Module 7

Implementing containers in Azure

Module Overview

- Implementing Windows and Linux containers in Azure
- Implementing Azure Container Service

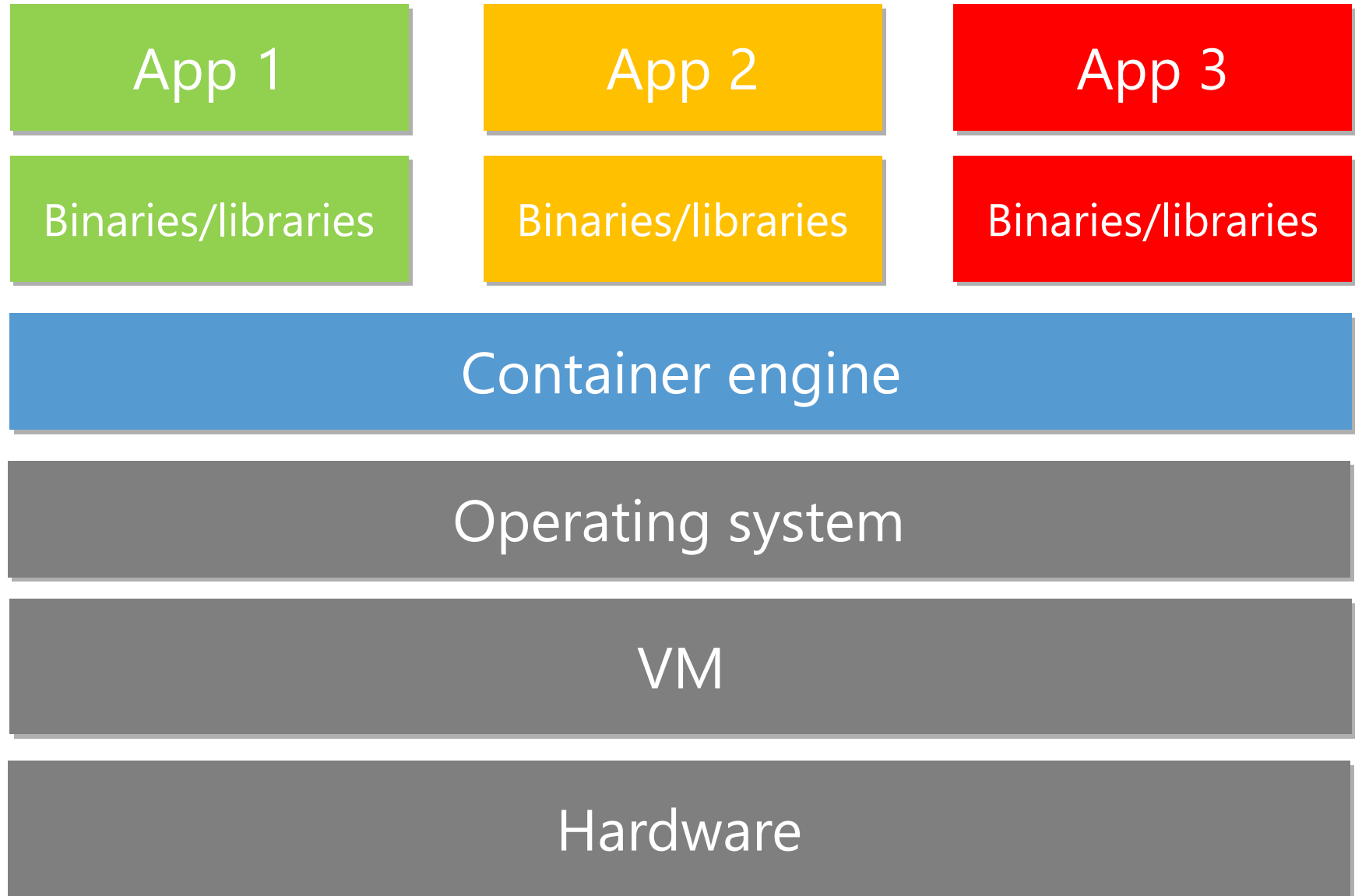# Lesson 1: Implementing Windows and Linux containers in Azure

- Demonstration: Preparing the lab environment
- Introduction to containers
- Introduction to Docker
- Implementing Docker hosts in Azure
- Deploying containers on Azure VMs
- Demonstration: Installing a Docker host and containers on an Azure VM
- Creating multicontainer applications with Docker Compose
- Implementing Azure Container Registry

Demonstration: Preparing the lab environment

In this demonstration, you will learn how to prepare the lab environment

Note: To prepare the lab environment for this module, you must complete this task

Introduction to containers

App 1	App 2	App 3
Binaries/libraries	Binaries/libraries	Binaries/libraries

Container engine

Operating system

VM

Hardware

Introduction to Docker

- Docker terminology:
  - Image
  - Container
  - Dockerfile
- Docker toolbox
  - Docker client
  - Docker Engine
  - Docker Compose
  - Docker Machine
  - Docker Registry
  - Kitematic

