**Session 01: Azure DevOps Learning Plan**

**1. Azure Services Overview**

* **Key Azure Services to Cover Initially:**
  + **Management Groups**
  + **Subscriptions**
  + **Resource Groups**
  + **Virtual Machines (VMs)**
  + **Storage**
* These services form the foundational knowledge required to start with Azure DevOps. We'll cover more Azure services in subsequent sessions.

**2. Design Tool: draw.io**

* We’ll use **draw.io** as a design tool to create visual representations of architecture.

**3. DevOps Basics**

* **DevOps** is a culture that promotes collaboration between development and operations teams. This session will cover the foundational aspects of DevOps, which we will dive deeper into in future sessions.

**4. Importance of Prompt Engineering**

* Understanding prompt engineering is crucial in the context of DevOps automation, AI, and cloud solutions.

**5. OS Knowledge: Windows and Linux**

* **Windows OS** knowledge is essential, and we will also cover **Linux fundamentals** needed for DevOps, as it’s a vital skill to support various cloud services and infrastructure management.

**6. Traditional Simple Deployment**

* We’ll start with a simple, traditional deployment to understand the basic flow. This will serve as the foundation for more advanced DevOps practices.

**7. Infrastructure as Code (IaC) – Terraform**

* **Terraform** will be introduced for Infrastructure as Code. If time allows, we will also briefly discuss **ARM Templates** and **Bicep** as alternative IaC tools.

**8. Git/GitHub/GitHub Actions**

* we’ll cover how to use **Git**, **GitHub**, and **GitHub Actions** for source control and CI/CD workflows.

**9. Azure DevOps**

* Deep dive into **Azure DevOps** for implementing DevOps pipelines, managing source control, and automating deployments.

**10. Azure PowerShell**

* **Azure PowerShell** will be covered for managing Azure resources programmatically through the command line.

**11. Docker & Kubernetes**

* Introduction to **Docker** for containerization and **Kubernetes** for container orchestration.

**12. Monitoring**

* **Monitoring** is critical for understanding the health of your system and will be discussed using Azure Monitor and other tools.

**Career Path After Completion:**

* After completing the sessions, you can apply for job roles such as:
  + **Azure + Terraform**
  + **Azure + PowerShell**
  + **Azure + Terraform + PowerShell**
  + **Azure + Azure DevOps**
  + **Azure + Azure DevOps + AKS (Azure Kubernetes Service)**
  + **Site Reliability Engineer (SRE)**

**Design Tool Explanation:**

**High-Level Design (HLD)**

* HLD provides a **high-level overview** of the system architecture, highlighting the major components, their interactions, and the overall system structure. It defines the **blueprint** of the solution without delving into the granular technical details.

**Low-Level Design (LLD)**

* LLD focuses on the **implementation specifics**. It provides detailed instructions on how to configure and implement each component described in the HLD. LLD addresses technical concerns such as exact configurations, settings, and system integration.

**DevOps Basics:**

* **DevOps** is a combination of **Development + Operations**, aimed at improving collaboration between development and IT operations teams.
* DevOps is both a **culture** and a **set of practices** designed to help organizations deliver applications and services more efficiently and reliably. More detailed discussions will follow in later sessions.

**Sample Basic Deployment:**

**Scenario: Creating a Windows VM and Installing IIS**

1. **Create a Windows VM** in Azure.
2. **Install IIS** (Internet Information Services) on the VM.
3. **Verify IIS Installation:**
   * Open a web browser and visit http://localhost. You should see the default IIS welcome page, indicating that IIS is installed correctly.
4. **Navigate to the IIS default web folder:**
   * Go to C:\inetpub\wwwroot.
5. **Copy the StreamFlix files:**
   * Copy all the extracted files from the StreamFlix folder into the C:\inetpub\wwwroot folder directly (no subfolders should be created).
   * Code available at session\_01 folder