SOP-AD-003: Azure VM Domain Join and Unjoin

Standard Operating Procedure: Azure VM Domain Join and Unjoin

Document ID:	SOP-AD-003
Title:	Azure VM Domain Join and Unjoin Operations
Category:	Active Directory
Version:	1.0
Status:	Draft
Author:	OberaConnect
Creation Date:	2026-01-12
Approval Date:	Pending

1.0 Purpose

This procedure documents the standard process for joining Windows Virtual Machines to an Active Directory domain or removing (unjoining) them from a domain within Azure tenant environments. This ensures consistent, secure, and documented domain membership changes across managed infrastructure.

2.0 Scope

This SOP applies to: - Windows Server VMs hosted in Azure (IaaS) - Domain join operations to on-premises or Azure-hosted Active Directory - Domain unjoin operations for decommissioning, migration, or troubleshooting - OberaConnect technicians and system administrators

3.0 Definitions

Term	Definition
AD	Active Directory - Microsoft directory service for Windows domain networks
Domain Controller (DC)	Server that responds to authentication requests within a Windows domain
FQDN	Fully Qualified Domain Name (e.g., server.domain.com)
NetBIOS	Legacy naming convention for Windows networks
SID	Security Identifier - unique identifier for security principals
Workgroup	Peer-to-peer network configuration (non-domain)
Azure Run Command	Azure feature to execute scripts on VMs without RDP

4.0 Roles & Responsibilities

Role	Responsibility
Technician	Execute domain join/unjoin operations, document changes
Domain Administrator	Provide domain credentials, approve membership changes
Azure Administrator	Ensure VM accessibility and proper networking
Change Manager	Approve changes in production environments

5.0 Prerequisites

5.1 For Domain Join

- ☐ Azure VM is running and accessible
- ☐ VM has network connectivity to Domain Controller(s)
- ☐ DNS configured to resolve domain (DC IP as primary DNS)
- ☐ Required ports open (see Section 5.3)
- ☐ Domain Administrator credentials available
- ☐ Computer account pre-staged in AD (optional but recommended)
- ☐ Target OU identified for computer object placement

5.2 For Domain Unjoin

- ☐ Azure VM is running and accessible
- ☐ Local Administrator account credentials available
- ☐ Domain Administrator credentials (for clean removal)
- ☐ VM snapshot taken for rollback capability
- ☐ Users notified of service impact
- ☐ No active user sessions on the server

5.3 Required Network Ports

Port	Protocol	Service
53	TCP/UDP	DNS
88	TCP/UDP	Kerberos
135	TCP	RPC Endpoint Mapper
389	TCP/UDP	LDAP
445	TCP	SMB
464	TCP/UDP	Kerberos Password Change
636	TCP	LDAPS
3268	TCP	Global Catalog
3269	TCP	Global Catalog SSL
49152-65535	TCP	RPC Dynamic Ports

6.0 Procedure

Part A: Domain Join Procedure

6.1 Pre-Join Verification Step 1: Verify VM Status

```
# Azure CLI - Check VM is running
az vm get-instance-view \
  --resource-group <RESOURCE_GROUP> \
  --name <VM_NAME> \
```

```
  --query "instanceView.statuses[?starts_with(code, 'PowerState')].displayStatus" \
  -o tsv
```

Expected output: `VM running`

Step 2: Verify Network Connectivity to Domain Controller

Via Azure Run Command:

```
az vm run-command invoke \
  --resource-group <RESOURCE_GROUP> \
  --name <VM_NAME> \
  --command-id RunPowerShellScript \
  --scripts "Test-NetConnection -ComputerName <DC_IP_OR_FQDN> -Port 389"
```

Or via RDP/PowerShell on VM:

```
# Test LDAP connectivity
Test-NetConnection -ComputerName <DC_IP_OR_FQDN> -Port 389

# Test DNS resolution
Resolve-DnsName <DOMAIN_FQDN>

# Test all required ports
$ports = @(53, 88, 135, 389, 445, 636, 3268)
foreach ($port in $ports) {
    Test-NetConnection -ComputerName <DC_IP> -Port $port
}
```

Step 3: Configure DNS Settings

Ensure VM DNS points to Domain Controller:

```
# View current DNS settings
Get-DnsClientServerAddress -InterfaceAlias "Ethernet*"

# Set DNS to Domain Controller (if needed)
Set-DnsClientServerAddress -InterfaceAlias "Ethernet" -ServerAddresses "<DC_IP>","<SECONDARY_DC_IP>"
```

6.2 Execute Domain Join Method 1: PowerShell (Recommended)

Via Azure Run Command:

```
az vm run-command invoke \
  --resource-group <RESOURCE_GROUP> \
  --name <VM_NAME> \
  --command-id RunPowerShellScript \
  --scripts "Add-Computer -DomainName '<DOMAIN_FQDN>' -OUPath 'OU=Servers,DC=domain,DC=com' -Credential
```

