•注意事项

1、预共享密钥的 s2s 连接地址不能用域名，只能用 IP 地址
2、每个客户配置：Set-ExecutionPolicy Bypass 防止评分时使用脚本评分，客户端无法执行 / 题目要求运行脚本，结果无法执行
3、台 Windows 配置：new-netfirewallrule -dis pass -name pass 放行所有规则，等同于直接关闭防火墙，但是防火墙实际上是开着的
4、配置 S2S VPN 时，建议在接口加上 -persistent 参数，这样它断开也会自动重连（除非 RemoteAccess 服务没有启动）
5、DNS 反向解析，必须要对所有 A 记录创建反解，如果服务域名是用 A 记录写的，也必须要写 PTR
6、客户端重新注册 ddns 的命令：ipconfig /registerdns
7、在检查环境时，可以先查看机器 时区、keymap，是否有预配置 IP、服务
8、安装 AD 域后，将 dns 服务器地址设置为本地 IP，如果没有 IPv6 地址，就取消 IPv6 网卡的监听，如果有，就把 IPv6 DNS 服务器地址也设置为本地
9、Server 2025 中，配置 failover 服务时，如果要连接 iSCSI 等磁盘，必须要先将 failover 集群创建出来后，再连接磁盘，否则创建 failover 集群时会在创建时一直卡住
10、如果子域、RODC 域的 DHCP 无法授权，需要在 林根域 DC 上用命令授权：Add-DhcpServerInDC -DnsName "lyon-router.lyon.paris.local" -IPAddress 10.40.0.1
11、有时 iscsi 配置后端口不会起来，需要手动重启一下服务（重启机器也不行）

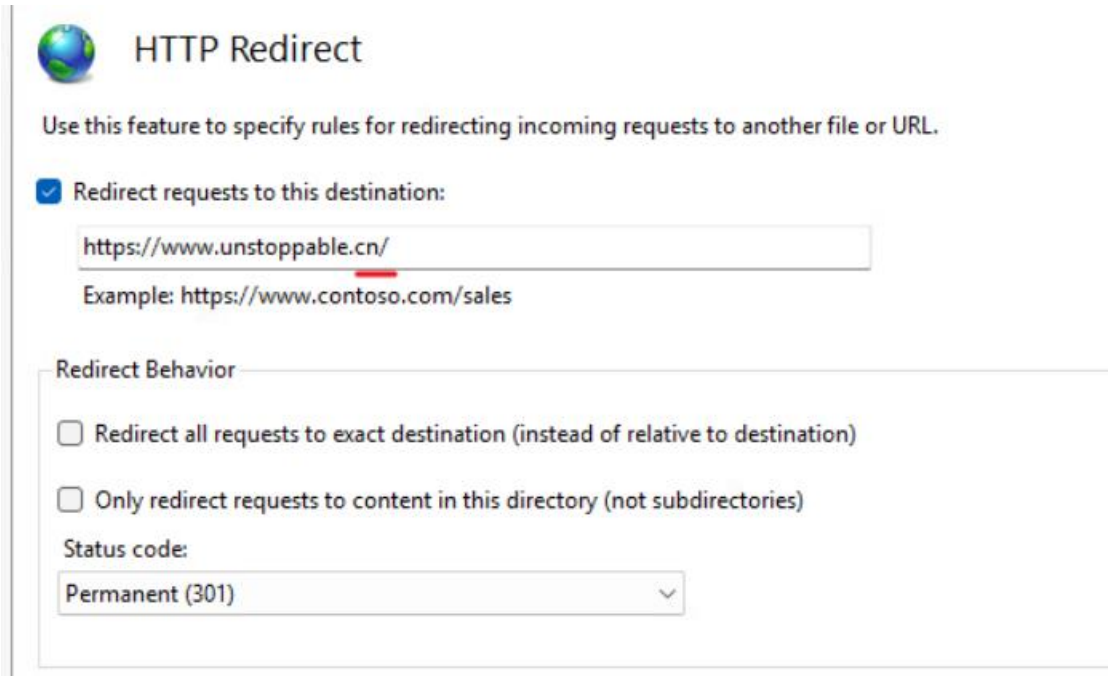
```
PS C:\Users\Administrator> Get-Service WinTarget

Status   Name               DisplayName
------   ----               -----------
Running  WinTarget          Microsoft iSCSI Target Server


PS C:\Users\Administrator> netstat -ano | findstr ":3260"
PS C:\Users\Administrator> restart-service wintarget
PS C:\Users\Administrator> netstat -ano | findstr ":3260"
  TCP    10.0.0.10:3260         0.0.0.0:0              LISTENING       8172
PS C:\Users\Administrator>
```

12、有时候使用 iSCSI 作为后端的 failover，没有自动创建出存储池，可以不创建存储池，在 failover 上之际将这块磁盘添加进去，然后开启维护模式，在服务器管理器中格式化该磁盘（Win + K 有时无法格式化）
13、一些需要计划任务运行 bat 脚本的，例如 DHCP 重启自动恢复，凭据可以指定 SYSTEM（勾选使用最高权限），否则每次它执行都会弹出一个 cmd 框，这样不太好，使用 SYSTEM 用户就不会有弹窗
14、Server 2025 中，adrms 增加了一个必要条件：1、必须能够验证证书吊销
15、如果有两个域、或者子域，建议将各自对端域的 Domain Admins 加入机器本地的 administrators 中：net localgroup Administrators “CN2026.wsc\Domain Admins” /add
16、Windows 域内机器会自动注册 dns 记录，但是假如这个机器有多个 IP，例如网关，那么它就只会注册靠近网关的 IP 的那条记录，其他的记录需要手动添加
17、RSAT 连接时，如果是同一个域的，最好用域名来连接，用 IP 有时候无法连接，例如 DNS
18、可以使用 netsh interface ip set address tun0（定义的 S2S 接口名称） 来指定 VPN 接口地址

19、DFS 根目录和创建的 DFS 文件夹中的共享文件夹的根目录不能一样，否则会出问题，会导致路径重叠，DFS 共享文件夹创建失败
20、在客户端创建证书申请，提交给 CA 签发的证书，有时会因为一些问题导致浏览器无法调用，这个时候可以在 certlm.msc 中将它导出为 pfx，然后再重新导入就能使用了。
21、关于 DNS 转发器，要看题目说的是转发到一个域，还是转发到一个机器，如果是转发到一个域，那就要做条件转发，如果是指向某个 dns 机器，那就是普通转发器
22、web index 文件不要用 echo 来写入内容，要用 set-content，否则就会像下面这样多两个字符
（Ansible 的 win_lineinfile 不会导致这样多出字符）

```
PS C:\> echo "China Web Site" > C:\inetpub\wwwroot\index.html
PS C:\> (Invoke-WebRequest -UseBasicParsing http://www.internet.com).Content
ÿþChina Web Site

PS C:\> Set-Content C:\inetpub\wwwroot\index.html "China Web Site"
PS C:\> (Invoke-WebRequest -UseBasicParsing http://www.internet.com).Content
China Web Site

PS C:\> _
```

23、Windows Web HTTP 重定向到 HTTPS 要使用 HTTP Redirect，不要用 HSTS，因为 HSTS 返回值是 200，HTTP 重定向返回值才是 301（新建一个 http 站点来做这个）
并且这里最好这样写，末尾加 / ，下面两个都不要勾选，这样重定向时就会携带 URI
并且记得这里一定要改成 301，默认是 302



24、子域配置 DFS，创建 DFS 复制组时是一定会出问题的，因为父域会把子域的那台机器用父域的域名去访问，而不是用子域的，需要在 DFS 服务器上添加一个子域的 suffix：
Set-DnsClientGlobalSetting -SuffixSearchList @("shanghai.cn","beijing.shanghai.cn")
25、就算题目没有要求修改密码策略，也必须要更改密码策略，将密码最小更改期限改为 0（默认为 1，这就导致只能修改一次密码，之后必须等待一天）
26、公网 DNS 如果有要求 NS 的 FQDN 的话，需要先设置域名后缀，然后创建一个 . 根域，主机的 FQDN 就会自动创建了。

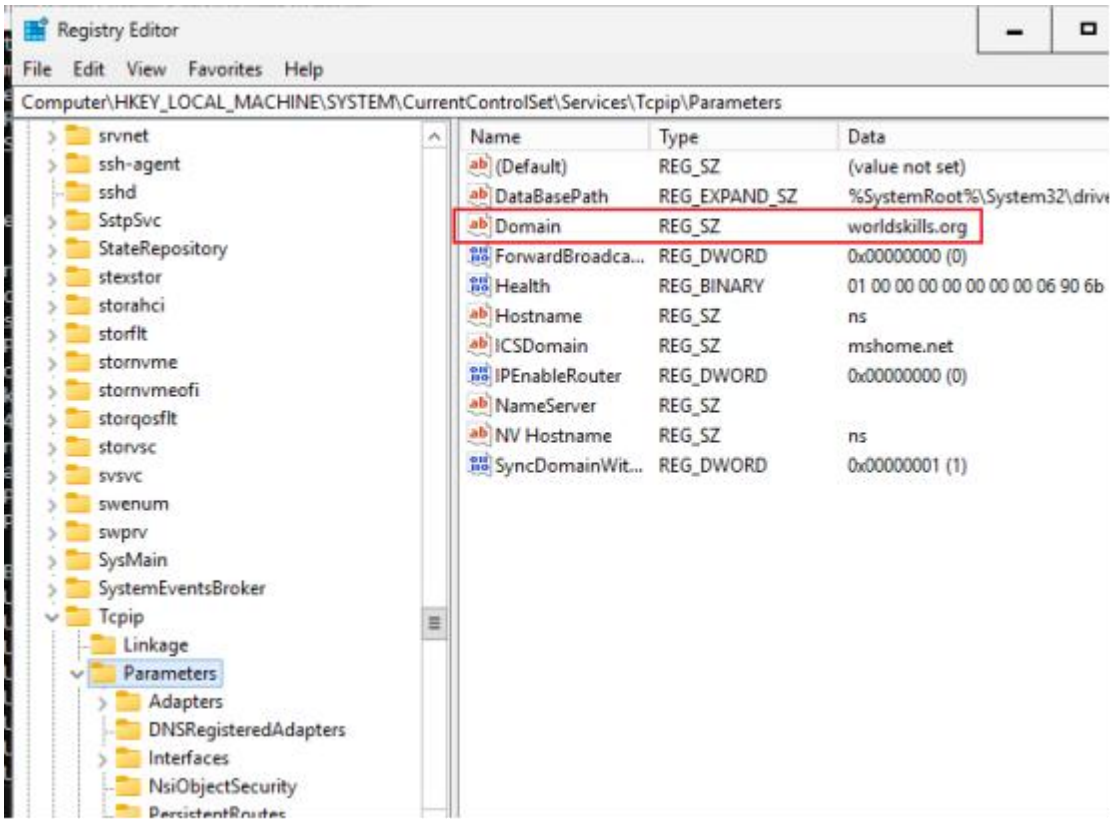
27、如果有两个域，共享文件夹记得给对端域的用户访问权限，共享文件夹权限不要用 Domain Users，要用 users（因为 Users 中有 Authenticated Users，能够包含对端域用户）
28、添加一些用户权限时，建议使用 Administrators、Users 这些。Administrators 同时包含本地管理员、域管理员；Users 同时包含本地用户，Authenticated Users，Authenticated Users 又同时包含域用户，对端做了域信任的域用户。
29、配置完组策略后，大部分都需要 gpupdate 一下策略，然后 logoff 才会生效

一、Powershell/cmd 命令：

1、初始化：

```
new-netfirewallrule -dis pass -name pass

Set-LocalUser administrator -PasswordNeverExpires $true

net user administrator /active:yes

powercfg /change     monitor-timeout-ac 0

powercfg /change     standby-timeout-ac 0

netdom renamecomputer (hostname) /newname:

net user administrator Skill39@Shanghai

Set-LocalUser user -PasswordNeverExpires $true
```

2、CLI 设置 suffix：



3、设置 IP、DNS：

```
# IPv4：
netsh interface ip set address Ethernet0 static 192.168.10.1
netsh interface ipv4 set dnsservers Ethernet0 static 192.168.10.1 primary validate=no

# IPv6：
netsh interface ipv6 add address Ethernet0 2001:db8:10::1/64
netsh interface ipv6 add route ::/0 Ethernet0 2001:db8:10::fffe
netsh interface ipv6 set dnsservers Ethernet0 static 2001:db8:10::1 validate=no

## 删除地址、dns：
# IPv4:
netsh interface ip set address Ethernet0 dhcp
netsh interface ipv4 delete dnsservers Ethernet0 all

# IPv6：
netsh interface ipv6 delete address Ethernet0 2001:db8:10::1
netsh interface ipv6 delete route ::/0 Ethernet0 2001:db8:10::fffe
netsh interface ipv6 delete dnsservers Ethernet0 all
```

•关闭 IPv6 临时地址（只会在客户端开启，服务端不会开启）：

```
Set-NetIpv6Protocol -UseTemporaryAddresses Disabled
```

4、安装软件包：

```
Install-WindowsFeature routing -IncludeManagementTools -Restart
```

5、共享文件夹：

共享权限 不对 SYSTEM、Administrators 做限制，这里只做白名单，其他应该在 NTFS 中去限制

```
# 创建共享：
net share Groups=D:\DFS\Groups /grant:everyone,full

# 定义多个组：
net share Groups=D:\DFS\Groups /grant:Users,full /grant:Sales,full

# 删除共享：
net share test \\HQS1 /delete
```

```
# 对共享文件夹设置拒绝权限：
Block-SmbShareAccess -Name "MKT" -AccountName "PARIS\User1" -Force
```

```
# 对文件夹下的所有子文件、子文件夹 关闭继承（并保留权限）：
icacls c:\share\personal\ /inheritance:d /T /C
## /inheritance:r 就是禁用继承的同时删除继承来的权限（但是一般都是全删）

# 对文件夹下的所有子文件、子文件夹 开启继承：
icacls c:\share\personal\ /inheritance:e /T /C
```

可以这样查看参数：

```
PS C:\Users\Administrator.UNSTOPPABLE> icacls /? | findstr -i inher
    replaces ACLs with default inherited ACLs for all matching files.
        Inheritance options for the integrity ACE may precede the level
    /inheritance:e|d|r
        e - enables inheritance
        d - disables inheritance and copy the ACEs
        r - remove all inherited ACEs
            Inherited denials
            Inherited grants
        inheritance rights may precede either form and are applied
                (OI) - object inherit
                (CI) - container inherit
                (IO) - inherit only
                (NP) - don't propagate inherit
                (I) - permission inherited from parent container
PS C:\Users\Administrator.UNSTOPPABLE>
```

6、findstr -i 参数，忽略大小写搜索

7、一键安装所有 RSAT 工具：

```
Get-WindowsCapability -Online -name RSAT* | % {Add-WindowsCapability -Online -Name $_.Name -Source D:\LanguagesAndOptionalFeatures\ -LimitAccess}
# Tab 简化：get-windowsca -on -na RSAT* | % {add-windowsca -on -n $_.name d:\l -li}
```

8、DNS：

1）添加/删除 DNS 区域：

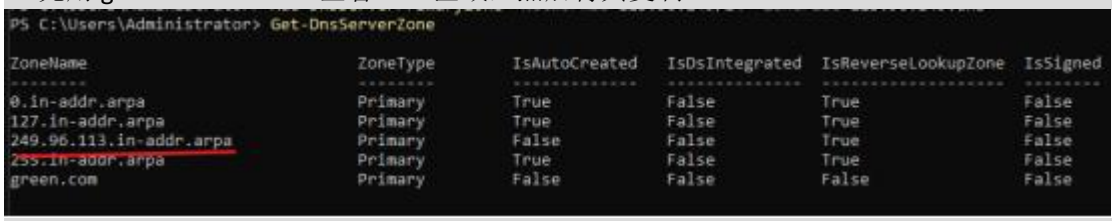
```
# 添加一个普通主区域：
Add-DnsServerPrimaryZone -Name test1.com -zonefile test1.com.dns

# 添加辅助区域：
Add-DnsServerSecondaryZone -Name test1.com -ZoneFile test1.com.dns -MasterServers 192.168.10.1

# 添加普通反向区域：
## 注意不能写全 192.168.10.0，不能这样写，必须写：192.168.10/24，最后一位 IP 要省略
Add-DnsServerPrimaryZone -NetworkID 192.168.10/24 -ZoneFile 192.168.10.dns

# 添加 AD 集成主区域：
Add-DnsServerPrimaryZone -Name test1.com -ReplicationScope Domain  //这里也可以写 'Enterprise'

# 添加一个普通主区域：
Add-DnsServerPrimaryZone -NetworkID 192.168.10 -ZoneFile "10.168.192.in-addr.arpa.dns"

# 添加 AD 集成反向区域：
Add-DnsServerPrimaryZone -NetworkID "192.168.10.0/24" -ReplicationScope "Domain"

# 删除 dns 区域/AD 集成区域：
Remove-DnsServerZone -Name test1.com -Force
```

2）添加 记录：

A 记录：
Add-DnsServerResourceRecordA -ZoneName "wsc2026.org" -Name "www" -IPv4Address "192.168.10.10"

CNAME 记录：
Add-DnsServerResourceRecordCName -ZoneName "wsc2026.org" -Name "ftp" -HostNameAlias "www.wsc2026.org."

PTR 记录：
先用 get-dnsserverzone 查看 PTR 区域，然后将其复制：



添加记录：
Add-DnsServerResourceRecordPtr -ZoneName 249.96.113.in-addr.arpa -PtrDomainName pubsrv01.green.com -Name 225
-PtrDomainName 参数指定域名
-Name 指定地址（注意要反着写，如果 PTR 区域是/16，并且要写的地址是 192.168.10.1，那么这里-Name 就要写 1.10 ）

删除记录：
Remove-DNSServerResourceRecord -Name external -ZoneName paris.com -RRType A // 在 paris.com 区域中删除名为 external 的 A 记录
或者：
Remove-DNSServerResourceRecordA -Name external -ZoneName paris.com

3）指定 根提示、转发器、条件转发器、委派：

指定根提示：
Add-DnsServerRootHint -NameServer ns.worldskills.org -IPAddress 86.20.1.1

指定转发器：
Set-DnsServerForwarder -IPAddress 10.20.20.20
转发失败后，不允许使用根提示（不指定默认就是 true）：
Set-DnsServerForwarder -IPAddress 10.20.20.20 -UseRootHint $false

指定条件转发器：
Add-DnsServerConditionalForwarderZone -Name "beijing.cn" -MasterServers 10.20.20.20

指定委派（这种方式不能用于 根区域：.）：

```
### 首先得先有 shanghai.cn 区域：
Add-DnsServerPrimaryZone -zoneName gz.shanghai.cn -ZoneFile gz.shanghai.cn.dns
Add-DnsServerZoneDelegation -Name "shanghai.cn" -ChildZoneName "gz" -NameServer "dns1.gz.shanghai.cn" -IPAddress 10.30.30.10

## 指定委派（在根 . 区域下，并且其他区域也适用）
add-dnsserverprimaryzone -zonename . -zonefile root.dns
Add-DnsServerResourceRecord -ZoneName . -Name green.com -NS -NameServer pubsrv01.green.com.
Add-DnsServerResourceRecordA -ZoneName . -Name pubsrv01.green.com -IPv4Address 113.96.249.225

### 例如，在根区域上为 shanghai.cn 配置委派：
Add-DnsServerPrimaryZone -ZoneName . -ZoneFile root.dns
Add-DnsServerZoneDelegation -Name . -ChildZoneName shanghai.cn -NameServer ns.shanghai.cn -IpAddress 86.20.3.254
```

```
## 删除：
## 删除委派：
Remove-DnsServerZoneDelegation -Name "." -ChildZoneName "shanghai.cn" -Force

## 删除转发器：
Remove-DnsServerForwarder -IPAddress 10.20.20.20 -Force

## 删除条件转发区域：
就是正常删除区域的命令
Remove-DnsServerZone -Name "beijing.cn" -Force
## 删除根提示：
Remove-DnsServerRootHint -NameServer "root.skills.cn" -IPAddress 10.0.0.10 -Force
```

4）域内客户端自动注册 dns 记录：

```
# 如果有多个 IP，例如网关，只会注册一个地址
ipconfig /registerdns
```

•清除 DNS 缓存：

```
Clear-DnsServerCache -Force
```

9、DHCP：

```
授权 DHCP 服务器：
Add-DhcpServerindc -DnsName “dc.shanghai.cn” -IPAddress 192.168.1.1
```

```
# 这里的 dnsname 写 FQDN（命令在 dc 上执行）
# 查看已经授权的 dhcp 服务器：
get-dhcpserverindc
```