Implementing Docker hosts in Azure

- Use the Custom Script VM Extension or cloud-init:
  - Implement with Azure Resource Manager template, Azure PowerShell, or Azure CLI
  - Intended for Linux Azure VMs
- Provision a Docker Azure VM from Azure Marketplace:
  - Intended for Windows or Linux Azure VMs
- Run the Docker Machine Azure driver:
  - Download from docker.com (Windows, Linux, or Mac OS X)
  - Run **docker-machine create --driver azure**
  - Use the *--azure image* parameter to specify the intended image
  - Intended for Linux Azure VMs
- Use OneGet provider:
  - Intended for Windows Azure VMs
- Deploy an AKS cluster:
  - Intended for clusters of Linux container hosts

Deploying containers on Azure VMs

- Connect to an Azure VM Docker host:
  - Docker Machine
  - RDP
  - SSH
- Use the Docker client to:
  - Create containers
  - Stop containers
  - Remove containers
  - Create images
  - Browse for images

Demonstration: Installing a Docker host and containers on an Azure VM

In this demonstration, you will learn how to install a Docker host and containers on an Azure VM

Creating multicontainer applications with Docker Compose

- Install Docker Compose:
  - Included by default when using:
    - Azure Marketplace Docker images
    - The Azure VM Docker extension
    - Docker Machine
- Create **docker-compose.yml**:
  - Include all images
  - Specify container dependencies
  - Specify deployment parameters
- Run **docker-compose up**

Implementing Azure Container Registry

- Create the Azure Container Registry service:
  - SKU: Basic, Standard, and Premium
  - Authentication: Azure AD and Admin user
- Use the Azure Container Registry service:
  - **docker login**
  - **docker pull**
  - **docker tag**
  - **docker push**
  - **docker pull** or **docker run**
  - **docker rmi**

Lab A: Implementing containers on Azure VMs

- Exercise 1: Implementing Docker hosts on Azure VMs

- Exercise 2: Deploying containers to Azure VMs

- Exercise 3: Deploying multicontainer applications with Docker Compose to Azure VMs

- Exercise 4: Implementing Azure Container Registry

Logon Information
Virtual machine: **20533E-MIA-CL1**
User name: **Admin**
Password: **Pa55w.rd**

Estimated Time: 30 minutes

Lab Scenario

Adatum Corporation plans to implement some of its applications as Docker containers on Azure VMs. To optimize this implementation, you intend to combine multiple containers by using Docker Compose. Adatum would also like to deploy its own private Docker registry in Azure to store containerized images. Your task is to test the functionality of tools that facilitate deployment of Docker hosts and Docker containers. You also need to evaluate Azure Container Registry.

Lab Review

- Which method would you use when deploying Docker hosts on Azure VMs?

- What authentication and authorization method do you intend to use when implementing Azure Container Registry?

Lesson 2: Implementing Azure Container Service

- Overview of container-clustering solutions in Azure
- Creating and managing an ACS Docker Swarm cluster
- Creating and managing an ACS Kubernetes cluster
- Creating and managing an ACS DC/OS cluster
- Creating and managing an AKS cluster
- Demonstration: Creating an AKS cluster

