Amazon Elastic Compute Cloud (EC2) Simple Systems Manager (SSM)

Lightweight configuration management of EC2

November 2015

Notices

Contents

Abstract

When you move from a relatively static and homogeneous computing environment where you have a small number of persistent, well-known servers (or instances, using Amazon Elastic Compute Cloud (EC2) terminology) to a larger and more dynamic and heterogeneous environment, you may need to think about managing the configuration of those instances in a new way.

This whitepaper will describe how Simple Systems Manager (SSM) can be used to manage your Window Server fleet in EC2.

Introduction

Amazon EC2 Simple Systems Manager (SSM) is an Amazon Web Services (AWS) feature that facilitates the automatic configuration of AWS Elastic Compute Cloud (EC2) instances running Microsoft Windows. Configuration of Windows EC2 instances can be specified at launch time or can occur while the instance is running.

SSM currently supports the following configuration scenarios that allow for:

- Automated joining of instance to a Microsoft Active Directory domain
- MSI Package installation, repair and uninstallation
- PowerShell Module installation
- Delivery of Performance Monitor, Event Log, IIS Log, and custom log file data to CloudWatch and CloudWatch Logs.

SSM Requirements

SSM is implemented through the EC2Config service already included with Windows Server Amazon Machine Images (AMIs). By default, the EC2Config service will poll SSM every 5 minutes for configuration documents. These documents (in JSON format) contain the details of the configuration and how it is to be applied.

Please visit the SSM documentation at http://docs.aws.amazon.com/AWSEC2/latest/WindowsGuide/ec2-configuration-manage.html for up-to-date information about region availability, requirements, and limitations. In order to gain the automated configuration benefits of SSM, make sure EC2Config version 3.2.97 or greater is installed. One way to verify the version is to view properties of the EC2Config.exe running process in TaskManager or locally via the (ps) PowerShell cmdlet.

```
ps ec2* |Select-Object fileversion
```

It is also possible to use the Get-Process (ps) PowerShell cmdlet remotely to find the version of the EC2Config service on remote computers. For example:

```
invoke-command -ComputerName Server01 {ps ec2* |Select-
Object fileversion}
```

Instances must connect to SSM through the Internet via the public endpoint. Please note that SSM does not currently provide a private endpoint for access from within the VPC.

Getting Started

Any Windows instance based on the latest version of the AWS Public AMIs include a compatible version of the EC2Config Service. Existing instances that were previously launched may require an update to the latest version of the EC2Config service.

To update to the latest version of EC2Config, first obtain the latest version at https://aws.amazon.com/developertools/5562082477397515. Once downloaded, you can unzip and run EC2Install.exe (with –q for quiet installation) to initiate the upgrade. This process can be automated with existing or third-party software distribution tools.

