"
    },
    "subnet": {
        "id": "[variables('subnetRef')]"
    }
},
networkSecurityGroup: {
    "id": "[resourceId('Microsoft.Network/networkSecurityGroups', variables('networkSecurityGroupName'))]"
},
{
    "type": "Microsoft.Compute/virtualMachines",
    "apiVersion": "2021-07-01",
```

This screenshot shows the same logic app "OnboardAutomatorP1st" with a slightly modified configuration for the "Resource Provision (VM)" step. The "Parameters" tab in the right-hand pane now displays the following JSON configuration:

```
"name": "[parameters('vmName')]",
"location": "[parameters('location')]",
"dependsOn": [
    "[resourceId('Microsoft.Network/networkInterfaces', variables('networkInterfaceName'))]"
],
"properties": {
    "hardwareProfile": {
        "vmSize": "Standard_DS1_v2"
    },
    "storageProfile": {
        "imageReference": {
            "publisher": "Canonical",
            "offer": "UbuntuServer",
            "sku": "18.04-LTS",
            "version": "latest"
        },
        "osDisk": {
            "createOption": "FromImage"
        }
    },
    "osProfile": {
```

Microsoft Azure

OnboardAutomatorP1st | Logic app designer

Resource Provision (VM)

Parameters

```
[{"osProfile": {"computerName": "[parameters('vmName')]", "adminUsername": "[parameters('adminUsername')]", "adminPassword": "[parameters('adminPassword')]"}}, {"networkProfile": {"networkInterfaces": [{"id": "[resourceId('Microsoft.Network/networkInterfaces', variables('networkInterfaceName'))]"}]}}, {"outputs": {"adminUsername": {"type": "string", "value": "[parameters('adminUsername')]"}}]
```

Send an email (V2)

Logic app designer

Admin Password (VM)

Resource Provision (VM)

Send an email (V2)

OnboardAutomatorP1st | Logic app designer

Resource Provision (VM)

Parameters

```
[{"outputs": {"adminUsername": {"type": "string", "value": "[parameters('adminUsername')]"}}}]
```

Deployment Mode: Complete

Wait For Deployment: Yes

Connected to vivekvash1507@gmail.com#EXT#@vivekvash1507@gmail.onmicrosoft.com.

Change connection

---

## Welcome Email

**Description:**

Send a welcome email to the new hire with access details.

**Configuration:**

- **To:** @{body('Create\_user')?['userPrincipalName']}
- **Subject:** Welcome to the Team, @{body('Create\_user')?['givenName']}
- **Body:**

Dear @{body('Create\_user')?['displayName']},

Welcome to our team! We are excited to have you on board.

Here are your access details: -

Username: @{body('Create\_user')?['userPrincipalName']}

Temporary Password: @{variables('Useraccountpassword')}

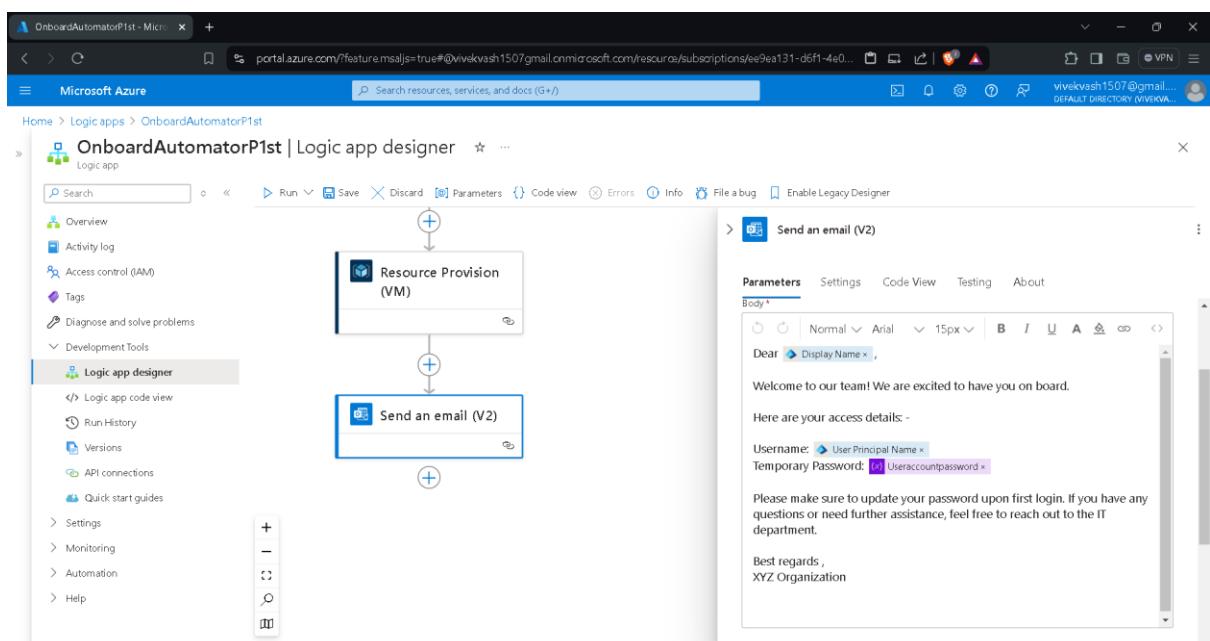
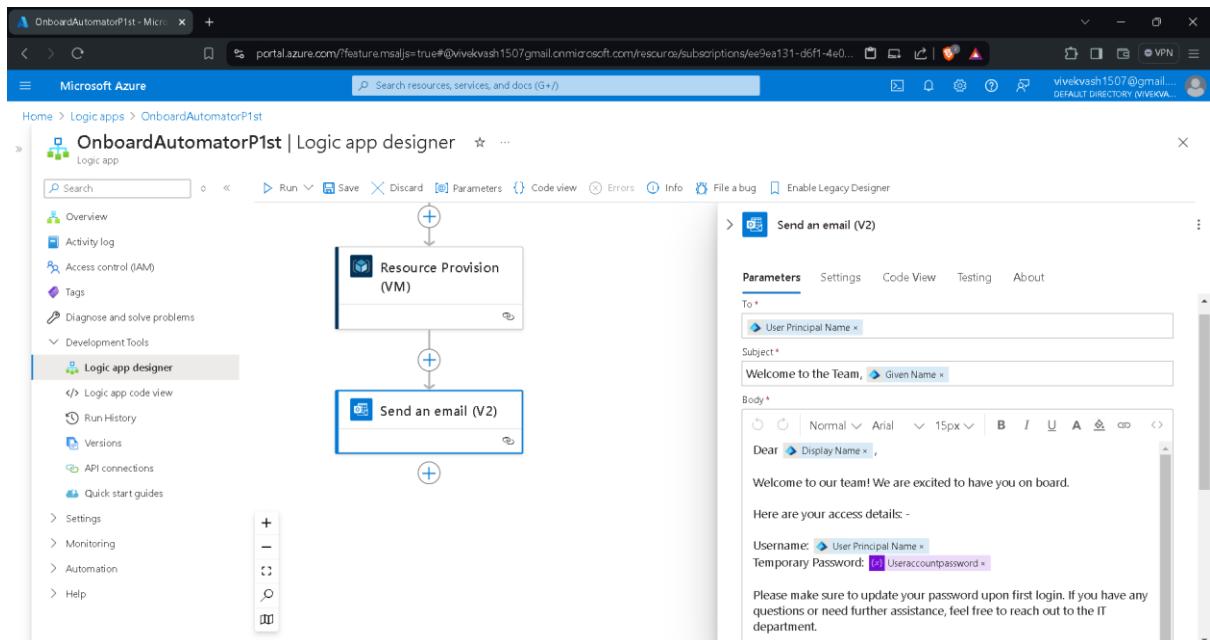
Please make sure to update your password upon first login. If you have any questions or need further assistance, feel free to reach out to the IT department.

