Lightweight provisioning system for multi-machine test environments

Vladimir Averkin, Microsoft
Muhammad Usman Janjua, Microsoft
Sunil Kutty, Microsoft
Personal Experience:
Multi-domain test topology with varying trusts

MULTI-DOMAIN TRUST TOPOLOGY DEPLOYMENT USING MACHINE FACTORY

Symbol Count Description
Topology Elements

- Multi-Domain trust topology defined using VM factory XML constructs
- Automated setup for domain controllers and domain members
- Hybrid Forest/External trusts defined and setup between domains
- Domain Admin, Domain Users and Local admin/user accounts created unattended
Personal Experience:
Multi-domain test topology with varying trusts

Challenges ...

1. Acquiring multiple machines.
2. Installing and configuring correct OS.
3. Creating forests and domain controllers (DCs).
5. Configuring member machines for different DCs.
6. Creating unique user-accounts across DCs and member machines.
7. Enable users across machines to enable cross-domain management.
8. Manually do all of the above correctly and consistently within a day or two.
9. What are the odds of getting everything right 😊?
Personal Experience: Multi-domain test topology with varying trusts

Topology Builder provided:

1. A mechanism to crisply define my required topology
2. Ability to specify fine details (e.g., password/naming patterns)
3. A highly robust and repeatable process yielding exact output.
4. Productivity increase in orders of scale.
   - A manual error-prone setup of 2-3 days became a 2-hour deliverable
5. A risk free environment for rigorous testing.
   - Based on virtual machines, I could do destructive testing and still be able to recover the state
   - In worst case, an exact new setup was available within 2 hours
6. Independence from a central test environment
Application Development

- Implement
- Build
- Deploy
- Debug
- Test
- Configure
Testing single machine application

- You have created a single-machine application.
- And you have written automated tests.
- You run the tests, and all tests have passed.
- Are you ready to ship?
- Probably not.
- Testing on one machine/configuration does not guarantee it will work on another machine or in different configuration.
- You need to run tests on multiple machines and configurations to ensure the quality of your release.
Testing distributed applications

- If your application is distributed: web-based, multi-tiered, parallel compute etc.
- You need multiple machines for testing.
- The machines have to be configured in a certain way.
- A static set of machines does not work very well:
  - Configuration changes accumulate
  - Hard-coded settings can go unnoticed
  - Environment dependencies can seep through
Main Ideas So Far

- Reliable application testing requires testing on different configurations.
- For some applications you need more than one machine to test on.
- Deploying new machines for each test pass gives more reliable results.
- Manual machine deployment and configuration is complex and time-consuming.
- Automated machine provisioning system saves your resources and provides higher test quality and coverage.
The Solution - Machine Factory

- Automates machine deployment and configuration.
- Starts with sysprepped Windows VHD.
- Can deploy Hyper-V based virtual machines.
- Can deploy VHD Boot physical machines.
- Configuration is driven by CMD scripts.
- Configuration can include tasks that require multiple reboots.
DEMO
<?xml version="1.0" encoding="utf-8"?>
<Topology Name="DeploymentTest">
  <Description>A set of server SKUs for Server Manager Deployment testing.</Description>
  <Script>
    # Set global variables for configuration tasks.
    $global:fqdn = 'mtest.microsoft.com'
    $global:domain = 'mtest'
    $global:user = 'msuser'
    $global:password = 'Admin01'
  </Script>
  <RunOnFirstLogon>
    <SetNetworkLocationPrivate/>
    <EnablePing/>
    <EnableRemoteDesktop/>
    <JoinDomainWithRetry DomainName="$fqdn" UserName="$domain\$user" Password="$password"/>
    <AddUserToLocalAdministrators UserName="$domain\$user"/>
    <SetAutoLogonAndReboot DomainName="$domain" UserName="$user" Password="$password"/>
  </RunOnFirstLogon>
  <Machine Count="1">
    <Id>Server</Id>
    <NameSchema>"[WIN]-AADD-NNN"</NameSchema>
    <MemorySizeMB>"1024"</MemorySizeMB>
    <VirtualProcessors>"2"</VirtualProcessors>
    <AdministratorPassword>P@ssw0rd"</AdministratorPassword>
    <VHD>\VHD\Windows8\serverdatacenter_en-us.vhd</VHD>
  </Machine>
  <Machine Count="0">
    <Id>Client</Id>
    <NameSchema>"[WIN]-AADD-NNN"</NameSchema>
    <MemorySizeMB>"1024"</MemorySizeMB>
    <VirtualProcessors>"2"</VirtualProcessors>
    <AdministratorPassword>P@ssw0rd"</AdministratorPassword>
    <VHD>\VHD\Windows8\serverstandard_en-us.vhd</VHD>
  </Machine>
</Topology>
Deployment Process