Via RDP/PowerShell on VM (interactive):

```
# Join domain with specific OU placement
Add-Computer -DomainName "domain.com" `
  -OUPath "OU=Servers,DC=domain,DC=com" `
  -Credential (Get-Credential) `
  -Restart -Force

# OR simple join (places in default Computers container)
Add-Computer -DomainName "domain.com" -Credential (Get-Credential) -Restart
```

Method 2: System Properties GUI (via RDP)

1. Connect to VM via RDP or Azure Bastion
2. Open **System Properties**: `sysdm.cpl`
3. Click **Computer Name** tab
4. Click **Change...**
5. Select **Domain** radio button
6. Enter domain name (e.g., `domain.com`)
7. Click **OK**
8. Enter Domain Administrator credentials when prompted
9. Click **OK** on success message
10. Restart the server when prompted

6.3 Post-Join Verification Step 1: Verify Domain Membership

```
# Check domain membership
(Get-WmiObject Win32_ComputerSystem) | Select-Object Name, Domain, PartOfDomain

# Expected output:
# Name          Domain         PartOfDomain
# ----          ------         ------------
# SERVERNAME    domain.com     True
```

Step 2: Verify Computer Object in AD

```
# From Domain Controller or machine with RSAT
Get-ADComputer -Identity "<COMPUTER_NAME>" | Select-Object Name, DistinguishedName, Enabled
```

Step 3: Test Domain Authentication

```
# Test user authentication
runas /user:DOMAIN\username "cmd.exe"

# Test group policy application
gpupdate /force
gpresult /r
```

Step 4: Verify DNS Registration

```
# Force DNS registration
ipconfig /registerdns

# Verify A record exists
Resolve-DnsName <COMPUTER_NAME>.<DOMAIN_FQDN>
```

Part B: Domain Unjoin Procedure

6.4 Pre-Unjoin Preparation Step 1: Take VM Snapshot (Critical)

```
# Create snapshot for rollback
az snapshot create \
  --resource-group <RESOURCE_GROUP> \
  --name "<VM_NAME>-PreUnjoin-$(date +%Y%m%d)" \
  --source "/subscriptions/<SUB_ID>/resourceGroups/<RG>/providers/Microsoft.Compute/disks/<OS_DISK_NAME
```

Step 2: Document Current Configuration

```
# Export current configuration
$config = @{
```

```
    ComputerName = $env:COMPUTERNAME
    Domain = (Get-WmiObject Win32_ComputerSystem).Domain
    IPAddress = (Get-NetIPAddress -AddressFamily IPv4 | Where-Object {$_.InterfaceAlias -like "Ethernet
    DNS = (Get-DnsClientServerAddress -InterfaceAlias "Ethernet*" -AddressFamily IPv4).ServerAddresses
    DateTime = Get-Date
}
$config | ConvertTo-Json | Out-File "C:\PreUnjoinConfig.json"
```

Step 3: Verify Local Administrator Access

```
# Ensure local admin account is enabled and password is known
Get-LocalUser -Name "Administrator" | Select-Object Name, Enabled

# Enable if disabled
Enable-LocalUser -Name "Administrator"

# Set password if needed (do this BEFORE unjoin!)
Set-LocalUser -Name "Administrator" -Password (ConvertTo-SecureString "<PASSWORD>" -AsPlainText -Force)
```

Step 4: Check for Active Sessions

```
# List active sessions
query user

# List open file handles (if file server)
Get-SmbOpenFile | Select-Object ClientComputerName, Path
```

6.5 Execute Domain Unjoin Method 1: PowerShell - Clean Unjoin (Recommended)

This method properly removes the computer account from AD:

Via Azure Run Command:

```
az vm run-command invoke \
  --resource-group <RESOURCE_GROUP> \
  --name <VM_NAME> \
  --command-id RunPowerShellScript \
  --scripts "Remove-Computer -UnjoinDomainCredential (New-Object PSCredential('<DOMAIN>\<ADMIN_USER>',
```

Via RDP/PowerShell on VM (interactive):

```
# Clean unjoin - removes computer object from AD
Remove-Computer -UnjoinDomainCredential (Get-Credential) -WorkgroupName "WORKGROUP" -Force -Restart
```

Method 2: PowerShell - Force Unjoin (When DC Unavailable)

Use when Domain Controller is not accessible:

```
# Force unjoin without contacting DC (computer object remains in AD)
Remove-Computer -WorkgroupName "WORKGROUP" -Force -Restart

# NOTE: Manually delete computer object from AD later
```

Method 3: System Properties GUI (via RDP)

