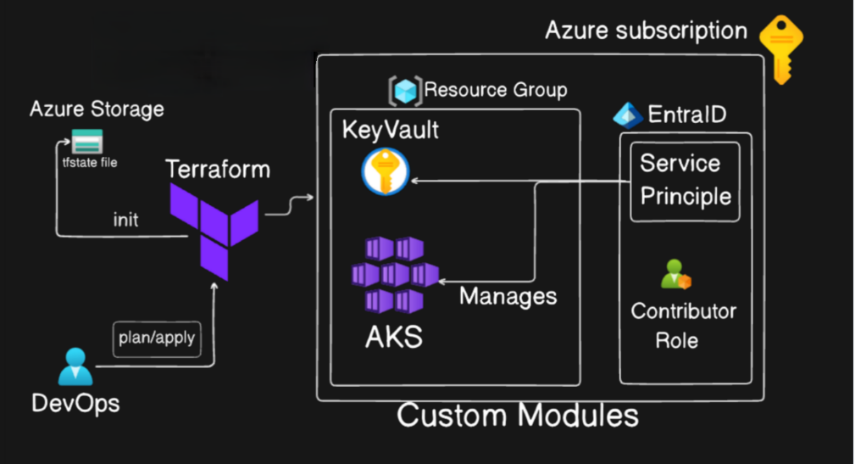
**1.Create an Azure Kubernetes Cluster:**

**1. Design a Terraform configuration to provision an Azure Kubernetes Service (AKS) cluster.**

**2. Ensure the cluster is scalable and resilient.**

My project focuses on designing and implementing a highly scalable, resilient, and secure Azure Kubernetes Service (AKS) infrastructure, entirely managed using Infrastructure as Code (IaC) with Terraform. A core aspect of this design is a modular Terraform architecture, utilizing dedicated custom modules for AKS, Azure Key Vault, and Azure AD Service Principals. This approach ensures reusability, simplifies maintenance, and enables consistent deployments, aligning with best practices for scalable and efficient infrastructure management.

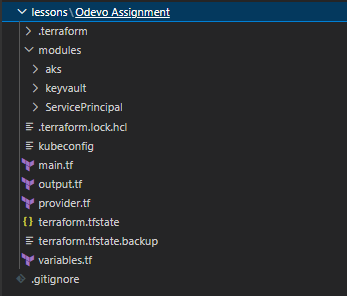


**🧱💡 Why Use Terraform Modules?**

**✅ Benefits**

* **Reusability**: Write once, use across environments (dev, QA, prod).
* **Separation of Concerns**: Keeps code clean and componentized.
* **Maintainability**: Easier to troubleshoot and extend individual components.
* **Scalability**: Modules enable consistent scaling of infrastructure components without duplicating code, making complex environments manageable as they grow.

**Modules Structure:**



**🧩 Modules Created**

* **☸️Aks**: Deploys a scalable and resilient AKS cluster, configured for high availability with features like autoscaling and integration with Azure Monitor.
* **🔒key vault**: Creates a secure Azure Key Vault instance for centralizing sensitive values such as Service Principal secrets, credentials, and certificates, enhancing overall security compliance.
* **🛡️Service Principal**: Automates the creation and configuration of an Azure Active Directory (Azure AD) Service Principal, providing a secure identity for Terraform authentication and other automated tools to access Azure resources with defined roles.

In this setup, I’ve provisioned and managed a Kubernetes environment on Azure using Terraform. The core infrastructure includes an AKS cluster, odevo-aks-cluster, deployed in the **Sweden Central** region. For **resilience and high availability**, the cluster utilizes Standard\_D2s\_v3 VMs (2 vCPUs, 8 GiB RAM) and is strategically deployed across **Availability Zones 1, 2, and 3**. While currently configured with 1 node, the underlying design allows for seamless scaling to accommodate increased load. To support **observability**, Azure Monitor is integrated by linking the AKS cluster to a Container Insights workspace, enabling comprehensive logs and metrics collection for ongoing performance and health monitoring. For improved management and lifecycle control, virtual network components, including the default node pool and VNet, are organized into a separate resource group, odevorg-nrg. **Secure access to secrets** is facilitated by the odevovault Key Vault, provisioned within the Odevorg resource group for locality and simplified access control. Additionally, Azure automatically provisions NetworkWatcherRG for network monitoring tools. All these resources were created using a modular Terraform approach, ensuring the infrastructure is reusable, consistent, and easy to manage across different environments.

**✅Deployed Azure Resources:**

**Resource Groups:**

* defaultresourcegroup-sec: Contains ContainerInsights workspace (defaultworkspace-550ea751-3a3c-4a25-8773-93d9ec783ca5-sec) for AKS monitoring.

[**ContainerInsights(defaultworkspace-550ea751-3a3c-4a25-8773-93d9ec783ca5-sec)**](https://portal.azure.com/#@raghumalangarigmail.onmicrosoft.com/resource/subscriptions/550ea751-3a3c-4a25-8773-93d9ec783ca5/resourceGroups/defaultresourcegroup-sec/providers/Microsoft.OperationsManagement/solutions/ContainerInsights(defaultworkspace-550ea751-3a3c-4a25-8773-93d9ec783ca5-sec))

[**defaultworkspace-550ea751-3a3c-4a25-8773-93d9ec783ca5-sec**](https://portal.azure.com/#@raghumalangarigmail.onmicrosoft.com/resource/subscriptions/550ea751-3a3c-4a25-8773-93d9ec783ca5/resourceGroups/defaultresourcegroup-sec/providers/microsoft.operationalinsights/workspaces/defaultworkspace-550ea751-3a3c-4a25-8773-93d9ec783ca5-sec)

* NetworkWatcherRG: Houses NetworkWatcher\_swedencentral, automatically provisioned by Azure for network diagnostics.

[**NetworkWatcher\_swedencentral**](https://portal.azure.com/#@raghumalangarigmail.onmicrosoft.com/resource/subscriptions/550ea751-3a3c-4a25-8773-93d9ec783ca5/resourceGroups/NetworkWatcherRG/providers/Microsoft.Network/networkWatchers/NetworkWatcher_swedencentral)

* Odevorg: Contains the primary AKS cluster (odevo-aks-cluster) and the Key Vault (odevovault).

[**odevo-aks-cluster**](https://portal.azure.com/#@raghumalangarigmail.onmicrosoft.com/resource/subscriptions/550ea751-3a3c-4a25-8773-93d9ec783ca5/resourceGroups/odevorg/providers/Microsoft.ContainerService/managedClusters/odevo-aks-cluster)[**odevovault**](https://portal.azure.com/#@raghumalangarigmail.onmicrosoft.com/resource/subscriptions/550ea751-3a3c-4a25-8773-93d9ec783ca5/resourceGroups/odevorg/providers/Microsoft.KeyVault/vaults/odevovault)

* odevorg-nrg: Groups networking components related to AKS, such as the default node pool (aks-defaultpool-28341885-vmss) and the Virtual Network (aks-vnet-42319859).

