Production Server Deployment and Configuration

Purpose

This SOP outlines the standard procedure for deploying and configuring new production servers in the Azure cloud environment. It ensures consistency, security compliance, and proper integration with monitoring systems.

Scope

Applies to all new Linux (Ubuntu 22.04) and Windows Server 2022 instances deployed to production resource groups.

Prerequisites

- Azure subscription with appropriate RBAC permissions

- Access to company VPN or jump host

- Approved change request ticket

- Network security group rules pre-approved by security team

Step 1: Create Azure Virtual Machine

Navigate to Azure Portal and create a new Virtual Machine with the following specifications:

For Linux Servers:

- Resource Group: PROD-RG-{region}

- VM Name: {application}-{environment}-{number}

- Region: Select based on application requirements

- Availability Zone: Distribute across zones for HA

- Image: Ubuntu Server 22.04 LTS

- Size: Standard_D2s_v3 (adjust based on workload)

- Authentication: SSH public key (no password authentication)

For Windows Servers:

- Image: Windows Server 2022 Datacenter

- Authentication: Local admin account with complex password

- Enable Azure AD Join if required

Configure networking:

- Virtual Network: PROD-VNET-{region}

- Subnet: PROD-SUBNET-{tier} (web, app, or data)

- Public IP: None (use bastion or VPN for access)

- NSG: Apply PROD-NSG-{tier} template

Enable managed identity for Azure resource access.

Step 2: Install Base Packages and Updates

Linux Servers

SSH into the server and execute:

`bash

sudo apt update && sudo apt upgrade -y

sudo apt install -y azure-cli docker.io docker-compose-v2 fail2ban ufw

sudo apt install -y monitoring-agents log-forwarder

`

Configure automatic security updates:

`bash

sudo dpkg-reconfigure -plow unattended-upgrades

`

Windows Servers

Connect via RDP and run PowerShell as Administrator:

`powershell

Install-WindowsFeature -Name Web-Server, NET-Framework-45-Features

Install-PackageProvider -Name NuGet -Force

Install-Module -Name Az.Compute -Force

`

Run Windows Update and install all critical updates.

Step 3: Configure Security Hardening

Linux Security

1. Configure SSH:

- Disable root login

- Change default port to 2222

- Enable key-based authentication only

- Set LoginGraceTime to 60 seconds

2. Configure firewall (UFW):

- Allow SSH on port 2222 from VPN subnet only

- Allow application-specific ports

- Enable UFW with default deny policy

3. Configure fail2ban:

- Enable sshd jail

- Set maxretry to 3

- Set bantime to 3600 seconds

Windows Security

1. Configure Windows Firewall:

- Enable firewall for all profiles

- Create inbound rules for required ports only

- Block all other inbound traffic

2. Configure Local Security Policy:

- Set password complexity requirements

- Configure account lockout policy (3 attempts, 30 min lockout)

- Enable audit policies for security events

Step 4: Deploy Application

Container Deployment (Linux)

Pull and run the application container:

`bash

docker pull {acr-name}.azurecr.io/{app-name}:{version}

docker run -d --name {app-name} --restart unless-stopped \

-p {app-port}:{app-port} \

-e ENVIRONMENT=production \

-v /var/log/{app-name}:/var/log/app \

{acr-name}.azurecr.io/{app-name}:{version}

`

IIS Deployment (Windows)

1. Create application pool with .NET CLR version

2. Deploy application files to C:\inetpub\wwwroot\{app-name}

3. Configure IIS site bindings (HTTP/HTTPS)

4. Set application pool identity to managed service account

Step 5: Configure Monitoring and Logging

Install and configure monitoring agent:

1. Install Azure Monitor Agent

2. Configure data collection rules:

- Performance counters (CPU, memory, disk, network)

- Application event logs

- Custom application metrics

3. Set up alert rules:

- CPU > 80% for 5 minutes

- Memory > 85% for 5 minutes

- Disk space < 10%

- Application error rate > 1%

Configure log forwarding to Log Analytics workspace:

- Workspace: PROD-LOG-{region}

- Retention: 90 days

Step 6: Validate Deployment

Run validation checklist:

1. **Connectivity Tests**

- Ping from monitoring server

- Telnet to application port

- HTTP/HTTPS health check endpoint

2. **Security Validation**

- Run security baseline scan

- Verify no public IP assigned

- Confirm NSG rules applied correctly

- Test fail2ban/firewall rules

3. **Performance Baseline**

- Record initial CPU/memory usage

- Verify disk I/O performance

- Test network throughput

4. **Application Validation**

- Login functionality works

- Database connectivity confirmed

- External API integrations working

- Logging to central system confirmed

Step 7: Document and Handover

1. Update CMDB with new server details:

- Hostname, IP address, OS version

- Application name and version

- Owner and support team

- Backup schedule and retention

2. Create runbook entry:

- Startup/shutdown procedures

- Common troubleshooting steps

- Contact information for escalation

3. Schedule post-deployment review:

- 24-hour check (monitoring alerts)

- 7-day review (performance trends)

- 30-day assessment (stability confirmation)

Decision Points

If VM creation fails:

- Check Azure resource quotas

- Verify region availability

- Review subscription limits

- Contact Azure support if needed

If security hardening breaks application:

- Review application requirements

- Temporarily relax specific rules

- Document exception in security register

- Schedule security review within 48 hours

If monitoring agent fails to install:

- Check network connectivity to Azure endpoints

- Verify managed identity permissions

- Review proxy configuration

- Manual agent installation as fallback

If validation fails:

- Do not proceed to handover

- Create incident ticket

- Rollback to previous state if needed

- Schedule root cause analysis

Troubleshooting

SSH connection timeout:

- Verify NSG allows port 2222 from VPN subnet

- Check SSH service is running: `systemctl status ssh`

- Review /var/log/auth.log for connection attempts

Application not responding:

- Check container status: `docker ps`

- Review application logs in /var/log/{app-name}

- Verify database connectivity

- Check resource utilization (CPU/memory)

High CPU usage:

- Identify process: `top` or `htop`

- Check for runaway processes

- Review application logs for errors

- Consider scaling up VM size

Disk space critical:

- Check log rotation configuration

- Clean old logs: `find /var/log -name "*.log" -mtime +30 -delete`

- Expand disk in Azure Portal

- Move data to separate data disk

References

- Azure Well-Architected Framework

- CIS Benchmarks for Ubuntu/Windows

- Company Security Policy v2.3

- Change Management Procedure CM-001

- Incident Response Plan IR-2024

Revision History

- v1.0 (2024-01-15): Initial release

- v1.1 (2024-03-20): Added Windows Server procedures

- v1.2 (2024-06-10): Updated security hardening steps

Visual Diagram (mermaid)

Visual diagram available as SVG: sop-20260716052940-mermaid.svg