1. Connect to VM via RDP or Azure Bastion
2. Open **System Properties**: `sysdm.cpl`
3. Click **Computer Name** tab
4. Click **Change...**
5. Select **Workgroup** radio button
6. Enter workgroup name (e.g., `WORKGROUP`)

7. Click **OK**
8. Enter Domain Administrator credentials when prompted
9. Click **OK** on success message
10. Restart the server when prompted

6.6 Post-Unjoin Tasks Step 1: Verify Workgroup Membership

```
# Verify no longer domain-joined
(Get-WmiObject Win32_ComputerSystem) | Select-Object Name, Domain, PartOfDomain

# Expected output:
# Name          Domain      PartOfDomain
# ----          ------      ------------
# SERVERNAME    WORKGROUP   False
```

Step 2: Update DNS Settings (if needed)

```
# Change DNS to public DNS or other servers
Set-DnsClientServerAddress -InterfaceAlias "Ethernet" -ServerAddresses "8.8.8.8","8.8.4.4"
```

Step 3: Clean Up AD Computer Object (if force unjoin)

From Domain Controller:

```
# Find and remove stale computer object
Get-ADComputer -Identity "<COMPUTER_NAME>" | Remove-ADComputer -Confirm:$false
```

Step 4: Remove Cached Domain Credentials

```
# Clear cached credentials
cmdkey /list
cmdkey /delete:Domain:target=<DOMAIN_NAME>
```

7.0 Verification & Quality Checks

7.1 Domain Join Verification Checklist

- ☐ `PartOfDomain` returns `True`
- ☐ Computer object exists in correct OU in AD
- ☐ DNS A record created for computer
- ☐ Domain users can authenticate to server
- ☐ Group Policy applies successfully (`gpresult /r`)
- ☐ Server appears in AD Users and Computers

7.2 Domain Unjoin Verification Checklist

- ☐ `PartOfDomain` returns `False`
- ☐ Domain shows as `WORKGROUP`
- ☐ Local Administrator login works
- ☐ Computer object removed from AD (if clean unjoin)
- ☐ DNS settings updated appropriately
- ☐ Services using domain accounts are reconfigured or stopped

---

8.0 Troubleshooting

Issue	Resolution
"The specified domain either does not exist or could not be contacted"	Verify DNS points to DC, test port 389 connectivity, check firewall rules
"Access is denied" during join	Verify domain admin credentials, check if user has "Add workstations to domain" permission
"The machine account quota has been exceeded"	Contact domain admin to increase quota or pre-stage computer account
Computer object in wrong OU	Move object in AD Users and Computers, or unjoin/rejoin with `-OUPath` parameter
Can't unjoin - DC unavailable	Use force unjoin method, manually clean up AD object later
Local admin account locked after unjoin	Boot to Safe Mode, use Azure Serial Console, or restore from snapshot
"The trust relationship between this workstation and the primary domain failed"	Unjoin and rejoin the domain, or reset computer account in AD
Group Policy not applying after join	Run `gpupdate /force`, verify DNS, check OU GPO links
Azure Run Command times out	Increase timeout, use Azure Bastion/RDP instead

8.1 Azure-Specific Troubleshooting

Issue	Resolution
VM not starting after domain join	Check boot diagnostics, may need to restore from snapshot
Network connectivity lost after join	Review NSG rules, verify no conflicting IP addresses
Run Command returns "VM not running"	Start VM first: `az vm start --resource-group <RG> --name <VM>`
DNS not resolving after Azure VNet change	Restart VM, flush DNS: `ipconfig /flushdns`

9.0 Related Documents

Document	Description
SOP-AD-001	Domain Troubleshooting
SOP-AD-002	Mac AD Integration
SOP-AZ-001	Azure VM Administration
SOP-NET-xxx	Network Firewall Port Configuration

9.1 External References

- Microsoft: Join a Computer to a Domain
- Azure Run Command Documentation
- Active Directory Ports and Protocols

10.0 Revision History

Version	Date	Author	Change Description
1.0	2026-01-12	OberaConnect	Initial document creation

11.0 Approval

Name	Role	Signature	Date
	Technical Lead Operations Manager		

Appendix A: Quick Reference Commands

Domain Join (One-Liner)

```
Add-Computer -DomainName "domain.com" -Credential (Get-Credential) -Restart -Force
```

Domain Unjoin (One-Liner)

```
Remove-Computer -UnjoinDomainCredential (Get-Credential) -WorkgroupName "WORKGROUP" -Force -Restart
```

Check Domain Status

```
(Get-WmiObject Win32_ComputerSystem).Domain
```

Azure CLI - Run Command Template

```
az vm run-command invoke --resource-group <RG> --name <VM> --command-id RunPowerShellScript --scripts "
```