[**aks-defaultpool-28341885-vmss**](https://portal.azure.com/#@raghumalangarigmail.onmicrosoft.com/resource/subscriptions/550ea751-3a3c-4a25-8773-93d9ec783ca5/resourceGroups/odevorg-nrg/providers/Microsoft.Compute/virtualMachineScaleSets/aks-defaultpool-28341885-vmss)[**aks-vnet-42319859**](https://portal.azure.com/#@raghumalangarigmail.onmicrosoft.com/resource/subscriptions/550ea751-3a3c-4a25-8773-93d9ec783ca5/resourceGroups/odevorg-nrg/providers/Microsoft.Network/virtualNetworks/aks-vnet-42319859)[**kubernetes**](https://portal.azure.com/#@raghumalangarigmail.onmicrosoft.com/resource/subscriptions/550ea751-3a3c-4a25-8773-93d9ec783ca5/resourceGroups/odevorg-nrg/providers/Microsoft.Network/loadBalancers/kubernetes)

**☸️AKS Cluster Details:**

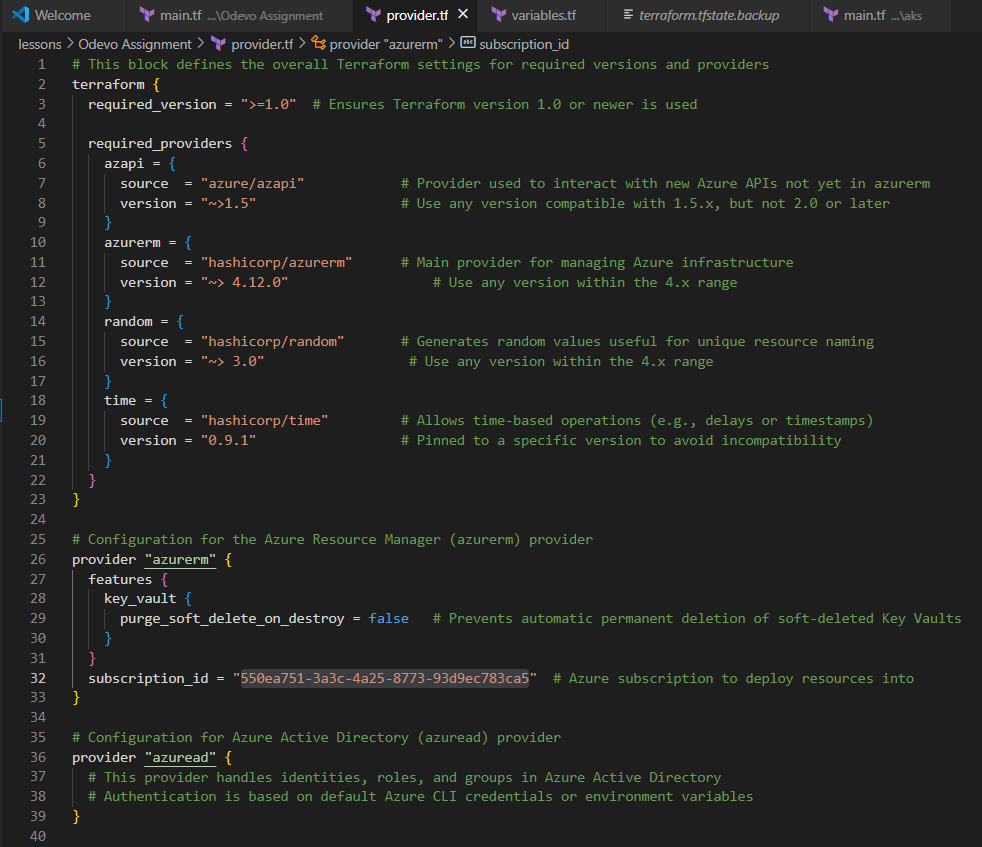
* **Location: Sweden Central**
* **VM Size: Standard\_D2s\_v3 (2 vCPUs, 8 GiB RAM)**
* **🌐Availability Zones: 1, 2, 3**
* **Current Node Count: 1 node (1/1 nodes ready)**
* **Cluster Name: odevo-aks-cluster**

**✅ Terraform files:**

**🔌provider.tf**

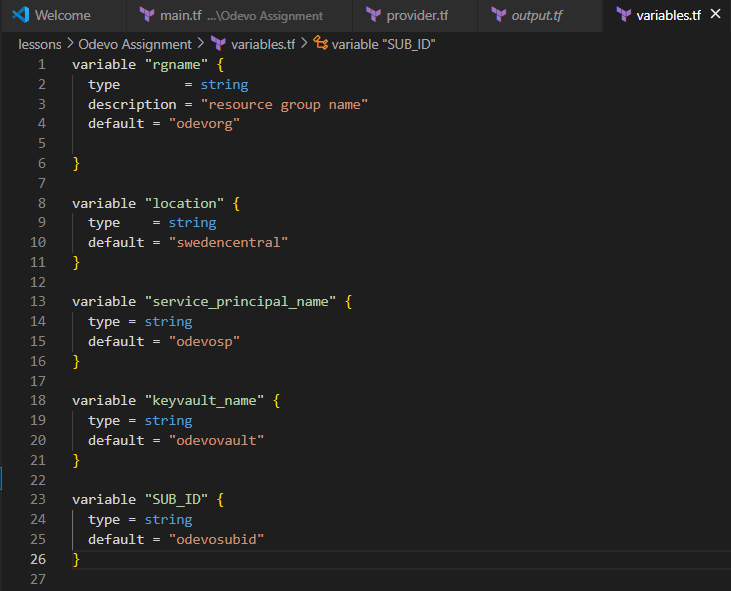
👉"This file defines the cloud providers we're interacting with, in this case, Azure (azurerm), along with specific versions. It also configures global settings like the Azure subscription ID and features like Key Vault soft delete."

👉Self-correction/Add-on: "Specifying provider versions (~> 4.12.0 for azurerm) is crucial for consistent deployments."

****

**🧱variables.tf**

👉 "This file declares all the input variables for our root module. This makes the configuration flexible and reusable, allowing us to easily change parameters like resource group names, locations, or service principal names without modifying the core logic."

