Azure Fundamentals Workshop

Workshop Overview

Duration: 4-6 hours
Level: Beginner to Intermediate

Module 1: Azure Role-Based Access Control (RBAC)

What is RBAC?

RBAC is Azure's authorization system that helps you manage who has access to Azure resources, what they can do with those resources, and what areas they have access to.

Core Principles

- **Least Privilege**: Grant only the minimum permissions needed
- **Separation of Duties**: Different roles for different responsibilities
- **Defense in Depth**: Multiple layers of security

RBAC Components

1. Security Principal (Who)

- **Users**: Individual people with Azure AD accounts
- **Groups**: Collections of users
- **Service Principals**: Applications or services
- **Managed Identities**: Automatically managed identities for Azure services

2. Role Definition (What)

Collections of permissions defining allowed actions.

Built-in Roles (Common Examples):

- **Owner**: Full access including ability to delegate access
- **Contributor**: Can create and manage resources but cannot grant access
- **Reader**: View-only access
- **User Access Administrator**: Manage user access without managing resources

3. Scope (Where)

The boundary where access applies:

```
Management Group
  └── Subscription
    └── Resource Group
      └── Resource
```

RBAC Hierarchy

Permissions inherit downward:

- Assignment at Management Group → applies to all subscriptions below
- Assignment at Subscription → applies to all resource groups below
- Assignment at Resource Group → applies to all resources within
- Assignment at Resource → applies only to that resource

Hands-On Exercise 1

Scenario: Grant a developer read-only access to a specific resource group.

Steps:

1. Navigate to the Resource Group in Azure Portal
2. Click "Access control (IAM)"
3. Click "+ Add" → "Add role assignment"
4. Select "Reader" role
5. Select the user
6. Review and assign

Module 2: Microsoft Entra ID Fundamentals

What is Microsoft Entra ID?

Microsoft Entra ID (formerly Azure AD) is Microsoft's cloud-based identity and access management service.

Core Components

Users

- **Cloud-only identities**: Created directly in Entra ID

- **Synchronized identities**: Synced from on-premises AD
- **Guest users**: External users invited to collaborate

User Attributes:

- Display name, User Principal Name (UPN)
- Job title, department, location
- Contact information
- License assignments

Groups

Types:

- **Security Groups**: Manage access to resources
- **Microsoft 365 Groups**: Collaboration with shared mailbox, calendar, files

Membership Types:

- **Assigned**: Manually add members
- **Dynamic User**: Auto-populate based on user attributes
- **Dynamic Device**: Auto-populate based on device attributes

Example Dynamic Rule:

```
(user.department -eq "Engineering") -and (user.country -eq "USA")
```

Device Management

Join Types:

- **Entra ID Registered**: Personal devices (BYOD)
- **Entra ID Joined**: Corporate-owned, cloud-only
- **Hybrid Entra ID Joined**: Corporate-owned, joined to both on-prem AD and Entra ID

Benefits:

- Single sign-on to cloud resources
- Conditional access policies
- Device compliance enforcement

Guest Access

Invite external users to collaborate on resources.

Process:

1. Invite guest user via email
2. Guest accepts invitation
3. Guest account created in your directory
4. Assign appropriate permissions

Best Practices:

- Regular access reviews
- Set expiration dates
- Limit permissions appropriately

Self-Service Password Reset (SSPR)

Requirements:

- Entra ID Premium P1 or P2 license
- Users must register authentication methods

Authentication Methods (require at least 2):

- Mobile app notification
- Mobile app code
- Email
- Mobile phone (SMS)
- Office phone
- Security questions

Configuration Steps:

1. Navigate to Entra ID → Password reset
2. Enable SSPR for selected or all users
3. Configure authentication methods
4. Set registration requirements
5. Configure notifications