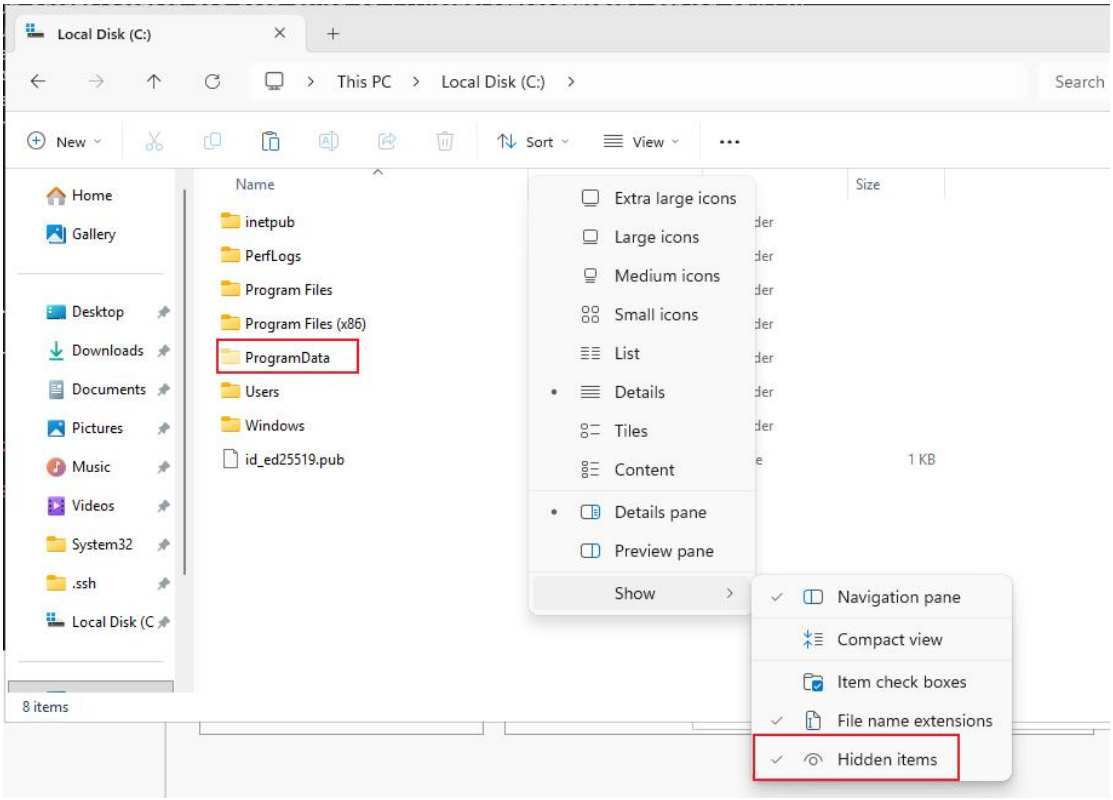
```
PS C:\Users\Administrator> Get-DhcpServerInDC

IPAddress            DnsName
---------            -------
192.168.1.1          dc.shanghai.cn
192.168.1.2          file.shanghai.cn


PS C:\Users\Administrator>
```

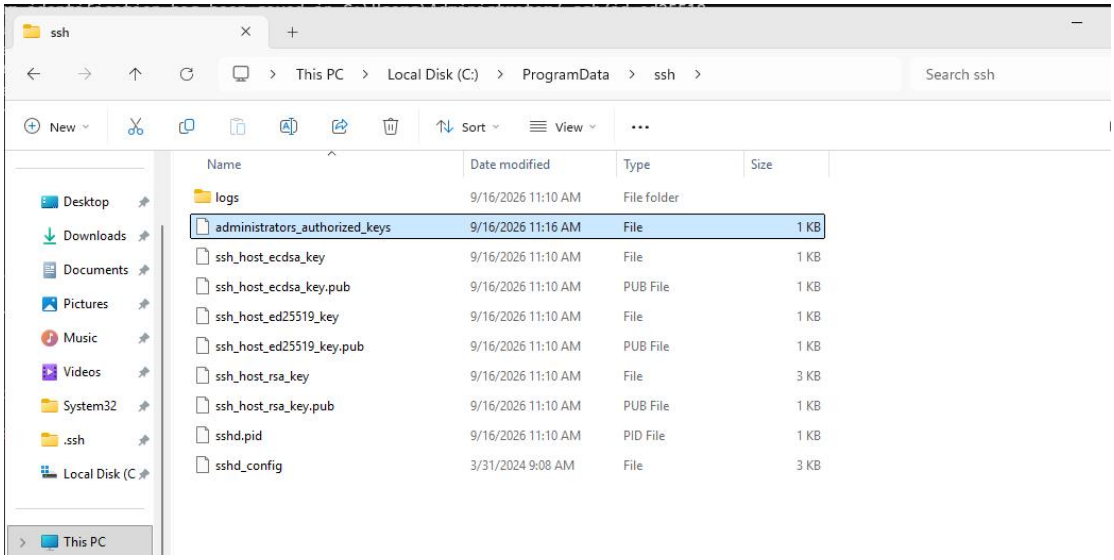
•如果子域、RODC 域的 DHCP 无法授权，需要在 林根域 DC 上用命令授权：

```
Add-DhcpServerInDC -DnsName "lyon-router.lyon.paris.local" -IPAddress 10.40.0.1
```

10、配置 ssh 密钥对连接：

注意：普通用户直接放在 家目录 .ssh\authorized_keys 中
如果是 Administrator 用户，必须放在 C:\ProgramData\ssh\administrators_authorized_keys
直接把客户端公钥复制进去即可





1）先在 sshd 上开启 sshd：

```
start-service sshd
set-service sshd -starttype automatic
```

2）客户端生成密钥对：

```
ssh-keygen
 -t 指定算法
 -b 指定长度 bit
```

然后把公钥拷给服务端，重命名为 authorized_keys 或 administrators_authorized_keys 就能连接了

•测试连接：

```
PS C:\Users\user> ssh administrator@1.1.1.2
Microsoft Windows [Version 10.0.26100.32230]
(c) Microsoft Corporation. All rights reserved.

administrator@WIN-9GCNA7VCALS C:\Users\Administrator>
administrator@WIN-9GCNA7VCALS C:\Users\Administrator>
administrator@WIN-9GCNA7VCALS C:\Users\Administrator>
administrator@WIN-9GCNA7VCALS C:\Users\Administrator>
administrator@WIN-9GCNA7VCALS C:\Users\Administrator>exit
Connection to 1.1.1.2 closed.
PS C:\Users\user>
```

11、设置机器时区：

用命令设置机器时区：

```
C:\Users\Administrator>tzutil /l | findstr China
China Standard Time

C:\Users\Administrator>tzutil /s "China Standard Time"

C:\Users\Administrator>
```

二、脚本：

1、给 Powershell 传递参数，然后 powershell 调用 cmd 脚本，将参数传入 cmd 脚本中：

*import_users.ps1 - Notepad

File Edit Format View Help

```
param(
[string]$path
)


cmd /c c:\user.bat -path $path
```

cmd 脚本中，powershell 的变量用 %，自定义的变量用 %%

user.bat - Notepad

File Edit Format View Help

```
FOR /F "tokens=1-20 delims=," %%a in (%2) do dsadd ou "ou=%%g,dc=shanghai,dc=cn"

FOR /F "tokens=1-20 delims=," %%a in (%2) do dsadd group "cn=%%h,ou=%%g,dc=shanghai,dc=cn"

FOR /F "tokens=1-20 delims=," %%a in (%2) do dsadd user "cn=%%c,ou=%%g,dc=shanghai,dc=cn" -

FOR /F "tokens=1-20 delims=," %%a in (%2) do net user %%c /active:yes

FOR /F "tokens=1-20 delims=," %%a in (%2) do net user %%c "%%f"

FOR /F "tokens=1-20 delims=," %%a in (%2) do powershell set-aduser 'cn=%%c,ou=%%g,dc=shangh
```

2、读取用户表，创建用户：

用 $_.group 这样就能够读取到用户表属性

3、创建用户脚本：

5. Create a powershell script and save it in C:\create_user.ps1, the script must create the following users according to the table.

USER PREFIX	NUMBERS	PASSWORD	OU	GROUP MEMBERSHIPS
mkt	1 – 20	<<Default Password>>	MKT	MKT
sales	1 – 20	<<Default Password>>	SALES	SALES
tech	1 – 20	<<Default Password>>	TECH	TECH
hr	1 – 20	Skill39@LyonSkill39@Lyon	HR	HR

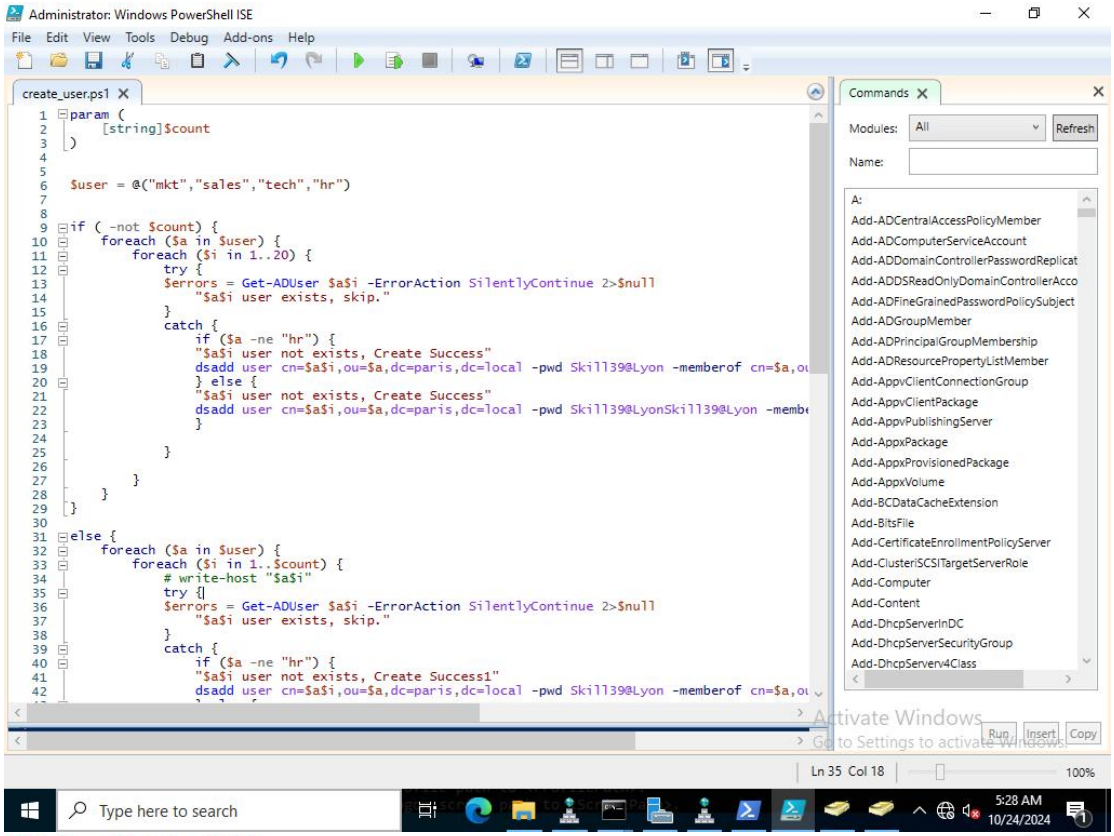
e.g. username: mkt13

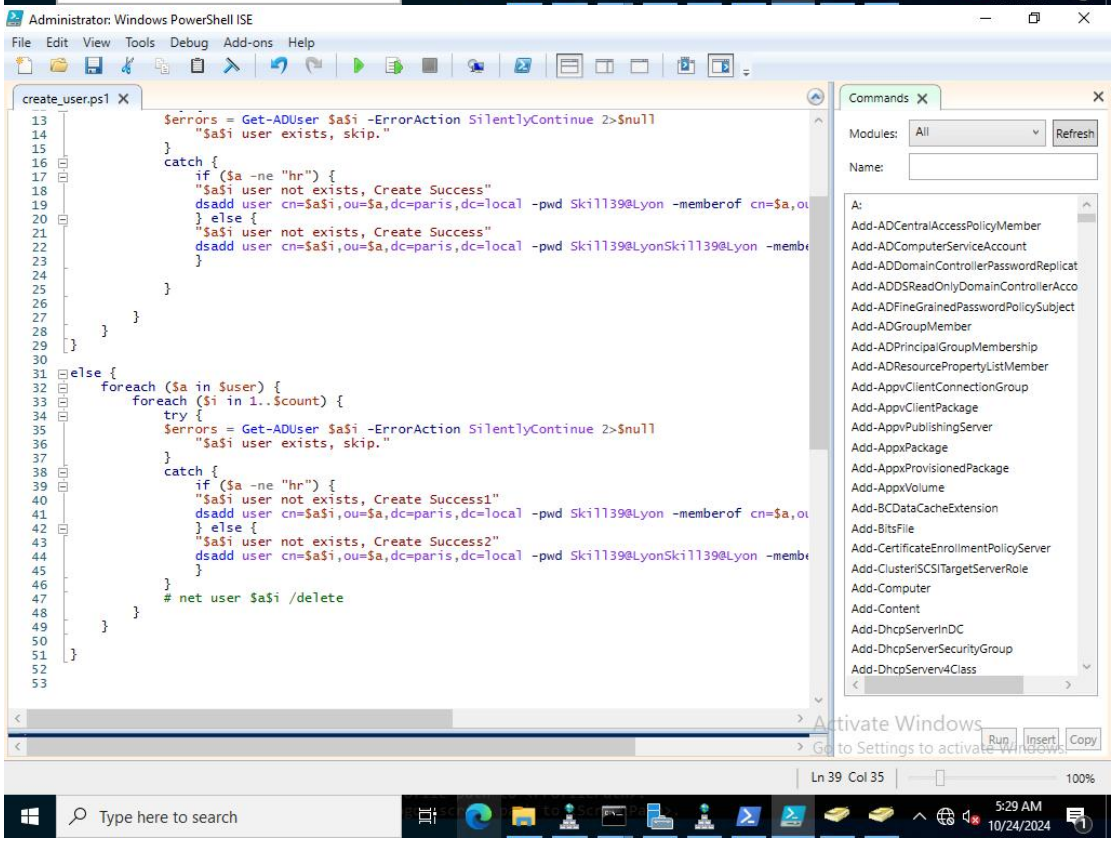
Ensure that your script is runnable, taking in an argument “-count”, when used with “-count X”, X is a number indicating the last number for user creation.

The script should ensure that users created do not need to change password on next logon, and it should ignore creating a user if it already exists.

Sample run: C:\create_user.ps1 -count 5

Note: If you are unable to create the powershell script the users can be created manually but you will not receive marks for the creation of the automated script


Administrator: Windows PowerShell ISE
File Edit View Tools Debug Add-ons Help
create_user.ps1
param (
[string]$count
)
$user = @("mkt","sales","tech","hr")
if (-not $count) {
foreach ($a in $user) {
foreach ($i in 1..20) {
try {
$errors = Get-ADUser ai -ErrorAction SilentlyContinue 2>$null
"ai user exists, skip."
}
catch {
if ($a -ne "hr") {
"ai user not exists, Create Success"
dsadd user cn=ai,ou=$a,dc=paris,dc=local -pwd Skill39@Lyon -memberof cn=$a,ou
} else {
"ai user not exists, Create Success"
dsadd user cn=ai,ou=$a,dc=paris,dc=local -pwd Skill39@LyonSkill39@Lyon -membe
}
}
}
}
else {
foreach ($a in $user) {
foreach ($i in 1..$count) {
# write-host "ai"
try {
$errors = Get-ADUser ai -ErrorAction SilentlyContinue 2>$null
"ai user exists, skip."
}
catch {
if ($a -ne "hr") {
"ai user not exists, Create Success1"
dsadd user cn=ai,ou=$a,dc=paris,dc=local -pwd Skill39@Lyon -memberof cn=$a,ou
Commands
Modules: All Refresh
Name:
A:
Add-ADCentralAccessPolicyMember
Add-ADComputerServiceAccount
Add-ADDomainControllerPasswordReplicat
Add-ADDSReadOnlyDomainControllerAcco
Add-ADFineGrainedPasswordPolicySubject
Add-ADGroupMember
Add-ADPrincipalGroupMembership
Add-ADResourcePropertyListMember
Add-AppvClientConnectionGroup
Add-AppvClientPackage
Add-AppvPublishingServer
Add-AppxPackage
Add-AppxProvisionedPackage
Add-AppxVolume
Add-BCDataCacheExtension
Add-BitsFile
Add-CertificateEnrollmentPolicyServer
Add-ClusteriSCSITargetServerRole
Add-Computer
Add-Content
Add-DhcpServerInDC
Add-DhcpServerSecurityGroup
Add-DhcpServerv4Class
Activate Windows
Go to Settings to activate Windows.
Run Insert Copy
Ln 35 Col 18
100%
Type here to search
5:28 AM
10/24/2024


Administrator: Windows PowerShell ISE
File Edit View Tools Debug Add-ons Help
create_user.ps1
$errors = Get-ADUser ai -ErrorAction SilentlyContinue 2>$null
"ai user exists, skip."
}
catch {
if ($a -ne "hr") {
"ai user not exists, Create Success"
dsadd user cn=ai,ou=$a,dc=paris,dc=local -pwd Skill39@Lyon -memberof cn=$a,ou
} else {
"ai user not exists, Create Success"
dsadd user cn=ai,ou=$a,dc=paris,dc=local -pwd Skill39@LyonSkill39@Lyon -membe
}
}
}
}
else {
foreach ($a in $user) {
foreach ($i in 1..$count) {
try {
$errors = Get-ADUser ai -ErrorAction SilentlyContinue 2>$null
"ai user exists, skip."
}
catch {
if ($a -ne "hr") {
"ai user not exists, Create Success1"
dsadd user cn=ai,ou=$a,dc=paris,dc=local -pwd Skill39@Lyon -memberof cn=$a,ou
} else {
"ai user not exists, Create Success2"
dsadd user cn=ai,ou=$a,dc=paris,dc=local -pwd Skill39@LyonSkill39@Lyon -membe
}
}
# net user ai /delete
}
}
}
Commands
Modules: All Refresh
Name:
A:
Add-ADCentralAccessPolicyMember
Add-ADComputerServiceAccount
Add-ADDomainControllerPasswordReplicat
Add-ADDSReadOnlyDomainControllerAcco
Add-ADFineGrainedPasswordPolicySubject
Add-ADGroupMember
Add-ADPrincipalGroupMembership
Add-ADResourcePropertyListMember
Add-AppvClientConnectionGroup
Add-AppvClientPackage
Add-AppvPublishingServer
Add-AppxPackage
Add-AppxProvisionedPackage
Add-AppxVolume
Add-BCDataCacheExtension
Add-BitsFile
Add-CertificateEnrollmentPolicyServer
Add-ClusteriSCSITargetServerRole
Add-Computer
Add-Content
Add-DhcpServerInDC
Add-DhcpServerSecurityGroup
Add-DhcpServerv4Class
Activate Windows
Go to Settings to activate Windows.
Run Insert Copy
Ln 39 Col 35
100%
Type here to search
5:29 AM
10/24/2024

三、BGP：

•注意：windows bgp 邻居 不要同时建立，要一个一个单独建立，否则会出问题

建立好一个之后，再去写下一个邻居的地址

1、IPv4 BGP：

```
Install-RemoteAccess -RoleType RoutingOnly
Add-BgpRouter -BgpIdentifier 1.1.1.1 -LocalASN 501 -TransitRouting Enabled
```

2、IPv6 BGP：

先用 netsh interface ipv6 set interface 开启 routing 路由转发
然后 set-bgprouter -ipv6routing enabled -localipv6address [随便指定一个 ipv6 地址]
然后就能够建立 IPv6 BGP 邻居了

•BGP 路由聚合：

```
# 创建路由聚合：
Add-BgpRoute-Aggregate -Prefix 2001:db8:20::/48 -SummaryOnly Enabled

# 然后将路由通告出去：
Add-BgpCustomRoute -Network 2001:db8:20::/48
```

四、CA：

注意，配置 CA 前，要按题目先确认 是否加域、是否要改 WorkGroup

1、导入信任 根证书：

```
certutil -addstore -f root c:\root1.cer
certutil -addstore -f root c:\sub1.cer
```

2、子 CA 手动安装证书：

```
certutil -installcert c:\sub1.cer
```

3、使用 命令申请证书：

```
get-certificate -Template Web -SubjectName cn=www.paris.local -DnsName www.paris.local -CertStoreLocation cert:\LocalMachine\My

# 如果是多个字段：
get-certificate -Template Web -SubjectName "cn=www.paris.local,cn=test.paris.local" -DnsName "www.paris.local","test.paris.local" -CertStoreLocation cert:\LocalMachine\My
```

4、安装 根 CA，设置 CACommonName：

```
Install-AdcsCertificationAuthority -CACommonName worldskills-CA

## 指定 CA 类型为 企业 CA、企业子 CA、独立 CA
-CAType EnterpriseRootCA //企业根 CA
-CAType EnterpriseSubordinateCA  //企业子 CA
-CAType StandaloneRootCA  //独立根 CA
## 不指定类型时，WORKGROUP 环境 默认是独立根 CA，域环境 默认是企业根 CA（这个点是 GPT 说的，具体不太确定，建议加上 CAType）
```