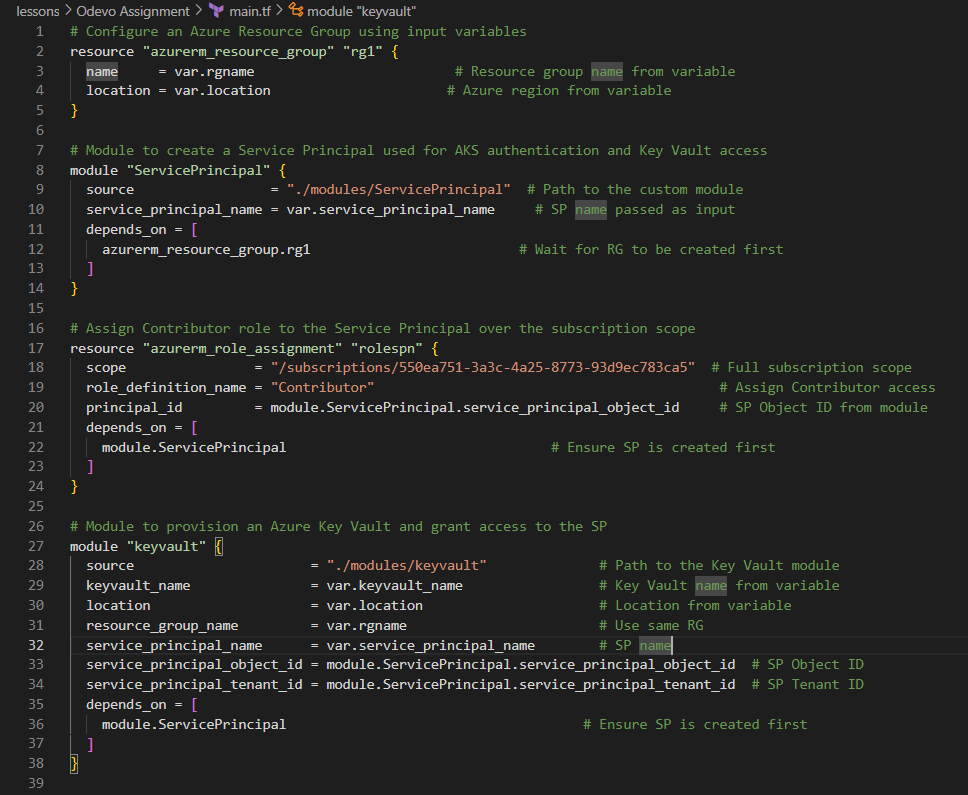


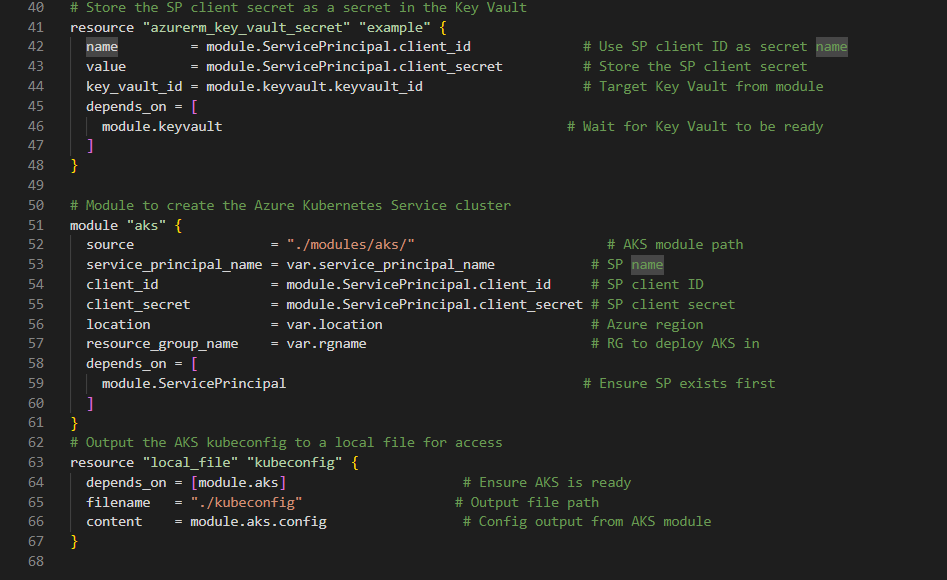
**🏗️main.tf**

👉 "This is the orchestration file where I bring together all the custom modules. Here, I define the resource group and then call our ServicePrincipal, keyvault, and aks modules, passing in the necessary variables. I also manage dependencies to ensure resources are created in the correct order."

👉 "Crucially, the main.tf also shows how the Service Principal client secret is stored in Key Vault, reinforcing our security posture."

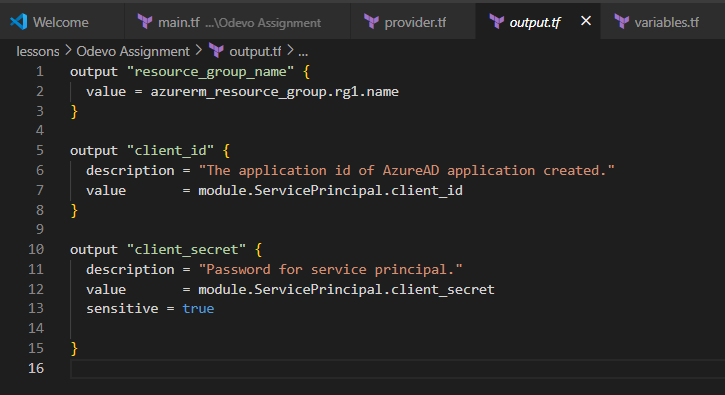
👉 "Finally, it includes a local\_file resource to output the kubeconfig to your local machine for easy access to the cluster, ensuring all sensitive outputs are correctly handled."





**🎯output.tf**

👉 "This file defines the values that will be outputted after Terraform applies the configuration. I specifically output sensitive information like the Service Principal client secret and the kubeconfig details, explicitly marking them as sensitive = true to prevent accidental exposure in the console or state file, adhering to security best practices."



👉 “Each of my custom modules, like aks, keyvault, and ServicePrincipal, follows the standard Terraform module structure with their own main.tf, variables.tf, and output.tf files."

👉 "For instance, the aks module's main.tf contains all the specific resource definitions for the AKS cluster itself. Its variables.tf defines the configurable inputs for that module (like Kubernetes version, node pool sizes, networking details), allowing for a highly customizable deployment of the cluster."

👉 “Similarly, the keyvault module handles the provisioning of the Key Vault resource and its associated access policies, while the ServicePrincipal module automates the creation of the Azure AD Service Principal."

