**Experiment no. 9**

**Aim**: To study and implement Identity and Access Management (IAM) practices on AWS/Azure cloud.

**Requirements**: Azure account

**Theory**:

Microsoft Azure IAM, also known as Access Control (IAM), is the product provided in Azure for RBAC and governance of users and roles. Identity management is a crucial part of cloud operations due to security risks that can come from misapplied permissions. Whenever you have a new identity (a user, group, or service principal) or a new resource (such as a virtual machine, database, or storage blob), you should provide proper access with as limited of a scope as possible. Here are some of the questions you should ask yourself to maintain maximum security:

1. Who needs access?

Granting access to an identity includes both human users and programmatic access from applications and scripts. If you are utilizing Azure Active Directory, then you likely want to use those managed identities for role assignments. Consider using an existing group of users or making a new group to apply similar permissions across a set of users, as you can then remove a user from that group in the future to revoke those permissions.

Programmatic access is typically granted through Azure service principals. Since it’s not a user logging in, the application or script will use the App Registration credentials to connect and run any commands.

2. What role do they need?

Azure IAM uses roles to give specific permissions to identities. Azure has a number of built-in roles based on a few common functions:

* **Owner** – Full management access, including granting access to others
* **Contributor** – Management access to perform all actions except granting access to others
* **User Access Administrator** – Specific access to grant access to others
* **Reader** – View-only access

These built-in roles can be more specific, such as “Virtual Machine Contributor” or “Log Analytics Reader”. However, even with these specific pre-defined roles, the principle of least privilege shows that you’re almost always giving more access than is truly needed.

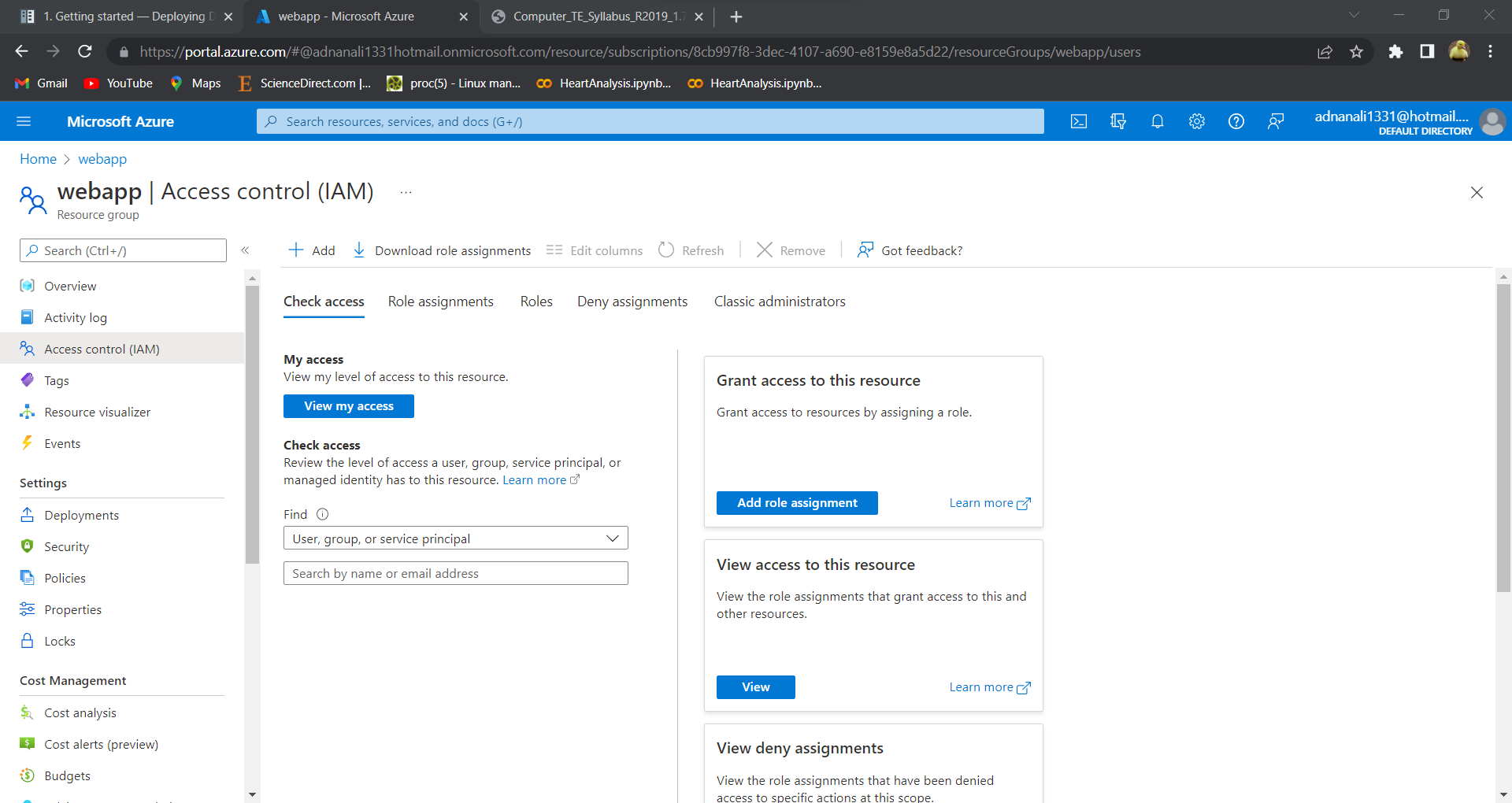
For even more granular permissions, you can create Azure custom roles and list specific commands that can be run.

