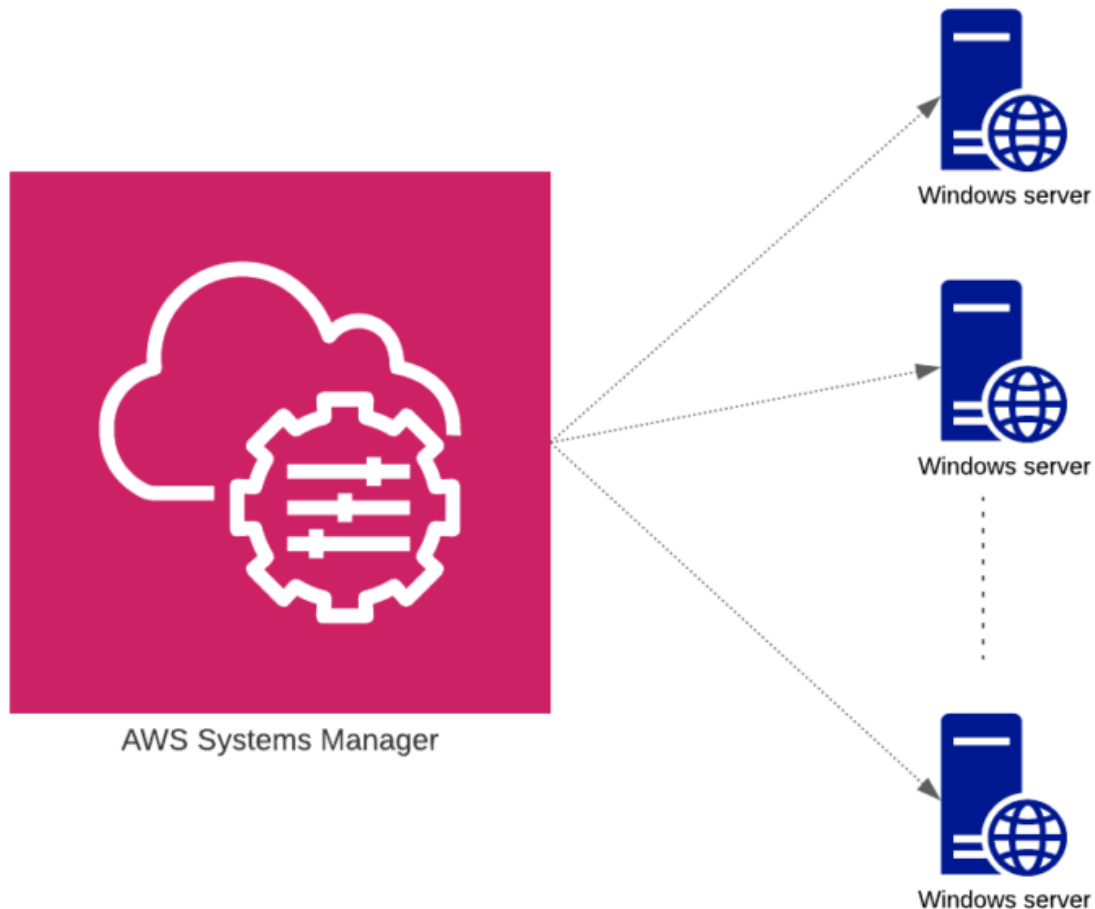


# AWS SSM - How to Manage On-Premise Windows 10 machine



## What is the AWS Systems Manager service?

AWS Systems Manager is an AWS service that can be used to view and control AWS cloud and on-premise infrastructure. By installing and configuring AWS Systems Manager Agent (SSM Agent) on an EC2 instance, an on-premise server, or a virtual machine we can update, manage and configure different software and applications.

## Use Case – How to manage a software (un)installation on Windows 10 machine using AWS Systems Manager service.

### Step 1 – Create a Hybrid Activation.

To set up servers and on-premise virtual machines (VMs) in a hybrid environment as managed instances, we need to create a managed-instance activation. After we

successfully complete the activation, we immediately receive an Activation Code and Activation ID. We specify this Code/ID combination when we install AWS Systems Manager SSM Agent on servers and VMs. The Activation Code and Activation ID provides secure access to the Systems Manager service from the managed instances.

To control and manage on-premise servers or virtual machines, we need to create a managed-instance activation. Login to AWS Console, navigate to AWS SSM service, and to hybrid activation. For this demo, we are keeping the default configuration values.

AWS Console -> AWS SSM Service -> Hybrid Activation -> Create Activation

Save the Activation Code and Activation ID to use later for setting up the SSM agent.

## Create activation

### Activation setting

Create a new activation. After you complete the activation, you receive an activation code and ID. Use the code and ID to register SSM Agent on hybrid and on-premises servers or virtual machines. [Learn more](#)

#### Activation description- *Optional*

Maximum 256 characters.

#### Instance limit

Specify the total number of servers and VMs that you want to register with AWS. The maximum is 1000.

Maximum number is 1000.