**☸️Azure Kubernetes cluster:**

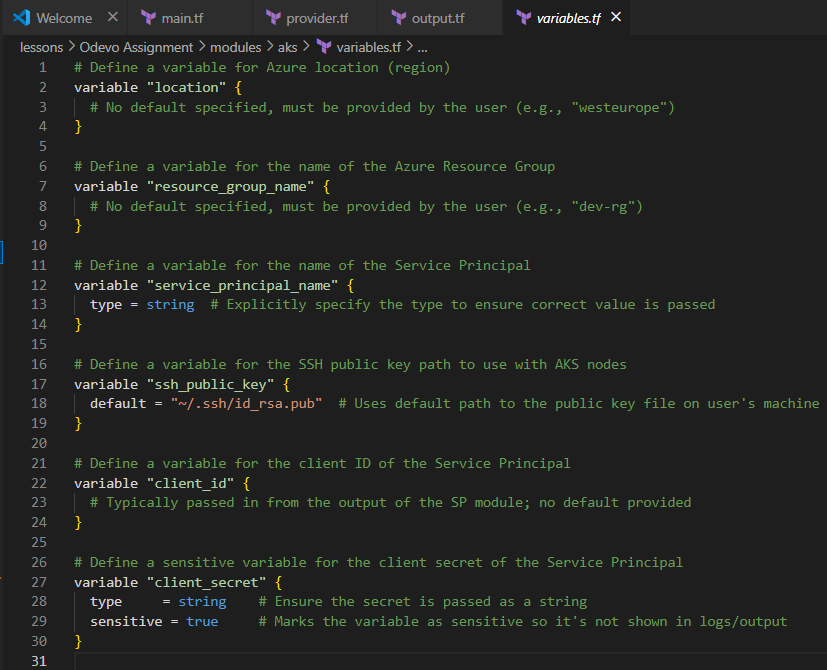
**modules/aks/variables.tf**

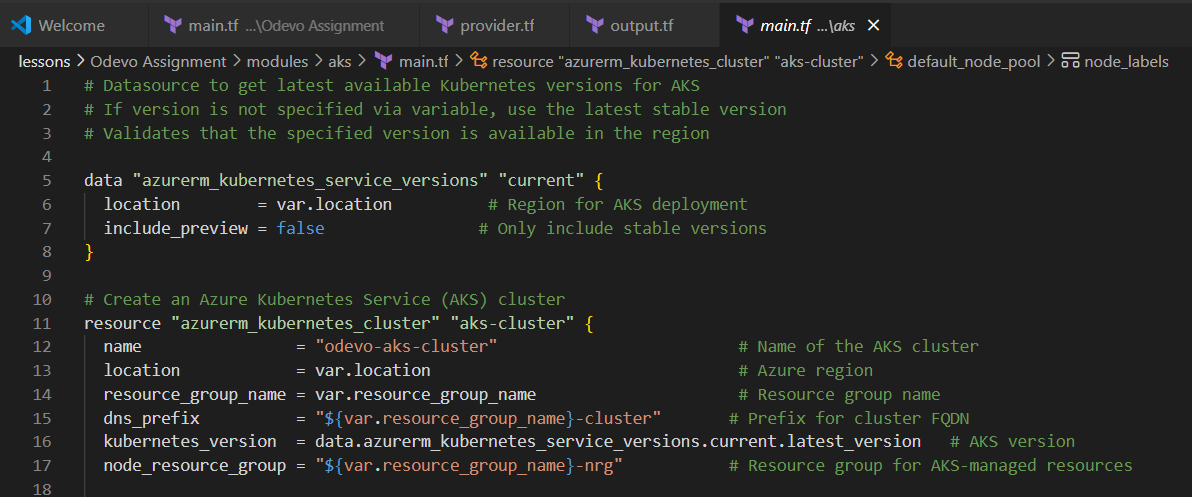
**modules/aks/main.tf**

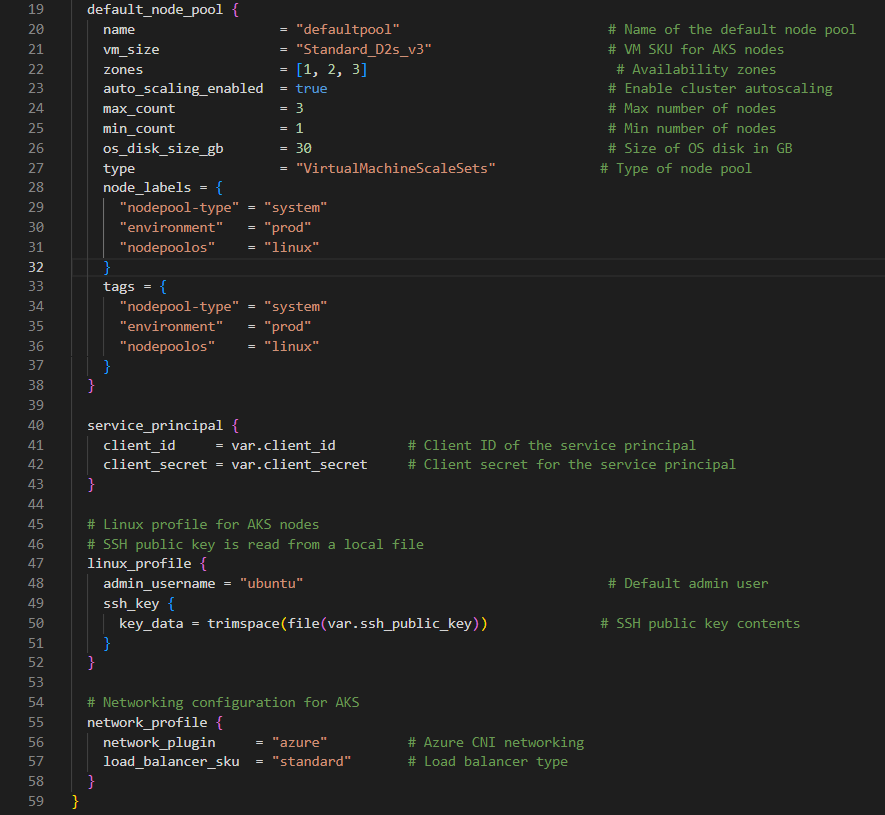
**modules/aks/output.tf**

**✅** This module is responsible for provisioning an Azure Kubernetes Service (AKS) cluster using Terraform. It includes all essential configurations.

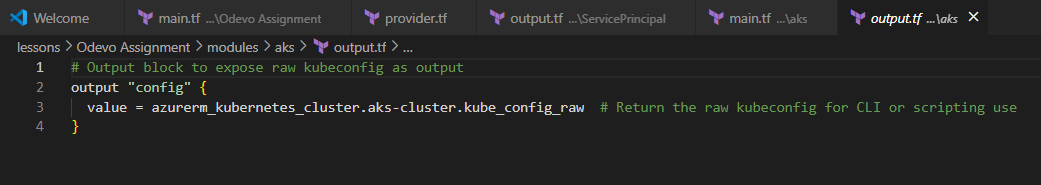
**🧱variables.tf**

****

**🏗️main.tf  
**

****

**🎯output.tf**



**🔒keyvault:**

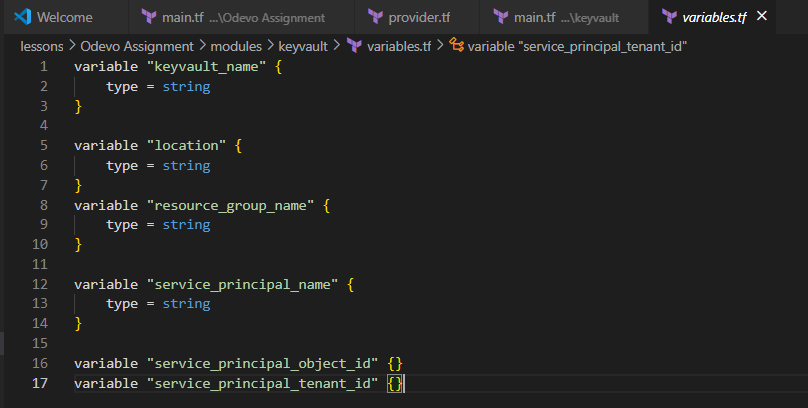
**modules/keyvault/variables.tf**

**modules/keyvault/main.tf**

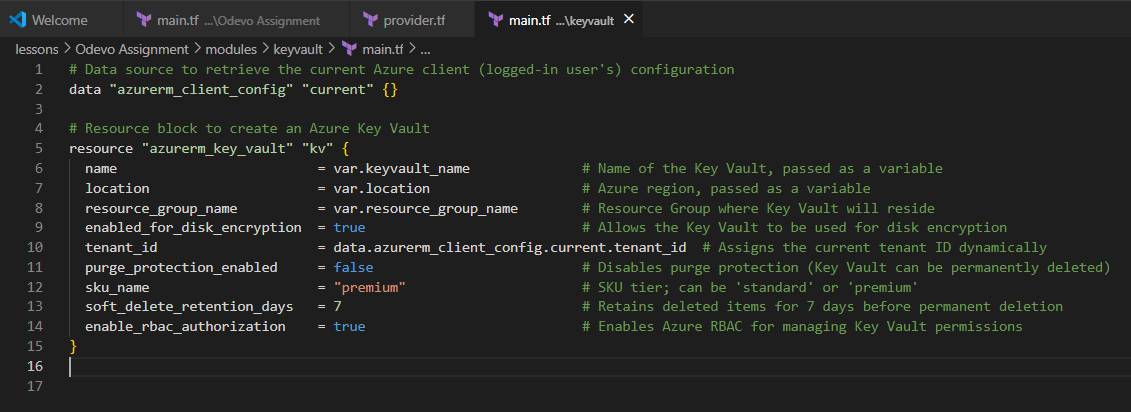
**modules/keyvault/output.tf**

**✅**This module is designed to provision and configure an Azure Key Vault resource using Terraform. The Key Vault stores sensitive values securely, such as Service Principal secrets, SSH keys, certificates, and other secrets required by your infrastructure and applications.

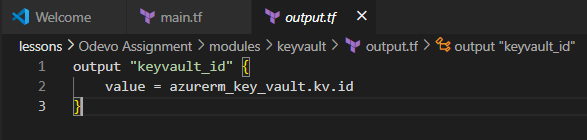
**🧱variables.tf**



**🏗️main.tf**



**🎯output.tf**



**🛡️Service Principal:**

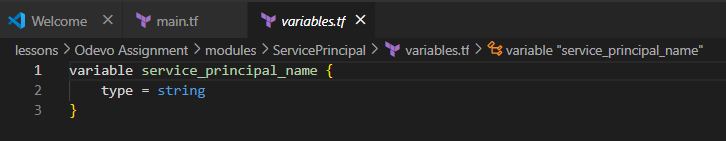
**modules/ServicePrincipal/variables.tf**

**modules/ServicePrincipal /main.tf**

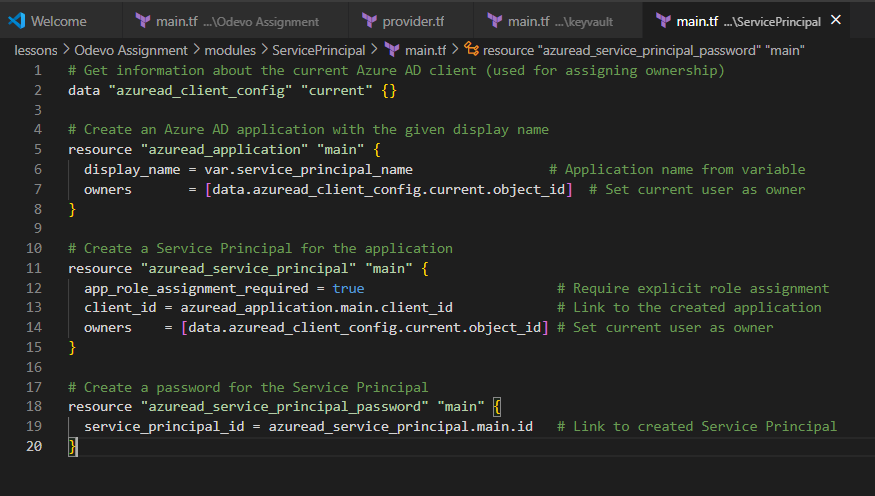
**modules/ServicePrincipal/output.tf**

**✅**This module automates the creation and configuration of an Azure Active Directory (Azure AD) Service Principal using Terraform. The service principal acts as a security identity used by applications, hosted services, and automated tools (like Terraform) to access Azure resources.