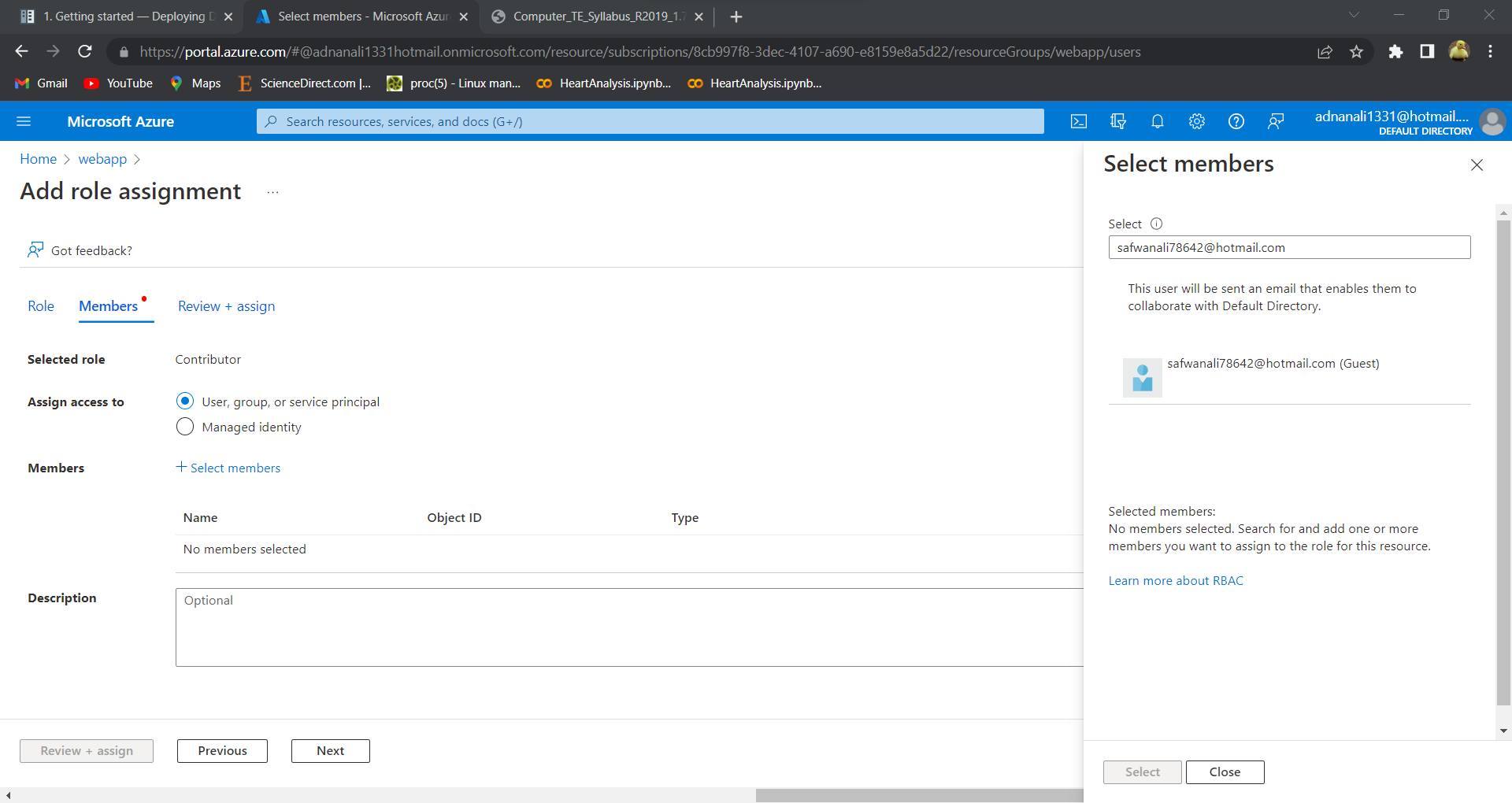
3. Where do they need access?