10、创建存储池：

```
# 先添加 4 块磁盘

$PhysicalDisks=(Get-PhysicalDisk -CanPool $True)
或获取指定几块磁盘：
## 这里一共有 4 块盘，编号分别是 5、6、7、8（用 Get-PhysicalDisk 查看编号）
$PhysicalDisks=5,6,7 | %{Get-PhysicalDisk -DeviceNumber $_}

# 创建存储池：
New-StoragePool -FriendlyName HRDKOREA-POOL -StorageSubSystemFriendlyName "Windows Storage*" -PhysicalDisks $PhysicalDisks

## 创建虚拟磁盘直接分配最大大小
New-VirtualDisk -StoragePoolFriendlyName HRDKOREA-POOL -FriendlyName RAID5 -ResiliencySettingName Parity -ProvisioningType Fixed -UseMaximumSize
或者精简分配，但是要手动指定大小上限：
New-VirtualDisk -StoragePoolFriendlyName HRDKOREA-POOL -FriendlyName RAID5 -ResiliencySettingName Parity -ProvisioningType Thin -Size 20GB
```

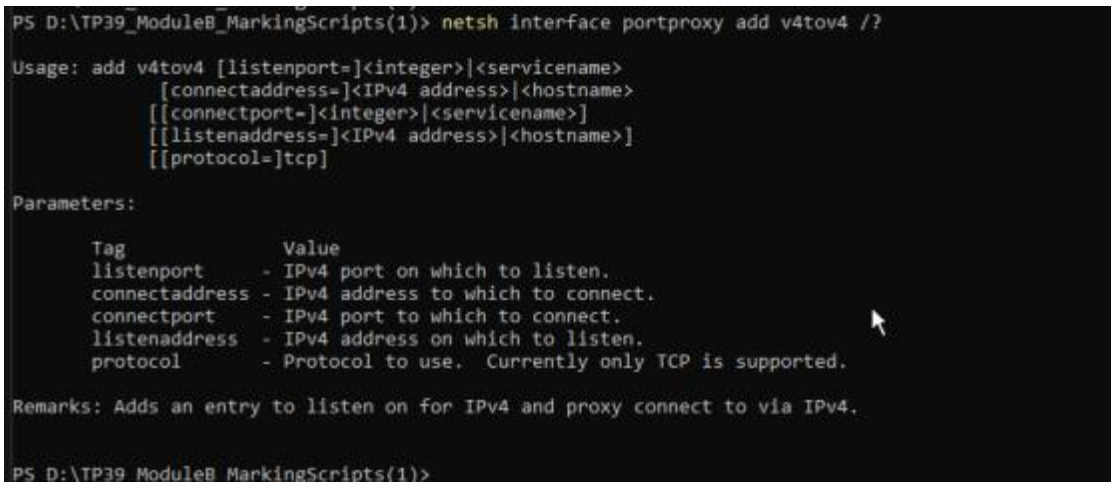
```
Set-VirtualDisk -FriendlyName RAID5 -IsManualAttach $False

get-disk //查看磁盘号码
Initialize-Disk -Number 10 -PartitionStyle GPT

New-Partition -DiskNumber 10 -UseMaximumSize -DriveLetter Z

Format-Volume -DriveLetter Z -Filesystem NTFS

get-physicaldisk //先查看物理磁盘，指定一个要被设置为热备的盘
## 将一块不在存储池中的磁盘设置为热备盘
Add-PhysicalDisk -StoragePoolFriendlyName "HRDKOREA-POOL" -PhysicalDisks (Get-PhysicalDisk -DeviceNumber 8) -Usage HotSpare

删除卷：
Get-Partition -DriveLetter Z | Remove-Partition
```

11、加入域：

```
Add-Computer -DomainName shanghai.cn
```

12、使用 windows 内置的端口转发来实现代理（它目前只支持 TCP 端口的代理）

```
netsh interface portproxy add v4tov4 listenaddress=0.0.0.0 listenport=80 connectaddress=10.0.1.100 connectport=80
netsh interface portproxy add v4tov4 listenaddress=0.0.0.0 listenport=443connectaddress=10.0.1.100 connectport=443
```



五：AD 域：

1、安装 AD 域控：

1）安装新林：

```
install-addsforest
```

2）新增域控到现有域：

新域控的 DNS 必须要有 SRV 记录才能被使用

```
Install-ADDSDomainController -InstallDNS
## 可选参数：
-noglobalcatalog //不作为 GC
-readonlyreplica //RODC
## Windows 第二个域控有时候不会自己创建 DNS 和 SRV 记录，此时可以先从主服务器复制 DNS 区域，然后手动生成 SRV 记录：
nltest /dsregdns  //生成 SRV 记录
ipconfig /registerdns
restart-service netlogon
```

3）在现有林中创建新域：

```
Install-ADDSDomain
Install-ADDSDomain -NewDomainName "company.cn" -ParentDomainName "skills.com" -DomainType TreeDomain -InstallDNS
```

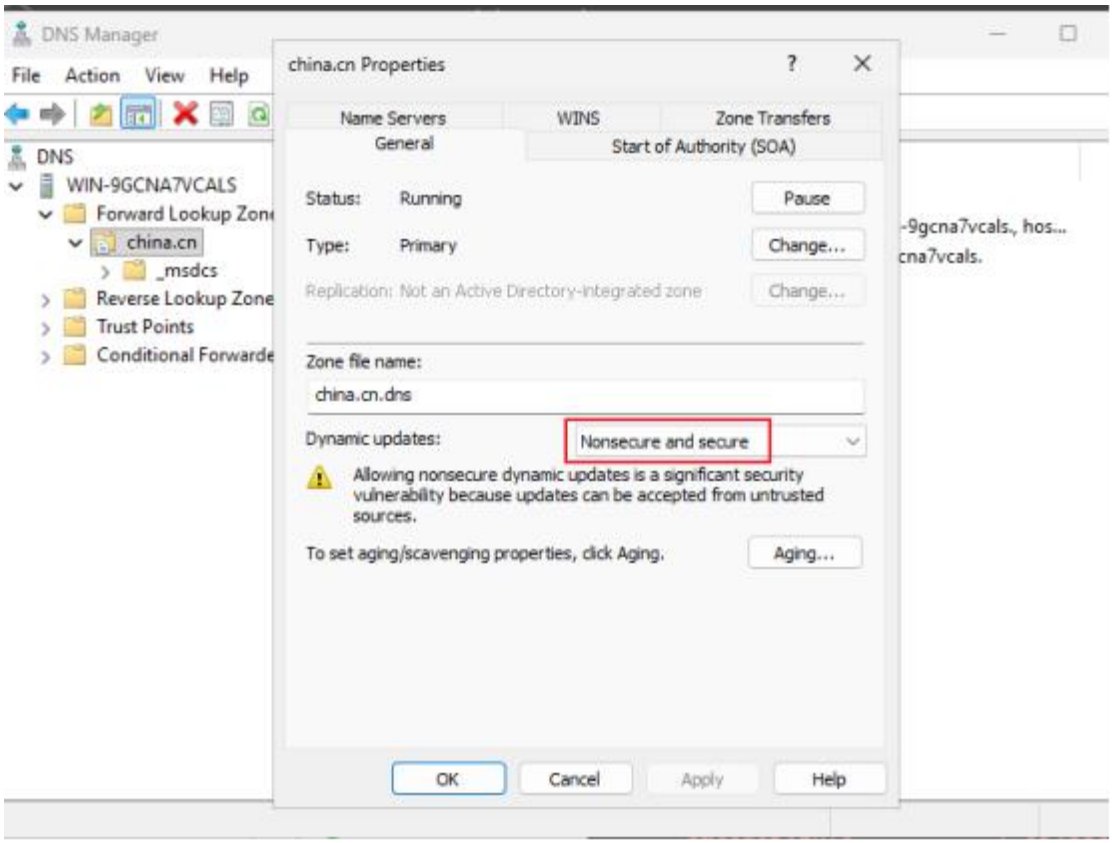
4）创建子域：

```
Install-ADDSDomain -NewDomainName "gz" -ParentDomainName "skills.com" -DomainType ChildDomain -InstallDNS
```

5）创建新林，但是本机不安装 dns：

DC：
install-addsforest -installdns:$false

DNS 机器：
install-windowsfeature dns -in..
然后必须创建出 DNS 区域（不需要记录），并且将动态更新设置为 Nonsecure and secure



然后在 DC 上将 dns 服务器地址指向 DNS 机器，然后打开停止并启动 netlogon 服务：
restart-service netlogon
这样 _msdcs 记录就自动生成了，然后可以再在 DC 上自动生成主机名记录：
ipconfig /registerdns
然后这样其他机器就能够加域了

六、VPN：

•证书记得要加上 IKE 字段

1、s2s：

•自定义 IKEv2 算法：

记得加 persistent 参数，让 vpn 接口持久化建立连接

设置 VPN 接口算法：

Add-VpnS2SInterface -Name fw.shanghai.cn -Destination fw.shanghai.cn -Protocol IKEv2 -AuthenticationMethod MachineCertificates -EncryptionMethod AES256 -IntegrityCheckMethod SHA256 -DHGroup Group14 -CipherTransformConstants AES256 -Persistent -CustomPolicy

可选：

-AuthenticationTransformConstants SHA256128

-PfsGroup PFS2048

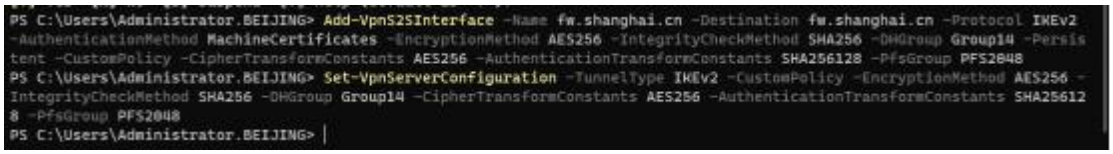
同时要设置 RRAS 全局 IKEv2 算法为一样的值：

Set-VpnServerConfiguration -TunnelType IKEv2 -CustomPolicy -EncryptionMethod AES256 -IntegrityCheckMethod SHA256 -DHGroup Group14 -CipherTransformConstants AES256

可选：

-AuthenticationTransformConstants SHA256128

-PfsGroup PFS2048



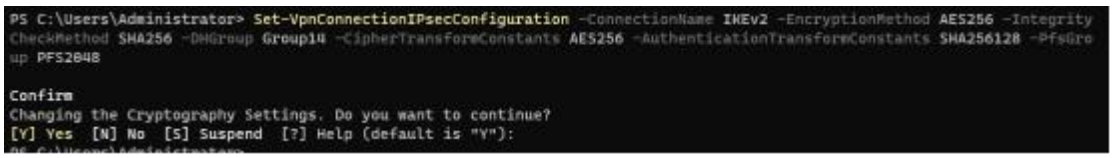
如果有拨号 VPN，要同时修改 VPN 客户端的算法：

Set-VpnConnectionIPsecConfiguration -ConnectionName IKEv2 -EncryptionMethod AES256 -IntegrityCheckMethod SHA256 -DHGroup Group14 -CipherTransformConstants AES256

可选：

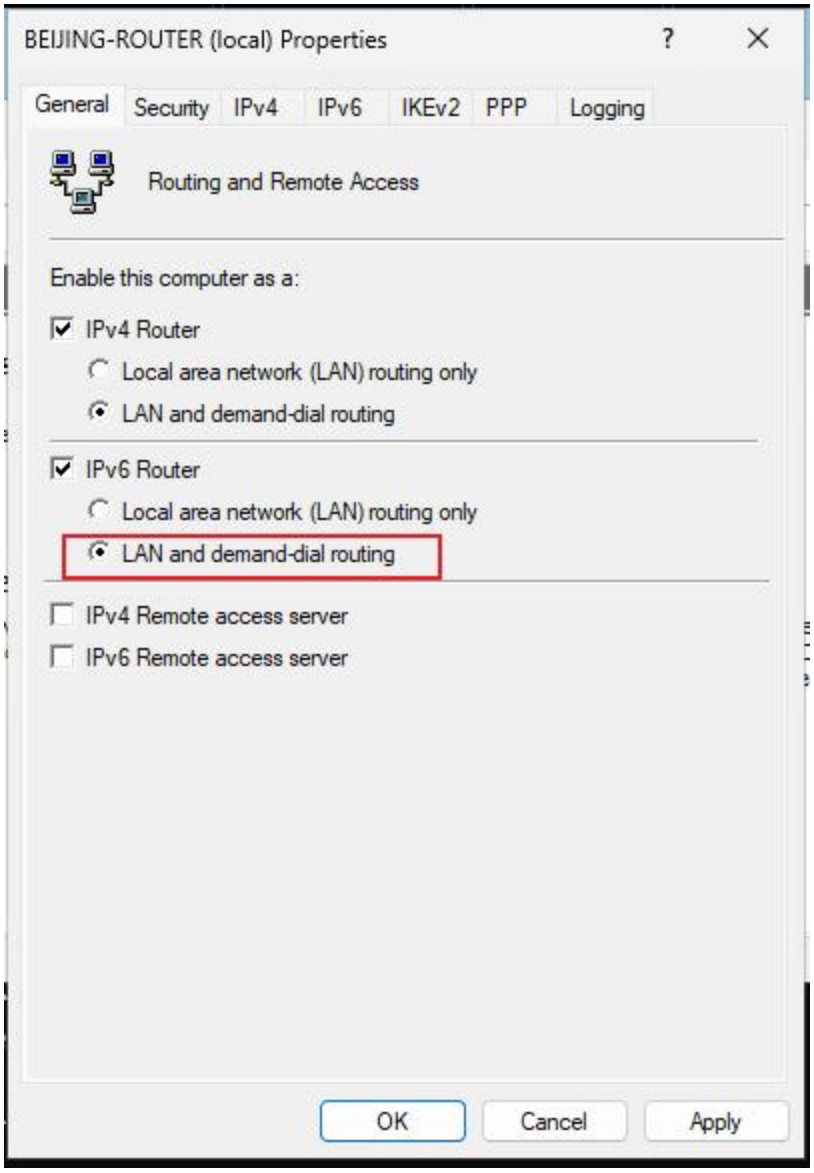
-AuthenticationTransformConstants SHA256128

-PfsGroup PFS2048



•让 S2S 能够通过 IPv6 流量：

要开启 IPv6 路由转发

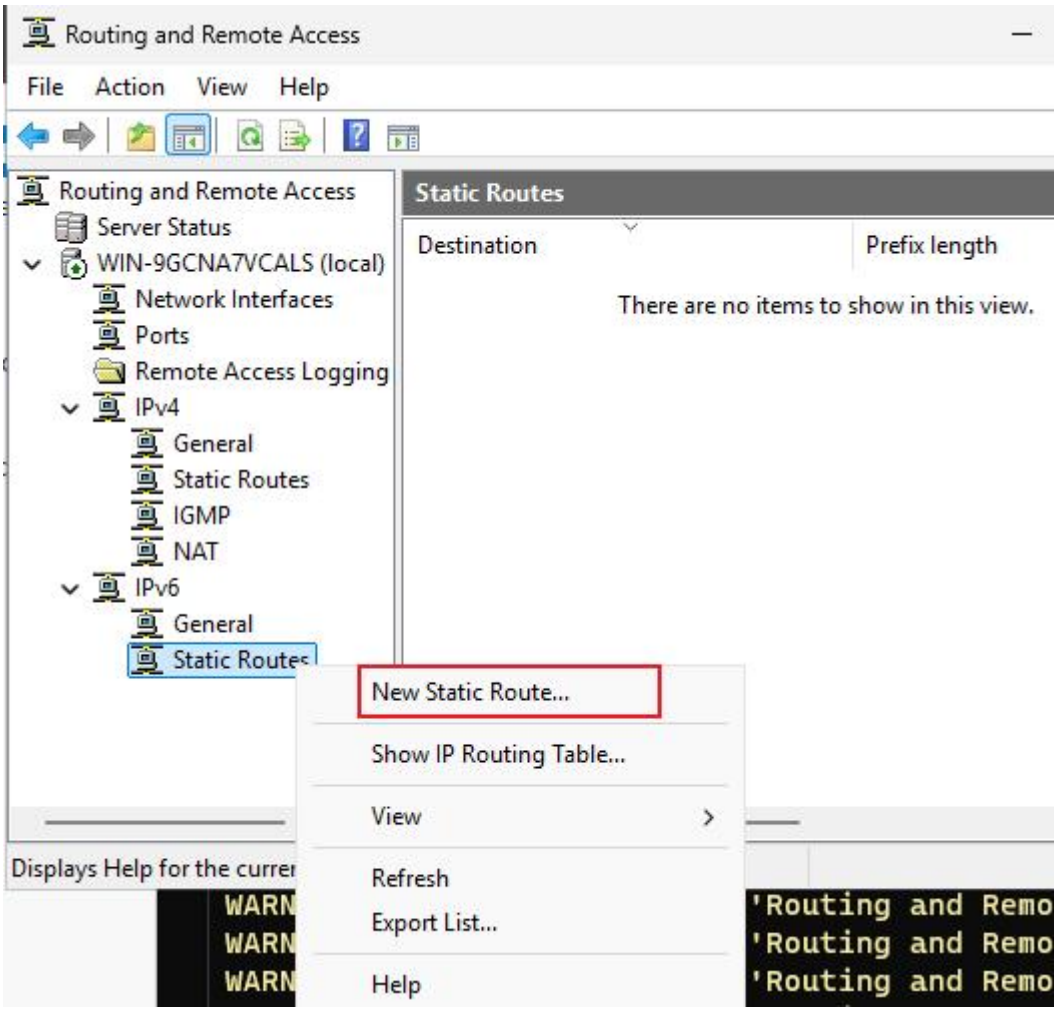


有时图形化开启会报错，需要用命令开启：

```
netsh ras set type ipv4rtrtype=LANANDDD ipv6rtrtype=LANANDD rastype=none
```

```
PS C:\Users\Administrator> netsh ras set type ipv4rtrtype=LANANDDD ipv6rtrtype=LANANDDD rastype=none
The Remote Access Service must be restarted for the changes to take effect.

PS C:\Users\Administrator>
```

然后添加 IPv6 路由指向 S2S 接口就行:



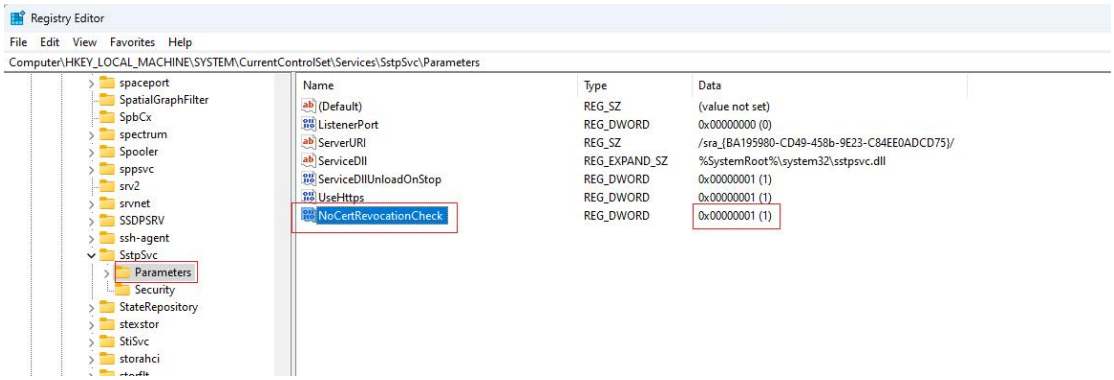
2、Remote Access:

•SSTP 关闭证书吊销验证:

修改注册表，不验证证书吊销列表

regedit

在 HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\SstpSvc\Parameters 下新建 DWORD(32 位) “NoCertRevocationCheck”，值为 1