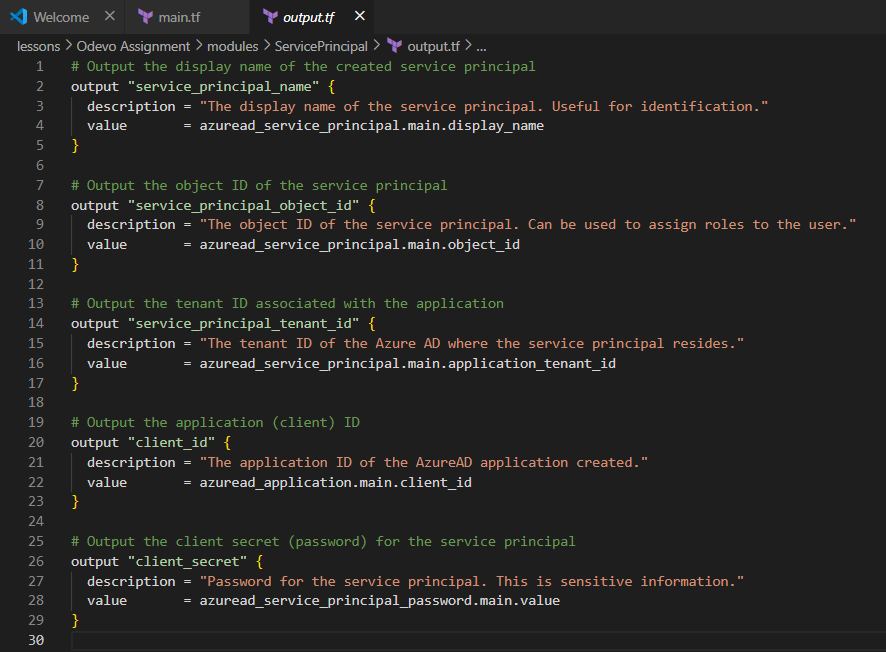
**🧱variables.tf**

****

**🏗️main.tf**



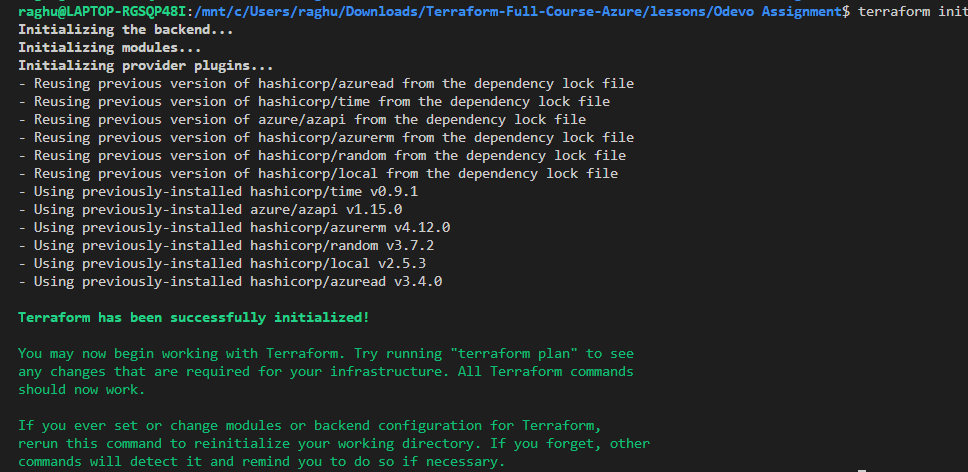
**🎯output.tf**



**Terraform Commands and Their Purpose:**

**✅terraform init**

* **Purpose:** This command initializes a Terraform working directory. It performs several essential setup tasks, including:
  + Downloading and configuring the necessary provider plugins (e.g., azurerm, azapi, random, time).
  + Setting up the backend for the Terraform state file. In your project's design, the state file (. tfstate file) is configured to be stored securely in Azure Storage. This is crucial for collaborative environments and for maintaining a reliable record of your deployed infrastructure.
* **Result (Screenshot):** The document provides a screenshot of the terraform init command output. This output typically shows successful initialization, provider plugin downloads, and backend configuration.

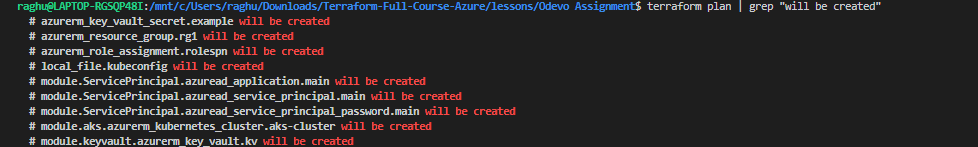


**✅terraform plan**

**Purpose:** This command creates an execution plan, which is a preview of the changes Terraform will make to your infrastructure to match the desired state defined in your configuration files. It does not make any changes to your actual cloud resources; it only shows you what *would* happen.

**Result (Screenshot):** The screenshot for terraform plan would display a detailed summary of resources to be added, changed, or destroyed. For your project, this would include the creation of:

* Resource groups (e.g., Odevorg, odevorg-nrg).
* An Azure AD Service Principal.
* An Azure Key Vault (odevovault).
* An AKS cluster (odevo-aks-cluster) and its associated networking components.
* Log Analytics Workspace and Network Watcher related resources



**✅terraform apply**

**Purpose:** This command executes the actions proposed in a Terraform plan (or generates and executes a new plan if no plan file is specified). This is the step where Terraform actually provisions or modifies your Azure resources to match your .tf configuration files. You are prompted to confirm the changes before they are applied.

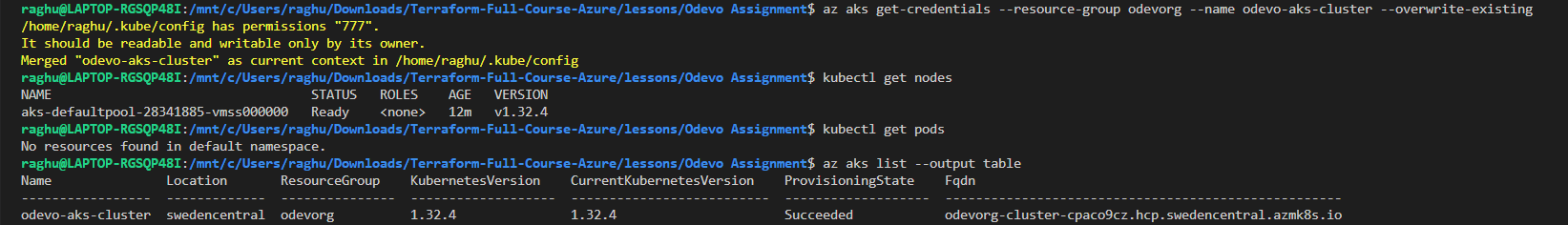
**Result (Screenshot):** The terraform apply screenshot shows the progress of resource creation and ultimately indicates whether the apply operation was successful, listing the number of resources added, changed, or destroyed

.



