Project “Epimoni” Scope

Project Title: Azure Global Network Deployment and Management

Objectives:

- Demonstrate the ability to deploy Azure resources using ARM templates.
- Demonstrate the ability to deploy Azure resources using the Azure Portal.
- Demonstrate the ability to manage Azure resources using Azure Cloud Shell, PowerShell, Azure CLI.
- Achieve a globally peered network with VMSS/autoscaling and load balancing.

Deliverables:

- Begin with several locally peered networks in Azure.
- The end result is a fully configured globally peered network in Azure.
- Load balancing setup for Azure resources like VMs and web apps.
- Autoscaling implemented via Virtual Machine Scale Sets (VMSS) and Azure Container Apps (ACAs).
- Creation of individual VMs and application of NSG rules at VNET/NIC levels.
- Effective GitHub repository management showcasing continuous updates and documentation.
- Running web apps in specific networks, shielded behind load balancers.
- Implementation of resource deallocation strategies to minimize costs during non-peak hours.

Technical Requirements:

- Azure PAYG subscription and access to Azure services.
- Knowledge of Azure networking, VMSS, ACAs, and load balancing.
- Proficiency in using ARM templates for resource deployment.
- Familiarity with NSG configurations and Azure network security.
- Creation and management of Alerts and Action Groups for cost and resource utilization
- Understanding of GitHub for version control and repository management.
- Ability to configure and manage web apps in Azure.

Timeline and Milestones:

- Week 1:
  - Set up basic network infrastructure and VM deployment using ARM templates.
  - Document the process and take initial screenshots.
- Week 2:
  - Implement load balancing and autoscaling. Begin setting up web apps and integrating them with the network.
  - Document the process and take screenshots for the highlight reel and repo
- Week 3:
  - Finalize configurations, apply NSG rules, and optimize resource deallocation.
  - Complete documentation and prepare the GitHub repository for public viewing.

Resources Needed:

- Azure Cloud Services (Networking, Compute, Storage).
- ARM Templates for resource deployment.
- Alerts and Action Groups for cost optimization and monitoring
- GitHub account for repository management.
- Documentation tools (e.g., Microsoft Word, Markdown for GitHub README).
- Screen capture tool for documentation and highlight reel.

Potential Challenges:

- Learning and implementing unfamiliar Azure services and features.
- Time management for extensive project documentation.
- Ensuring clear and concise documentation that effectively explains technical configurations and decisions.

Additional Considerations:

- Regularly review Azure service updates and documentation to ensure the use of current best practices.
- Consider setting up alerts or monitoring in Azure to keep track of resource utilization and performance.
- Plan for regular project reviews to assess progress and adjust the scope as needed.

This scope document serves as a dynamic blueprint for your project. It's normal for some aspects to evolve as you progress, so be flexible and update the document as needed.