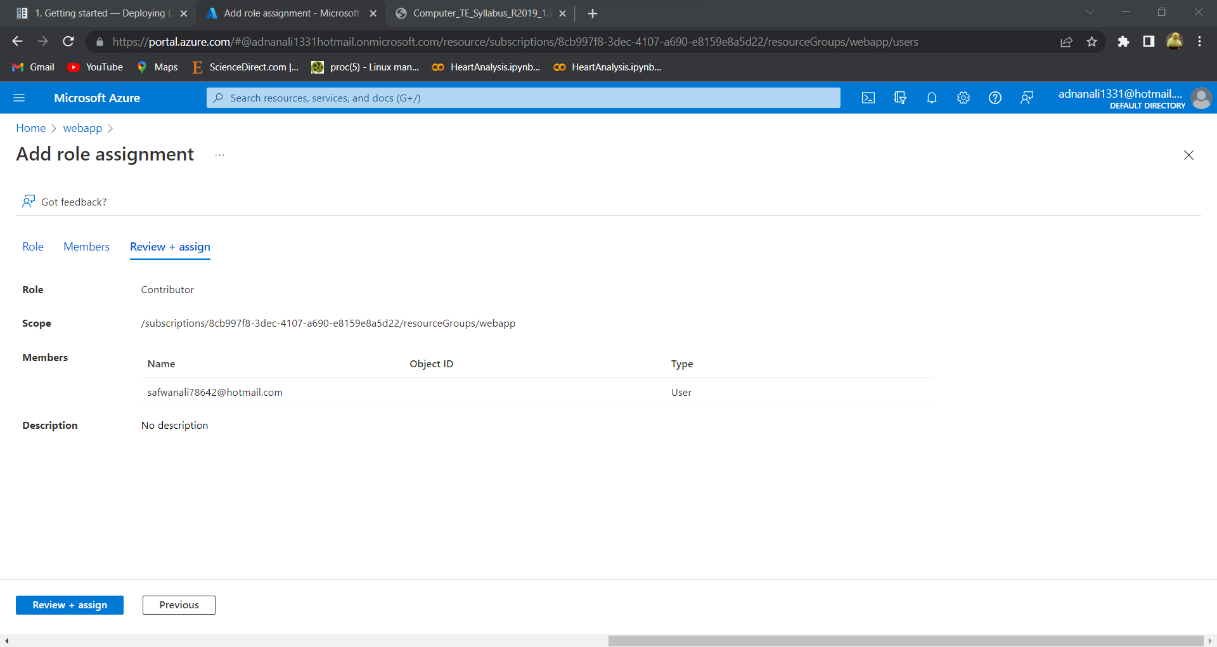
The final piece of an Azure IAM permission set is deciding the specific resource that the identity should be able to access. This should be at the most granular level possible to maintain maximum security. For example, a Cloud Operations Manager may need access at the management group or subscription level, while a SQL Server utility may just need access to specific database resources. When creating or assigning the role, this is typically referred to as the “scope” in Azure.

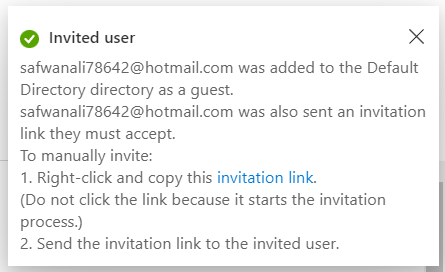
The scope of a role is to always think twice before using the subscription or management group as a scope. The scale of your subscription is going to come into consideration, as organizations with many smaller subscriptions that have very focused purposes may be able to use the subscription-level scope more frequently. On the flip side, some companies have broader subscriptions, then use resource groups or tags to limit access, which means the scope is often smaller than a whole subscription.