Hands-On Exercise 2

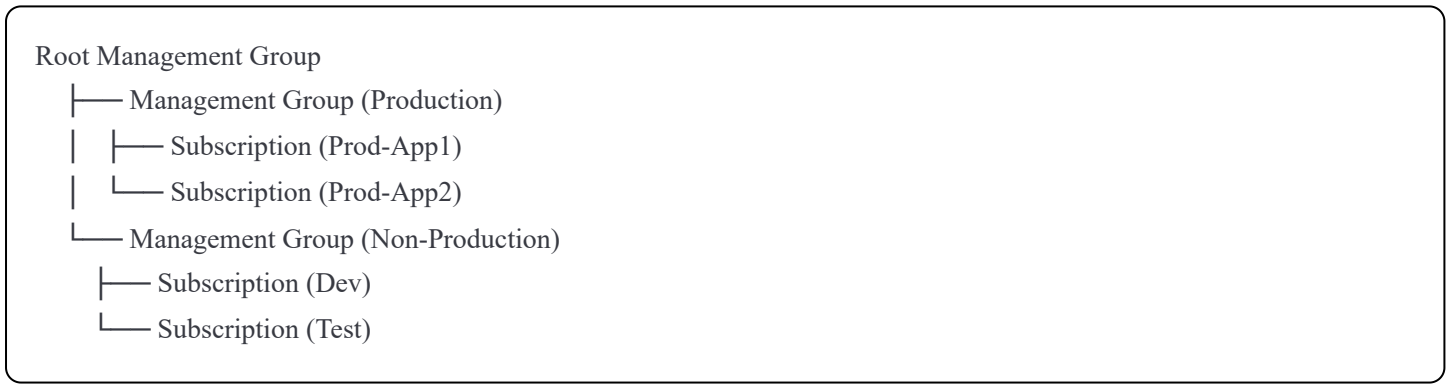
Task: Create a security group and add three users.

Steps:

1. Navigate to Entra ID → Groups
2. Click "+ New group"
3. Select "Security" as group type
4. Name the group "Workshop-Team"
5. Add members
6. Create the group

Module 3: Azure Governance

Organizational Hierarchy



Management Groups

Containers for managing access, policies, and compliance across multiple subscriptions.

Key Features:

- Up to 6 levels of depth
- 10,000 management groups per directory
- Inherit policies and RBAC downward

Use Cases:

- Separate production from non-production
- Organizational structure (by department, geography)
- Apply governance at scale

Subscriptions

Logical containers for resources and billing.

Functions:

- Billing boundary
- Access control boundary
- Resource organization

Common Patterns:

- One subscription per environment (Dev, Test, Prod)
- One subscription per department
- One subscription per project

Resource Groups

Containers for resources that share the same lifecycle.

Best Practices:

- Group resources by lifecycle (deploy, update, delete together)
- One resource can exist in only one resource group
- Resources can communicate across resource groups
- Can contain resources from multiple regions

Naming Convention Example:

```
rg-{workload}-{environment}-{region}-{instance}
rg-webapp-prod-eastus-001
```

Hands-On Exercise 3

Task: Create a management group structure.

Steps:

1. Navigate to Management Groups
2. Create "Production" management group
3. Create "Non-Production" management group
4. Move appropriate subscriptions into each group

Module 4: Policy Enforcement

Azure Policy

Service to create, assign, and manage policies that enforce rules over resources.

Key Concepts:

- **Policy Definition**: The rule (JSON format)
- **Policy Assignment**: Applying the policy to a scope
- **Compliance**: Resources evaluated against policies

Built-in Policy Examples

- Allowed locations for resources
- Allowed virtual machine SKUs
- Require tag and its value
- Audit VMs that do not use managed disks

Custom Policy Structure

```json
```

```
{
 "properties": {
  "displayName": "Require tag on resources",
  "policyType": "Custom",
  "mode": "Indexed",
  "parameters": {
   "tagName": {
    "type": "String",
    "metadata": {
     "displayName": "Tag Name"
    }
   }
  },
  "policyRule": {
   "if": {
    "field": "[concat('tags[', parameters('tagName'), ']')]",
    "exists": "false"
   },
   "then": {
    "effect": "deny"
   }
  }
 }
}
```

Policy Effects

- **Deny**: Block non-compliant resource creation/update
- **Audit**: Log non-compliant resources
- **Append**: Add fields to resources
- **Modify**: Add, update, or remove tags
- **DeployIfNotExists**: Deploy resources if they don't exist
- **AuditIfNotExists**: Audit if related resources don't exist

Resource Locks

Prevent accidental deletion or modification of critical resources.

Lock Types:

- **CanNotDelete**: Users can read and modify but not delete
- **ReadOnly**: Users can only read, no modifications or deletions

Lock Hierarchy:

- Locks apply to all resources within the scope
- Child resources inherit locks from parents
- Most restrictive lock applies

Example Scenarios:

- Lock production subscription with CanNotDelete
- Lock critical resource groups with ReadOnly during change freeze

Resource Tags

Key-value pairs for organizing and managing resources.