七、adfs、wap：

•证书字段：

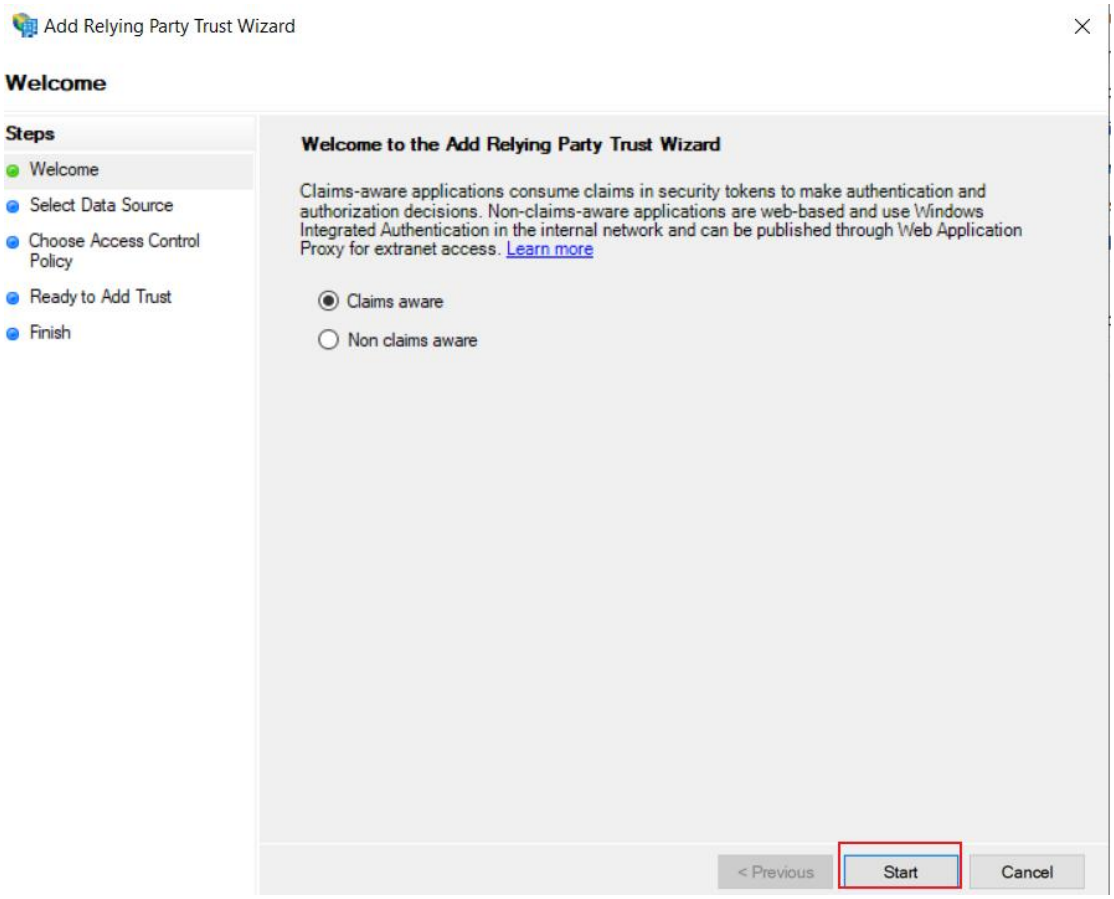
```
# 证书必须要有以下字段（可以全部写在一张证书中）：
ADFS：
1、enterpriseregistration.test.com
2、连接名称，例如：adfs.test.com

wap:
1、wap 必须要加上 wap 机器当前的主机名，例如：Server2.test.com
2、必须要包含 wap 代理域名的 外网 URL、内网 URL

workfolder:
1、workfolder 连接名称，例如：work.test.com

最后加上 *.域名
```

1、workfolder、adfs、wap：

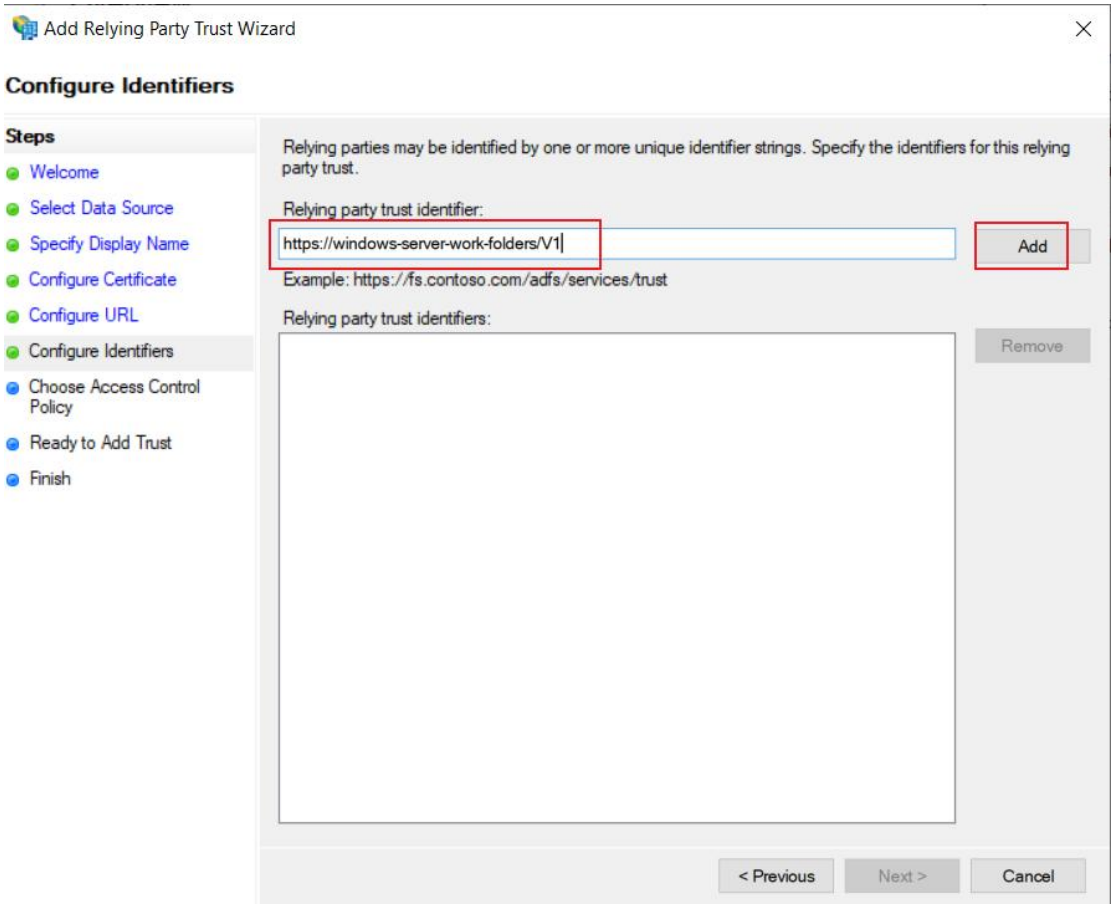
```
# ADFS 必须有两条 DNS 记录：
一条是：enterpriseregistration 例如：enterpriseregistration.test.com
一条是可以自定义的 adfs 连接域名 例如：adfs.test.com
都指向 ADFS 自己
# 证书也要包含这些字段
```

•配置 adfs：

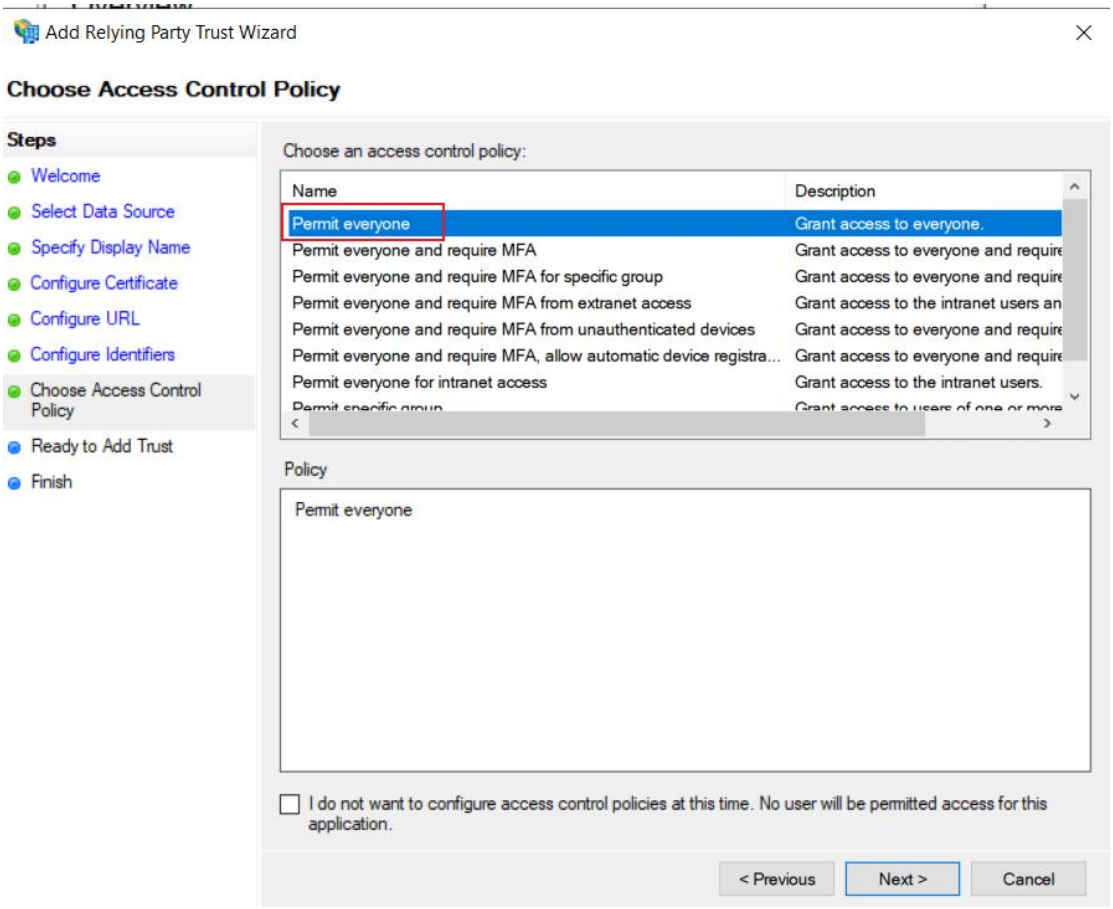
创建声明规则



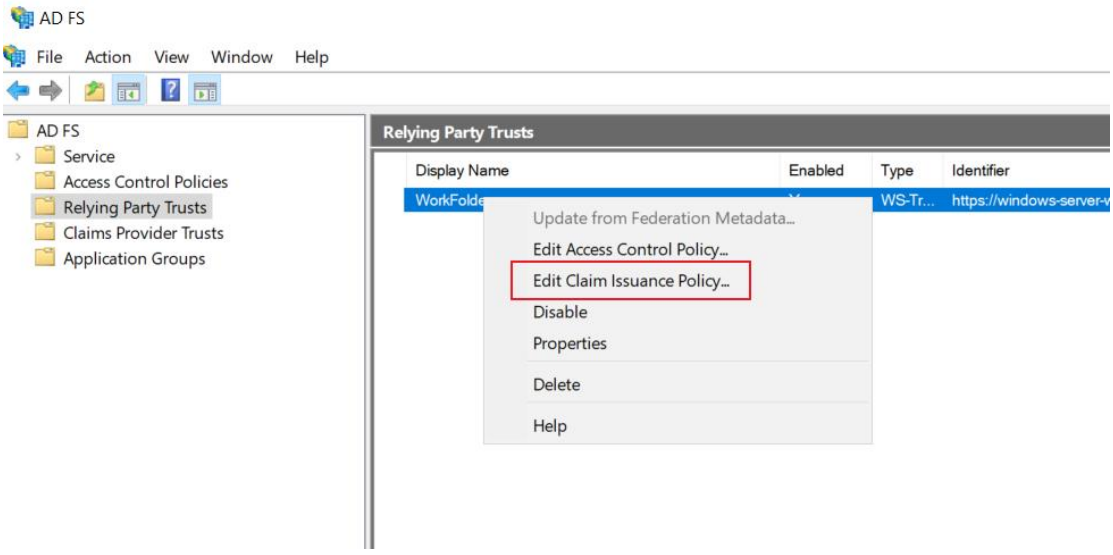
https://windows-server-work-folders/V1

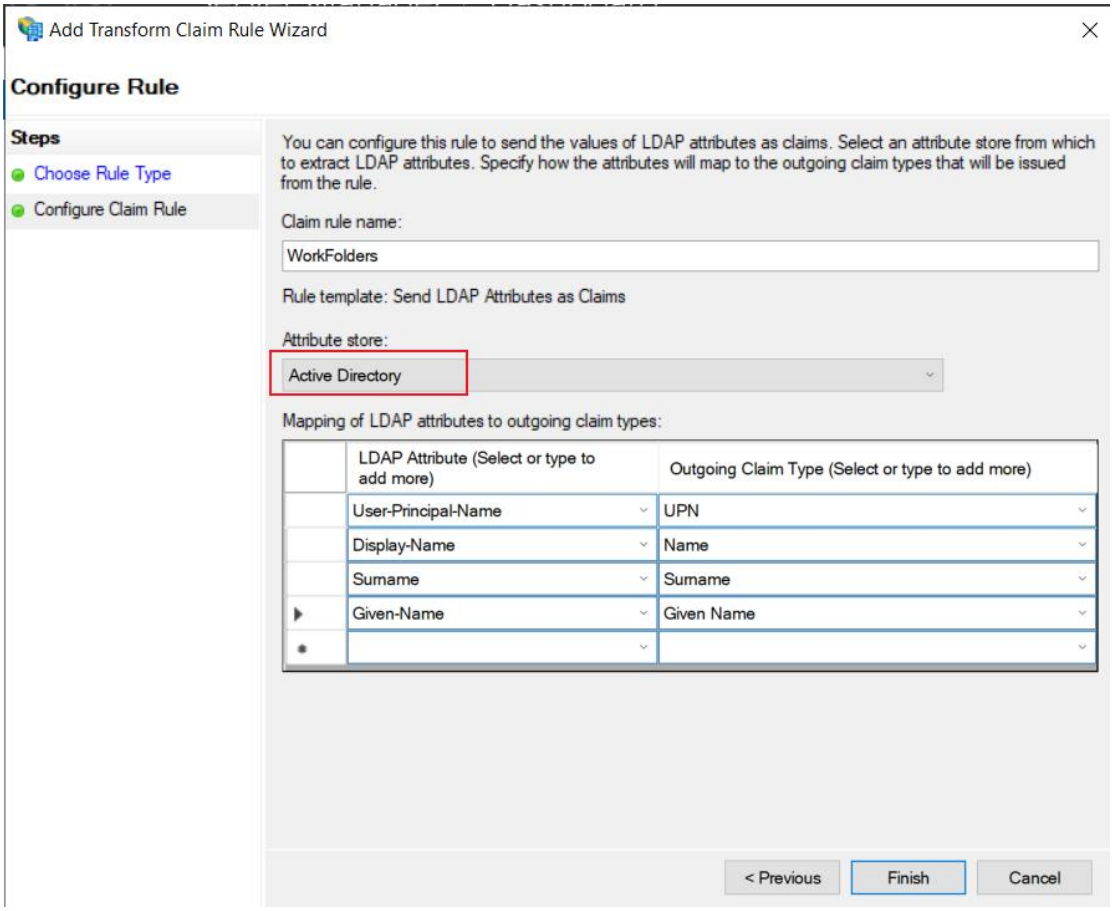


放行权限这里，everyone 只能放行本地的域，如果要放行对端域，需要用 Permit Specific group 来同时指定本地和对端


Add Relying Party Trust Wizard
Choose Access Control Policy
Steps
Welcome
Select Data Source
Specify Display Name
Configure Certificate
Configure URL
Configure Identifiers
Choose Access Control Policy
Ready to Add Trust
Finish
Choose an access control policy:
Name
Description
Permit everyone
Grant access to everyone.
Permit everyone and require MFA
Grant access to everyone and require
Permit everyone and require MFA for specific group
Grant access to everyone and require
Permit everyone and require MFA from extranet access
Grant access to the intranet users an
Permit everyone and require MFA from unauthenticated devices
Grant access to everyone and require
Permit everyone and require MFA, allow automatic device registra...
Grant access to everyone and require
Permit everyone for intranet access
Grant access to the intranet users.
Policy
Permit everyone
I do not want to configure access control policies at this time. No user will be permitted access for this application.
< Previous
Next >
Cancel

添加依赖方属性：


AD FS
File
Action
View
Window
Help
AD FS
Service
Access Control Policies
Relying Party Trusts
Claims Provider Trusts
Application Groups
Relying Party Trusts
Display Name
Enabled
Type
Identifier
WS-Tr...
https://windows-server-
Update from Federation Metadata...
Edit Access Control Policy...
Edit Claim Issuance Policy...
Disable
Properties
Delete
Help



•可以用命令导出，然后复制到 Server Core 导入：

```
(Get-AdfsRelyingPartyTrust).IssuanceTransformRules
```

```
PS C:\Users\Administrator.WSC2026> (Get-AdfsRelyingPartyTrust).IssuanceTransformRules
@RuleTemplate = "LdapClaims"
@RuleName = ""
c:[Type == "http://schemas.microsoft.com/ws/2008/06/identity/claims/windowsaccountname", Issuer == "AD AUTHORITY"]
 => issue(store = "Active Directory", types = ("http://schemas.xmlsoap.org/ws/2005/05/identity/claims/upn"), query = ";userPrincipalName;{0}", param = c.Value);
```

```
# 在 adfs 上复制导出的内容：
Set-AdfsRelyingPartyTrust -TargetName "WorkFolders" -IssuanceTransformRules '@RuleName="WorkFolders"
c:[Type=="http://schemas.microsoft.com/ws/2008/06/identity/claims/windowsaccountname",Issuer=="AD AUTHORITY"]=>issue(store="Active Directory",types=("http://schemas.xmlsoap.org/ws/2005/05/identity/claims/upn","http://schemas.xmlsoap.org/ws/2005/05/identity/claims/name","http://schemas.xmlsoap.org/ws/2005/05/identity/claims/surname","http://schemas.xmlsoap.org/ws/2005/05/identity/claims/givenname"),query=";userPrincipalName,displayName,sn,givenName;{0}",param=c.Value);'
```

•授权 adfs、开启测试页面：

```
Grant--AdfsApplicationPermission --ServerRoleIdentifier "https://windows--server--work-folders/V1" --AllowAllRegisteredClients --ScopeNames openid,profile
```

```
Set-AdfsProperties -EnableIdPInitiatedSignonPage $true

# 如果内网客户端也要使用 adfs 认证（关闭它的 windows 身份认证）：
Set-AdfsGlobalAuthenticationPolicy -PrimaryIntranetAuthenticationProvider @("FormsAuthentication","MicrosoftPassportAuthentication")
```