Instances must be launched with an Identity and Access Management (IAM) role to take advantage of SSM and to have required permissions granted. An example of an IAM role with the necessary actions is shown below.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1415763493000",
      "Effect": "Allow",
      "Action": [
        "ssm:DescribeAssociation",
        "ssm:ListAssociations",
        "ssm:ListDocuments",
        "ssm:GetDocument",
        "ssm:UpdateAssociationStatus",
        "ds:CreateComputer"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
```

SSM Scenarios

All features of SSM can be implemented by associating a configuration JSON document with an EC2 instance running Microsoft Windows. Only one configuration document can be associated with an instance at a time. A single configuration document can specify multiple configurations. For example, one document could specify some MSI files to install and configure the instance to forward various performance counters to Amazon CloudWatch.

If you disassociate a configuration document from an instance, this doesn't change the configuration of the instance. To change the configuration of an instance after you disassociate a configuration document, you must create a new configuration document that describes the configuration tasks (for example, uninstalling software), and then associate it with the instance.

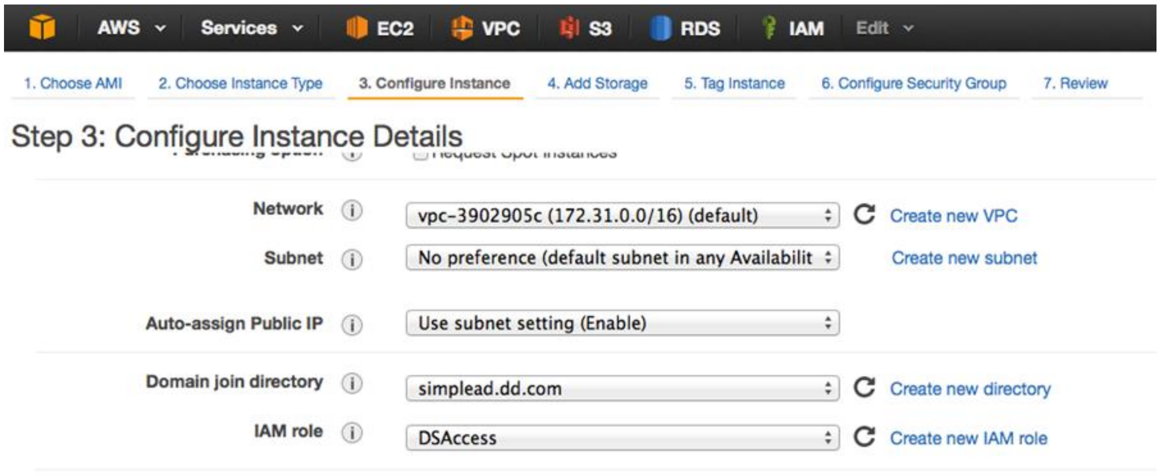
Domain Join

For domain joins, SSM requires the use of AWS Directory Services. You can use AWS Directory Service to create three types of directories. AWS Directory Service for Microsoft Active Directory (Enterprise Edition), or Microsoft AD, is a managed Microsoft Active Directory, powered by Windows Server 2012 R2. Simple AD is a stand-alone, managed directory, powered by a Samba 4 Active Directory Compatible Server. AD Connector is a directory gateway that allows you to proxy directory requests like authentication requests, as well as user or group lookups to your existing, on-premises Microsoft Active Directory. Please refer to Directory Services documentation for related issues at http://aws.amazon.com/documentation/directory-service/. Please note, joining an Active Directory requires a reboot.

With AWS Management Console

SSM enables users to join EC2 instances running Microsoft Windows to their AWS Directory Service (AWS DS) domains at scale via a secure mechanism. Customers may join manually, via CloudFormation templates, or user data often with hard-coded credentials. SSM aims to make it easier for administrators by leveraging configuration documents which can be associated to an instance at launch or while it is running without the need to hard-code credentials. Furthermore, customers with “on-premises only” AD environments need a solution. The AD Connector feature in AWS Directory Service coupled with the domain join function of SSM provides customers a secure and automated way to perform domain joins with their on-premises directory.

The AD domain join feature is available directly in the AWS Management Console. Simply choose the AWS Directory Services (AWS DS) directory in the “Domain join directory” dropdown as shown below. SSM will automatically connect to the directory service with a machine account on the DNS address associated with the AWS DS directory.



Step 3: Configure Instance Details

Network: vpc-3902905c (172.31.0.0/16) (default)
Subnet: No preference (default subnet in any Availabilit
Auto-assign Public IP: Use subnet setting (Enable)
Domain join directory: simplead.dd.com
IAM role: DSAccess

Figure 1: Domain join directory, AWS Console

With AWS Tools for PowerShell or AWS CLI

First, ensure that your instance has been started with the appropriate IAM role. See “Getting Started”.

Create a JSON document accepted by SSM for domain join. Here is an example “my-config.json”:

```
{
  "schemaVersion":"1.0",
  "description": "Domain Join",
  "runtimeConfig": {
      "aws:domainJoin": {
          "properties":{
          "directoryId": "d-906XXX0367",
          "directoryName": "ad.example.com",
             "directoryOU" : "OU=AWS,DC=ad,DC=example,DC=com",
          "dnsIpAddresses": ["172.31.53.1","172.31.26.167"]
          }
      }
  }
}
```

Then, from a machine with the latest AWS Tools for PowerShell installed (see http://docs.aws.amazon.com/powershell/latest/userguide/pstools-getting-set-up.html), execute the following PowerShell commands:

```
$doc = Get-Content my-config.json | Out-String

New-SSMDocument -Content $doc -Name "my-custom-config"