**✅kubectl get nodes**

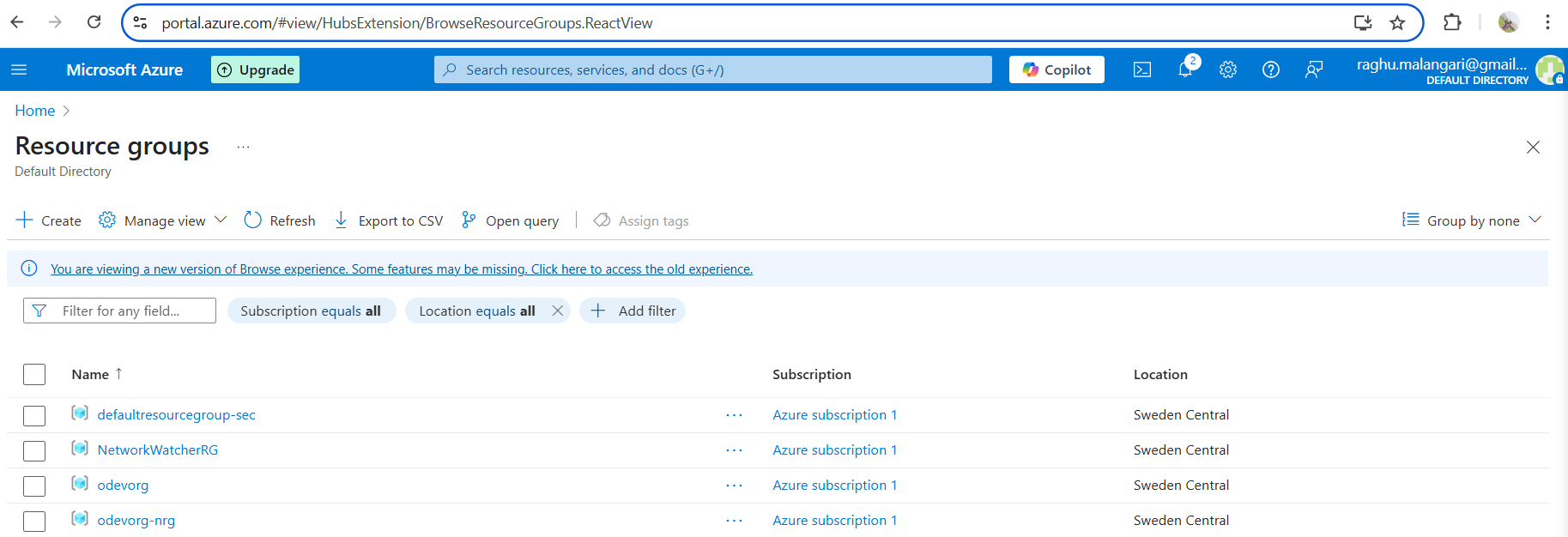
The kubectl get nodes command would also confirm that the Kubernetes cluster is running and its nodes are ready, matching the 1/1 ready nodes count reported in your Azure portal overview.