•配置 wap：

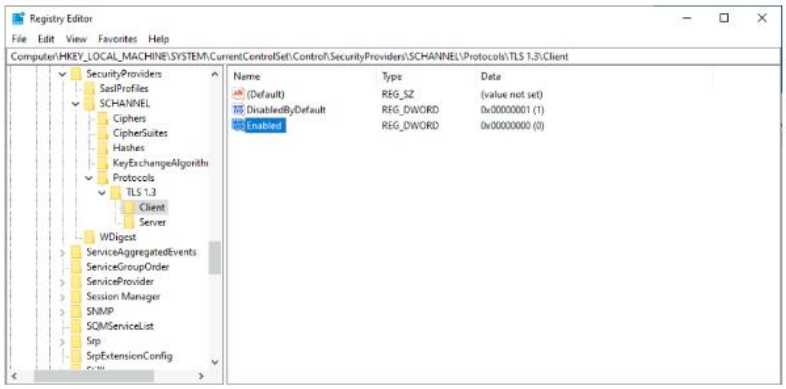
•禁用 TLS 1.3：

Windows 2022 的新功能之一是支持 TLS 1.3，这就是罪魁祸首。
ADFS 似乎无法与 TLS 1.3 正常工作。

要在 WAP 服务器上禁用 TLS 1.3，请添加以下注册表项：

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\TLS 1.3]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\TLS 1.3\Client]
"DisabledByDefault"=dword:00000001
"Enabled"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\TLS 1.3\Server]
```

如下图所示：

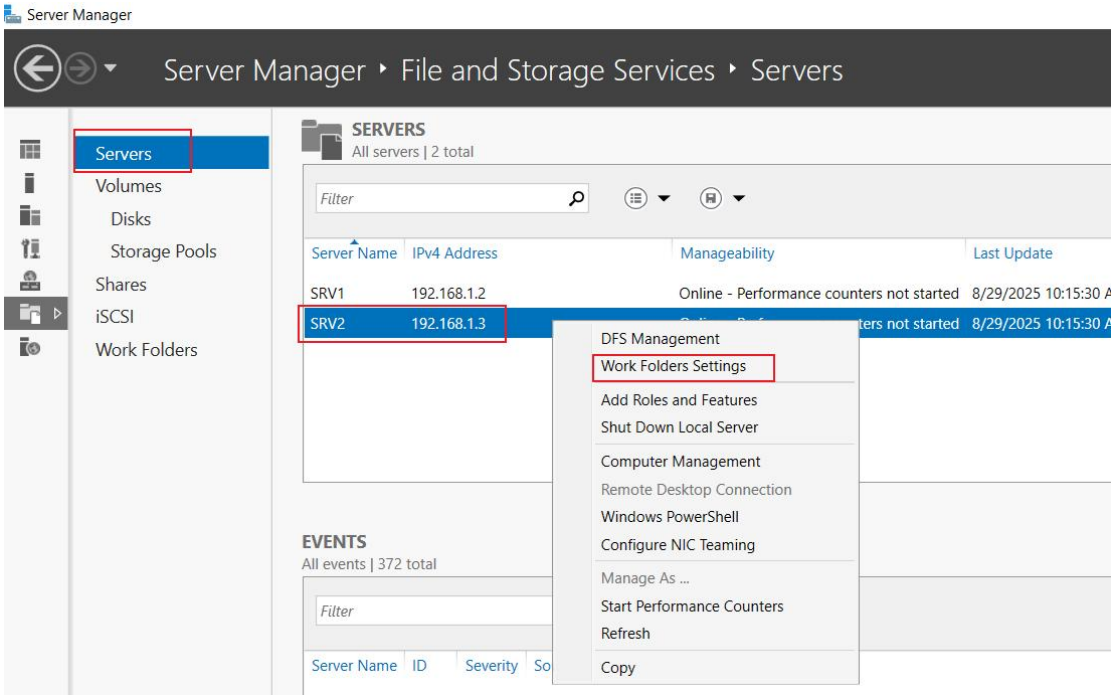


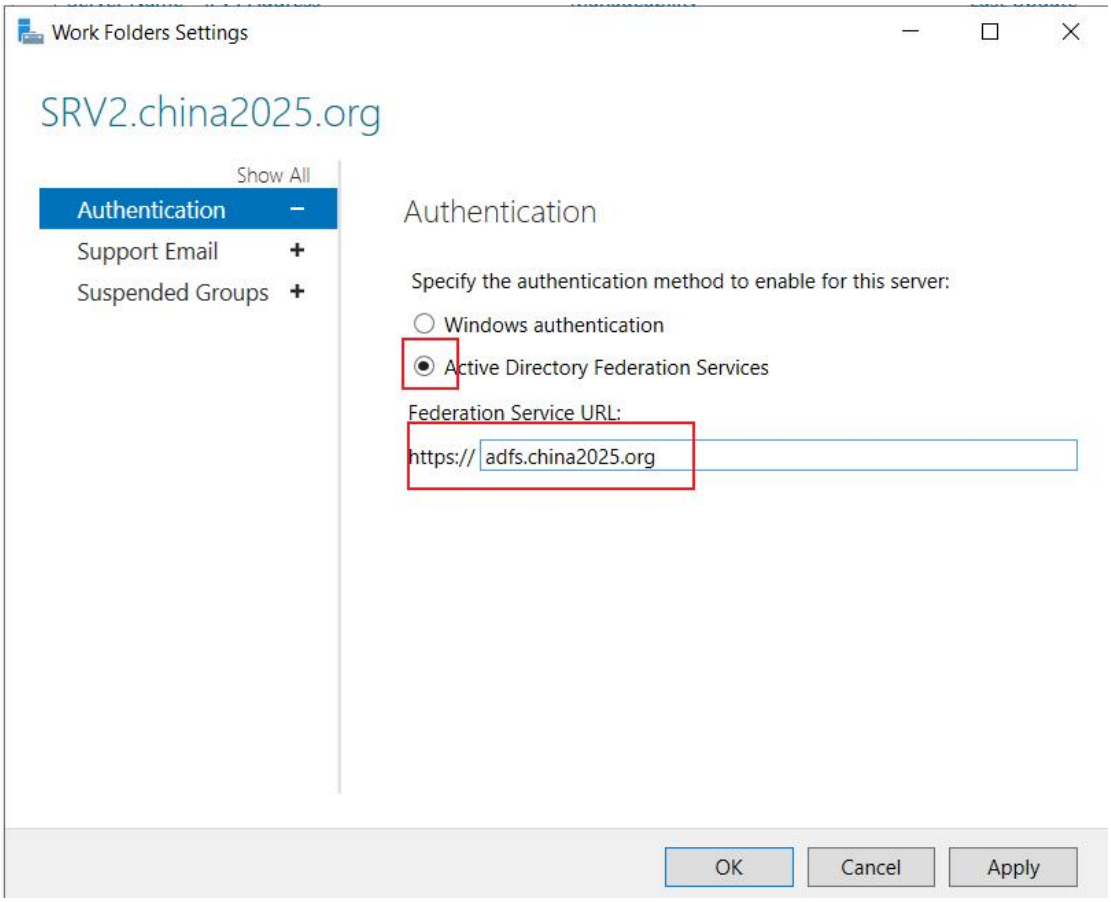
添加这些注册表项后，WAP 代理配置向导成功完成。

•workfolder：

•注意，workfolder 必须要有一个 443 站点绑定了 443 证书，否则无法使用

设置 workfolder 使用 adfs 进行认证：





点击 OK

2、Web、adfs、wap：

•配置 adfs：

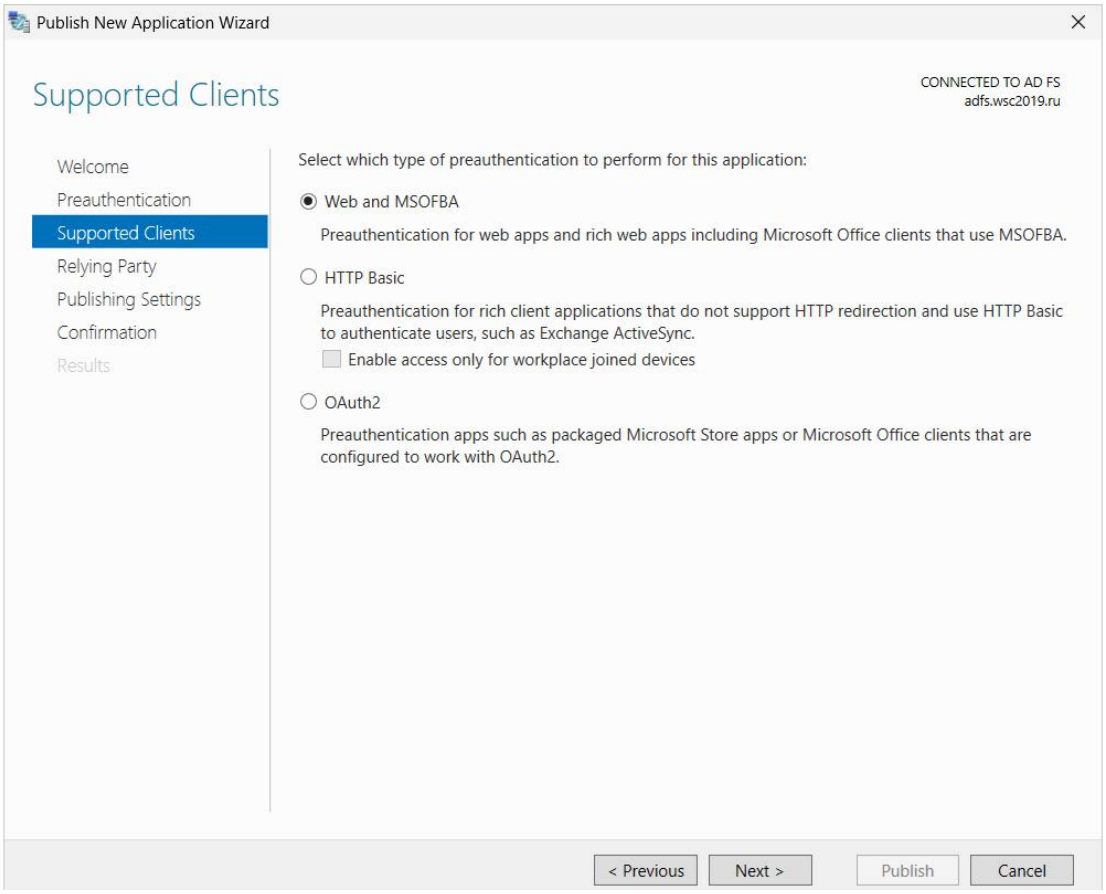
创建声明感知：
这里的 URL 要写外网的

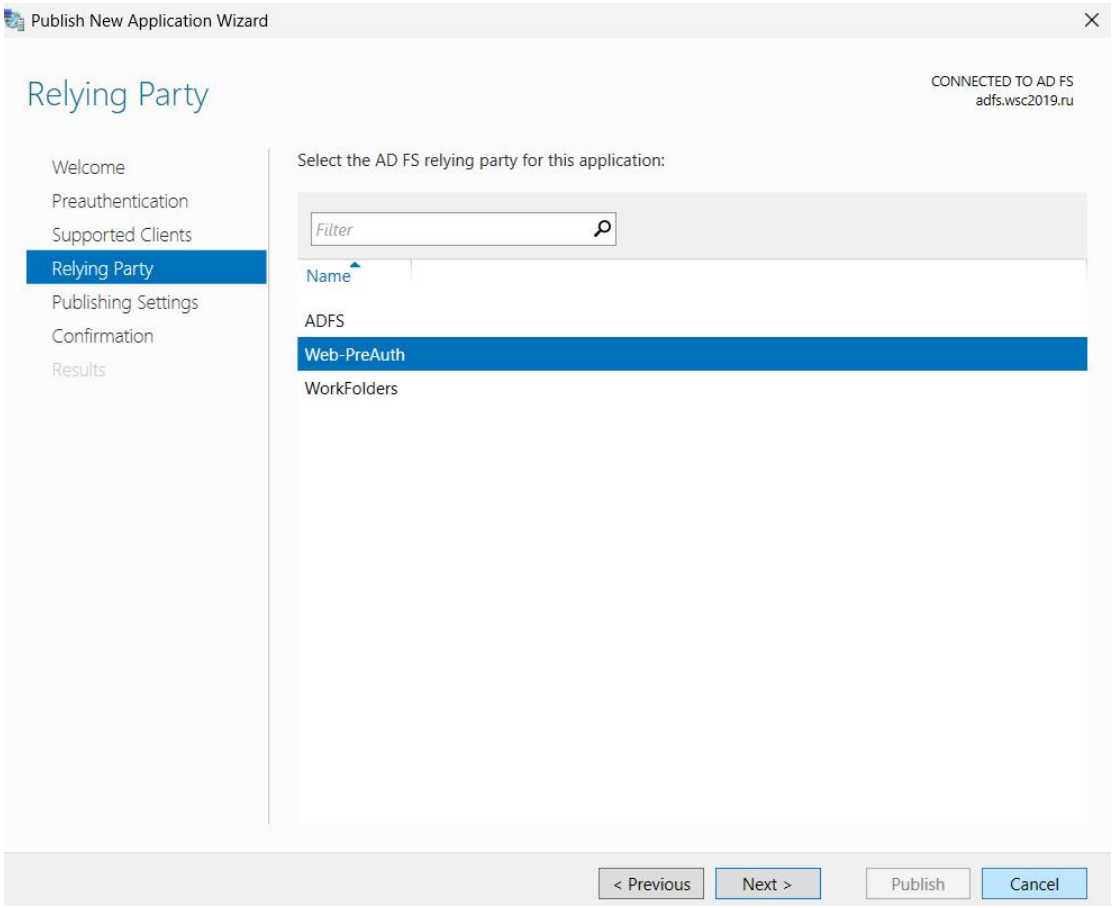
```
Add-AdfsRelyingPartyTrust -Name "Web-PreAuth" -Identifier "https://external.cn2026.mo/" -WSFedEndpoint "https://external.cn2026.mo/" -AccessControlPolicyName "Permit everyone"

# 如果要指定组：
Add-AdfsRelyingPartyTrust -Name "Web-PreAuth" -Identifier "https://external.cn2026.mo/" -WSFedEndpoint "https://external.cn2026.mo/" -AccessControlPolicyName "Permit specific group" -AccessControlPolicyParameters "CN2026\WebUsers"
```

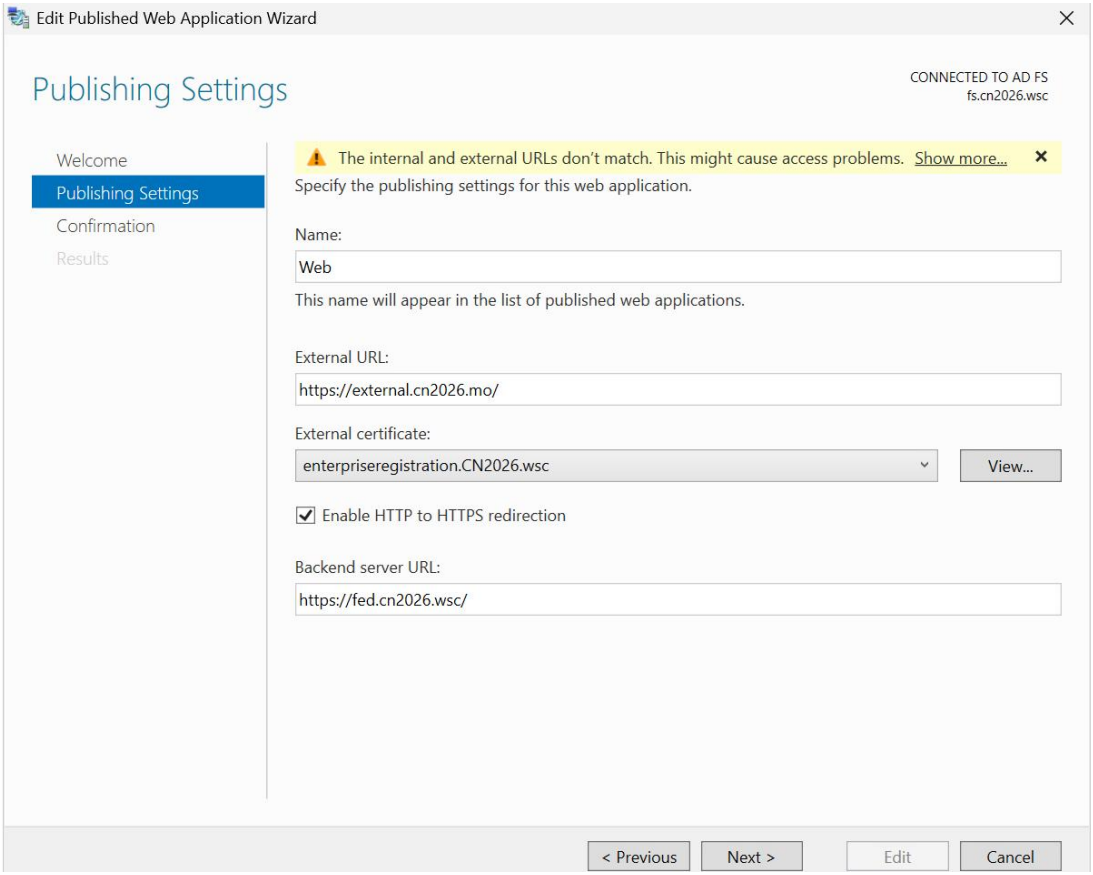
测试过了，虽然都是声明感知，但是 web 这里不需要像 wap 一样配置 UPN 那些字段，可以直接代理

```
Grant--AdfsApplicationPermission --ServerRoleIdentifier "https://external.cn2026.mo/" --AllowAllRegisteredClients --ScopeNames openid,profile
```

•配置 WAP


Publish New Application Wizard
Supported Clients
CONNECTED TO AD FS
adfs.wsc2019.ru
Welcome
Preauthentication
Supported Clients
Relying Party
Publishing Settings
Confirmation
Results
Select which type of preauthentication to perform for this application:
Web and MSOFBA
Preauthentication for web apps and rich web apps including Microsoft Office clients that use MSOFBA.
HTTP Basic
Preauthentication for rich client applications that do not support HTTP redirection and use HTTP Basic to authenticate users, such as Exchange ActiveSync.
Enable access only for workplace joined devices
OAuth2
Preauthentication apps such as packaged Microsoft Store apps or Microsoft Office clients that are configured to work with OAuth2.
< Previous
Next >
Publish
Cancel


Publish New Application Wizard
Relying Party
CONNECTED TO AD FS
adfs.wsc2019.ru
Welcome
Preauthentication
Supported Clients
Relying Party
Publishing Settings
Confirmation
Results
Select the AD FS relying party for this application:
Filter
Name
ADFS
Web-PreAuth
WorkFolders
< Previous
Next >
Publish
Cancel



八、RDS：

•注意：RDC 不能装在 ServerCore 上，会报错

•图形化有时无法创建 rd 会话集合，需要用命令创建：

用命令创建的合集默认允许访问的用户组：Domain Users

```
# 需要使用命令创建：
New-RDSessionCollection -ConnectionBroker server1.shanghai.org -CollectionName RDS -SessionHost Server2.shanghai.org,Server3.shanghai.org
```

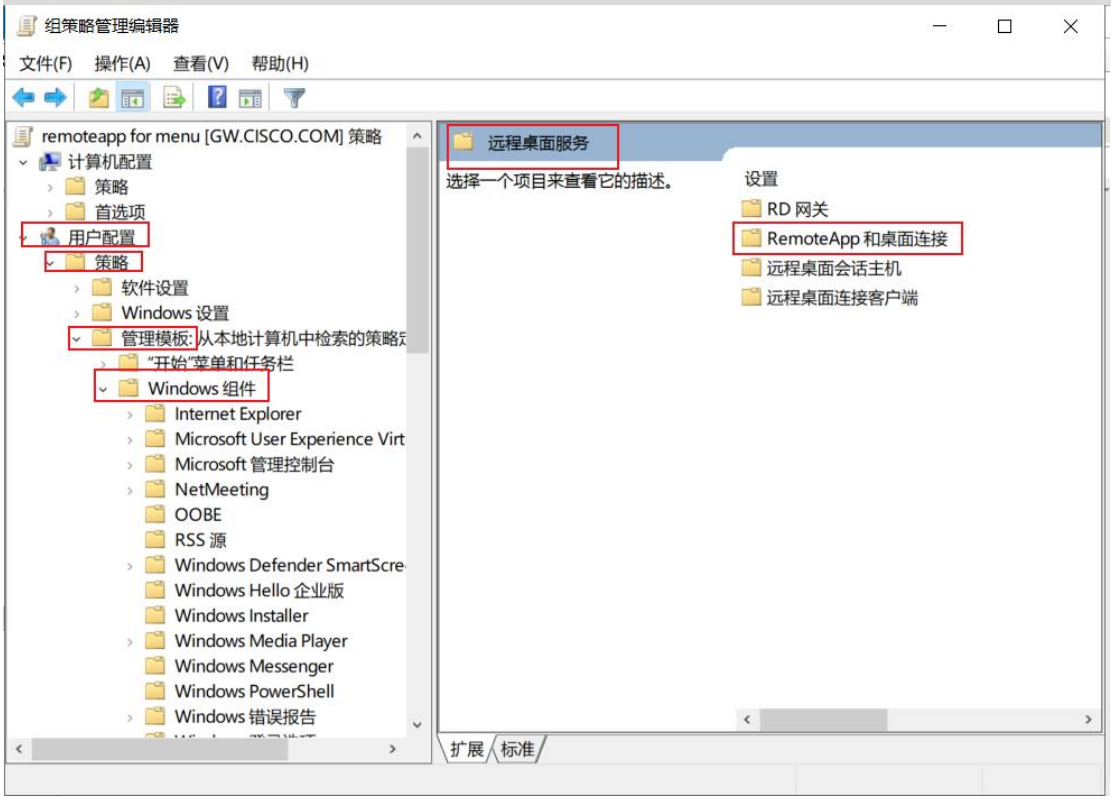
```
PS C:\Users\Administrator> Get-RDServer -ConnectionBroker Server1.shanghai.org

Server                                             Roles
------                                             -----
SERVER1.SHANGHAI.ORG                               {RDS-CONNECTION-BROKER}
Server4.shanghai.org                               {RDS-WEB-ACCESS, RDS-GATEWAY}
Server3.shanghai.org                               {RDS-RD-SERVER}
Server2.shanghai.org                               {RDS-RD-SERVER}