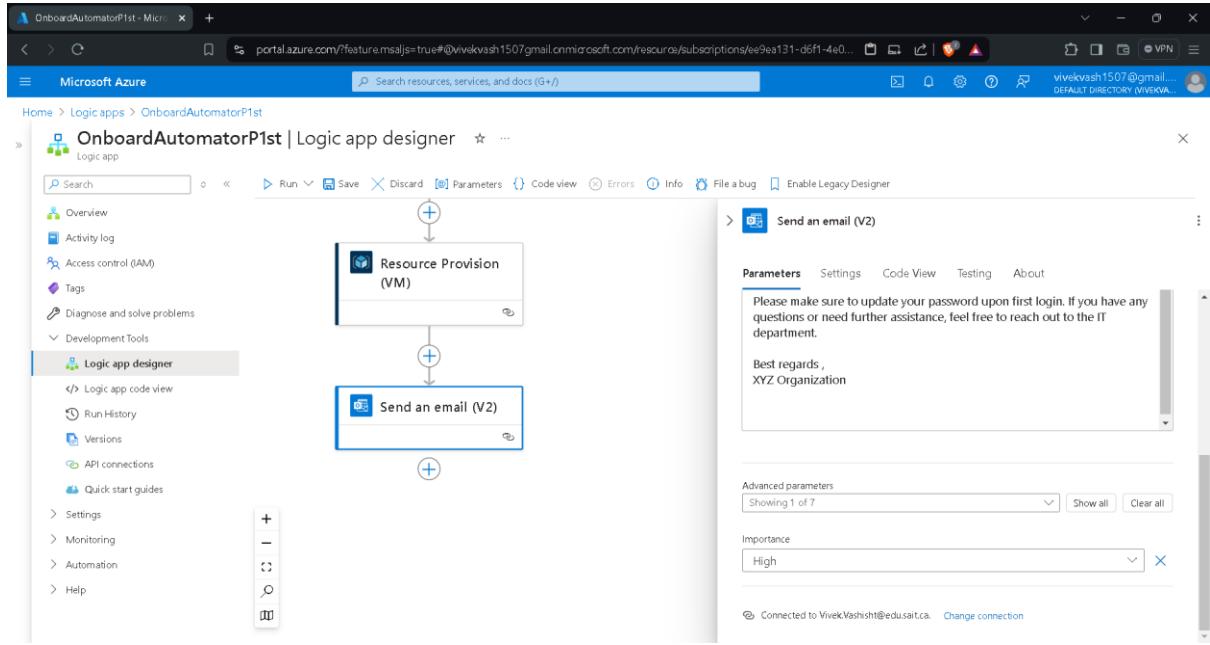
Best regards,

XYZ Organization

- **Importance:** High

## Screenshot:





## Monitoring and Review

### Description:

Monitor and review the onboarding process through Logic Apps run history.

- Sent a test email and verified the successful execution of the workflow and receipt of the welcome email.

### Screenshot:

Status	Start time	Identifier	Duration	Static Results
Succeeded	7/20/2024, 10:53 PM	08584800692886309337787945907...	33.81 Seconds	
Succeeded	7/20/2024, 10:32 PM	08584800705173386748779241003...	50.69 Seconds	
Succeeded	7/20/2024, 10:25 PM	08584800709827803430165224881...	36.23 Seconds	
Succeeded	7/20/2024, 9:17 PM	08584800750466127781652454546...	34.19 Seconds	
Failed	7/20/2024, 8:51 PM	08584800766214246601441887837...	33.83 Seconds	
Failed	7/20/2024, 8:43 PM	0858480077094464091278809505B...	1.04 Seconds	

**OnboardAutomatorP1st | Run History**

Logic app

Run Details Resubmit Cancel Run Refresh Info File a bug Enable Legacy Designer

When a new email arrives (V) → Cleaned Email Body → User Credentials (Name) → User Credentials (User Input) → User Credentials (Job Title) → User Account Password → Create user → GroupD → Check if Department is IT

```

graph TD
    Start((When a new email arrives (V))) --> CleanedEmailBody[Cleaned Email Body]
    CleanedEmailBody --> UserCredentialsName[User Credentials (Name)]
    UserCredentialsName --> UserCredentialsInput[User Credentials (User Input)]
    UserCredentialsInput --> UserCredentialsJobTitle[User Credentials (Job Title)]
    UserCredentialsJobTitle --> UserAccountPassword[User Account Password]
    UserAccountPassword --> CreateUser[Create user]
    CreateUser --> GroupD[GroupD]
    GroupD --> CheckIfDepartmentIsIT[Check if Department is IT]
  
```

**OnboardAutomatorP1st | Run History**

Logic app

Run Details Resubmit Cancel Run Refresh Info File a bug Enable Legacy Designer

Check if Department is IT → If True: Finance Group → If False: Product & Group → Check if Department is Marketing → If True: Marketing Group → If False: Sales → Check if Department is Research → If True: Research Group → If False: Admin → Select User → Assign Group → Add User to Group

```

graph TD
    Start((Check if Department is IT)) --> IfTrueFinance[If True: Finance Group]
    IfTrueFinance --> IfFalseProduct[If False: Product & Group]
    IfFalseProduct --> IfFalseMarketing[Check if Department is Marketing]
    IfFalseMarketing --> IfFalseMarketingTrue[If True: Marketing Group]
    IfFalseMarketingTrue --> IfFalseMarketingFalse[If False: Sales]
    IfFalseMarketingFalse --> IfFalseResearch[Check if Department is Research]