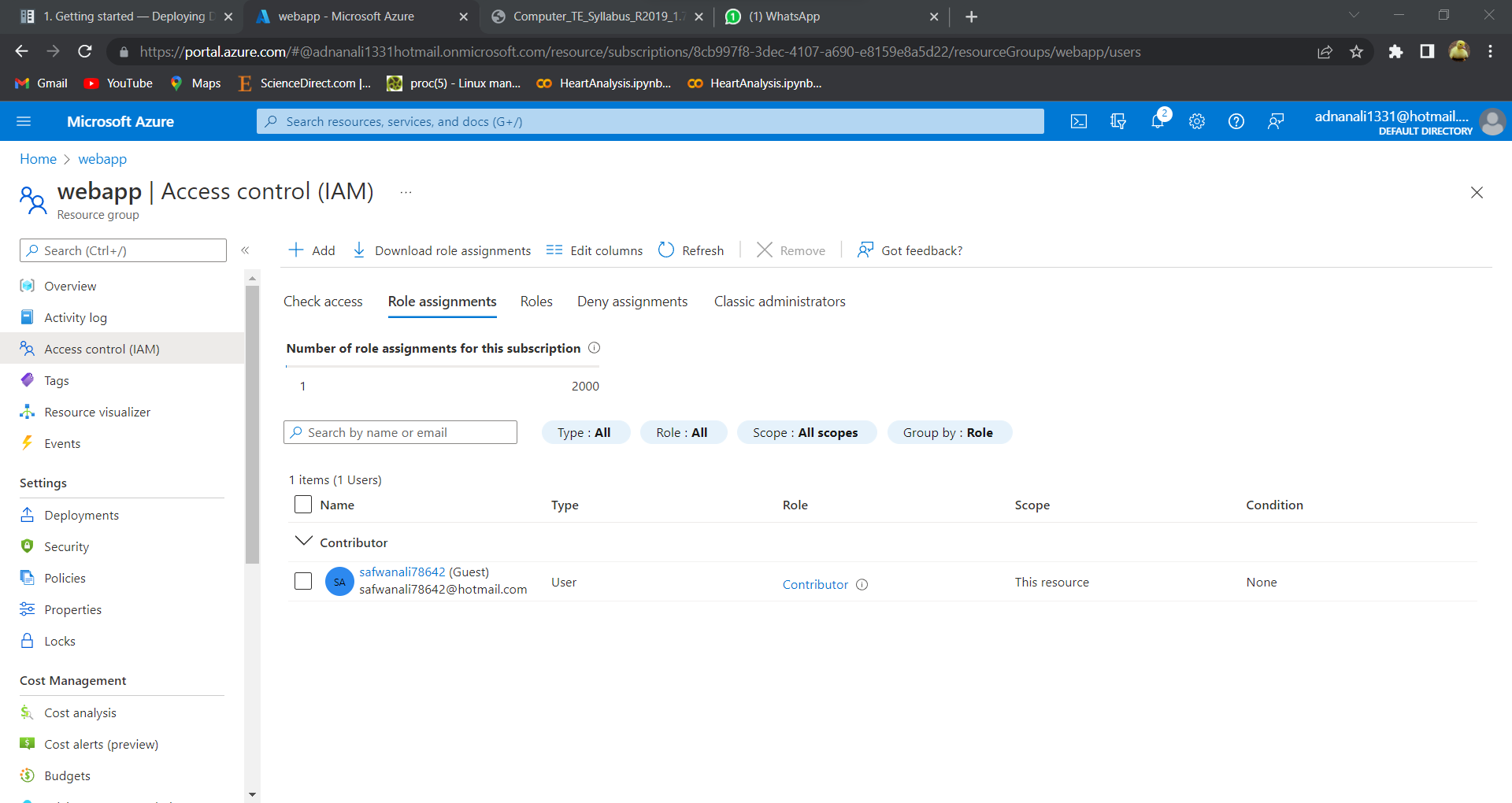
**Output**:











**Conclusion**: We have successfully implemented Identity and Access Management (IAM) practices on Azure cloud.