PS C:\Users\Administrator>
```

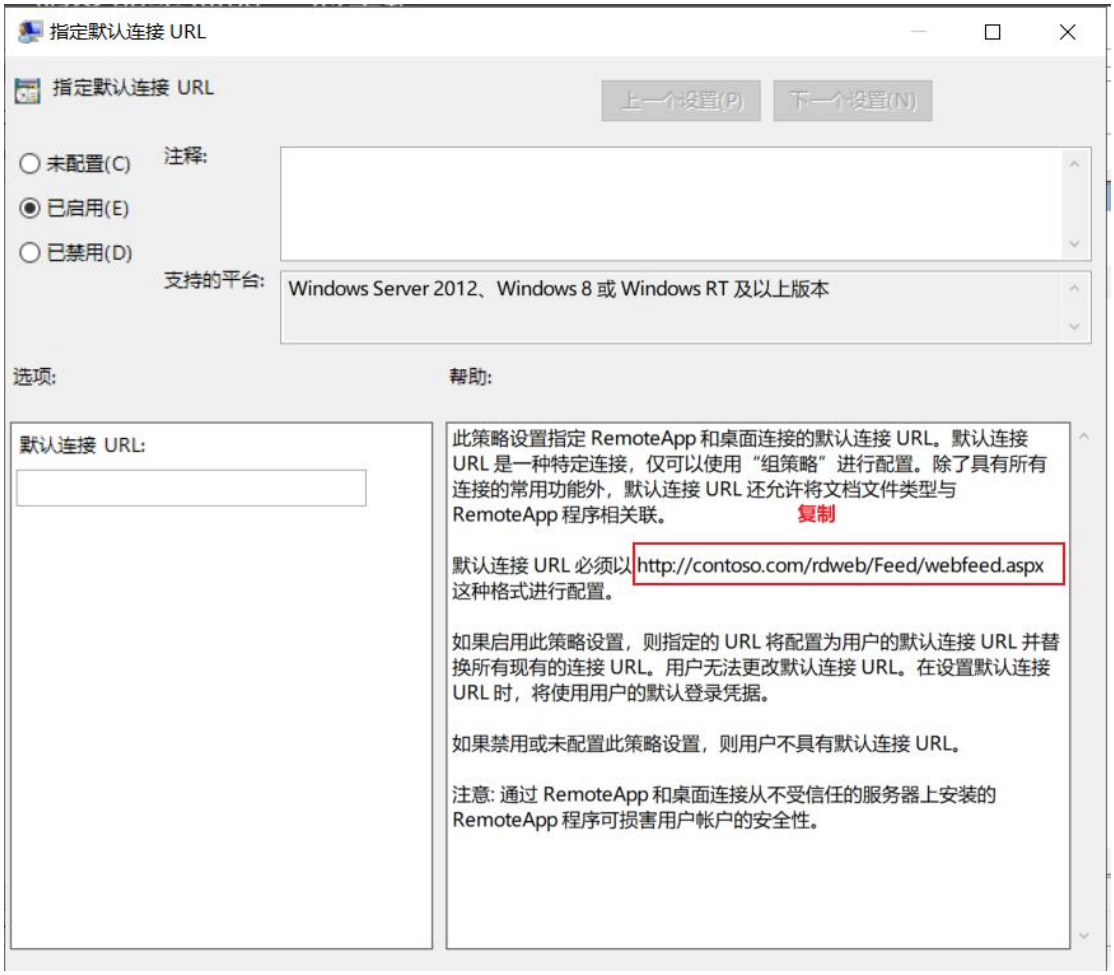
•如果有两个域，配置 RDS 后，可以设置允许两个域的用户组访问（必须命令行设置、需要先双向信任）：

```
Set-RDSessionCollectionConfiguration -CollectionName "RemoteApps" -UserGroup @("PARIS\Domain Users","LYON\Domain Users")
```

•自动连接 RemoteApp：

```
# 路径
用户配置 > 策略 > 管理模板 > Windows 组件 > 远程左面服务 > RemoteApp 和桌面连接 > 指定默认连接 URL
```

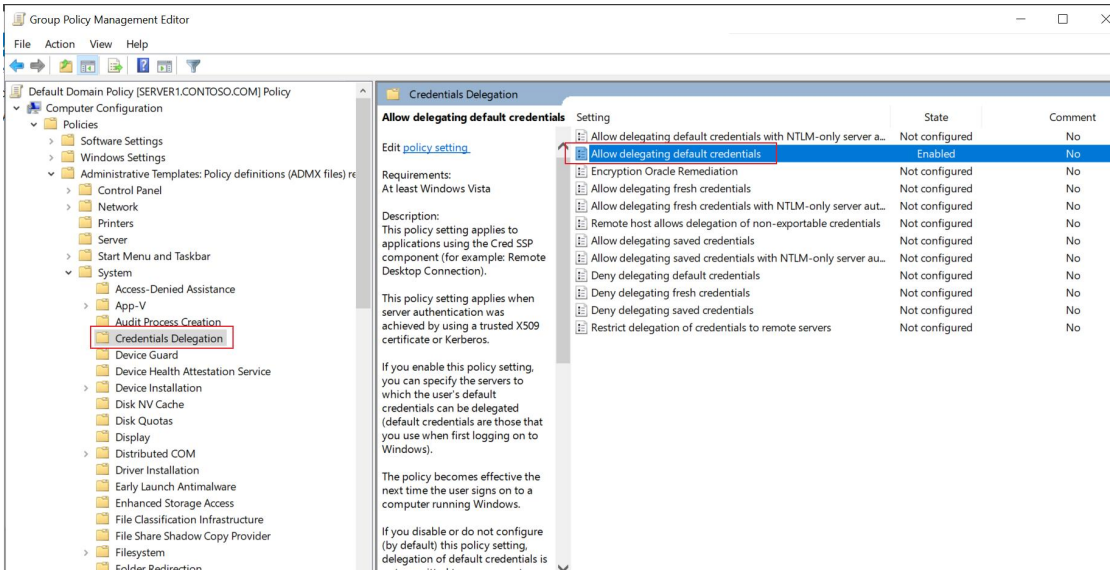


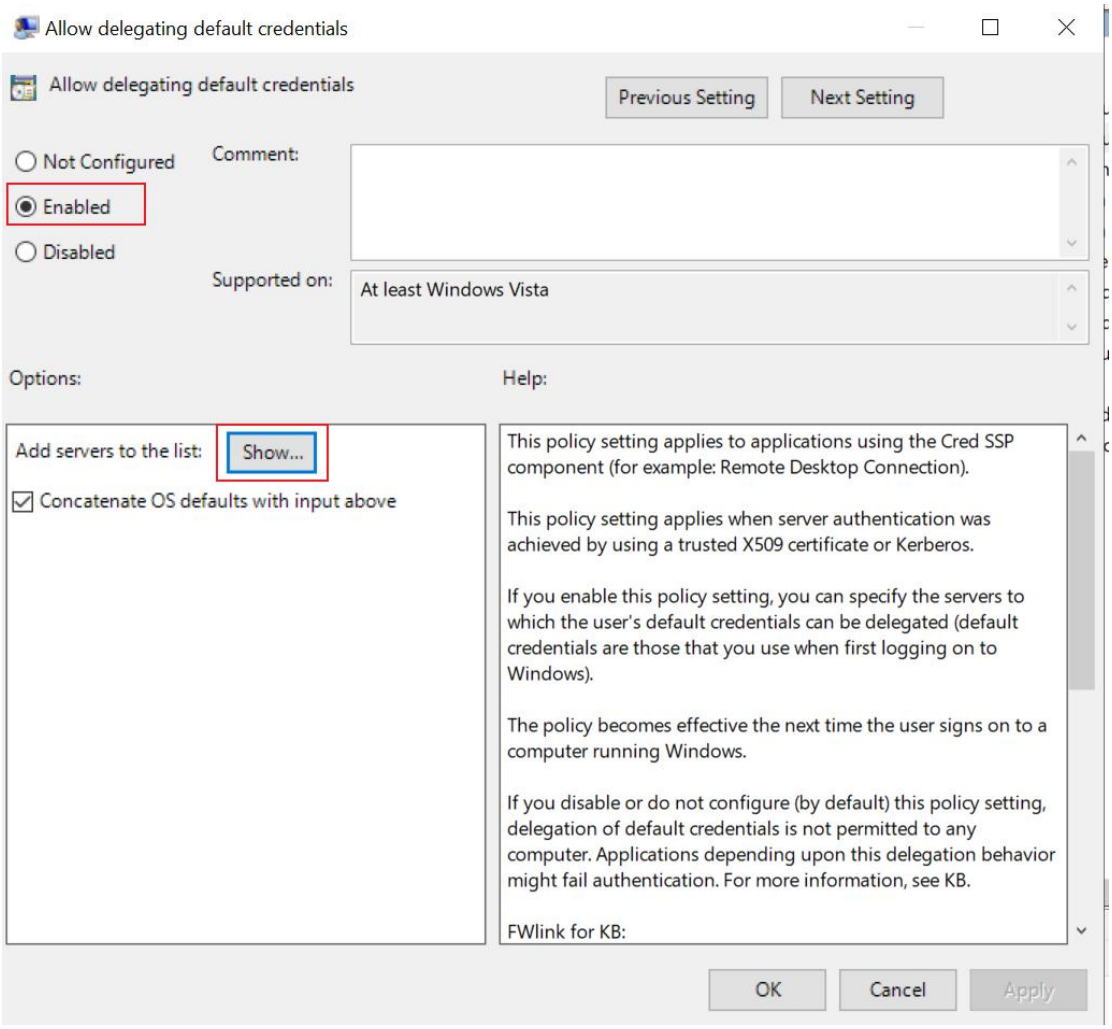


•（SSO，只对域内客户端永久有效，只有域内客户端、服务器能够生效）可以让它不弹身份认证框：

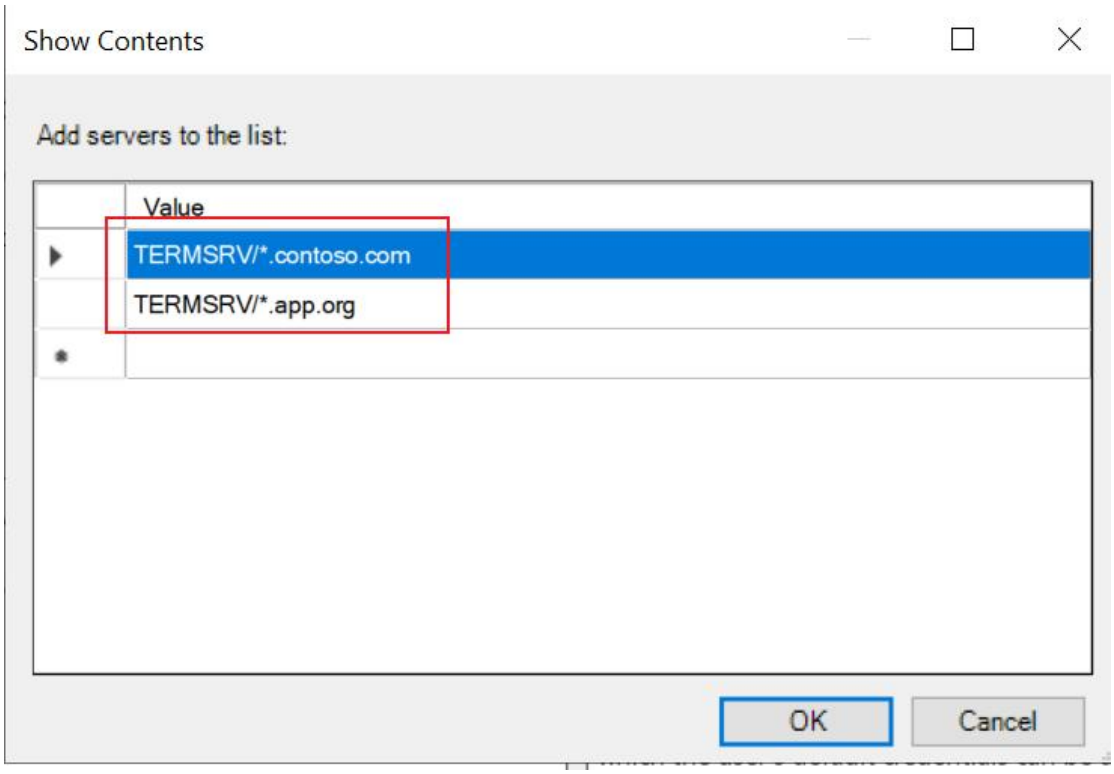
在域控上编辑组策略：

```
# 组策略路径：
Computer Configuration - Administrative Templates - System - Credentials Delegation - "Allow delegating default credentials"
## 中文路径：
计算机配置 - 管理模板 - 系统 - 凭据委派 - 允许委派默认凭证
```

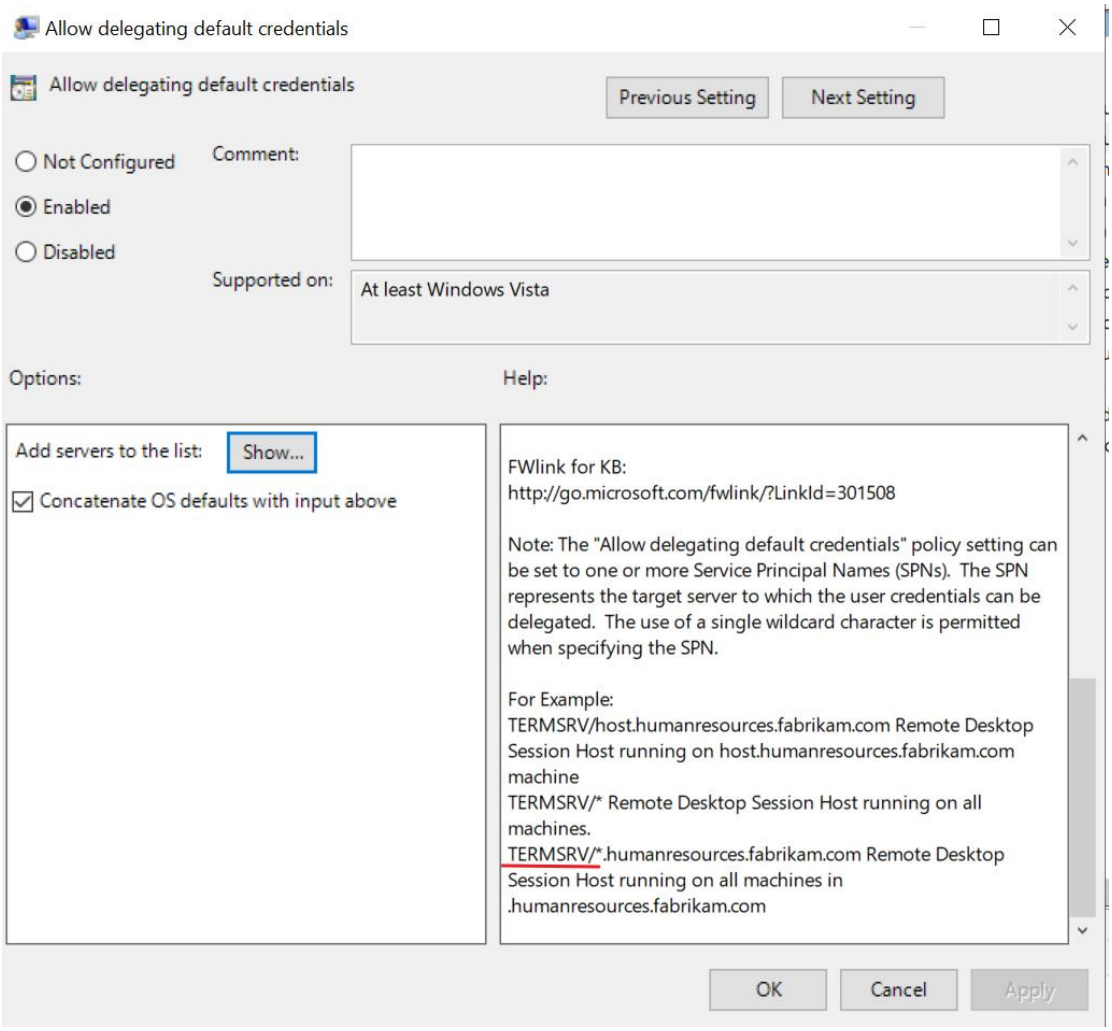




这里写 RDweb 和 RD 网关的域名，前面需要用 * 号，因为有时 RD 会话主机与 RD 网关不在同一台机器，就要再写上 RD 会话主机的域名，比较麻烦，所以可以直接写 *（不能直接整个写 *，不写任何域名）


Show Contents
Add servers to the list:
Value
TERMSRV/*.contoso.com
TERMSRV/*.app.org
OK
Cancel

这个前缀在说明中能抄：

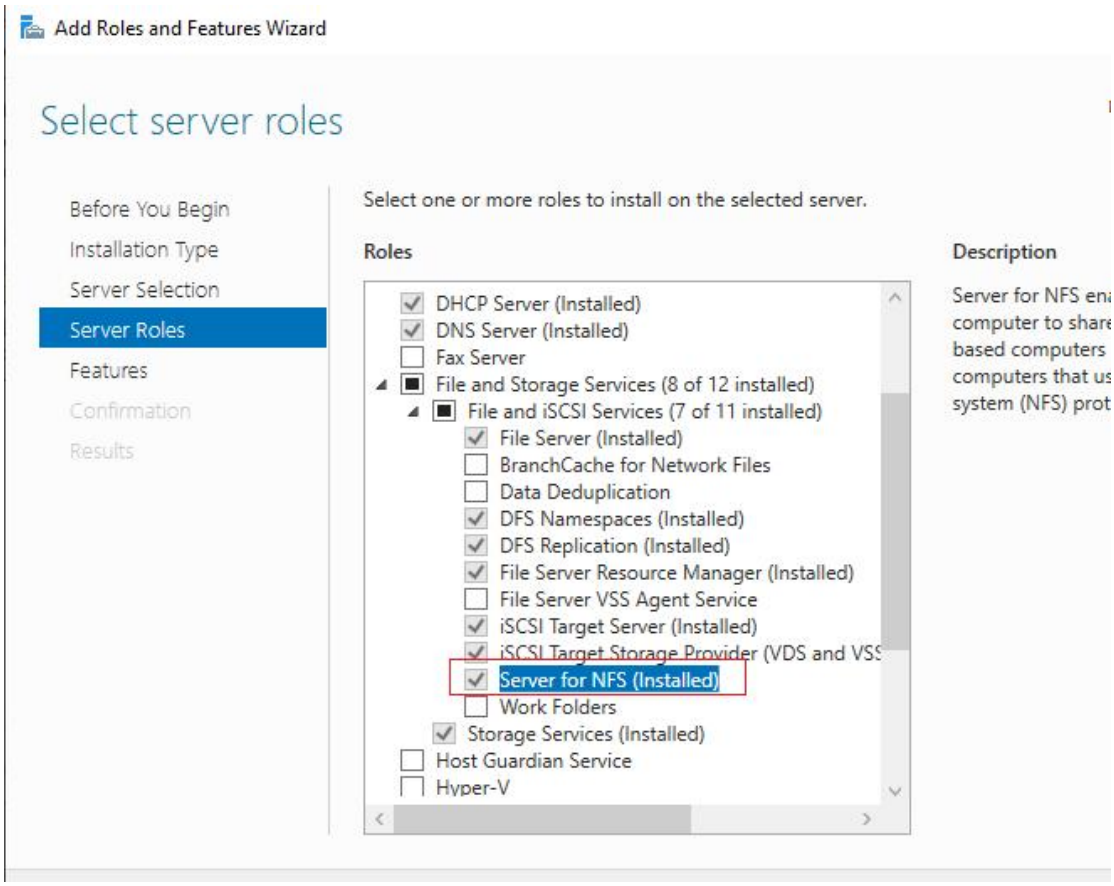


•客户端同步组策略后 必须要重启机器才会生效，logoff 不行（对于 server 和客户端都能够生效）

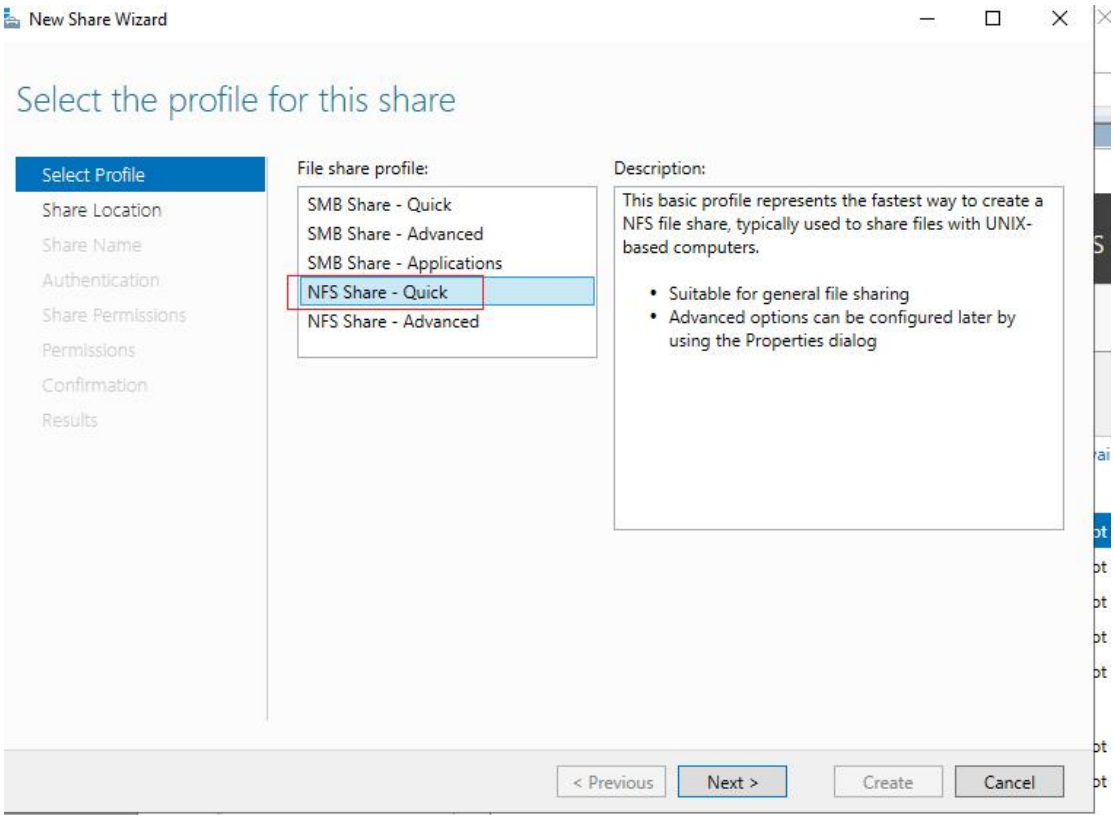
九、磁盘：

1、NFS 共享：

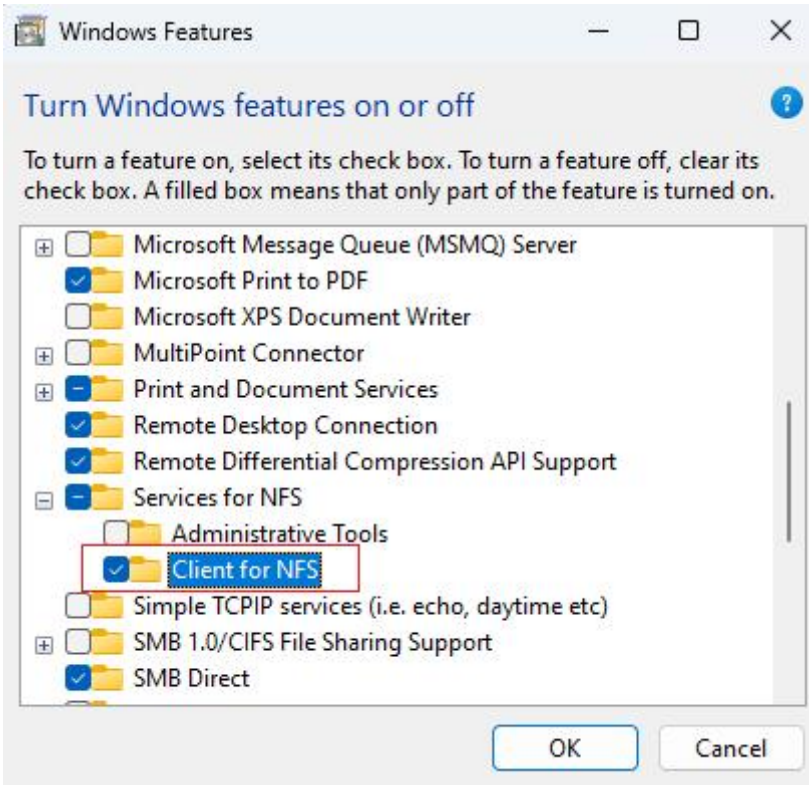
服务端：


Add Roles and Features Wizard
Select server roles
Before You Begin
Installation Type
Server Selection
Server Roles
Features
Confirmation
Results
Select one or more roles to install on the selected server.
Roles
DHCP Server (Installed)
DNS Server (Installed)
Fax Server
File and Storage Services (8 of 12 installed)
File and iSCSI Services (7 of 11 installed)
File Server (Installed)
BranchCache for Network Files
Data Deduplication
DFS Namespaces (Installed)
DFS Replication (Installed)
File Server Resource Manager (Installed)
File Server VSS Agent Service
iSCSI Target Server (Installed)
Server for NFS (Installed)
Work Folders
Storage Services (Installed)
Host Guardian Service
Hyper-V
Description

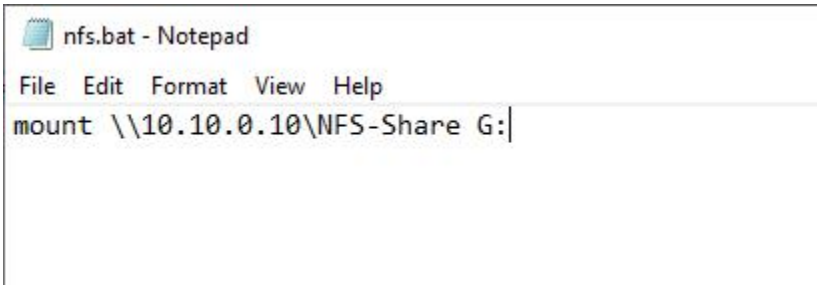
创建 NFS 共享：


New Share Wizard
Select the profile for this share
Select Profile
Share Location
Share Name
Authentication
Share Permissions
Permissions
Confirmation
Results
File share profile:
SMB Share - Quick
SMB Share - Advanced
SMB Share - Applications
NFS Share - Quick
NFS Share - Advanced
Description:
This basic profile represents the fastest way to create a NFS file share, typically used to share files with UNIX-based computers.
Suitable for general file sharing
Advanced options can be configured later by using the Properties dialog
< Previous
Next >
Create
Cancel

windows 要开启 NFS 客户端 功能：



•配置 自动挂载：



编写组策略：
将该脚本放到 public 的 desktop，开启自动运行这个脚本

2、iSCSI：

十、WSL：

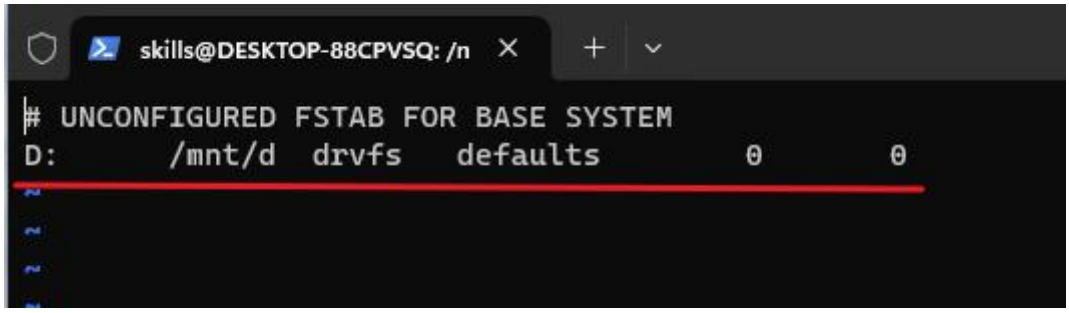
1、预安装的 wsl 可能无法正常使用：

修改 locale，否则 root 用户无法正常使用命令：
vi /etc/default/locale

