SSM CloudWatch Configuration – Part 4 of 5

Sivaprasad Padisetty(October 2015)

*Amazon CloudWatch is a monitoring service for AWS cloud resources and the applications that run on AWS. It can be used to collect and track metrics, collect log files, and set alarms. Amazon CloudWatch can monitor AWS resources such as Amazon EC2 instances, Amazon DynamoDB tables, and Amazon RDS DB instances, as well as custom metrics generated by applications and services, and any log files generated by the applications. CloudWatch can be used to gain system-wide visibility into resource utilization, application performance, and operational health. You can use these insights to react and keep your application running smoothly. Fourth in the series of SSM blogs, this blog covers the task ‘aws:psModule’. You can find part three* [*here*](http://www.padisetty.com/2015/10/ssm-powershell-modules-and-scripts-part.html)*.*

# Schema for aws:cloudWatch

"aws:cloudWatch": {

"properties": {

"EngineConfiguration": {

"PollInterval": "00:00:15",

"Components": [

{

"Id": "ApplicationEventLog",

"FullName": "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",

"Parameters": {

"LogName": "Application",

"Levels": "*value*"

}

},

{

"Id": "CloudWatch",

"FullName": "AWS.EC2.Windows.CloudWatch.CloudWatch.CloudWatchOutputComponent,AWS.EC2.Windows.CloudWatch",

"Parameters": {

"AccessKey": "*access-key-id*",

"SecretKey": "*secret-access-key*",

"Region": "*region*",

"NameSpace": "*namespace*"

}

},

{

"Id": "CloudWatchLogs",

"FullName": "AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",

"Parameters": {

"AccessKey": "*access-key-id*",

"SecretKey": "*secret-access-key*",

"Region": "*region*",

"LogGroup": "*group*",

"LogStream": "*stream*"

}

},

{

"Id": "CustomLogs",

"FullName": "AWS.EC2.Windows.CloudWatch.CustomLog.CustomLogInputComponent,AWS.EC2.Windows.CloudWatch",

"Parameters": {

"LogDirectoryPath": "*path*",

"TimestampFormat": "*value*",

"Encoding": "*value*",

"Filter": "",

"CultureName": "*locale*",

"TimeZoneKind": "*value*",

"LineCount": "*value*"

}

},

{

"Id": "ETW",

"FullName": "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",

"Parameters": {

"LogName": "Microsoft-Windows-WinINet/Analytic",

"Levels": "*value*"

}

},

{

"Id": "IISLogs",

"FullName": "AWS.EC2.Windows.CloudWatch.CustomLog.CustomLogInputComponent,AWS.EC2.Windows.CloudWatch",

"Parameters": {

"LogDirectoryPath": "*path*",

"TimestampFormat": "*value*",

"Encoding": "*value*",

"Filter": "",

"CultureName": "*locale*",

"TimeZoneKind": "*value*",

"LineCount": "*value*"

}

},

{

"Id": "PerformanceCounter",

"FullName": "AWS.EC2.Windows.CloudWatch.PerformanceCounterComponent.PerformanceCounterInputComponent,AWS.EC2.Windows.CloudWatch",

"Parameters": {

"CategoryName": "*name*",

"CounterName": "*name*",

"InstanceName": "*name*",

"MetricName": "*name*",

"Unit": "*unit*",

"DimensionName": "*name*",

"DimensionValue": "*value*"

}

},

{

"Id": "SecurityEventLog",

"FullName": "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",

"Parameters": {

"LogName": "Security",

"Levels": "*value*"

}

},

{

"Id": "SystemEventLog",

"FullName": "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",

"Parameters": {

"LogName": "System",

"Levels": "*value*"

}

}

],

"Flows": {

"Flows": [

"*source*,*destination*",

"(*source1*, *source2*),*destination*",

"*source*, (*destination1*,*destination2*)"

]

}

}

}

}

# CloudWatch Custom Metrics Sample

By default, CloudWatch captures free metrics at the host level. To capture the metrics from within the guest, the agent running inside the guest should be configured. A sample to collect the memory usage metric is provided.

function CMApplyMetrics ($instance)

{

$doc = @"

{

"schemaVersion": "1.0",

"description": "Instance configuration",

"runtimeConfig": {

"aws:cloudWatch": {

"description": "CloudWatch Metrics Demo",

"properties": {

"EngineConfiguration": {

"PollInterval": "00:00:15",

"Components": [

{

"Id": "PerformanceCounter",

"FullName": "AWS.EC2.Windows.CloudWatch.PerformanceCounterComponent.PerformanceCounterInputComponent,AWS.EC2.Windows.CloudWatch",