New-SSMAssociation -InstanceId i-1a2b3c4d -Name "my-custom-config"
```

The EC2Config Service on the target instance (i-la2b3c4d in this example) will poll the SSM service within 10 minutes and implement the configuration specified in the JSON document. If the domain join was successful, the instance will reboot. Further details are available at http://docs.aws.amazon.com/AWSEC2/latest/WindowsGuide/ec2-join-aws-domain.html.

Amazon CloudWatch and Logs Integration

You can use SSM to configure integration with Amazon CloudWatch and Amazon CloudWatch Logs on multiple instances to monitor their log files. You can send Windows Server messages in the application, system, security, and Event Tracing for Windows logs to Amazon CloudWatch Logs.

When you enable logging for the first time, entries in the application, system, security, and Event Tracing for Windows logs generated within the previous minute are sent before new entries are sent; entries that occurred before this time are not included. If you disable logging and then later re-enable logging, entries are sent from where it left off. For custom log files and Internet Information Services (IIS) logs, all entries are sent from the beginning. In addition, performance counter data can be sent to Amazon CloudWatch.

To configure an instance to forward information to CloudWatch and CloudWatch logs, you need to launch the instance with a role that has access to CloudWatch and SSM.

For an example IAM role, which combines SSM and CloudWatch with the necessary actions, see below:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1415763493000",
      "Effect": "Allow",
      "Action": [
        "ssm:DescribeAssociation",
        "ssm:ListAssociations",
        "ssm:ListDocuments",
        "ssm:GetDocument",
        "ssm:UpdateAssociationStatus",
        "ds:CreateComputer"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Sid": "Stmt1414002531000",
      "Effect": "Allow",
      "Action": [
        "cloudwatch:PutMetricData"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Sid": "Stmt1414002720000",
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:DescribeLogGroups",
        "logs:DescribeLogStreams",
        "logs:PutLogEvents"
      ],
      "Resource": [
        "*"
```

```
        ]
      }
    ]
  }
```

Next, create a JSON document that specifies the Windows Event Log, Performance Counters, and Log files to forward to CloudWatch. For example:

```
{
   "schemaVersion":"1.0",
   "description":"Example CloudWatch Logs tasks",
   "runtimeConfig":{
      "aws:cloudWatch":{
         "properties":{
            "EngineConfiguration":{
               "PollInterval":"00:00:15",
               "Components":[
                  {
                     "Id":"ApplicationEventLog",

"FullName":"AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC
2.Windows.CloudWatch",
                     "Parameters":{
                        "LogName":"Application",
                        "Levels":"1"
                     }
                  },
                  {
                     "Id":"SystemEventLog",
"FullName":"AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC
2.Windows.CloudWatch",
                     "Parameters":{
                        "LogName":"System",
                        "Levels":"1"
                     }
                  },
                  {
                     "Id":"SecurityEventLog",

"FullName":"AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC
2.Windows.CloudWatch",
                     "Parameters":{
                        "LogName":"Security",
                        "Levels":"1"
                     }
                  },
                  {
                     "Id":"ETW",

"FullName":"AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC
2.Windows.CloudWatch",
                     "Parameters":{
                        "LogName":"Microsoft-Windows-WinINet/Analytic",
                        "Levels":"1"
                     }
                  },
                  {
                     "Id":"PerformanceCounter",

"FullName":"AWS.EC2.Windows.CloudWatch.PerformanceCounterComponent.Performanc
eCounterInputComponent,AWS.EC2.Windows.CloudWatch",
                     "Parameters":{
                        "CategoryName":"Memory",
                        "CounterName":"Available MBytes",
                        "InstanceName":"",
                        "MetricName":"AvailableMemory",
                        "Unit":"Megabytes",
                        "DimensionName":"",
                        "DimensionValue":""
                     }
                  },
                  {
                      "Id": "CustomLogs",
                      "FullName":
"AWS.EC2.Windows.CloudWatch.CustomLog.CustomLogInputComponent,AWS.EC2.Windows
.CloudWatch",
                      "Parameters": {
                          "LogDirectoryPath": "C:\\Program
Files\\Amazon\\Ec2ConfigService\\Logs\\",
                          "TimestampFormat": "MM/dd/yyyy HH:mm:ss",
                          "Encoding": "UTF-8",
                          "Filter": "",
                          "CultureName": "en-US",
                          "TimeZoneKind": "Local",
                          "LineCount": "15"
                      }
                  },

                  {
                     "Id":"CloudWatchLogs",

"FullName":"AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.C
loudWatch",
                     "Parameters":{
                        "Region":"us-east-1",
                        "LogGroup":"Default-Log-Group",
                        "LogStream":"{instance_id}"
                     }
                  },
                  {
                     "Id":"CloudWatch",

"FullName":"AWS.EC2.Windows.CloudWatch.CloudWatch.CloudWatchOutputComponent,A
WS.EC2.Windows.CloudWatch",
                     "Parameters":{
                        "Region":"us-east-1",
                        "NameSpace":"Windows/Default"
                     }
                  }
               ],
               "Flows":{
                  "Flows":[
                     "PerformanceCounter,CloudWatch",
                     "(CustomLogs, ETW, SystemEventLog, ApplicationEventLog,
SecurityEventLog),CloudWatchLogs"
```

```
                    ]
                }
            }
        }
    }
}
```

Complete details of the sections in the document can be found in the SSM documentation at http://docs.aws.amazon.com/AWSEC2/latest/WindowsGuide/ec2-configuration-cwl.html.