```
LANGUAGE=en_US.UTF-8
LC_CTYPE="en.US.UTF-8"
```

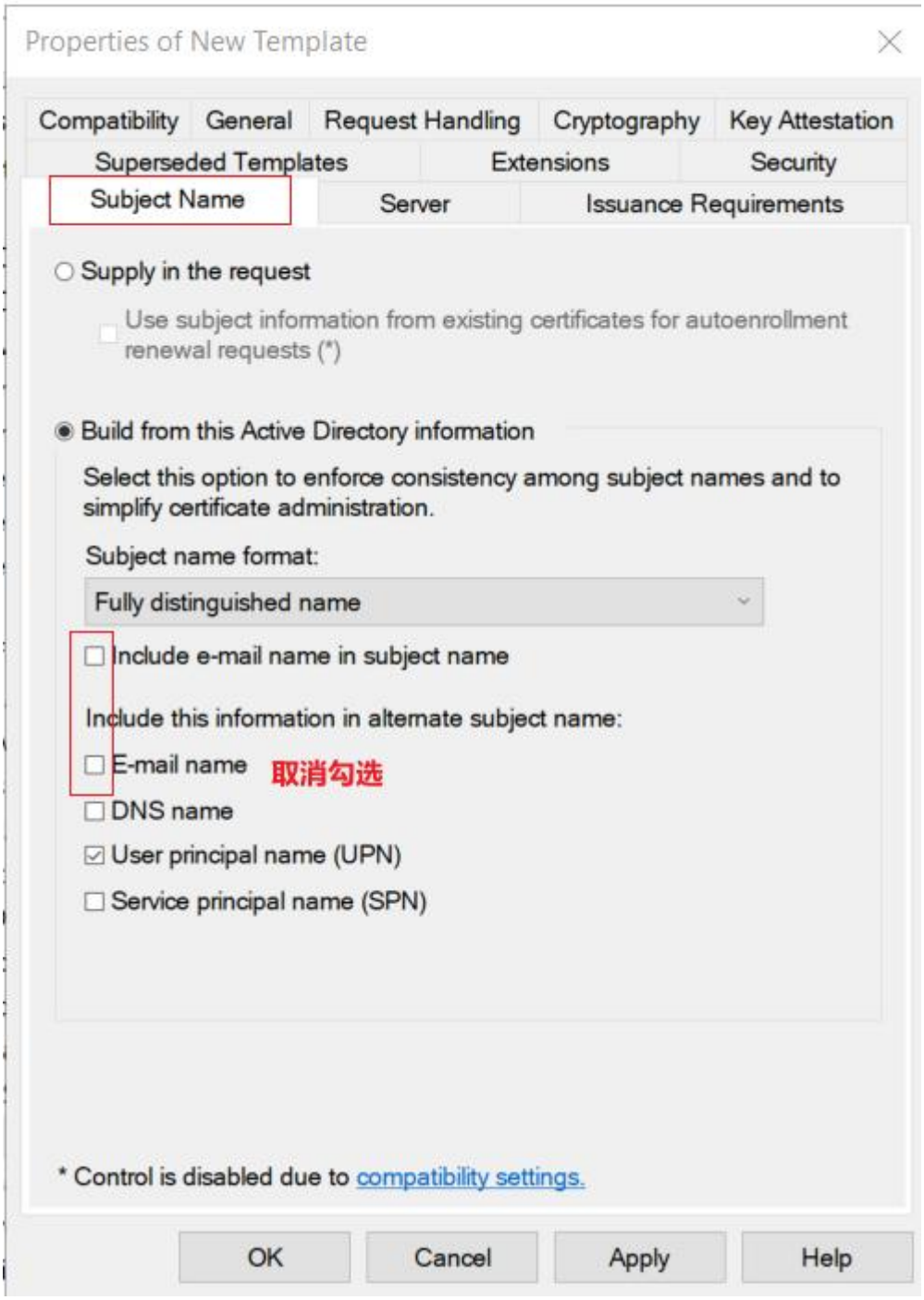
然后 wsl --shutdown 重新再打开 wsl 即可

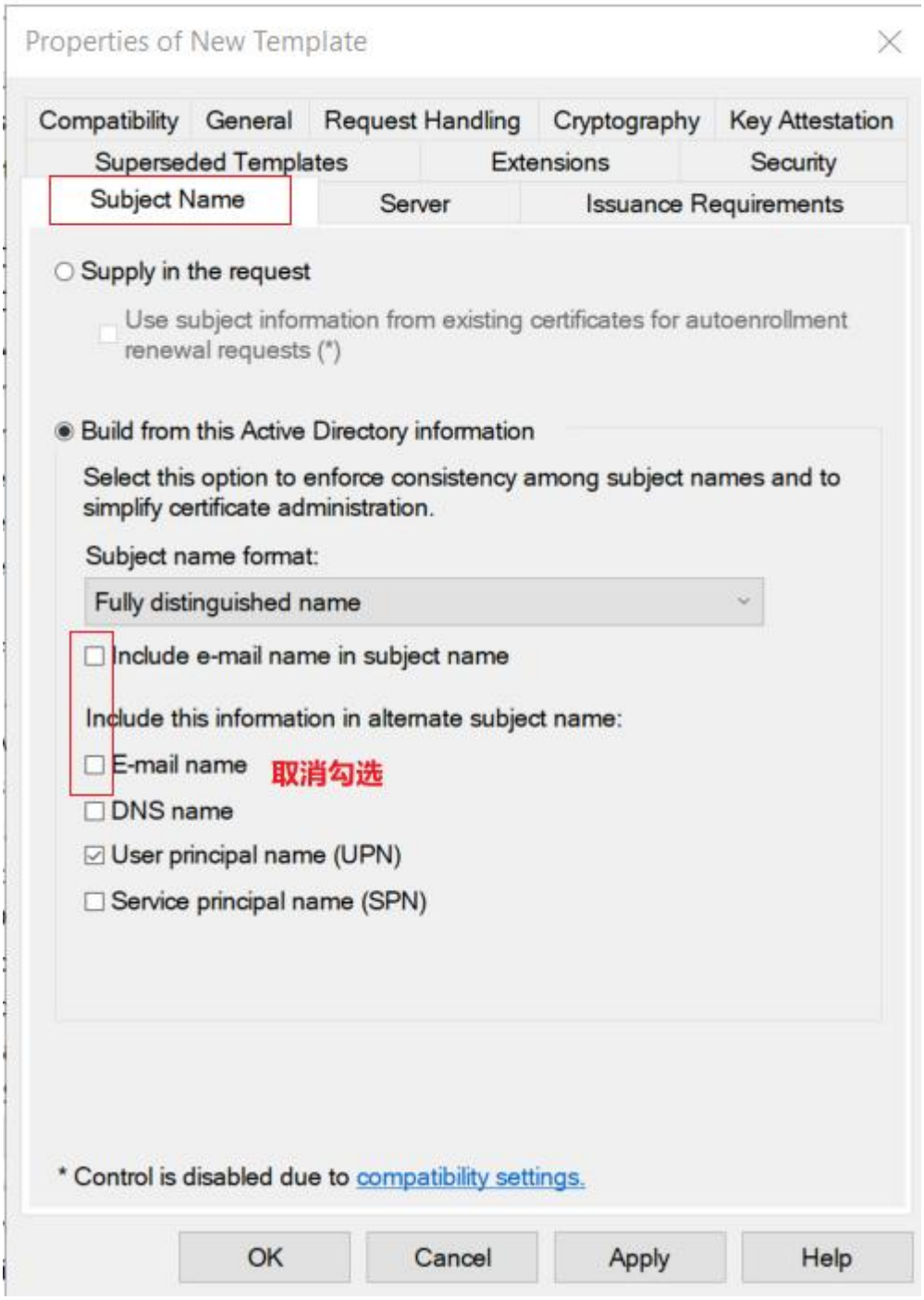
挂载镜像：
先挂载 debian 镜像到虚拟机上
格式：drvfs

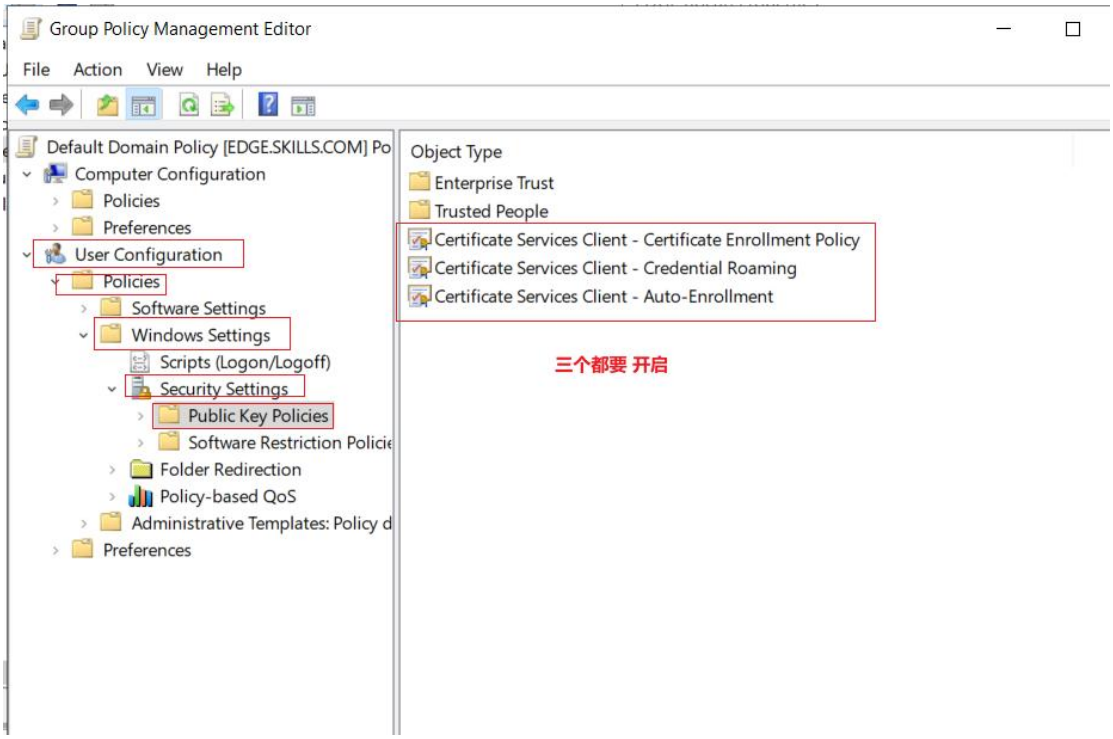


十一、组策略：

1、自动颁发用户证书：


Properties of New Template
Compatibility
General
Request Handling
Cryptography
Key Attestation
Superseded Templates
Extensions
Security
Subject Name
Server
Issuance Requirements
Supply in the request
Use subject information from existing certificates for autoenrollment renewal requests (*)
Build from this Active Directory information
Select this option to enforce consistency among subject names and to simplify certificate administration.
Subject name format:
Fully distinguished name
Include e-mail name in subject name
Include this information in alternate subject name:
E-mail name
取消勾选
DNS name
User principal name (UPN)
Service principal name (SPN)
* Control is disabled due to compatibility settings.
OK
Cancel
Apply
Help


Properties of New Template
Compatibility
General
Request Handling
Cryptography
Key Attestation
Superseded Templates
Extensions
Security
Subject Name
Server
Issuance Requirements
Supply in the request
Use subject information from existing certificates for autoenrollment renewal requests (*)
Build from this Active Directory information
Select this option to enforce consistency among subject names and to simplify certificate administration.
Subject name format:
Fully distinguished name
Include e-mail name in subject name
Include this information in alternate subject name:
E-mail name
取消勾选
DNS name
User principal name (UPN)
Service principal name (SPN)
* Control is disabled due to compatibility settings.
OK
Cancel
Apply
Help



十二：Ansible：

1、IIS：

```
# 可以使用的模块：
win_iis_website  //只能创建 http 站点
microsoft.iis.website  //可以同时创建 80 和 443 站点
win_file  //state: directory 幂等创建目录；state: touch 幂等创建文件
win_lineinfile  //幂等将内容写入文件
```

（1）同时创建 80 和 443 站点：

这里写入 index.html，可以用模块：win_copy: content

```yaml
---
- name: web
  hosts: web-srv.shanghai.local
  gather_facts: false
  tasks:
    - name:
      include_vars:
        file: /opt/ansible/website.yml
        name: web

    - name:
      debug:
        var: web.website[1]

    - name: Create WebSite Root Directory
      win_file:
        path: "C:\\{{ item.domain_prefix }}"
        state: directory
      loop: "{{ web.website }}"
      register: dir

    - name: Create WebSite Index
      win_file:
        path: "C:\\{{ item.domain_prefix }}\\index.html"
        state: touch
      loop: "{{ web.website }}"
      when: dir.changed

    - name: Create WebSite Index
      win_lineinfile:
        path: "C:\\{{ item.domain_prefix }}\\index.html"
        line: "{{ item.content }}"
      loop: "{{ web.website }}"

    - name: Create a website with two bindings, set all values and start it
      microsoft.iis.website:
        name: "{{ item.domain_prefix }}"
        state: started
        physical_path: "C:\\{{ item.domain_prefix }}"
        application_pool: DefaultAppPool
        bindings:
          set:
            - ip: '*'
              port: 80
              hostname: "{{ item.domain_prefix }}.shanghai.local"
            - ip: '*'
              port: 443
              hostname: "{{ item.domain_prefix }}.shanghai.local"
              protocol: https
              #use_sni: true
              #use_ccs: false
              certificate_hash: c001ab7743f7385bceb2339fe1105e50026e657d
              certificate_store_name: MY
      loop: "{{ web.website }}"

    - name: Stop Default WebSite
      microsoft.iis.website:
        name: "Default Web Site"
        state: Stopped
```

这里的 certificate_hash 就是证书指纹
能同时配置 http 和 https 是因为这里 bindings 定义了两个端口（ansible-doc 中有模板）

(2) 创建 Web（复杂需求）：

Playbook: web.yml

- Create /data/ansible/web.yml to build the IIS web sites of the web group.
  - The sites, bindings, paths and contents are read from /data/ansible/files/websites.yml.
  - Each site is served over HTTPS with the certificate of the company, and the certificate is requested from the issuing CA by the playbook itself.
  - A site that already exists with the correct settings must not be recreated.

还没全部完成，大体的框架是做出来了

```yaml
---
- name: web
  hosts: web
  gather_facts: false
  tasks:
    - name: Read yml
      set_fact:
        name: "{{ lookup('file','/data/ansible/files/websites.yml') | from_yaml }}"

    - name: bind
      set_fact:
        bind: "{{ item }}"
      loop: "{{ name.sites }}"

    - name: Get Certificate
      include_tasks: add.yml
      loop: "{{ name.sites }}"
      loop_control:
        loop_var: web
      when: "inventory_hostname in web.hosts"

```

add.yml：