Initial Configuration
- Factory.xml
- Media.xml
- Tasks.xml

Factory Admin

Source OS Media

User Desktop

Execute

Install-Topology

1. Locate Hyper-V Host
2. Copy OS VHD
3. Create Deployment Tasks & unattend.xml
4. Create VM(s) based on Topology.xml
5. Start VM(s)

File Share with Machine Factory
Automated configuration

Unattend.xml

RunOnce

CMD
Creating new configuration task

- Start with a CMD script
- Declare any files that you need to copy
- Check if reboot is expected or required
- Define if the script needs to take parameters.
- Add manifest to `<MachineFactory>\Config\Tasks.xml`
- Put a reference to your task in topology XML.
- Specify parameter values, if needed
<Task Name="SetNetworkLocationPrivate">
  <Description>Set network location type to private.</Description>
  <Copy Source="Tools\sleep.exe" Destination="0:\Tools"/>
  <Copy Source="Tools\SetNetworkLocationPrivate.exe" Destination="0:\Tools"/>
  <Execute>
    <![CDATA[
      REM Wait for network to come online.
      %SystemDrive%\Tools\sleep.exe 30

      REM Log network state for diagnostics
      ipconfig

      REM Skip execution on Server Core, since SetNetworkLocationPrivate.exe does not work on this SKU.
      (wmic path win32_operatingsystem get OperatingSystemSKU /value) | ^
      find "14" & if NOT errorlevel 1 (exit /b 0)

      REM Service 'netprofm' is not yet started on Vista/WS2k8 by the time of execution.
      net start netprofm

      %SystemDrive%\Tools\SetNetworkLocationPrivate.exe

      REM --- Do not show network location UI on logon.
      reg add "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\NetworkList\NewNetworks" ^
       /v NetworkList /t REG_MULTI_SZ /d "" /f
    ]]>}
  </Execute>
</Task>
Plug-ability
Benefits

• **Light-weight and Simple**
  - 55 Kilobytes of download size.
  - Copy on any Windows 7 or higher machine, target a hyper-V server and start building topologies.
  - XML-based interfaces and powershell/cmd based tasks
  - Portable: Deploy on laptop and create topologies from any location

• **Cost-effective and Powerful**
  - No proprietary software required, just windows 7 or higher OS.
  - Single machine environments to complex multi-machine topologies

• **Extensible and Pluggable**
  - If a task is scriptable, it will fit in
  - XML interfaces allow easy plugging into existing test infrastructure
Try it yourself

You can download the tool from

http://mfactory.codeplex.com/
References

MICROSOFT CORPORATION. 2012, Windows PowerShell

MICROSOFT CORPORATION. 2012, Windows Management Instrumentation

CodePlex 2009, Windows PowerShell Management Library for Hyper-V
http://pshyperv.codeplex.com

MICROSOFT CORPORATION. 2010, Remote Desktop Connection Manager – RDCMan 2.2

MICROSOFT CORPORATION. 2012, Script Convert-WindowsImage.ps1 - WIM2VHD for Windows 8!
http://gallery.technet.microsoft.com/scriptcenter/Convert-WindowsImageps1-0fe23a8f

Markin et al. 2010, Test Environment Configuration in a Complex World
Questions?