Notice, it is possible to specify an AccessKey and SecretKey for access to CloudWatch and CloudWatch Logs. However, this is not the recommended approach. The EC2Config service will attempt to use the privileges of any role assigned to the EC2 instance.

Once the document is created, execute the following PowerShell commands to create and associate the JSON document with the target instance:

```
$doc = Get-Content my-config.json | Out-String

New-SSMDocument -Content $doc -Name "my-custom-config"

New-SSMAssociation -InstanceId i-1a2b3c4d -Name "my-custom-config"
```

Or with the latest AWS CLI, execute the following commands:

```
aws ssm create-document --content file://path/to/my-config.json --name "my-
custom-config"

aws ssm create-association --instance-id i-1a2b3c4d –name "my-custom-config"
```

MSI Deployment

SSM also supports downloading an MSI file from S3 or a public endpoint and installing, repairing, or uninstalling that MSI. This is implemented in a similar fashion as the other SSM features. Simply create a configuration document and associate it with the target instance(s).

Here is an example MSI configuration document:

```
{
    "schemaVersion": "1.0",
    "description": "MSI Install Script",
    "runtimeConfig": {
        "aws:applications": {
            "properties": [
                {
                "action": "Install",
                "source":
"https://S3region.amazonaws.com/ddbucketofun/MSIs/7z938-x64.msi",
                "sourceHash":
"7C8E873991C82AD9CFCDBDF45254EA6101E9A645E12977DCD518979E50FDEDF3"
                },
                {
                "action": "Install",
                "source": "http://location.s3.amazonaws.com/Firefox/Firefox-
33.0.2/Firefox-33.0.2-en-US.msi",
                "parameters":
"INSTALLEVEL=1000 custompath=\"c:\\folder with space\""
                }
            ]
        }
    }
}
```

Repair and Uninstall are also supported values for the *action* property. The *sourceHash* property is optional and uses a SHA256 hash of the MSI file to validate file contents and authenticity. You can use the Get-FileHash PowerShell command (requires PowerShell 4.0) to determine the SHA256 hash value.

Install PowerShell Modules

Publicly-available PowerShell modules are generally zip files. They should be referenced in the configuration document with a URL. The following JSON describes PowerShell modules to install on your instance. For each module, `source` is the URL of the module and `runCommand` specifies the PowerShell command to use for installation, if needed.

In the example below, runCommand is a single string pointing to a command line:

```
{
  "schemaVersion": "1.0",
  "description": "Install the Chocolatey package provider and use it to
install 7zip and Google Chrome",
  "runtimeConfig": {
    "aws:psModule": {
      "properties": [
        {
          "description": "Install a Windows update PS module and install .NET
4 updates.",
          "source":
"https://gallery.technet.microsoft.com/scriptcenter/2d191bcd-3308-4edd-9de2-
88dff796b0bc/file/41459/43/PSWindowsUpdate.zip",
          "runCommand": "Get-WUInstall -ServiceID 9482f4b4-e343-43b6-b170-
9a65bc822c77 -Title \".NET Framework 4\" -AcceptAll"
        },
        {
          "description": "Install Chocolatey package provider and use it to
install 7zip and GoogleChrome.",
          "runCommand": [
            "$url = 'https://chocolatey.org/install.ps1'",
            "iex ((new-object net.webclient).DownloadString($url))",
            "choco install -y 7zip",
            "choco install -y GoogleChrome"
          ]
        }
      ]
    }
  }
}
```

Note, runCommand can also accept an array of strings, for example:

```
"runCommand": [
            "$url = 'https://my.test.com/MyConfigurationScript.ps1'",
            "iex ((new-object net.webclient).DownloadString($url))"
            ]
```

In the example above, we are using SSM to not only install Chocolatey, but also execute “choco install Google Chrome” and “choco install 7zip” to install Google Chrome and 7zip. Chocolatey is a command-line based package manager for Windows which can be used to install a variety of publicly available installation packages. For more information, see http://chocolatey.org.

Troubleshooting

The EC2Config service provides detailed information, warning and error messages in the Windows Event Log. Open the following file to view success or error messages associated with your assigned SSM configuration documents.

