Troubleshooting Guide - Azure DevOps VMSS with Microsoft Prebuilt Images

This comprehensive troubleshooting guide helps you diagnose and resolve common issues with the Azure DevOps VMSS template deployment and operation.

Quick Diagnostic Checklist

Before diving into specific issues, run this quick diagnostic checklist:

```
# 1. Check Azure CLI authentication
az account show

# 2. Verify resource group exists
az group show --name <your-resource-group>

# 3. Check VMSS status
az vmss show --resource-group <rg> --name <vmss-name> --query "provisioningState"

# 4. Check VMSS instances
az vmss list-instances --resource-group <rg> --name <vmss-name> --output table

# 5. Test network connectivity
az vmss run-command invoke \
  --resource-group <rg> \
  --name <vmss-name> \
  --command-id RunPowerShellScript \
  --instance-id 0 \
  --scripts "Test-NetConnection dev.azure.com -Port 443"
```

Deployment Issues

Issue 1: Marketplace Terms Not Accepted

Symptoms:

```
Error: The subscription is not registered for the offer 'visualstudio2022' with pub-
lisher 'MicrosoftVisualStudio'
```

Root Cause: Visual Studio marketplace images require explicit terms acceptance.

Solution:

```
# Accept terms for Visual Studio 2022 Enterprise
az vm image terms accept \
  --publisher MicrosoftVisualStudio \
  --offer visualstudio2022 \
  --plan vs-2022-ent-latest-ws2022

# Accept terms for Visual Studio 2022 Build Tools
az vm image terms accept \
  --publisher MicrosoftVisualStudio \
  --offer visualstudio2022 \
  --plan vs-2022-buildtools-latest-ws2022

# Accept terms for Visual Studio 2019 Enterprise
az vm image terms accept \
  --publisher MicrosoftVisualStudio \
  --offer visualstudio2019 \
  --plan vs-2019-ent-latest-ws2019

# Verify terms acceptance
az vm image terms show \
  --publisher MicrosoftVisualStudio \
  --offer visualstudio2022 \
  --plan vs-2022-ent-latest-ws2022
```

Prevention: Always accept marketplace terms before deployment.

Issue 2: Insufficient Azure Permissions

Symptoms:

```
Error: Authorization failed. The client does not have authorization to perform action
'Microsoft.Compute/virtualMachineScaleSets/write'
```

Root Cause: Insufficient permissions on the resource group or subscription.

Solution:

```
# Check current permissions
az role assignment list --assignee $(az account show --query user.name -o tsv) --out-
put table

# Assign Contributor role to resource group
az role assignment create \
  --assignee <user-or-service-principal> \
  --role Contributor \
  --scope /subscriptions/<subscription-id>/resourceGroups/<resource-group>

# For service principal deployment
az role assignment create \
  --assignee <service-principal-id> \
  --role Contributor \
  --scope /subscriptions/<subscription-id>/resourceGroups/<resource-group>
```

Required Permissions:

- `Microsoft.Compute/virtualMachineScaleSets/*`

- `Microsoft.Network/*`
- `Microsoft.Storage/*`
- `Microsoft.Resources/*`

Issue 3: VM Quota Exceeded

Symptoms:

```
Error: Operation could not be completed as it results in exceeding approved quota for
resource type 'standardDSv3Family' in region 'East US'
```

Root Cause: Insufficient VM quota for the requested VM size and region.

Solution:

```
# Check current quota usage
az vm list-usage --location "East US" --output table

# Check specific VM family quota
az vm list-usage --location "East US" --query "[?contains(name.value, 'standardDSv3Fam-
ily')]"

# Request quota increase (requires Azure portal)
echo "Request quota increase through Azure portal:"
echo "https://portal.azure.com/#blade/Microsoft_Azure_Support/HelpAndSupportBlade/news-
upportrequest"
```

Workarounds:
- Use smaller VM sizes (`Standard_D2s_v3` instead of `Standard_D4s_v3`)
- Deploy in different regions
- Use different VM families (`Standard_B` series for testing)

Issue 4: Network Configuration Errors

Symptoms:

```
Error: Subnet 'subnet-agents' is not found in virtual network 'vnet-devops'
```

Root Cause: Network configuration mismatch or missing network resources.