"Parameters": {

"CategoryName": "Memory",

"CounterName": "Available MBytes",

"InstanceName": "",

"MetricName": "Memory",

"Unit": "Megabytes",

"DimensionName": "",

"DimensionValue": ""

}

},

{

"Id": "CloudWatch",

"FullName": "AWS.EC2.Windows.CloudWatch.CloudWatch.CloudWatchOutputComponent,AWS.EC2.Windows.CloudWatch",

"Parameters": {

"Region": "us-east-1",

"NameSpace": "SSMDemo"

}

}

],

"Flows": {

"Flows":["PerformanceCounter,CloudWatch"]

}

}

}

}

}

}

"@

Write-Verbose "CMApply instanceid=$($instance.InstanceId)"

SSMAssociate $instance $doc $cred

}

Sample doc, produced by above function.

PS C:\temp\ssm> $doc

{

"schemaVersion": "1.0",

"description": "Instance configuration",

"runtimeConfig": {

"aws:cloudWatch": {

"description": "CloudWatch Metrics Demo",

"properties": {

"EngineConfiguration": {

"PollInterval": "00:00:15",

"Components": [

{

"Id": "PerformanceCounter",

"FullName": "AWS.EC2.Windows.CloudWatch.PerformanceCounterComponent.PerformanceCounterInputComponent,AWS.EC2.Windows.CloudWatch",

"Parameters": {

"CategoryName": "Memory",

"CounterName": "Available MBytes",

"InstanceName": "",

"MetricName": "Memory",

"Unit": "Megabytes",

"DimensionName": "",

"DimensionValue": ""

}

},

{

"Id": "CloudWatch",

"FullName": "AWS.EC2.Windows.CloudWatch.CloudWatch.CloudWatchOutputComponent,AWS.EC2.Windows.CloudWatch",

"Parameters": {

"Region": "us-east-1",

"NameSpace": "SSMDemo"

}

}

],

"Flows": {

"Flows": [

"PerformanceCounter,CloudWatch"

]

}

}

}

}

}

}

# CloudWatch Logs Sample

In addition to custom metrics, CloudWatch can also collect the logs. This includes event logs, IIS logs and custom logs. The sample below collects the event logs from EC2CofigService into the “SSM-Log-Group” log group. Each instance log is stored in a separate stream. The function CMApplyLogs is straight forward to follow.

function CMApplyLogs ($instance)

{

$doc = @"

{

"schemaVersion": "1.0",

"description": "Instance configuration",

"runtimeConfig": {

"aws:cloudWatch": {

"description": "CloudWatch Logs Demo",

"properties": {

"EngineConfiguration": {

"PollInterval": "00:00:15",

"Components": [

{

"Id": "SSMLogs",

"FullName": "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",

"Parameters": {

"LogName": "EC2ConfigService",

"Levels": "7"

}

},

{

"Id": "CloudWatchLogs",

"FullName": "AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",

"Parameters": {

"Region": "us-east-1",

"LogGroup": "SSM-Log-Group",

"LogStream": "{instance\_id}"

}

}

],

"Flows": {

"Flows": ["SSMLogs,CloudWatchLogs"]

}

}

}

}

}

}

"@

Write-Verbose "CMApply instanceid=$($instance.InstanceId)"

SSMAssociate $instance $doc -Credential $cred

}

Sample doc, produced by the above function.

PS C:\temp\ssm> $doc

{

"schemaVersion": "1.0",

"description": "Instance configuration",

"runtimeConfig": {

"aws:cloudWatch": {

"description": "CloudWatch Logs Demo",

"properties": {

"EngineConfiguration": {

"PollInterval": "00:00:15",

"Components": [

{

"Id": "SSMLogs",

"FullName": "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",

"Parameters": {

"LogName": "EC2ConfigService",

"Levels": "7"

}

},

{

"Id": "CloudWatchLogs",

"FullName": "AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",

"Parameters": {

"Region": "us-east-1",

"LogGroup": "SSM-Log-Group",

"LogStream": "{instance\_id}"

}

}

],

"Flows": {

"Flows": [

"SSMLogs,CloudWatchLogs"

]

}

}

}

}

}

}

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

Explore & Enjoy!

/Siva