    IfFalseResearch --> IfFalseResearchTrue[If True: Research Group]
    IfFalseResearchTrue --> IfFalseResearchFalse[If False: Admin]
    IfFalseResearchFalse --> SelectUser[Select User]
    SelectUser --> AssignGroup[Assign Group]
    AssignGroup --> AddUserGroup[Add User to Group]
  
```

OnboardAutomatorP1st | Run History

Run Details | Resubmit | Cancel Run | Refresh | Info | File a bug | Enable Legacy Designer

OnboardAutomatorP1st

Run History

Start time	Duration
7/20/2024, 10:53 PM	33.81 Seconds
7/20/2024, 10:32 PM	50.69 Seconds
7/20/2024, 10:25 PM	3.623 Seconds
7/20/2024, 9:17 PM	34.19 Seconds
7/20/2024, 8:51 PM	33.83 Seconds
7/20/2024, 8:43 PM	1.04 Seconds

## Testing the Project

**Test Email:**

**Screenshot:**

New Employee Onboarding

Vivek Vashishth <vivekvashish1507@gmail.com>  
to vivekvashish1507  
Hi Team,  
Please onboard the following new employee:  
Name: John Doe  
Department: IT  
Position: Software Engineer  
Start Date: 7 November 2024

Best,  
HR Team

Virus-free [www.avg.com](http://www.avg.com)

## User Created:

### Screenshot:

The screenshot shows the Azure portal interface for a user named John Doe. The top navigation bar includes links for Home, Users, and Groups. The main content area displays the user's basic information, such as User principal name (john.doe@vivekvash1507@gmail.onmicrosoft.com), Object ID (25d4511d-422c-49cd-b702-85837568be88), and Created date time (Jul 20, 2024, 10:53 PM). It also shows Group memberships (1), Applications (0), Assigned roles (0), and Assigned licenses (0). Below this, there is a 'My Feed' section with two items: 'Account status' (Enabled) and 'B2B invitation'. A sidebar on the left provides navigation links for Overview, Audit logs, Sign-in logs, Diagnose and solve problems, Manage, and Troubleshooting + Support.

## Assigned to Group Based on the New Employee's Department:

### Screenshot:

The screenshot shows the Azure portal interface for the 'IT' group. The top navigation bar includes links for Home, Default Directory, Groups, Groups, All groups, and IT. The main content area displays the group's members under the 'Members' tab. The 'Direct members' section shows one member, John Doe, with details: Name (John Doe), Type (User), Email (john.doe@vivekvash1507@gmail.onmicrosoft.com), and User type (Member). A sidebar on the left provides navigation links for Overview, Diagnose and solve problems, Manage, Properties, Members (selected), Owners, Roles and administrators, Administrative units, Group memberships, Applications, Licenses, Azure role assignments, Activity, and Troubleshooting + Support.