Common Tag Examples:

- **Environment**: Production, Development, Test
- **CostCenter**: Finance, Engineering, Marketing
- **Owner**: email@company.com
- **Project**: ProjectX, ProjectY
- **ApplicationName**: WebApp, Database

Tag Management:

- Maximum 50 tags per resource
- Tag name: 512 characters (128 for storage accounts)
- Tag value: 256 characters
- Tags are not inherited by child resources

Enforcing Tags with Policy:

```
Policy: "Require a tag and its value on resources"
Effect: Deny or Append
```

Resource Movement

Moving resources between resource groups or subscriptions.

Considerations:

- Not all resource types can be moved

- Source and destination subscriptions must be in same tenant
- Validate move operation before executing
- Resources are locked during move (read-only)

Move Process:

1. Verify resource type supports move
2. Check dependencies
3. Validate the move
4. Execute the move
5. Update any automation or configurations

Hands-On Exercise 4

Task: Apply a policy to enforce tagging.

Steps:

1. Navigate to Azure Policy
2. Select "Definitions"
3. Search for "Require a tag on resources"
4. Click "Assign"
5. Select scope (resource group)
6. Specify tag name: "Environment"
7. Set effect to "Deny"
8. Assign the policy
9. Test by trying to create a resource without the tag

Module 5: Cost Governance

Azure Cost Management Overview

Tools and features to monitor, allocate, and optimize Azure spending.

Key Features

1. Cost Analysis

Visualize and analyze spending patterns.

Views Available:

- Accumulated costs over time
- Daily costs
- Costs by service
- Costs by resource group
- Costs by location

Filtering Options:

- Time range (daily, monthly, custom)
- Resource group
- Service name
- Location
- Tags

2. Budgets

Set spending limits and get alerted when thresholds are reached.

Configuration:

- Set budget amount (monthly, quarterly, annually)
- Define alert conditions (50%, 75%, 90%, 100%)
- Configure notification emails
- Integrate with action groups for automation

Example Budget:

```
Name: Production-Monthly-Budget
Amount: $10,000
Period: Monthly
Alerts:
  - 80% threshold → Email team
  - 90% threshold → Email management
  - 100% threshold → Trigger automation to scale down
```

3. Cost Allocation

Use tags to track spending by business units.

Strategy:

- Tag all resources with cost center
- Create cost views filtered by tags
- Export data for chargeback/showback
- Regular reporting to stakeholders

4. Recommendations

Azure Advisor provides cost optimization suggestions:

- Right-size underutilized VMs
- Shut down idle resources
- Purchase reserved instances
- Use Azure Hybrid Benefit

Cost Optimization Best Practices

Resource Optimization

- **Auto-shutdown**: Schedule VMs to turn off during non-business hours
- **Right-sizing**: Match VM sizes to actual usage
- **Spot VMs**: Use for fault-tolerant workloads (up to 90% discount)
- **Reserved Instances**: Commit to 1 or 3 years for 40-60% savings

Monitoring and Alerts

- Set up budgets for each subscription
- Configure cost alerts
- Regular review of cost analysis reports
- Track spending trends

Governance

- Use policies to prevent expensive SKUs
- Require tags for cost allocation
- Implement approval workflows for expensive resources
- Regular access reviews to remove unused permissions

Hands-On Exercise 5

Task: Create a budget with alerts.

Steps:

1. Navigate to Cost Management + Billing
2. Select "Budgets"
3. Click "+ Add"
4. Enter budget details:
    - Name: "Workshop-Budget"
    - Amount: $500
    - Reset period: Monthly
5. Set alert conditions:
    - 80% actual
    - 100% forecasted
6. Add email addresses for notifications
7. Create the budget

Workshop Summary

Key Takeaways

RBAC:

- Use built-in roles when possible
- Apply least privilege principle
- Understand scope inheritance

Entra ID:

- Central identity management for Azure
- Groups simplify access management
- SSPR reduces helpdesk burden

Governance:

- Management groups for large-scale organization

- Resource groups for lifecycle management
- Clear naming conventions

Policy Enforcement:

- Azure Policy for compliance at scale
- Resource locks for critical resources
- Tags for organization and cost allocation

Cost Management:

- Regular monitoring prevents surprises
- Budgets and alerts for proactive management
- Optimization recommendations save money

Additional Resources

- Azure RBAC Documentation
- Microsoft Entra ID Documentation
- Azure Governance Documentation
- Azure Policy Documentation
- Cost Management Documentation

Next Steps

1. Practice exercises in your own Azure subscription
2. Explore Azure Advisor recommendations
3. Review Microsoft Learn modules for deeper dives
4. Implement governance framework in your organization
5. Consider Azure certifications (AZ-104, AZ-305)