

4.2. Student Handout

Student Handout: Introduction to Microsoft Azure and AKS

Overview

This handout provides a concise summary of the key concepts covered in the session on Microsoft Azure and Azure Kubernetes Service (AKS). It includes examples to illustrate each topic.

Microsoft Azure

What is Microsoft Azure?

- **Definition:** Azure is a cloud computing platform by Microsoft that provides a range of services including computing, analytics, storage, and networking.
- **Purpose:** Allows businesses to rent computing resources over the internet instead of maintaining physical hardware.

Why Use Azure?

1. **Scalability:** Easily scale resources up or down.
 - Example: An e-commerce site can handle increased traffic during sales.
 - Example: A startup can scale its infrastructure as it grows.
 - Example: A seasonal business can adjust resources based on demand.
2. **Cost-Effective:** Pay only for what you use.
 - Example: Avoid upfront costs for hardware.
 - Example: Reduce costs by shutting down unused resources.
 - Example: Use reserved instances for predictable workloads.
3. **Global Reach:** Deploy applications closer to users.
 - Example: A global app can reduce latency by using regional data centers.

- Example: A multinational company can comply with data residency requirements.
- Example: A media company can deliver content faster to international audiences.

4. **Security:** Built-in tools for data and application security.

- Example: Use Azure Security Center for threat protection.
 - Example: Implement multi-factor authentication with Azure Active Directory.
 - Example: Encrypt data at rest and in transit.
-

Kubernetes

What is Kubernetes?

- **Definition:** An open-source platform for automating deployment, scaling, and management of containerized applications.
- **Functionality:** Manages containers, ensuring consistent application performance across environments.

Why Use AKS for Kubernetes Orchestration?

1. **Managed Service:** Reduces operational overhead.

- Example: Azure manages the Kubernetes control plane.
- Example: Automatic updates and patching of the Kubernetes environment.
- Example: Simplified cluster management through Azure Portal.

2. **Integration with Azure Services:** Seamless integration with other Azure tools.

- Example: Use Azure Monitor for cluster insights.
- Example: Integrate with Azure DevOps for CI/CD pipelines.
- Example: Secure access with Azure Active Directory.

3. **Cost-Effective:** Only pay for the resources used.

- Example: AKS service is free; pay for VMs and storage.
 - Example: Optimize costs with auto-scaling.
 - Example: Use spot instances for non-critical workloads.
-

Setting Up an Azure Subscription and AKS Prerequisites

Steps to Set Up an Azure Subscription:

1. **Sign Up for Azure:** Create a free account on the Azure website.
 - Example: Receive free credits to explore Azure services.
 - Example: Access to a wide range of Azure products.
 - Example: Start with a pay-as-you-go model.
 2. **Install Azure CLI:** Tool for interacting with Azure from the terminal.
 - Example: Use CLI commands to manage resources.
 - Example: Automate tasks with scripts.
 - Example: Access Azure services programmatically.
 3. **Set Up Resource Groups:** Organize related resources.
 - Example: Group resources by project or department.
 - Example: Simplify resource management and billing.
 - Example: Apply policies and permissions at the group level.
-

Creating and Configuring an AKS Cluster

Steps to Create an AKS Cluster:

1. **Using Azure Portal:**
 - Example: Create a cluster with a few clicks.
 - Example: Customize node size and count.
 - Example: Enable monitoring and logging during setup.
 2. **Using Azure CLI:**
 - Example: Create a resource group with `az group create`.
 - Example: Deploy a cluster with `az aks create`.
 - Example: Generate SSH keys for secure access.
-

Managing Role-Based Access Control (RBAC) on Azure

Key Concepts of RBAC:

1. **Roles:** Define permissions for users.
 - Example: "Owner" role for full access.
 - Example: "Contributor" role for resource management.
 - Example: "Reader" role for view-only access.
 2. **Role Assignments:** Assign roles to users or groups.
 - Example: Assign "Reader" role to a team member for monitoring.
 - Example: Grant "Contributor" role to a developer for deployment.
 - Example: Use custom roles for specific permissions.
-

Deploying Applications to AKS

Steps to Deploy an Application:

1. **Create a Deployment YAML File:** Define application settings.
 - Example: Specify container image and replicas.
 - Example: Set environment variables for the application.
 - Example: Configure resource limits and requests.
 2. **Apply the Deployment:** Deploy using `kubectl apply`.
 - Example: Deploy a web application with a single command.
 - Example: Update deployments with new configurations.
 - Example: Rollback to previous versions if needed.
 3. **Expose the Application:** Make it accessible via a service.
 - Example: Use a LoadBalancer service for external access.
 - Example: Create a ClusterIP service for internal communication.
 - Example: Implement NodePort for direct node access.
-

Implementing Ingress Controllers and Load Balancing

Steps to Implement an Ingress Controller:

1. Install the NGINX Ingress Controller:

- Example: Deploy using a single `kubectl apply` command.
- Example: Manage traffic routing with ingress rules.
- Example: Support for SSL termination and URL-based routing.

2. Create an Ingress Resource: Define routing rules.

- Example: Route traffic to different services based on URL paths.
 - Example: Use host-based routing for multiple domains.
 - Example: Configure TLS for secure connections.
-

Using Persistent Volumes in AKS

Steps to Use Persistent Volumes:

1. Create a Persistent Volume Claim (PVC): Request storage.

- Example: Define storage size and access mode.
- Example: Use Azure Disks for block storage.
- Example: Use Azure Files for shared file storage.

2. Mount the PVC to a Pod: Specify in the pod's YAML file.

- Example: Persist data for a database application.
 - Example: Share data between multiple pods.
 - Example: Backup and restore data using persistent volumes.
-

Monitoring and Scaling AKS Workloads

Steps to Monitor and Scale Workloads:

1. Enable Monitoring: Use Azure Monitor for insights.

- Example: Track CPU and memory usage of pods.
- Example: Set up alerts for resource thresholds.
- Example: Visualize metrics with dashboards.

2. **Auto-Scaling:** Adjust pod count based on usage.

- Example: Use Horizontal Pod Autoscaler for scaling pods.
- Example: Implement Vertical Pod Autoscaler for resource adjustments.
- Example: Scale based on custom metrics with Prometheus.

Conclusion

This handout provides a summary of the key concepts and steps involved in using Microsoft Azure and AKS. By understanding these fundamentals, you can effectively deploy and manage applications in the cloud. Keep practicing to enhance your skills in cloud computing and Kubernetes orchestration.