Solution:

```
# Check if VNet exists
az network vnet show --resource-group <rg> --name <vnet-name>

# List subnets in VNet
az network vnet subnet list --resource-group <rg> --vnet-name <vnet-name> --output tabl
e

# Create missing subnet
az network vnet subnet create \
  --resource-group <rg> \
  --vnet-name <vnet-name> \
  --name <subnet-name> \
  --address-prefixes 10.0.1.0/24

# For existing VNet integration, verify parameters
az network vnet show \
  --resource-group <existing-vnet-rg> \
  --name <existing-vnet-name> \
  --query "subnets[].{Name:name, AddressPrefix:addressPrefix}"
```

Agent Registration Issues

Issue 5: Agent Registration Fails

Symptoms:
- Agents don't appear in Azure DevOps agent pool
- PowerShell script reports authentication errors
- Agents show as "Offline" in Azure DevOps

Root Cause: Authentication, network, or configuration issues.

Diagnostic Steps:

```
# Check agent configuration logs
az vmss run-command invoke \
  --resource-group <rg> \
  --name <vmss-name> \
  --command-id RunPowerShellScript \
  --instance-id 0 \
  --scripts "Get-Content C:\temp\devops-agent-config.log -Tail 50"

# Check agent summary
az vmss run-command invoke \
  --resource-group <rg> \
  --name <vmss-name> \
  --command-id RunPowerShellScript \
  --instance-id 0 \
  --scripts "Get-Content C:\temp\agent-config-summary.txt"

# Check agent service status
az vmss run-command invoke \
  --resource-group <rg> \
  --name <vmss-name> \
  --command-id RunPowerShellScript \
  --instance-id 0 \
  --scripts "Get-Service 'vstsagent*'"
```

Solutions:

5.1 PAT Token Issues

```
# Verify PAT token permissions in Azure DevOps:
# 1. Go to User Settings → Personal Access Tokens
# 2. Check token expiration
# 3. Verify scopes include "Agent Pools (read, manage)"
# 4. Test token with REST API

curl -u :<PAT_TOKEN> \
  "https://dev.azure.com/<organization>/_apis/distributedtask/pools?api-version=6.0"
```

5.2 Azure DevOps URL Format

```
# Correct format examples:
#   https://dev.azure.com/myorganization
#   https://myorganization.visualstudio.com
#   https://dev.azure.com/myorganization/myproject
#   https://dev.azure.com/myorganization/
```

5.3 Agent Pool Permissions

```
# Check agent pool security in Azure DevOps:
# 1. Go to Organization Settings → Agent pools
# 2. Select your pool → Security
# 3. Verify user/service has "Administer" permissions
```

Issue 6: Network Connectivity to Azure DevOps

Symptoms:

```
Error: Unable to connect to Azure DevOps services
Test-NetConnection dev.azure.com -Port 443 : Failed
```

Root Cause: Network restrictions blocking outbound connectivity.

Diagnostic Steps:

```
# Test connectivity from VMSS instance
az vmss run-command invoke \
  --resource-group <rg> \
  --name <vmss-name> \
  --command-id RunPowerShellScript \
  --instance-id 0 \
  --scripts @test-connectivity.ps1
```

Create `test-connectivity.ps1`:

```
# Test Azure DevOps connectivity
$endpoints = @(
    "dev.azure.com",
    "vstsagentpackage.azureedge.net",
    "login.microsoftonline.com"
)

foreach ($endpoint in $endpoints) {
    Write-Host "Testing connectivity to $endpoint..."
    try {
        $result = Test-NetConnection -ComputerName $endpoint -Port 443 -Information-
Level Quiet
        if ($result) {
            Write-Host "  $endpoint - Connected" -ForegroundColor Green
        } else {
            Write-Host "  $endpoint - Failed" -ForegroundColor Red
        }
    } catch {
        Write-Host "  $endpoint - Error: $($_.Exception.Message)" -ForegroundColor Red
    }
}

# Test DNS resolution
Write-Host "`nDNS Resolution Test:"
try {
    $dnsResult = Resolve-DnsName dev.azure.com
    Write-Host "  DNS Resolution successful" -ForegroundColor Green
    Write-Host "IP Addresses: $($dnsResult.IPAddress -join ', ')"
} catch {
    Write-Host "  DNS Resolution failed: $($_.Exception.Message)" -ForegroundColor Red
}
```

Solutions:

6.1 Azure Firewall Configuration

```
# Check Azure Firewall rules
az network firewall show --resource-group <firewall-rg> --name <firewall-name>

# Required outbound rules for Azure DevOps:
# - HTTPS (443) to *.dev.azure.com
# - HTTPS (443) to *.visualstudio.com
# - HTTPS (443) to vstsagentpackage.azureedge.net
# - HTTPS (443) to login.microsoftonline.com
```

