SSM Automated Domain Join Configuration – Part 5 of 5

Sivaprasad Padisetty (October 2015)

*This is the last of the five part series, discusses the automated domain join which is defined by the task ‘aws:domainJoin’. Part four can be found* [*here*](http://www.padisetty.com/2015/10/ssm-cloudwatch-configuration-part-4-of-5.html)*. In addition to joining the instance to the domain, it also adds a user to the administrators group. This means the domain user can login into the instance as an administrator and perform admin tasks with domain credential and completely avoiding instance level passwords. Neat thing about this is, you don’t have to specify the password for domain join!*

*Note: You can either join to a Simple AD based domain or to an Active Directory via AD Connector. This is not the way to directly join Active Directory and bypass AWS Directory Service.*

# Schema for aws:domainJoin

"aws:domainJoin": {

"properties": {

"directoryId": "id",

"directoryName": "name",

"directoryOU": "OU=organizationalUnit,DC=domainComponent,DC=domainComponent,DC=domainComponent",

"dnsIpAddresses": ["ip1","ip2",...]

}

}

# AutoDomainJoin Function

This helper function domain joins and adds the specified user to the local administrators group. Domain join is performed by using the task “aws:domainJoin”. The user can be added to the administrators group by the task “aws:psModule”, using the “net” command. A document with these two tasks is constructed. Then the document is passed to SSMAssociated which creates the actual document, associates with the instances and waits for the convergence.

function AutoDomainJoin (

$Instance,

[string]$DirectoryId,

[string]$DirectoryName,

[string]$DNSIP1,

[string]$DNSIP2,

[string]$AdminUser) #User to be added to the administrators group

{

$doc = @"

{

"schemaVersion": "1.0",

"description": "Auto Domain Join",

"runtimeConfig": {

"aws:domainJoin": {

"properties": {

"directoryId": "$DirectoryId",

"directoryName": "$DirectoryName",

"dnsIPAddresses": [

"$DNSIP1",

"$DNSIP2"

]

}

},

"aws:psModule": {

"description": "Install and run ps modules.",

"properties": [

{

"runCommand": "net localgroup administrators $AdminUser /add"

}

]

}

}

}

"@

SSMAssociate $instance $doc -RetrySeconds 300 -Credential $cred

}

Locate the right directory by using the Get-DSDirectory (which is AWS Directory Service SDK). The script below simply picks the first one.

$d = Get-DSDirectory | select -First 1

The instance should be able to communicate with the DNS server associated with the directory. The easiest way is to create the instance in the same subnet. Instead of picking the first subnet, pick the corresponding subnet. Sample to match a subnet can be found at [link](http://www.padisetty.com/2014/05/powershell-bit-manipulation-and-network.html). Below script locates the right one.

function checkSubnet ([string]$cidr, [string]$ip)

{

$network, [int]$subnetlen = $cidr.Split('/')

$a = [uint32[]]$network.split('.')

[uint32] $unetwork = ($a[0] -shl 24) + ($a[1] -shl 16) + ($a[2] -shl 8) + $a[3]

$mask = (-bnot [uint32]0) -shl (32 - $subnetlen)

$a = [uint32[]]$ip.split('.')

[uint32] $uip = ($a[0] -shl 24) + ($a[1] -shl 16) + ($a[2] -shl 8) + $a[3]

$unetwork -eq ($mask -band $uip)

}

$subnet = Get-EC2Subnet | ? { checkSubnet $\_.CidrBlock $d.DnsIpAddrs[0]} | select -First 1

Finally, invoke the above AutoDomainJoin helper function with the details retrieved from Get-DSDirectory.

AutoDomainJoin -Instance $instance `

-DirectoryId $d.DirectoryId `

-DirectoryName $d.Name `

-DNSIP1 $d.DnsIpAddrs[0] `

-DNSIP2 $d.DnsIpAddrs[1] `

-AdminUser "$($d.ShortName)\\sivapadisetty"

The doc produced:

PS C:\temp\ssm> $doc

{

"schemaVersion": "1.0",

"description": "Auto Domain Join",

"runtimeConfig": {

"aws:domainJoin": {

"properties": {

"directoryId": "d-12341234f3",

"directoryName": "corp.amazonworkspaces.com",

"dnsIPAddresses": [

"172.16.1.100",

"172.16.0.107"

]

}

},

"aws:psModule": {

"description": "Install and run ps modules.",

"properties": [

{

"runCommand": "net localgroup administrators corp\\sivapadisetty /add"

}

]

}

}

}

You can find the code under “AWS/SSM” folder at <https://github.com/padisetty/Samples>.

Explore & Enjoy!

/Siva