## Personalized Ubuntu VM Provisioned for the New Employee:

### Screenshot:

The screenshot shows the Microsoft Azure portal interface for the 'VirtualMachines' resource group. The left sidebar includes options like Activity log, Access control (IAM), Tags, Resource visualizer, Events, Settings (Deployments, Security, Deployment stacks, Policies, Properties, Locks), Cost Management, Monitoring, Automation, and Help. The main area displays a table of resources under the 'Essentials' section. The table has columns for Name, Type, and Location. The resources listed are:

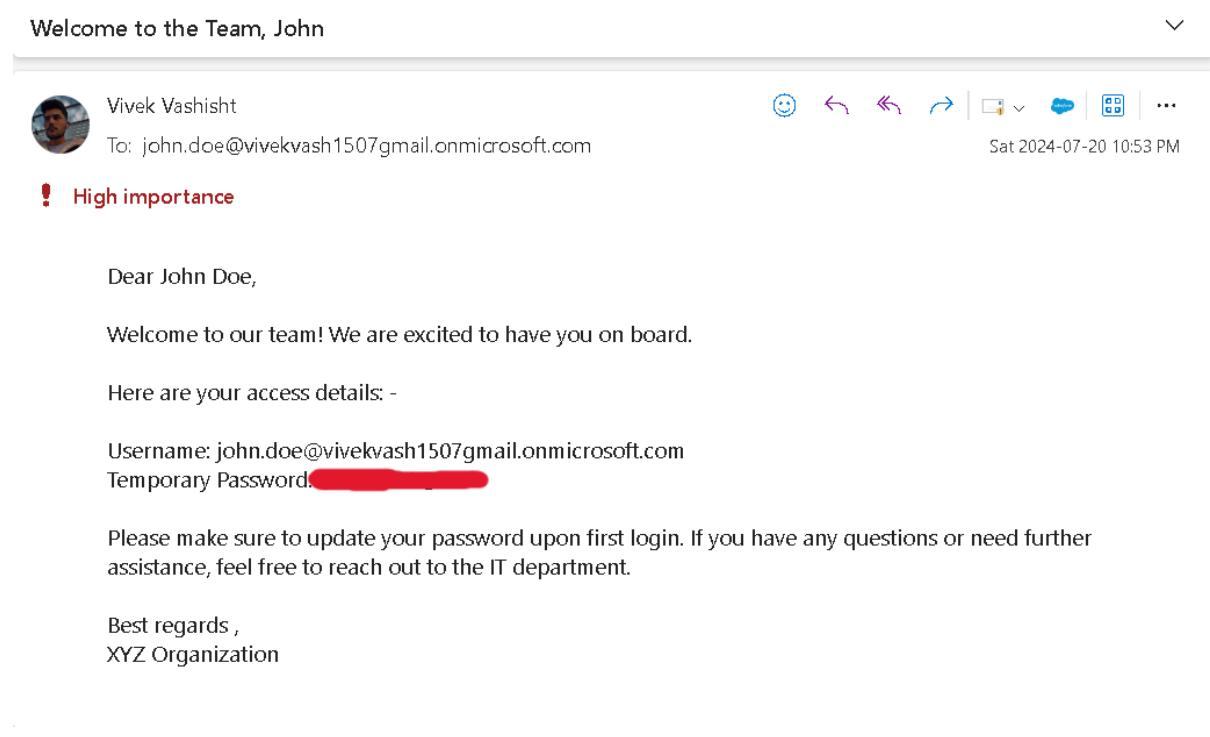
Name	Type	Location
UbuntuVM-John	Virtual machine	East US
UbuntuVM-John-nic	Network Interface	East US
UbuntuVM-John-nsg	Network security group	East US
UbuntuVM-John-pip	Public IP address	East US
UbuntuVM-John-vnet	Virtual network	East US
UbuntuVM-John_disk1_b1d02ab222314ba0a307eea04ce2161a	Disk	East US

The screenshot shows the 'Connect' page for the 'UbuntuVM-John' virtual machine. The left sidebar lists Connect options: Bastion, Networking, Settings, Availability + scale, Security, Backup + disaster recovery, Operations, Monitoring, Automation, and Help. The main area shows connection details: Admin username (john.doe), Port (change) (22), and Just-in-time policy (Unsupported by plan). Below this, two connection methods are recommended: 'SSH using Azure CLI' (Local machine, Azure portal) and 'Native SSH' (Local machine). Both have a 'Select' button and a heart icon. A link 'More ways to connect (3)' is also present.

## **Verification:**

Checked the Logic Apps run history for successful execution and received the welcome email with the expected details.

## **Screenshot:**



## **Conclusion**

The project successfully automated the onboarding process,

showcasing skills in

- 1) Azure Logic Apps
- 2) Microsoft Entra ID (Azure AD)
- 3) Azure Resource Manager

And Strong **Hands-on** Capabilities in the

(Manage Azure identities and governance)

Scope of **Microsoft Certified: Azure Administrator Associate (AZ-104)**