C:\Windows\System32\winevt\Logs\EC2ConfigService.evtx

Optionally, you can view the contents of the EC2Config Service Event Log by issuing the following PowerShell command:

```
Get-EventLog Ec2ConfigService | Sort-Object Index | Format-Table Message -
AutoSize -Wrap | Out-File -Width 240 "C:\Logs\PluginFramework.txt"
```

Other helpful log files are located at these locations:

C:\Program Files\Amazon\Ec2ConfigService\Logs
C:\Windows\System32\config\systemprofile\AppData\Local\Amazon\EC2Config

To verify instance permissions, look for the event log entry "Info: EC2Config configuration status:3;region:us-east-1;iam:0;authz:0" which provides information for the instance. If "iam" has value of "0", the instance does not have an IAM role attached. If "authz" has value of "0", the instance does not have permission to access SSM with its IAM role.

By default, event logging is turned on however file logging is disabled. To turn on file logging, uncomment the following line in log4net.config XML file:

C:\Program Files\Amazon\Ec2ConfigService\log4net.config

then restart ec2config:

```
net stop ec2config & net start ec2config
```

The log file location is at:

C:\Windows\System32\config\systemprofile\AppData\Local\Amazon\Ec2Config\Logs\Ec2ConfigPluginFramework.txt

In addition, the following table contains AWS CLI and PowerShell commands you can use to verify document creation and associations.

Action	AWS CLI	AWS Tools for Windows PowerShell
To view information about an association for a specific instance and configuration document. You can also use this command to view the status of an association.	describe-association	Get-SSMAssociation
To view information about a specified configuration document. You can also use this command to view the status of a configuration document, for example, **creating**.	describe-document	Get-SSMDocumentDescription
To view the contents of a specified configuration document.	get-document	Get-SSMDocument
To view a list of associations for a specified configuration document or a specified instance.	list-associations	Get-SSMAssociationList

Action	AWS CLI	AWS Tools for Windows PowerShell
To view a list of your configuration documents.	list-documents	Get-SSMDocumentList
Remove an existing association from an instance. Must be done before deleting the document.	delete-association	Remove-SSMAssociation

EC2Config-CLI

EC2Config includes a command-line tool called EC2Config-cli located at C:\Program Files\Amazon\Ec2ConfigService. It can be used to reapply the latest configuration for testing or convergence (bringing instance configuration to a desired state) at any later point in time.

Example use cases:

1. Fixing configuration state: An existing configuration installs 7zip. Later, a user accidently uninstalls 7zip. An administrator can reapply the config using EC2Config-cli.
2. Ensure instance configuration stays at desired/goal state: An administrator can create a scheduled task to call "ec2config-cli –a" to periodically apply a configuration so that the instance stays in desired state.
3. Dev-test: No need to wait for the 10 minute (polling interval) to test their configuration changes.

Example apply config command:

```
ec2config-cli /a
ec2config-cli -a
ec2config-cli --apply-configuration
```

FAQs

Are there any charges to use SSM service?

- No, SSM is provided for free. Charges do exist for other services such as Amazon CloudWatch, CloudTrail, AWSConfig, etc.

Can I associate a document to a running instance?

- Yes.

When should I use user data vs SSM to configure my instance?

- Use SSM to do automatic domain join, installing PowerShell modules and configuring CloudWatch settings. This can be done at any point in time (either during launch or on existing instances).
- User data functionality will continue to work the same way to run PowerShell scripts during a Windows launch, and user data will run before SSM does during instance launch.

Can I associate multiple configuration documents to an instance?

- No, an instance can only have one configuration document associated at a time.

How do I change instance configuration association?

- Step 1: Delete association via Remove-SSMAssociation
- Step 2: Associate the new configuration via New-SSMAssociation

What happens when I disassociate a configuration?

- The instance will remain in the current configuration state (nothing will be rolled back). This behavior is to keep this consistent with other Configuration Management tools. We will only apply the new document configuration and not attempt to uninstall or remove anything that was installed or changed from a prior association.

Conclusion

This Whitepaper introduced Amazon EC2 SSM which facilitates the automatic configuration of EC2 instances running Microsoft Windows. It covered various use-cases such as joining an instance to an Active Directory domain & Advanced Performance monitoring with Amazon CloudWatch.

SSM is a powerful configuration management tool that can be extended to many additional use cases. This whitepaper aimed at giving you a feel for the common use cases and hands on experience which will enable you to use SSM for advanced configuration of Windows instances running on EC2

Contributors

The following individuals contributed to this document:

- Justin Bradley, Solutions Architect, Amazon Web Services
- Lee Atkinson, Solutions Architect, Amazon Web Services

Further Reading

For additional help, please consult the following sources:

- SSM API Reference Guide
- Managing Windows Instance Configuration
- Joining a Windows Instance to an AWS Directory Service Domain
- SSM Documents
- Simple Domain Join
- AWS and Microsoft