Author: Baris Furtinalar

Draft version 0.1

## Lab: Managing Windows Servers with AWS Systems Manager

## Overview

This hands-on lab is designed to get you familiar with the concepts and the process of using AWS Systems Manager with Windows Servers. In the lab we provide “High Level Directions”. These are the steps you should use as the guidelines of what you need to build out. We encourage you to play around and test the steps in various ways. If you have a question about how something works, feel free to both ask a proctor and test it by trying various things.

This design was built off of feedback received from running other customer labs and we are very interested in getting your feedback at the end of the event today on your experience with this lab format.

## High Level Agenda

Customize & Configure Windows Servers

* User management
* Installing Server Roles and Features
* Backups

Implementing Security & Compliance

* Mitigating malware and threats (Patch Manager)
* Avoiding configuration drift (State Manager)

Monitoring Windows Servers

* Sending PerfMon data to CloudWatch Metrics

## Setup base infrastructure

In this first step, you setup EC2 Instances that you will manage with AWS Systems Manager. The instances need to have the ability to read from the AWS range of IP addresses for the region and have the IAM role assigned. Both are required for the instances to list under managed instances.

1. Login into your AWS account
2. Create an IAM role for instance to view ssm documents  
   <https://docs.aws.amazon.com/systems-manager/latest/userguide/systems-manager-access.html>
3. Make sure the instances have the ssm role assigned that you created above
4. Create 2 Windows 2016 servers from the image with image id ami-0241c63173d423f4d. (Launch Instance then choose Community AMIs, in the search box type ami-0241c63173d423f4d)
5. Add another tag to all instances (Tag Type=Webserver)
6. Open Port 80 & 443 as these servers will be webservers

## Customize & Configure Windows Servers

AWS Systems Manager provides you safe, secure remote management of your instances at scale without logging into your servers, replacing the need for bastion hosts, SSH, or remote PowerShell. This feature is called **Run Command**. Run Command enables you to automate common administrative tasks and perform ad hoc configuration changes at scale.

### User Management with Run Command

* AWS Systems Manager go to Actions and choose Run Command and click run command
* Choose the radio button on the left side of AWS-RunPowerShellScript document
* Copy & paste the below text to commands box

$Password = ConvertTo-SecureString "Password1" -AsPlainText –Force

New-LocalUser "User01" -Password $Password -FullName "User One" -Description "Lab User"

* On the targets section choose specify a tag
* On Tags section in the Tag Key input box write type and value box type WebServer (tags are case-sensitive)
* Scroll down and on the Output option section uncheck “Enable writing to an S3 bucket” to disable output logging to Amazon S3.
* Hit Run button.

### Installing Server Roles and Features using Run Command

* AWS Systems Manager go to Actions and choose Run Command and click run command
* Choose the radio button on the left side of AWS-RunPowerShellScript document
* Copy & paste the below text to commands box

Install-WindowsFeature -name Web-Server -IncludeManagementTools

* On the targets section choose specify a tag
* On Tags section in the Tag Key input box write type and value box type WebServer (tags are case-sensitive)
* Scroll down and on the Output option section uncheck “Enable writing to an S3 bucket” to disable output logging to Amazon S3.
* Hit Run button.

### Run Command to run Backups (Optional)

Using Run Command, you can take application-consistent snapshots of all Amazon Elastic Block Store (Amazon EBS) volumes attached to your Amazon EC2 Windows instances. The snapshot process uses the Windows Volume Shadow Copy Service (VSS) to take image-level backups of VSS-aware applications. For this initially **AWS-ConfigureAWSPackage** document is used to setup instance and for recurring backup operations **AWSEC2-CreateVssSnapshot** document is utilized.

The guide to creating consistent backups of your Windows instances can be found here:

<https://docs.aws.amazon.com/systems-manager/latest/userguide/integration-vss.html#integration-vss-console>

## Implementing Security & Compliance

Setting a reasonable goal for compliance levels is often a difficult concept. Part of this is “Patch management” and it is simply the practice of updating software with new pieces of code; most often to address vulnerabilities that could be exploited by hackers.

Although the practice sounds straightforward, patch management is not an easy process for most IT organizations. AWS Systems Manager simplifies operations and management security and compliance at scale. **AWS Systems Manager Patch Manager** automates the process of patching managed instances with security-related updates. **AWS Systems Manager Maintenance Windows** let you define a schedule for when to perform potentially disruptive actions on your instances such as patching an operating system, updating drivers, or installing software or patches

There are several SSM documents around patching. We will use the **AWS-RunPatchBaseline** SSM document to configure patching and compliance for our Windows Servers.