```yaml
- name: Create Web Directory
  win_file:
    path: "{{ web.physical_path }}"
    state: directory
  register: dir

- name: Create Web Directory-2
  win_file:
    path: "{{ web.physical_path }}-2"
    state: directory
  register: dir


- name: Create Certificate
  win_shell: |
    get-certificate -Template "{{ name.certificate.template }}" -SubjectName "CN={{ item.hostname }}" -Dnsname "{{ item.hostname }}" -CertStoreLocation "Cert:\LocalMachine\My
  "
  loop: "{{ web.bindings }}"
  when: dir.changed

- name: Web Content
  win_copy:
    content: "{{ web.content }}"
    dest: "{{ web.physical_path }}\\index.html"

- name: get certificate
  win_shell: |
    (get-childitem cert:\localmachine\my | where-object { $_.subject -eq "CN={{ item.hostname }}" }).thumbprint | select-object -first 1
  register: cert
  loop: "{{ web.bindings }}"
  changed_when: false

- name: Create WebiSite
  microsoft.iis.website:
    name: "{{ web.name }}"
    physical_path: "{{ web.physical_path }}"
    state: started

```

```yaml
- name: Create WebiSite-2
  microsoft.iis.website:
    name: "{{ web.name }}-2"
    physical_path: "{{ web.physical_path }}-2"
    state: started

- name: add http binding
  microsoft.iis.website:
    name: "{{ web.name }}-2"
    bindings:
      add:
        - ip: "*"
          port: "{{ item.port }}"
          protocol: "{{ item.protocol }}"
          hostname: "{{ item.hostname }}"
  loop: "{{ web.bindings }}"
  when: "item.protocol == 'http'"

- name: add https binding
  microsoft.iis.website:
    name: "{{ web.name }}"
    bindings:
      add:
        - ip: "*"
          port: "{{ item.port }}"
          protocol: "{{ item.protocol }}"
          hostname: "{{ item.hostname }}"
          certificate_hash: "{{ cert.results[0].stdout_lines[0] }}"
  loop: "{{ web.bindings }}"
  when: "item.protocol == 'https'"

- name: Http Redirect
  win_shell: |
    appcmd set config "{{ web.name }}-2" -section:system.webserver/httpredirect /enabled:true /destination:"https://{{ item.hostname }}/" /httpResponseStatus:"{{ web.redirect
_type }}"
  when: "web.redirect_http_to_https == 'true'"
```

（3）状态页面：

创建 /data/ansible/status.yml 文件，以便在网络组的每个节点上发布状态页面。

请使用 Jinja2 模板 /data/ansible/templates/status.html.j2。该模板未提供，您需要自行创建。

该页面会显示主机名、IP 地址、操作系统版本以及该页面生成的时间。这四个值均是从目标节点获取而来，而非手动输入到模板中。

生成的页面被部署到 C:\inetpub\status\index.html 文件夹中，并通过端口进行服务。

8080.

status.html.j2：

```
{{ inventory_hostname }}</br>
{{ ansible_host }}</br>
{{ ansible_facts.os_name }}</br>
{{ ansible_facts.date_time.date }} {{ ansible_facts.date_time.time }}
```

剧本：

```yaml
---
- name: Status
  hosts: all
  gather_facts: true
  tasks:
    - name: Install Web Service
      win_feature:
        name: Web-Server

    - name: Create Web Directory
      win_file:
        path: c:\inetpub\status
        state: directory

    - name: Web Index
      win_template:
        src: templates/status.html.j2
        dest: c:\inetpub\status\index.html

    - name: status web
      microsoft.iis.website:
        name: status
        physical_path: c:\inetpub\status
        bindings:
          set:
            - ip: "*"
              port: 8080
```

（4）如果只需要创建 80 站点：

使用 win_iis_website 会简单一点

```
35
36      - name: Creating WebSite
37        community.windows.win_iis_website:
38          name: "{{ item.domain_prefix }}"
39          state: started
40          port: 80
41          ip: '*'
42          hostname: "{{ item.domain_prefix }}.shanghai.local"
43          physical_path: "C:\\{{ item.domain_prefix }}"
44        loop: "{{ web.website }}"
45
46      - name: Stop Default WebSite
47        community.windows.win_iis_website:
48          name: "Default Web Site"
49          state: Stopped
```

2、创建用户、OU、组：

```
# 可以使用的模块：
microsoft.ad.user   //幂等创建 AD 域用户
microsoft.ad.group   //幂等创建 AD 域组
win_user   //幂等创建 本地用户
win_group   //幂等创建 本地组
```

（1）创建 OU、组、用户：

Playbook: ad.yml

- Create /data/ansible/ad.yml to build the directory objects of shintsrv01.
  - The organizational units, groups and users are read from /data/ansible/files/ad_objects.json. Nothing may be hard coded in the playbook.
  - Every user is placed in the organizational unit of the department, is a member of the group of the department, receives the email attribute and the home folder as described in Part 1.

  - Use the dedicated Active Directory collection of Ansible where a module exists. Fall back to a shell task only where no module is available.
- During marking the Experts will replace ad_objects.json with their own copy of the file, delete the objects of one department and run the playbook once.

这里脚本有一些比较复杂的点：
1、因为这里要用两个 loop，每个 loop 的变量都是 item，因此会冲突覆盖，因此要用 loop_control，将外层的 loop 的变量设置为 u
2、因为要用两个 loop，但是一个 tasks 中不能用两个 loop，因此要用 include_tasks 调用另一个 yml 剧本
3、因为 json 文件中给定的是一个范围，因此要用 range 读取 json 中的变量
4、range 输出的范围，例如 1 ~ 25，只会给出 1 ~ 24，因此末尾需要 + 1
5、因为变量是字符串，不能直接加减运算，因此要 | int 转为 数字格式
6、因为要求用户是 sales01 - 25，因此要配置 '%02d' | format(数字) 进行补 0
7、这里要定义 homedirectory 和 homedrive，要使用 attributes 来定义。

```yaml
---
- name: create user
  hosts: shintsrv01
  gather_facts: false
  tasks:
    - name: Read json
      set_fact:
        name: "{{ lookup('file','/data/ansible/ad_objects.json') | from_json }}"

    - name:
      debug:
        var: name.departments

    - name: base dn
      set_fact:
        dn: "{{ name.base_dn }}"

    - name: create user
      include_tasks: user.yml
      loop: "{{ name.departments }}"
      loop_control:
        loop_var: u
~
```

user.yml:

```yaml
- name: create user
  microsoft.ad.user:
    identity: "{{ u.user_prefix }}{{ '%02d' | format(item | int) }}"
    password: "{{ u.password }}"
    email: "{{ u.user_prefix }}{{ '%02d' | format(item | int) }}@{{ name.mail_domain }}"
    update_password: on_create
    attributes:
      set:
        homedirectory: "{{ name.home_share }}"
        homedrive: "{{ name.home_drive }}"
  loop: "{{ range(u.first,(u.last | int) + 1) | list }}"
```

（2）创建域用户：

```yaml
1  ---
2  - name: ansible
3    hosts: dc1.shanghai.local
4    gather_facts: false
5    tasks:
6      - name: read json file
7        set_fact:
8          user: "{{ lookup('file','/opt/ansible/users.json') | from_json }}"
9
10     - name: create user
11       microsoft.ad.user:
12         identity: "{{ item.name }}"
13         firstname: "{{ item.firstname }}"
14         surname: "{{ item.surname }}"
15         password: "{{ item.password }}"
16         email: "{{ item.name }}@shanghai.com"
17         update_password: on_create
18       loop: "{{ user }}"
19
```

必须要加 update_password：on_create 才是幂等创建，否则每次都会更新密码

（3）2024 题目：

```yaml
---
- name: Share
  hosts: all
  gather_facts: false
  vars:
    ansible_user: administrator
    ansible_password: "Skill39@Shanghai"
    ansible_port: 5985
    ansible_winrm_transport: basic
    ansible_connection: winrm
    ansible_winrm_server_cert_validation: ignore

  tasks:
    - name: Create read Group
      microsoft.ad.group:
        name: "{{ item.read }}"
        scope: global
        domain_username: "administrator@{{ item.site }}"
        domain_password: Skill39@Shanghai
        domain_server: "dc.{{ item.site }}.cn"
      run_once: true
      delegate_to: "dc.{{ item.site }}.cn"
      loop: "{{ project_shares }}"

    - name: Create write Group
      microsoft.ad.group:
        name: "{{ item.write }}"
        scope: global
        domain_username: "administrator@{{ item.site }}"
        domain_password: Skill39@Shanghai
        domain_server: "dc.{{ item.site }}.cn"
      run_once: true
      delegate_to: "dc.{{ item.site }}.cn"
      loop: "{{ project_shares }}"

    - name: Create Directory
      win_file:
        path: "C:\\project_share\\{{ item.name }}"
        state: directory
      when: "inventory_hostname in groups['{{ item.site }}']"
      loop: "{{ project_shares }}"
      tags: dir
:q
```

```yaml
    - name: Create Share
      win_share:
        name: "{{ item.name }}"
        path: "C:\\project_share\\{{ item.name }}"
        #read: "{{ item.read }}"
        #full: "{{ item.write }}"
        full: everyone
      when: "inventory_hostname in groups['{{ item.site }}']"
      loop: "{{ project_shares }}"
      tags: share

    - name: Disable inherited ACE's
      ansible.windows.win_acl_inheritance:
        path: "C:\\project_share\\{{ item.name }}"
        state: absent
      when: "inventory_hostname in groups['{{ item.site }}']"
      loop: "{{ project_shares }}"

```

```yaml
    - name: permission read
      win_acl:
        path: "C:\\project_share\\{{ item.name }}"
        user: "{{ item.read }}"
        rights: read
        type: allow
      when: "inventory_hostname in groups['{{ item.site }}']"
      loop: "{{ project_shares }}"

    - name: permission write
      win_acl:
        path: "C:\\project_share\\{{ item.name }}"
        user: "{{ item.write }}"
        rights: fullcontrol
        type: allow
      when: "inventory_hostname in groups['{{ item.site }}']"
      loop: "{{ project_shares }}"


    - name: permission write
      win_acl:
        path: "C:\\project_share\\{{ item.name }}"
        user: "Domain Admins"
        rights: fullcontrol
        type: allow
      when: "inventory_hostname in groups['{{ item.site }}']"
      loop: "{{ project_shares }}"

    - name: permission write
      win_acl:
        path: "C:\\project_share\\{{ item.name }}"
        user: SYSTEM
        rights: fullcontrol
        type: allow
      when: "inventory_hostname in groups['{{ item.site }}']"
      loop: "{{ project_shares }}"

```

或：

•优化剧本：

题目要求要同时对子域和当前域创建，并且这里有的 OU 路径还是 dc=lyon


```
---
- name: paris site share folder
 hosts: DMZ-1
 gather_facts: false
 tasks:
  - name: Create a group in a custom path
    microsoft.ad.ou:
      name: "{{ item.ou.split(',')[0].split('=')[1] }}"
      path: DC=paris,DC=local
      domain_server: SH-INTERNAL-1.paris.local
      domain_username: administrator@paris
      domain_password: Skills39
      state: present
    loop: "{{ file_shares }}"
    when: "'lyon' not in item.ou"

  - name: Create a read group in a custom path
    microsoft.ad.group:
      name: "{{ item.read }}"
      path: "{{ item.ou }}"
      scope: domainlocal
      #path: OU=groups,DC=ansible,DC=local
      domain_server: SH-INTERNAL-1.paris.local
      domain_username: administrator@paris
      domain_password: Skills39
      state: present
    loop: "{{ file_shares }}"
    when: "'lyon' not in item.ou"

  - name: Create a write group in a custom path
    microsoft.ad.group:
      name: "{{ item.write }}"
```

```
    path: "{{ item.ou }}"
    scope: domainlocal
    #path: OU=groups,DC=ansible,DC=local
    domain_server: SH-INTERNAL-1.paris.local
    domain_username: administrator@paris
    domain_password: Skills39
    state: present
  loop: "{{ file_shares }}"
  when: "'lyon' not in item.ou"

- name: Create Directory
  win_file:
    path: "c:\\project_share\\{{ item.name }}"
    state: directory
  loop: "{{ file_shares }}"
  when: "'lyon' not in item.ou"

- name: disable other user
  ansible.windows.win_acl_inheritance:
    path: "c:\\project_share\\{{ item.name }}"
    state: absent
  loop: "{{ file_shares }}"
  when: "'lyon' not in item.ou"



- name: write
  ansible.windows.win_acl:
    path: "c:\\project_share\\{{ item.name }}"
    user: "{{ item.write }}"
    rights: fullcontrol
    type: allow
    state: present
  loop: "{{ file_shares }}"
  when: "'lyon' not in item.ou"
  tags: t1

- name: write
  ansible.windows.win_acl:
    path: "c:\\project_share\\{{ item.name }}"
    user: SYSTEM
    rights: fullcontrol
    type: allow
    state: present
  loop: "{{ file_shares }}"
```

```
    when: "'lyon' not in item.ou"
    tags: t2

  - name: write
    ansible.windows.win_acl:
      path: "c:\\project_share\\{{ item.name }}"
      user: administrators
      rights: fullcontrol
      type: allow
      state: present
    loop: "{{ file_shares }}"
    when: "'lyon' not in item.ou"
    tags: t3

  - name: read
    ansible.windows.win_acl:
      path: "c:\\project_share\\{{ item.name }}"
      user: "{{ item.read }}"
      rights: read
      type: allow
      state: present
    loop: "{{ file_shares }}"
    when: "'lyon' not in item.ou"

  - name: Add secret share
    ansible.windows.win_share:
      name: "{{ item.name }}"
      path: "c:\\project_share\\{{ item.name }}"
      full: "{{ item.write }}"
      read: "{{ item.read }}"
    loop: "{{ file_shares }}"
    when: "'lyon' not in item.ou"
    tags: t4


- name: lyon site share folder
 hosts: BJ-INTERNAL-1
 gather_facts: false
 tasks:
  - name: Create a group in a custom path
    microsoft.ad.ou:
      name: "{{ item.ou.split(',')[0].split('=')[1] }}"
      path: DC=lyon,DC=paris,DC=local
      domain_server: BJ-INTERNAL-1.lyon.paris.local
```

```
    domain_username: administrator@lyon
    domain_password: Skills39
    state: present
  loop: "{{ file_shares }}"
  when: "'lyon' in item.ou"

- name: Create a read group in a custom path
  microsoft.ad.group:
    name: "{{ item.read }}"
    path: "{{ item.ou }}"
    scope: domainlocal
    #path: OU=groups,DC=ansible,DC=local
    domain_server: BJ-INTERNAL-1.lyon.paris.local
    domain_username: administrator@lyon
    domain_password: Skills39
    state: present
  loop: "{{ file_shares }}"
  when: "'lyon' in item.ou"

- name: Create a write group in a custom path
  microsoft.ad.group:
    name: "{{ item.write }}"
    path: "{{ item.ou }}"
    scope: domainlocal
    #path: OU=groups,DC=ansible,DC=local
    domain_server: BJ-INTERNAL-1.lyon.paris.local
    domain_username: administrator@lyon
    domain_password: Skills39
    state: present
  loop: "{{ file_shares }}"
  when: "'lyon' in item.ou"

- name: Create Directory
  win_file:
    path: "c:\\project_share\\{{ item.name }}"
    state: directory
  loop: "{{ file_shares }}"
  when: "'lyon' in item.ou"

- name: disable other user
  ansible.windows.win_acl_inheritance:
    path: "c:\\project_share\\{{ item.name }}"
    state: absent
  loop: "{{ file_shares }}"
```

```
  when: "'lyon' in item.ou"


- name: write
  ansible.windows.win_acl:
    path: "c:\\project_share\\{{ item.name }}"
    user: "{{ item.write }}"
    rights: fullcontrol
    type: allow
    state: present
  loop: "{{ file_shares }}"
  when: "'lyon' in item.ou"
  tags: t1

- name: write
  ansible.windows.win_acl:
    path: "c:\\project_share\\{{ item.name }}"
    user: SYSTEM
    rights: fullcontrol
    type: allow
    state: present
  loop: "{{ file_shares }}"
  when: "'lyon' in item.ou"
  tags: t2

- name: write
  ansible.windows.win_acl:
    path: "c:\\project_share\\{{ item.name }}"
    user: administrators
    rights: fullcontrol
    type: allow
    state: present
  loop: "{{ file_shares }}"
  when: "'lyon' in item.ou"
  tags: t3

- name: read
  ansible.windows.win_acl:
    path: "c:\\project_share\\{{ item.name }}"
    user: "{{ item.read }}"
    rights: read
    type: allow
    state: present
  loop: "{{ file_shares }}"
```

```
    when: "'lyon' in item.ou"

  - name: Add secret share
    ansible.windows.win_share:
      name: "{{ item.name }}"
      path: "c:\\project_share\\{{ item.name }}"
      full: "{{ item.write }}"
      read: "{{ item.read }}"
    loop: "{{ file_shares }}"
    when: "'lyon' in item.ou"
    tags: t4
```

3、备份文件：

```
# 可用的模块：
win_copy   //将 Ansible 本机的文件文件夹 拷给 Windows
win_robocopy   //在 Windows 和 Windows 之间复制文件、文件夹
# 示例：
community.windows.win_robocopy:
 src: C:\Data
 dest: \\server2\backup   //需要能够访问到
 recurse: true
delegate_to：Server1   //最好指定在哪台机器执行，指定 server1 就是将 server1 的文件夹复制过去
```

（1）备份 web 根目录、DFS 目录：

Ansible Controller

- This device is an Ansible Controller for managing servers.
- All tasks must be performed inside WSL. (Local User)
- All servers on the unstoppable networks should be added to the inventory.
  - Create "web" and "file" groups, and classify each into the web servers and file servers of the topology.
    - web: shdmzsrv01, shdmzsrv02, shintsrv01
    - file: shintsrv02, bjintsrv01
- The SSH protocol must be used when connecting to target nodes.
- Store server connection information in /etc/ansible/group_vars/all.
  - Encrypt the server information using vault.
  - Securely store the password in /etc/ansible/.vault_pass.
- The directory location for storing playbooks for automation is /data/ansible
- Create backup.yml to back up server devices.
  - When executing the playbook, display a prompt named 'Type' to select the target server group for backup. The input must be either "web" or "file".
    - If "web" is entered, back up the "C:\inetpub" folder of each server to "\\unstoppable.cn\share\org\iisbackup\<server name>\inetpub."
    - If "file" is entered, back up the "C:\share" folder of each server to the "C:\backup" folder.
  - If re-executed within 5 minutes for the same type, it should return "Less than 5 minutes have passed since the last backup" and perform no action.

这里脚本有一些比较复杂的点：
1、因为要调用两个 when，并且要同时匹配两个条件，因此要在 when 中用 and 来定义两个条件（如果同时写两个 when 条件，就是 or ，无论匹配上哪个条件都可）（或者用列表来写 when 的多个条件，这样用列表就等于用 when ... and ...）
2、并且还要 5 分钟内多次执行剧本将会退出并返回一段消息，所以这里剧本结尾创建一个文件，剧本开头去查看这个文件是否是 5 分钟内修改的，如果是，就 debug 一个信息，并且用 meta: end_play 终止剧本
3、在两台 windows 间复制文件、文件夹，用 win_robocopy 模块最好

```yaml
---
- name: task
  hosts: all
  gather_facts: false
  vars_prompt:
    - name: Type
      private: false
  tasks:
    - name: check
      stat:
        path: /data/ansible/{{ Type }}
      delegate_to: localhost
      run_once: true
      register: stat
      vars:
        ansible_shell_type: sh

    - name: check
      raw: |
        find /data/ansible -name {{ Type }} -mmin -5 -print -quit
      register: time
      delegate_to: localhost
      run_once: true
      changed_when: false
      vars:
        ansible_shell_type: sh

    - name: echo
      debug:
        msg: "Less than 5 minutes have passed since the last backup"
      when: time.stdout | trim != ''
      run_once: true

    - name: end
      meta: end_play
      when: time.stdout | trim != '' or Type != 'web' and Type != 'file'
      run_once: true

```

```yaml

  - name: create backup directory
    win_file:
      path: \\unstoppable.cn\share\org\iisbackup\{{ inventory_hostname }}\
      state: directory
    when: "Type == 'web' and inventory_hostname in groups['web']"

  - name: backup task 1
    win_robocopy:
      src: C:\inetpub
      dest: \\unstoppable.cn\share\org\iisbackup\{{ inventory_hostname }}\
      recurse: true
    when:
      - Type == 'web'
      - inventory_hostname in groups['web']

  - name: backup task 2
    win_robocopy:
      src: C:\share
      dest: C:\backup
      recurse: true
    when:
      - Type == 'file'
      - inventory_hostname in groups['file']


  - name: status
    file:
      path: /data/ansible/{{ Type }}
      state: touch
    delegate_to: localhost
    run_once: true
    vars:
      ansible_shell_type: sh

```