**Handout: Administering PaaS Compute Options**

This handout provides a structured guide to understanding and administering Azure PaaS (Platform-as-a-Service) compute options, focusing on Azure App Service Plans, scaling, deployment slots, and app security.

**1. Introduction to PaaS Compute Options**

Azure's PaaS offerings provide a managed environment for building, deploying, and scaling web and mobile applications efficiently. These services allow developers to focus on application development without the overhead of managing infrastructure.

**2. Understanding Azure App Service Plans**

**Definition**:

* An App Service Plan defines the set of compute resources for web apps, APIs, and mobile backends to run.
* It determines the **performance**, **price**, and **features** available.
* A single plan can host multiple apps, allowing resource sharing.

**Key Configuration Points**:

* **Region**: Location where resources are created.
* **Number and Size of VM Instances**.
* **Pricing Tier**:
  + **Shared Compute (Free/Shared)**: Basic, limited scalability.
  + **Dedicated Compute (Basic, Standard, Premium)**: Higher performance and scalability.
  + **Isolated**: Exclusive VMs in a dedicated VNet for maximum isolation.

**3. Scaling App Service Plans**

**Scale Up**:

* **Definition**: Increase the resources of the current VM (e.g., more CPU, memory).
* **Benefits**: More features (e.g., staging slots, autoscaling).

**Scale Out**:

* **Definition**: Add more VM instances to distribute traffic.
* **Options**:
  + **Manual Scaling**: Fixed number of instances.
  + **Auto-Scaling**: Adjust based on metrics like CPU, memory, HTTP requests, or scheduled times (e.g., peak hours).

**Configuration Considerations**:

* Scaling improves **availability** and **fault tolerance**.
* Combine metric-based and schedule-based rules for flexibility.

**4. Creating and Managing App Services**

**Implementing App Services**:

* **Types**: Web Apps, API Apps, Mobile Apps, and Function Apps.
* **Features**:
  + Fully managed environment, enabling rapid development.
  + Supports popular development languages such as .NET, Java, and Python.
  + Enterprise-grade security and compliance.

**Creating an App Service**:

* Ensure the app name is unique.
* Accessible via a default URL (azurewebsites.net), customizable with custom domains.
* Deploy using runtime stacks or Docker containers on Linux or Windows platforms.

**5. Deployment Slots**

**Definition**:

* Deployment slots are live environments for testing changes before production deployment.
* **Benefits**:
  + Continuous deployment with zero downtime.
  + Separate hostnames for each slot.
  + Automatic rollback options with auto-swap feature.

**Adding Deployment Slots**:

* Options to clone configurations from other slots.
* Consider sticky settings like custom domains, SSL certificates, and scaling configurations.
* Validate settings before swapping deployments to production.

**6. Securing App Services**

**Authentication and Authorization**:

* Enable built-in authentication with Azure AD or third-party providers.
* Default access is anonymous unless configured otherwise.

**Security Features**:

* Use **Diagnostic Logs** for troubleshooting failed requests.
* Add **SSL certificates** to enforce HTTPS.
* Implement **access control** through priority-based allow/deny lists.
* Store sensitive information in **Azure Key Vault** for enhanced security.

**7. Custom Domain Configuration**

**Steps**:

* Redirect from the default URL to a custom domain.
* Validate ownership through Azure.
* Update DNS registry (CNAME or A record) with your domain provider.
* Ensure the App Service Plan supports custom domains.

**8. Backup and Restore**

**Backup Features**:

* Manual or scheduled backups of app configuration, content, and connected databases.
* Available in **Standard** or **Premium** plans.
* Backup size can go up to 10 GB, with options for partial or full backups.
* Restore the app on-demand or create new instances from backup data.

**9. Demonstration**

**Creating and Testing App Services**:

* Step-by-step guide for creating a web app in the Azure Portal.
* Testing the app, adding deployment slots, and setting up backups.

**10. Summary and Resources**

**Key Takeaways**:

* Proper configuration of App Service Plans and scaling is critical for efficient resource management.
* Deployment slots enhance the continuous deployment pipeline.
* Secure apps using built-in Azure services and best practices.

**Further Learning**:

* **Microsoft Learn Modules**: Interactive guides and exercises for hands-on practice.
* **Documentation**: Azure resources for configuring and optimizing App Services.