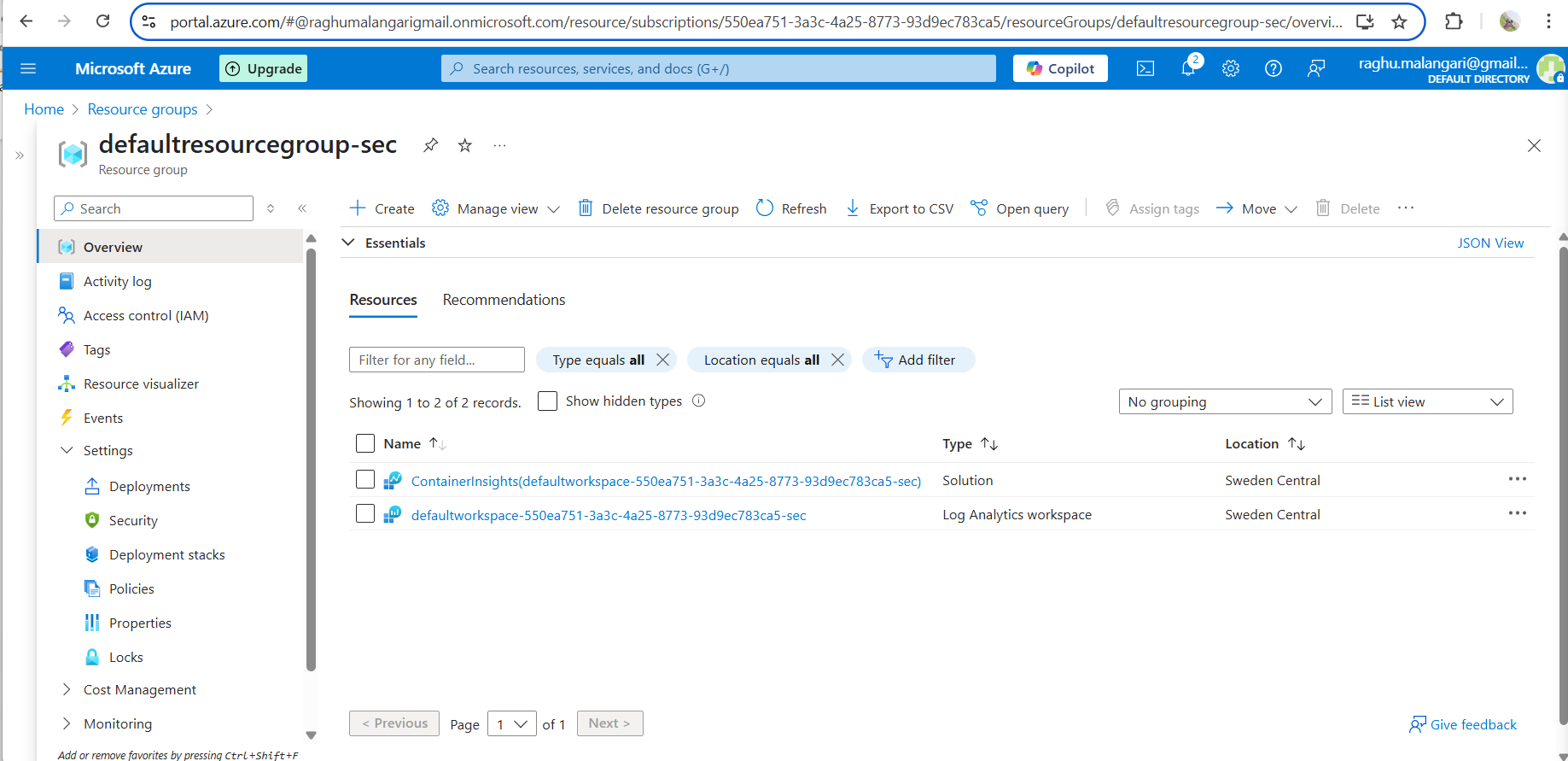


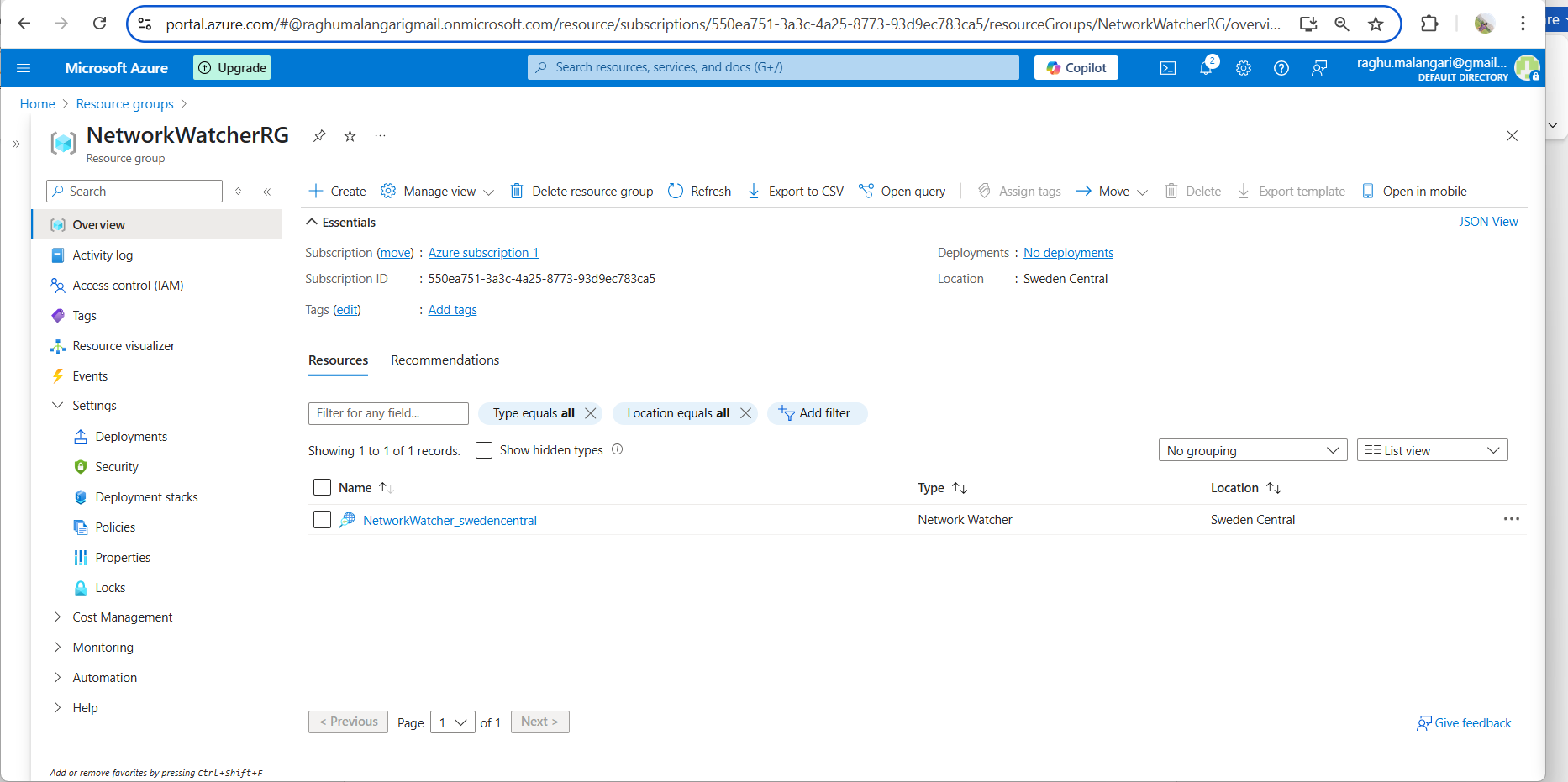
**✅ 🧭Resource Groups in Microsoft Azure Portal**:

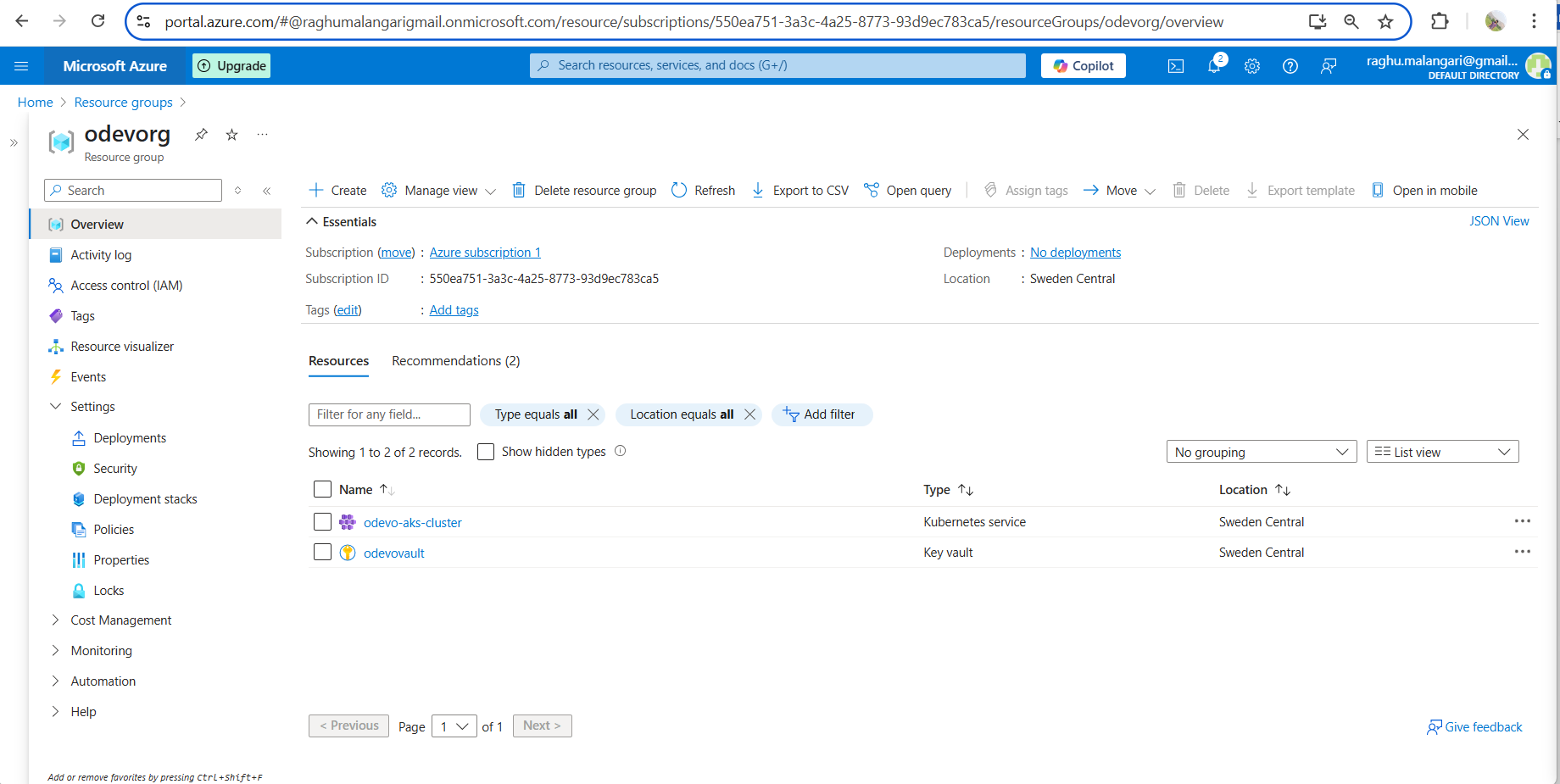
After a successful terraform apply, you would observe the following resources and their status in your Azure subscription, as demonstrated by the screenshots in your document:

