**Handout: Administering Azure Resources (AZ-104 Module)**

**Overview**

This module provides essential knowledge for managing Azure resources. Topics include the use of key tools (Azure Portal, PowerShell, Azure CLI), deploying resources with ARM templates, and understanding the advantages of Infrastructure as Code (IaC) within Azure.

**1. Azure Resource Management Tools**

**Azure Portal**

* **Purpose**: Provides a web-based graphical interface for creating, managing, and monitoring Azure resources.
* **Key Features**:
  + **User-Friendly Interface**: Customizable dashboards, resource views, and help menus.
  + **Resource Configuration**: Offers guided setup and advanced settings.
  + **Keyboard Shortcuts**: Efficiency features for experienced users.
  + **Hands-On**: Suitable for beginners and visual management needs.

**Azure Cloud Shell**

* **Purpose**: Browser-accessible CLI environment available with both Bash and PowerShell.
* **Capabilities**:
  + **Integrated Shell Options**: Allows for Azure management commands in both Bash and PowerShell.
  + **Temporary Session**: Cloud Shell uses a temporary environment; requires a storage account for persistence.
  + **Text Editor**: Built-in code editor within the Shell, beneficial for scripting.

**PowerShell for Azure**

* **Purpose**: Provides a command-line tool for advanced, scriptable management of Azure resources.
* **Key Elements**:
  + **Az Module**: Installable package containing Azure-specific cmdlets.
  + **Commands**: Offers a range of commands for creating, configuring, and managing resources.
  + **Remote Execution**: Ability to connect and manage Azure from a local or remote machine.

**Azure CLI**

* **Purpose**: A cross-platform command-line tool used for scripting and quick Azure management tasks.
* **Structure**:
  + **Groups and Subgroups**: Organized commands for specific resource actions.
  + **Interactive Use**: Great for repeated tasks or scripts that don’t require a GUI.
  + **Installation**: Can be run locally or from Cloud Shell.

**2. Configuring Resources with ARM Templates**

**ARM Templates Overview**

* **Definition**: JSON-based files defining Azure infrastructure as code, allowing consistent and automated resource provisioning.
* **Benefits**:
  + **Consistency**: Ensures the same configurations across multiple deployments.
  + **Automation**: Reduces manual input, supporting efficient Infrastructure as Code (IaC) practices.
  + **Reusability**: Template files can be reused and modified as needed.

**ARM Template Structure**

* **Schema**: Specifies the JSON schema version and defines the structure of the template.
* **Parameters**: Input fields for custom values, such as names, regions, or SKUs, which make templates versatile.
* **Resources**: The main section that defines resources like VMs, storage accounts, or networks.
* **Outputs**: Provides outputs post-deployment, such as resource IDs or connection strings.

**Using Azure Bicep**

* **Purpose**: Bicep is an alternative to JSON, offering a simpler, cleaner syntax for defining ARM templates.
* **Advantages**:
  + **Concise Syntax**: Bicep files are easier to read and write.
  + **Automatic Dependency Detection**: Detects resource dependencies, simplifying complex templates.
  + **Tool Support**: Visual Studio Code provides validation and IntelliSense for Bicep.

**3. Key Demonstrations of Resource Management Tools**

**Azure Portal Demonstration**

* **Resource Creation**: Using the Azure Portal to create, organize, and configure resources like VMs and storage accounts.
* **Resource Grouping and Navigation**: Grouping resources by purpose or department for better organization.
* **Custom Dashboards**: Customizing the portal for specific roles or tasks.

**Cloud Shell Demonstration**

* **Shell Configuration**: Setting up and navigating Cloud Shell with either Bash or PowerShell.
* **Basic Commands**: Practicing common commands in Azure PowerShell or CLI, like listing resources or creating a resource group.
* **Text Editor Use**: Using the built-in editor to create or modify scripts within Cloud Shell.

**PowerShell Demonstration**

* **Az Module Installation**: Steps to install the Az module for Azure resource management.
* **Common Cmdlets**:
  + **New-AzResourceGroup**: Create a new resource group.
  + **Get-AzVM**: Retrieve information about existing virtual machines.
  + **Set-AzVM**: Configure VM settings.
* **Session Management**: Connecting and managing sessions within Azure for efficient remote administration.

**Azure CLI Demonstration**

* **Installation and Verification**: Verifying the CLI installation, logging in, and checking subscription details.
* **Creating Resources**: Using CLI commands to create a resource group and verify its status.
* **Examples**:
  + **az group create**: Creates a new resource group.
  + **az vm create**: Provisions a virtual machine with specific configurations.
  + **az vm restart**: Restarts an existing virtual machine.