6.2 Network Security Group Rules

```
# Check NSG rules
az network nsg show --resource-group <rg> --name <nsg-name>

# Add rule for Azure DevOps if missing
az network nsg rule create \
  --resource-group <rg> \
  --nsg-name <nsg-name> \
  --name AllowAzureDevOpsHTTPS \
  --priority 1100 \
  --source-address-prefixes VirtualNetwork \
  --destination-address-prefixes Internet \
  --destination-port-ranges 443 \
  --access Allow \
  --protocol Tcp
```

6.3 Route Table Configuration

```
# Check route table (if using custom routing)
az network route-table show --resource-group <rg> --name <route-table-name>

# Ensure 0.0.0.0/0 routes to Azure Firewall
az network route-table route list \
  --resource-group <rg> \
  --route-table-name <route-table-name> \
  --output table
```

Chocolatey Package Installation Issues

Issue 7: Chocolatey Installation Fails

Symptoms:

```
Error: Failed to install Chocolatey package manager
Exception: Unable to download installation script
```

Root Cause: Network restrictions or PowerShell execution policy.

Diagnostic Steps:

```
# Check Chocolatey installation logs
az vmss run-command invoke \
  --resource-group <rg> \
  --name <vmss-name> \
  --command-id RunPowerShellScript \
  --instance-id 0 \
  --scripts "Get-Content C:\ProgramData\chocolatey\logs\chocolatey.log -Tail 20"
```

Solutions:

7.1 PowerShell Execution Policy

```
# Check current execution policy
Get-ExecutionPolicy -List

# Set execution policy for installation
Set-ExecutionPolicy Bypass -Scope Process -Force
```

7.2 Manual Chocolatey Installation

```
# Manual installation script
[System.Net.ServicePointManager]::SecurityProtocol = [System.Net.ServicePointManager]::
SecurityProtocol -bor 3072
$chocoInstallScript = Invoke-WebRequest -Uri "https://community.chocolatey.org/in-
stall.ps1" -UseBasicParsing
Invoke-Expression $chocoInstallScript.Content

# Verify installation
choco --version
```

Issue 8: Package Installation Failures

Symptoms:

```
Warning: Failed to install package: nodejs (Exit code: 1)
Error: The package was not found with the source(s) listed
```

Root Cause: Package name errors, version conflicts, or network issues.

Diagnostic Steps:

```
# Check specific package installation
az vmss run-command invoke \
  --resource-group <rg> \
  --name <vmss-name> \
  --command-id RunPowerShellScript \
  --instance-id 0 \
  --scripts "choco search nodejs --exact"
```

Solutions:

8.1 Verify Package Names

```
# Search for correct package names
choco search nodejs
choco search python
choco search docker

# Install with correct names
choco install nodejs -y
choco install python -y
choco install docker-desktop -y
```

8.2 Handle Version Conflicts

```
# Install specific versions
choco install nodejs --version=18.17.0 -y
choco install python --version=3.11.0 -y

# Force reinstall if conflicts
choco install nodejs --force -y
```

8.3 Alternative Package Sources

```
# Use alternative sources if needed
choco install nodejs --source=https://community.chocolatey.org/api/v2/ -y
```

Performance and Scaling Issues

Issue 9: Auto-scaling Not Working

Symptoms:

- VMSS doesn't scale out under load
- Instances don't scale in when idle
- Auto-scale rules not triggering

Diagnostic Steps:

```
# Check auto-scale settings
az monitor autoscale show \
  --resource-group <rg> \
  --name <vmss-name>-autoscale

# Check auto-scale history
az monitor autoscale profile list \
  --autoscale-name <vmss-name>-autoscale \
  --resource-group <rg>

# Check metrics
az monitor metrics list \
  --resource "/subscriptions/<sub-id>/resourceGroups/<rg>/providers/Microsoft.Compute/
virtualMachineScaleSets/<vmss-name>" \
  --metric "Percentage CPU" \
  --start-time 2024-01-01T00:00:00Z \
  --end-time 2024-01-01T23:59:59Z
```

Solutions:

9.1 Verify Auto-scale Rules

```
# List auto-scale rules
az monitor autoscale rule list \
  --autoscale-name <vmss-name>-autoscale \
  --resource-group <rg>

# Update scale-out rule
az monitor autoscale rule create \
  --resource-group <rg> \
  --autoscale-name <vmss-name>-autoscale \
  --condition "Percentage CPU > 75 avg 5m" \
  --scale out 1 \
  --cooldown 5

# Update scale-in rule
az monitor autoscale rule create \
  --resource-group <rg> \
  --autoscale-name <vmss-name>-autoscale \
  --condition "Percentage CPU < 25 avg 10m" \
  --scale in 1 \
  --cooldown 10
```

9.2 Check Instance Limits

```
# Verify instance count limits
az vmss show \
  --resource-group <rg> \
  --name <vmss-name> \
  --query "sku.capacity"

# Update capacity if needed
az vmss scale \
  --resource-group <rg> \
  --name <vmss-name> \
  --new-capacity 5
```

Issue 10: Slow Agent Performance

Symptoms:

- Build jobs take longer than expected
- High CPU usage on agents
- Memory pressure warnings

Diagnostic Steps:

```
# Check VM size and performance
az vmss show \
  --resource-group <rg> \
  --name <vmss-name> \
  --query "virtualMachineProfile.hardwareProfile.vmSize"

# Check performance metrics
az monitor metrics list \
  --resource "/subscriptions/<sub-id>/resourceGroups/<rg>/providers/Microsoft.Compute/
virtualMachineScaleSets/<vmss-name>" \
  --metric "Percentage CPU,Available Memory Bytes" \
  --start-time $(date -u -d '1 hour ago' +%Y-%m-%dT%H:%M:%SZ) \
  --end-time $(date -u +%Y-%m-%dT%H:%M:%SZ)
```

Solutions:

10.1 Upgrade VM Size

```
# Scale up to larger VM size
az vmss update \
  --resource-group <rg> \
  --name <vmss-name> \
  --set virtualMachineProfile.hardwareProfile.vmSize=Standard_D4s_v3

# Update instances with new size
az vmss update-instances \
  --resource-group <rg> \
  --name <vmss-name> \
  --instance-ids "*"
```

10.2 Optimize Agent Configuration

```
# Increase agent work folder cleanup
# Edit C:\agent\.agent file
{
  "workFolder": "C:\agent\_work",
  "cleanupWorkFolder": true,
  "cleanupWorkFolderAfterBuild": true
}
```

10.3 Monitor Resource Usage

```
# Set up performance monitoring
az monitor metrics alert create \
  --name "High CPU Usage" \
  --resource-group <rg> \
  --scopes "/subscriptions/<sub-id>/resourceGroups/<rg>/providers/Microsoft.Compute/
virtualMachineScaleSets/<vmss-name>" \
  --condition "avg Percentage CPU > 90" \
  --description "Alert when CPU usage is consistently high"
```

Security and Access Issues

Issue 11: Unable to Access VMs for Debugging

Symptoms:
- Cannot RDP to VMSS instances
- No public IP addresses assigned
- Network connectivity issues

Root Cause: Security configuration with private IPs only.

Solutions:

11.1 Use Azure Bastion

```
# Deploy Azure Bastion for secure access
az network bastion create \
  --resource-group <rg> \
  --name bastion-devops \
  --public-ip-address bastion-pip \
  --vnet-name <vnet-name> \
  --location <location>

# Connect via Azure portal → Bastion
```

11.2 Use Run Command for Debugging

```
# Execute PowerShell commands remotely
az vmss run-command invoke \
  --resource-group <rg> \
  --name <vmss-name> \
  --command-id RunPowerShellScript \
  --instance-id 0 \
  --scripts "Get-Process | Where-Object {$_.ProcessName -like '*agent*'}"
```

11.3 Temporary Public IP (Not Recommended for Production)

```
# Add public IP for debugging (temporary)
az vmss update \
  --resource-group <rg> \
  --name <vmss-name> \
  --set virtualMachineProfile.networkProfile.networkInterfaceConfigurations[0].ipCon-
figurations[0].publicIPAddressConfiguration.name=temp-pip

# Remove after debugging
az vmss update \
  --resource-group <rg> \
  --name <vmss-name> \
  --remove virtualMachineProfile.networkProfile.networkInterfaceConfigurations[0].ip-
Configurations[0].publicIPAddressConfiguration
```

Issue 12: Key Vault Integration Issues

Symptoms:

- Cannot retrieve secrets from Key Vault
- Managed identity authentication failures

Solutions:

12.1 Configure Managed Identity

```
# Enable system-assigned managed identity
az vmss identity assign \
  --resource-group <rg> \
  --name <vmss-name>

# Grant Key Vault access
az keyvault set-policy \
  --name <keyvault-name> \
  --object-id <managed-identity-principal-id> \
  --secret-permissions get list
```

12.2 Test Key Vault Access

```
# Test from VMSS instance
$response = Invoke-RestMethod -Uri 'http://169.254.169.254/metadata/identity/oauth2/
token?api-version=2018-02-01&resource=https://vault.azure.net/' -Method GET -Headers
@{Metadata="true"}
$KeyVaultToken = $response.access_token

$secret = Invoke-RestMethod -Uri "https://<keyvault-name>.vault.azure.net/secrets/
<secret-name>?api-version=2016-10-01" -Method GET -Headers @{Authorization="Bearer
$KeyVaultToken"}
```

Monitoring and Logging

Comprehensive Logging Setup

```
# Enable diagnostic settings for VMSS
az monitor diagnostic-settings create \
  --resource "/subscriptions/<sub-id>/resourceGroups/<rg>/providers/Microsoft.Compute/
virtualMachineScaleSets/<vmss-name>" \
  --name "vmss-diagnostics" \
  --logs '[{"category":"Administrative","enabled":true}]' \
  --metrics '[{"category":"AllMetrics","enabled":true}]' \
  --workspace "/subscriptions/<sub-id>/resourceGroups/<rg>/providers/Mi-
crosoft.OperationalInsights/workspaces/<workspace-name>"
```

Log Analysis Queries

```
// Azure DevOps agent registration events
Event
| where Source == "VSO Agent"
| where TimeGenerated > ago(1h)
| project TimeGenerated, Computer, EventLevelName, RenderedDescription

// Performance monitoring
Perf
| where ObjectName == "Processor" and CounterName == "% Processor Time"
| where TimeGenerated > ago(1h)
| summarize avg(CounterValue) by Computer, bin(TimeGenerated, 5m)

// Network connectivity issues
Event
| where Source == "Microsoft-Windows-Kernel-Network"
| where TimeGenerated > ago(1h)
| project TimeGenerated, Computer, EventLevelName, RenderedDescription
```

Emergency Procedures

Complete Environment Reset

```
# 1. Stop all VMSS instances
az vmss stop --resource-group <rg> --name <vmss-name>

# 2. Deallocate instances
az vmss deallocate --resource-group <rg> --name <vmss-name>

# 3. Update VMSS configuration
az vmss update --resource-group <rg> --name <vmss-name> --set <new-configuration>

# 4. Start instances
az vmss start --resource-group <rg> --name <vmss-name>

# 5. Update all instances
az vmss update-instances --resource-group <rg> --name <vmss-name> --instance-ids "*"
```

Agent Pool Recovery

```
# Remove offline agents from pool
az pipelines agent list --pool-id <pool-id> --organization https://dev.azure.com/<org>
--query "[?status=='offline'].id" -o tsv | \
xargs -I {} az pipelines agent delete --pool-id <pool-id> --agent-id {} --organization
https://dev.azure.com/<org>

# Force VMSS instance refresh
az vmss update-instances --resource-group <rg> --name <vmss-name> --instance-ids "*"
```

Getting Additional Help

Support Channels

1. **Azure Support**: For infrastructure and Azure service issues
2. **Azure DevOps Support**: For agent and pipeline issues
3. **Community Forums**: Stack Overflow, Azure DevOps Community
4. **GitHub Issues**: For template-specific problems

Information to Collect

When seeking help, provide:

```
# Environment information
az --version
az account show
az group show --name <rg>

# VMSS configuration
az vmss show --resource-group <rg> --name <vmss-name>

# Recent deployments
az deployment group list --resource-group <rg> --output table

# Error logs
az vmss run-command invoke \
  --resource-group <rg> \
  --name <vmss-name> \
  --command-id RunPowerShellScript \
  --instance-id 0 \
  --scripts "Get-Content C:\temp\devops-agent-config.log"
```

Escalation Matrix

Issue Type	First Contact	Escalation
Azure Infrastructure	Azure Support	Microsoft Premier Support
Azure DevOps Services	Azure DevOps Support	Microsoft Support
Template Issues	GitHub Issues	Community Forums
Network Connectivity	Network Team	Azure Networking Support
Security Issues	Security Team	Azure Security Center

Remember: Most issues can be resolved by checking logs, verifying configuration, and ensuring proper network connectivity. Always start with the diagnostic checklist before diving into specific troubleshooting steps.