Overview of container-clustering solutions in Azure
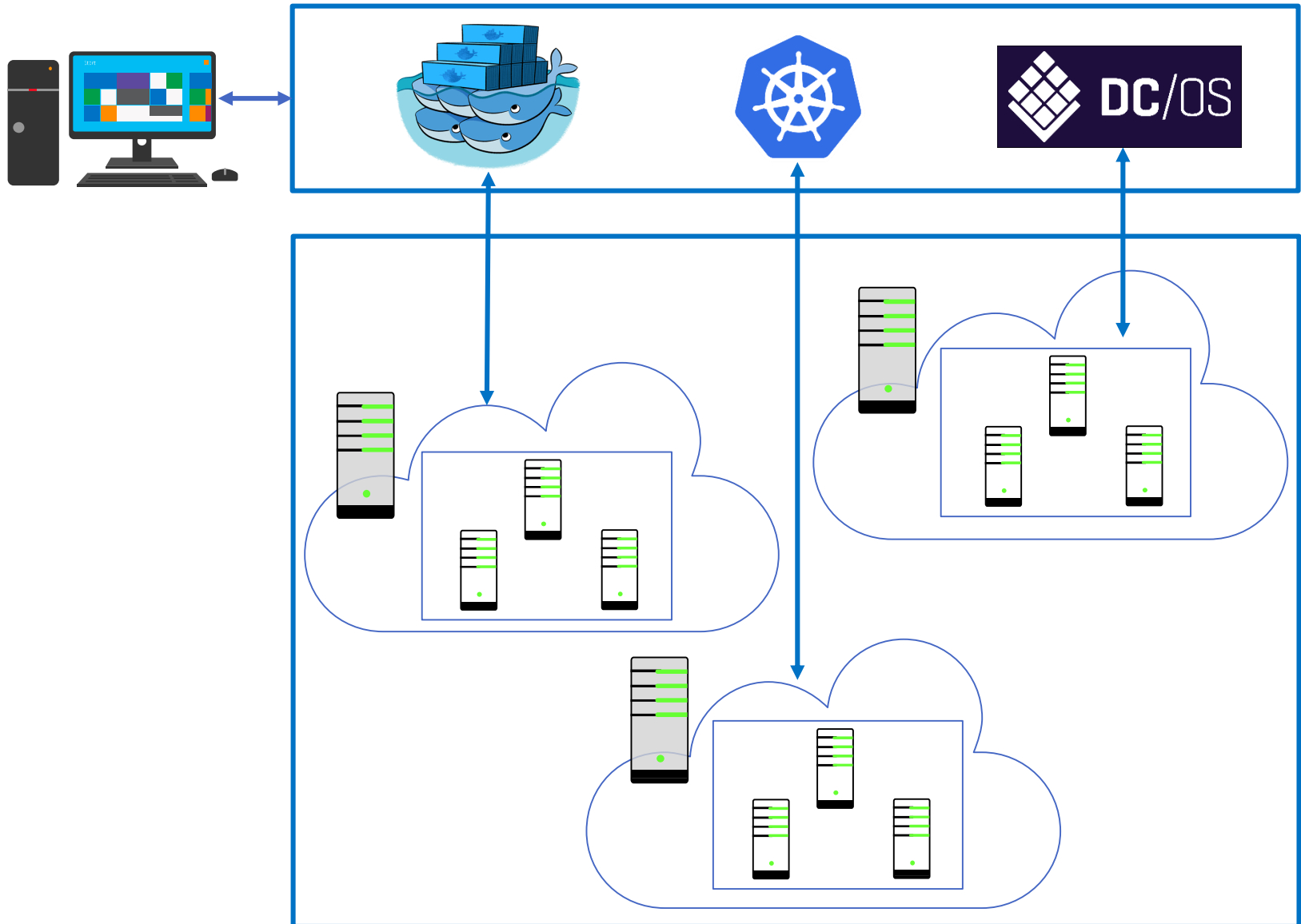
Creating and managing an ACS Docker Swarm cluster

1. Create a Docker Swarm cluster in ACS:
    - Requires an SSH RSA key
    - Specify the master and agent counts
    - Specify the agent VM size

2. Connect to the Docker Swarm cluster:
    - Establish an SSH tunnel to the master FQDN
    - Set **DOCKER_HOST** to :2375

3. Deploy containers to the Docker Swarm cluster:
    - Use **docker run** for individual containers
    - Use **docker-compose** for multicontainer applications

Creating and managing an ACS Kubernetes cluster

1. ## Create a Kubernetes cluster in ACS:
   - Requires an SSH RSA key
   - Requires an Azure AD service principal with the Contributor role to cluster resources
   - Specify the number of master and agent nodes
   - Specify the agent VM size

2. ## Connect to the Kubernetes cluster:
   - Install kubectl
   - Authenticate via SSH and download Kubernetes credentials

3. ## Deploy containers to the Kubernetes cluster:
   - Create a YAML-formatted manifest file
   - Run **kubectl create** and reference the manifest file name

Creating and managing an ACS DC/OS cluster

1. Create a DC/OS cluster in ACS:
   - Requires an SSH RSA key
   - Specify the number of master and agent nodes
   - Specify the agent VM size

2. Connect to the DC/OS cluster:
   - Establish an SSH tunnel
   - Connect to the DC/OS web portal via http://localhost
   - Install DC/OS CLI

3. Deploy containers to the DC/OS cluster:
   - Create a YAML-formatted Marathon configuration file
   - Run **dcos marathon app add** and reference the file name

Creating and managing an AKS cluster

1. ## Create an AKS cluster:

   - Requires an SSH RSA key

   - Requires an Azure AD service principal

   - Specify the number of agent nodes

   - Specify the agent VM size

2. ## Connect to the Kubernetes cluster:

   - Install kubectl

   - Authenticate via SSH and download Kubernetes credentials

3. ## Deploy containers to the Kubernetes cluster:

   - Create a YAML-formatted manifest file

   - Run **kubectl create** and reference the manifest file name

Demonstration: Creating an AKS cluster

In this demonstration, you will see how to implement an AKS cluster

Lab B: Implementing Azure Container Service (AKS)

- Exercise 1: Creating an AKS cluster
- Exercise 2: Managing an AKS cluster

Logon Information
Virtual machine: **20533E-MIA-CL1**
User name: **Admin**
Password: **Pa55w.rd**

Estimated Time: 30 minutes

Lab Scenario

Adatum is considering implementing containers on a larger scale by leveraging the capabilities that AKS offers. You want to test load balancing and scaling of a sample containerized application.

Lab Review

- What deployment methodology would you choose when deploying AKS clusters?
- What are the primary advantages of using AKS for deploying container clusters?

Module Review and Takeaways

- Review Question