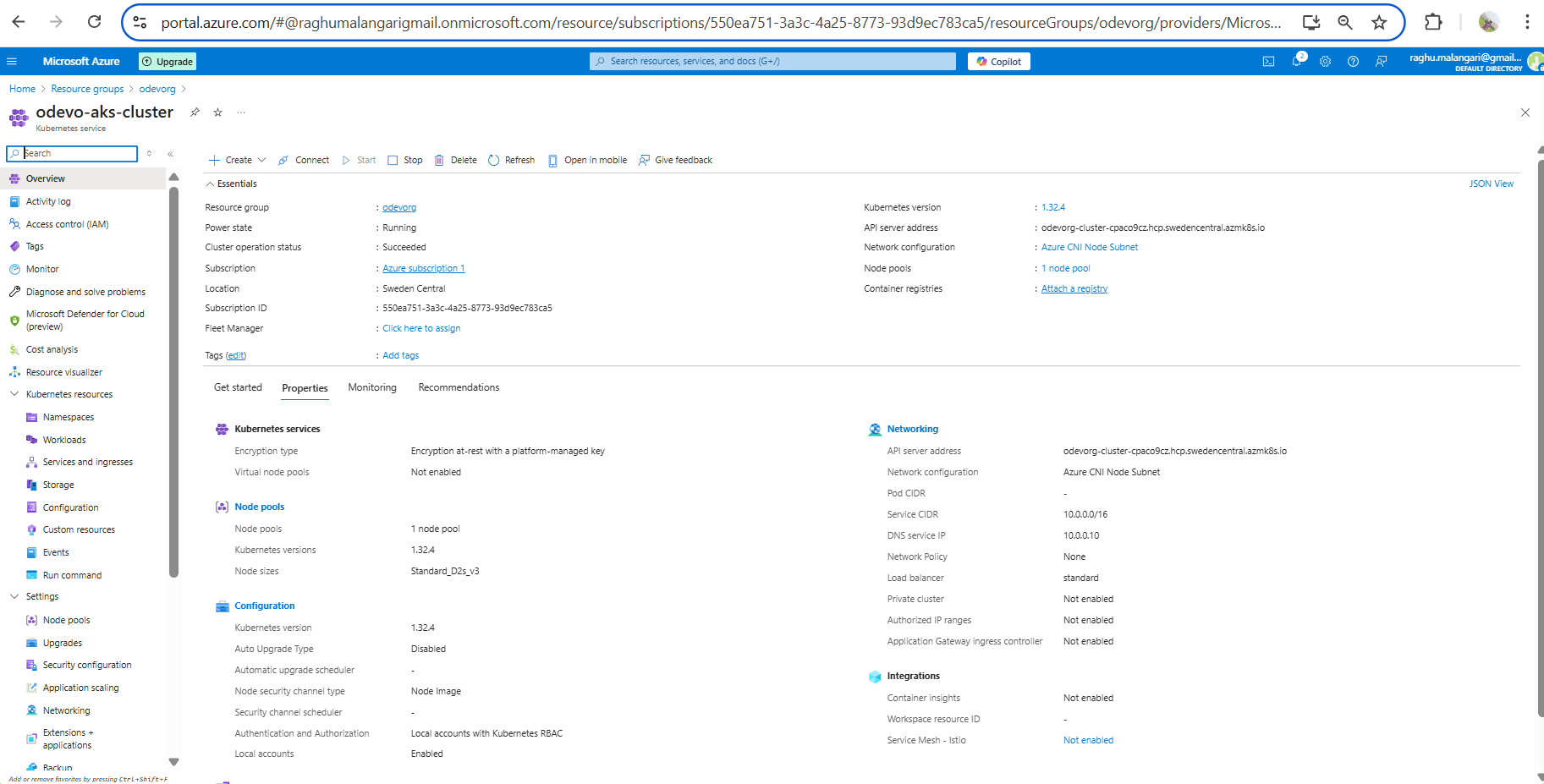
* **Resource Groups:** You would see multiple resource groups created to organize your infrastructure, including:
  + Odevorg: Containing the primary AKS cluster (odevo-aks-cluster) and the Key Vault (odevovault).
  + odevorg-nrg: Grouping networking components like the default node pool (aks-defaultpool-28341885-vmss) and the Virtual Network (aks-vnet-42319859).
  + defaultresourcegroup-sec: Hosting the ContainerInsights workspace (defaultworkspace-550ea751-3a3c-4a25-8773-93d9ec783ca5-sec) used for AKS monitoring.
  + NetworkWatcherRG: Automatically provisioned by Azure, containing NetworkWatcher\_swedencentral for network diagnostics.
* **Azure Kubernetes Service (AKS) Cluster:**
  + **Name:** odevo-aks-cluster.
  + **Location:** Sweden Central.
  + **VM Size:** Standard\_D2s\_v3 (2 vCPUs, 8 GiB RAM) for the nodes.
  + **Availability Zones:** The cluster nodes are deployed across Availability Zones 1, 2, and 3 for high availability.
  + **Node Count:** Initially, 1 node is provisioned and reported as ready.
* **Azure Key Vault:**
  + **Name:** odevovault.
  + **Location:** Within the Odevorg resource group.
* **Azure Active Directory (Azure AD) Service Principal:** While not directly visible as a "resource" in the main Azure portal overview (it's in Azure AD), the Service Principal would be created and configured with the Contributor role over your subscription scope, as defined in your main.tf.
* **Azure Monitor Integration:** The AKS cluster is linked to the ContainerInsights workspace for continuous logging and metrics collection, visible under the monitoring section of your AKS cluster in the portal.

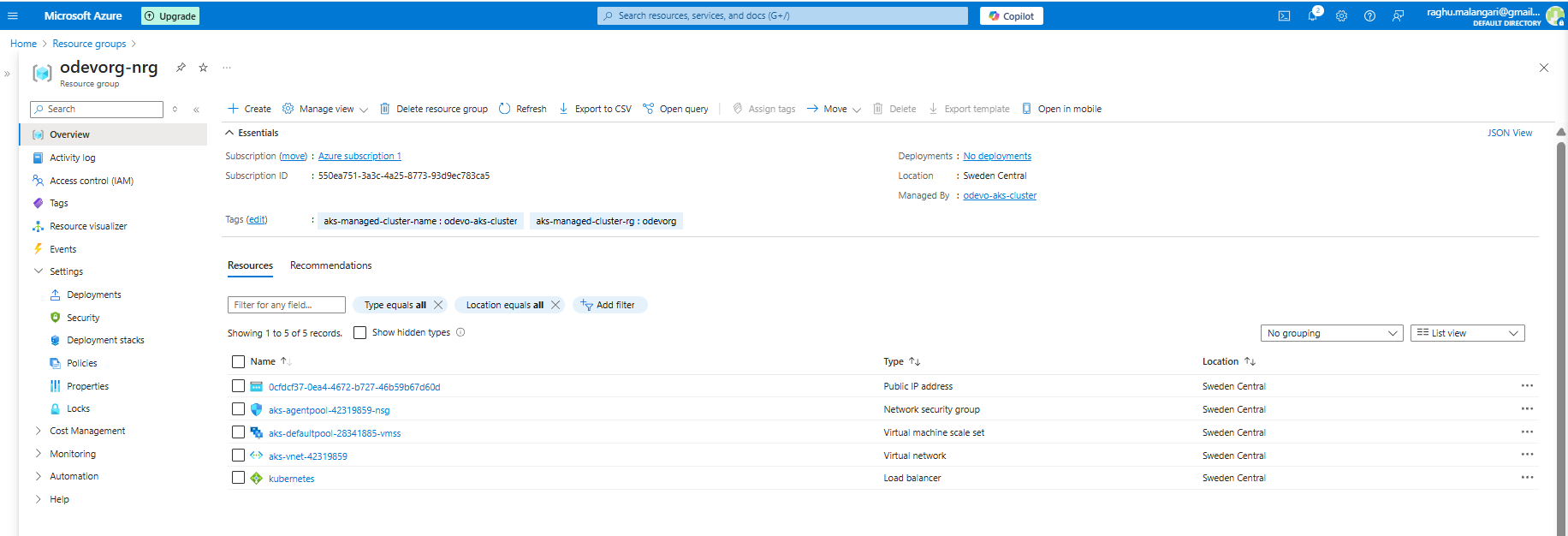












**Resources:**  
<https://registry.terraform.io/providers/hashicorp/azurerm/latest/docs/resources/kubernetes_cluster>

https://learn.microsoft.com/en-us/cli/azure/?view=azure-cli-latest