### Patch Management with AWS Systems Manager Patch Manager

* To use Patch Manager, complete the following tasks.
* Click Create patch baseline
* Give a **Name** to your baseline, supply a Description and choose the Operating System as **Windows.**
* On the **Approval Rules** page leave the default options.
* On Patch Exceptions in Approved patches compliance level – optional drop down menu choose Critical
* Hit Create Patch Baseline.
* On Patch Baselines choose the baseline you just created
* Click Actions 🡪 select Set default patch baseline

### 5.2. AWS Systems Manager Maintenance Windows

This section of the lab will continue with the Patch Manger section above. Please continue following the same walkthrough used above: <http://docs.aws.amazon.com/systems-manager/latest/userguide/sysman-patch-walkthrough.html>

* Create a role and name it SSMMaintenanceWindowRole   
  <https://docs.aws.amazon.com/systems-manager/latest/userguide/sysman-maintenance-perm-console.html>   
  This role will be used by the Maintenance Windows to run
* On the AWS Systems Manager console choose Maintenance window in the Actions menu.
* Create a maintenance window for patching
* Have it run Sunday at 01:00 UTC for 8 hours
* Register **Targets** to the patching maintenance windows by using tags  
  (Tag Type=Webserver)
* Click Create maintenance window
* On the maintenance windows list choose the window that you created.
* On the top right corner click **Actions**  🡪 Register Run Command Task.
* Choose the document AWS Run Patch Baseline
* Congratulations you finished your patching, compliance, and maintenance window work. You will need to wait until the maintenance window to see if it worked.

### State Manager for Configuration Drift Avoidance

Another important requirement of a sustainable security and compliance base line is maintaining a consistent state of your systems. **AWS Systems Manager State Manager** is a secure and scalable configuration management service that automates the process of keeping your Amazon EC2 and hybrid infrastructure in a state that you define.

In this section we will focus on improving the security profile of your servers by disabling SSH and RDP access to them. Using State Manager, you can continually have your configuration re-apply. So, in our example, if someone enables remote access to the server – it will automatically be removed when State Manager runs again. Detailed steps on State Manager are here: <http://docs.aws.amazon.com/systems-manager/latest/userguide/sysman-state-walk.html>

* Open the AWS Systems Manager console at <https://console.aws.amazon.com/systems-manager/>
* Create a new document that turns off remote desktop (RDP) service. Use this document https://s3-eu-west-1.amazonaws.com/aws-immersionday-labfiles/DisableRDP.txt  
    
  A guide to creating documents can be found here:  
  <http://docs.aws.amazon.com/systems-manager/latest/userguide/create-ssm-doc.html>
* On the AWS Systems Manager console choose State Manager in the Actions menu.
* Choose Create. Association.
* For Document, choose the document you previously created.
* For Document Version, choose **Default** version at runtime.
* For **Targets**, choose the instances to manage state.
* For Schedule, choose how often you want Systems Manager to apply this policy.
* (Optional) To send command output to an Amazon S3 bucket, On the Output Options section choose **Write to S3**.
* Choose Create Association.
* Choose the association you just created, and then choose Apply **Association Now**.
* See if you can do remote access session to the instance.
* (Optional) Create another document re-enable remote access  
  https://s3-eu-west-1.amazonaws.com/aws-immersionday-labfiles/EnableRDP.txt
* (Optional)Turn back on remote access with Run Command if you wish

## Monitoring Windows Servers

The most important part of Windows server monitoring is ensuring core resource components are being tracked. Having the ability to monitor key metrics such as CPU, memory and disk utilization greatly assists in overseeing resource consumption and knowing how usage growth will impact performance.

The Windows Server event logs contain a mass of useful information, but finding events that might indicate an operational issue or security breach from all the noise isn’t an easy task.

In this section we will focus on Amazon CloudWatch and its integration with AWS Systems Manager.

### Download the Sample Configuration File

Before you start please download the following sample file to your computer: [Download the Sample Configuration File](https://s3.amazonaws.com/ec2-downloads-windows/CloudWatchConfig/AWS.EC2.Windows.CloudWatch.json)

### Systems Manager State Manager to send Windows Event Logs and Performance Monitor Metrics

* Open the AWS Systems Manager console at <https://console.aws.amazon.com/systems-manager/>
* In the navigation pane, choose Systems Manager Services, State Manager on the Amazon EC2 console or on the AWS Systems Manager console choose State Manager in the Actions menu.
* Choose Create. Association.
* For Document, choose AWS-ConfigureCloudWatch.
* For Document Version, choose **Default** version at runtime.
* For **Targets**, choose the instances to integrate with CloudWatch. If you do not see an instance in this list, it might not be configured for Run Command. For more information, see [Systems Manager Prerequisites](https://docs.aws.amazon.com/systems-manager/latest/userguide/systems-manager-setting-up.html) in the AWS Systems Manager User Guide.
* For Schedule, choose how often you want Systems Manager to apply this policy. This indicates how often you want Systems Manager to ensure the integration with CloudWatch is still valid. This does not affect the frequency when the SSM Agent sends data to CloudWatch.
* For **Parameters, Status**, choose **Enabled**. For Properties, copy and paste the contents of JSON document you downloaded.
* (Optional) To send command output to an Amazon S3 bucket, On the Output Options section choose **Write to S3**.
* Choose Create Association.
* Choose the association you just created, and then choose Apply **Association Now**.

Awesome – you finished the labs - Important

When you are done with the hands-on lab you may want to delete the resources you created to avoid getting charged for the services you have created. If you used the free tier instance – they can run for the first year of your account without a charge but after 1 year they will start to bill.

Appendix

You can get some more details on Systems Manager here:   
<http://docs.aws.amazon.com/systems-manager/latest/userguide/systems-manager-setting-up.html>  
<https://github.com/awsdocs/aws-systems-manager-user-guide>