To register more than 1,000 managed instances in the current AWS account and Region, change your account settings to use advanced instances. [Learn more](#)

[Change setting](#)

#### IAM role

To enable communication between SSM Agent on your managed instances and AWS, specify an IAM role

- ☒ Use the default role created by the system  
(AmazonEC2RunCommandRoleForManagedInstances)
- ☐ Select an existing custom IAM role that has the required permissions

#### Activation expiry date

This date specifies when the activation expires. If you want to register additional managed instances after the expiry date, you must create a new activation. This expiry date has no impact on already registered and running instances.

The expiry date must be in the future, and not more than 30 days into the future

#### Default instance name- *Optional*

Specify a name to help you identify this managed instance when it is displayed in the console or when you call a List API.

Maximum 256 characters.

[Cancel](#)[Create activation](#)

☑ You have successfully created a new activation. Your activation code is listed below. Copy this code and keep it in a safe place as you will not be able to access it again.

**Activation Code** [REDACTED]

**Activation ID** [REDACTED]

You can now install amazon-ssm-agent and manage your instance using Run Command. [Learn more](#)

---

AWS Systems Manager > Activations

**Activations**

Q

ID	Description	Registered instances	Registration lir
fd233efe-de7c-49cf-9f92-1f3eac998dbc	Windows10Activation	0	1

## Step 2 – Install SSM on Windows 10 on-premise virtual machine.

Log on to the Windows Virtual Machine, and open Windows PowerShell in elevated (administrator) mode. Copy and Paste the following command block in Windows PowerShell. Replace the placeholder values with the Activation Code and Activation ID generated in step 1, and with the identifier of the AWS Region, we want to download SSM Agent from.

```
$code = "activation-code"
$id = "activation-id"
$region = "region"
$dir = $env:TEMP + "\ssm"
New-Item -ItemType directory -Path $dir -Force
cd $dir
(New-Object System.Net.WebClient).DownloadFile("https://amazon-ssm-
$region.s3.$region.amazonaws.com/latest/windows_amd64/AmazonSSMAgentSetup.e
xe", $dir + "\AmazonSSMAgentSetup.exe")
Start-Process .\AmazonSSMAgentSetup.exe -ArgumentList @("/q", "/log", "install.log",
"CODE=$code", "ID=$id", "REGION=$region") -Wait
Get-Content ($env:ProgramData + "\Amazon\SSM\InstanceData\registration")
Get-Service -Name "AmazonSSMAgent"
```

The command does the following:

- Downloads and installs SSM Agent onto the Windows 10 Virtual Machine.
- Registers the VM with the Systems Manager service.
- Returns a response to the request similar to the following.

```

PS C:\Users\abhi> $code = [redacted]
PS C:\Users\abhi> $id = [redacted]
PS C:\Users\abhi> $region = "ap-southeast-2"
PS C:\Users\abhi> $dir = $env:TEMP + "\ssm"
PS C:\Users\abhi> New-Item -ItemType directory -Path $dir -Force

Directory: C:\Users\abhi\AppData\Local\Temp

Mode                LastWriteTime         Length Name
----                -
d-----          7/6/2021 10:32 AM             ssm

PS C:\Users\abhi> cd $dir
PS C:\Users\abhi\AppData\Local\Temp\ssm> (New-Object System.Net.WebClient).DownloadFile("https://amazon-ssm-$region.s3-$region.amazonaws.com/latest/windows_amd64/AmazonSSMAgentSetup.exe",
dir + "\AmazonSSMAgentSetup.exe")
PS C:\Users\abhi\AppData\Local\Temp\ssm> Start-Process -AmazonSSMAgentSetup.exe -ArgumentList @("/q", "/log", "install.log", "CODE=$code", "ID=$id", "REGION=$region") -Wait
PS C:\Users\abhi\AppData\Local\Temp\ssm> Get-Content ($env:ProgramData + "\Amazon\SSM\InstanceData\registration")
{"ManagedInstanceID":"mi-0b0139cec2f9d9115","Region":"ap-southeast-2"}
PS C:\Users\abhi\AppData\Local\Temp\ssm> Get-Service -Name "AmazonSSMAgent"

Status  Name          DisplayName
-----  -
Running AmazonSSMAgent Amazon SSM Agent

PS C:\Users\abhi\AppData\Local\Temp\ssm>

```

The Windows VM is now a managed instance. This instance is now identified with the prefix “mi-“. We can view managed instances on the **Managed Instances** page in the Systems Manager console, by using the AWS CLI command.

```

abhishek@5CG933155Y:~$ aws ssm describe-instance-information --profile leaven
{
  "InstanceInformationList": [
    {
      "InstanceId": "mi-0b0139cec2f9d9115",
      "PingStatus": "Online",
      "LastPingDateTime": "2021-07-06T22:49:38.557000+12:00",
      "AgentVersion": "3.0.1295.0",
      "IsLatestVersion": true,
      "PlatformType": "Windows",
      "PlatformName": "Microsoft Windows 10 Pro",
      "PlatformVersion": "10.0.19042",
      "ActivationId": "fd233efe-de7c-49cf-9f92-1f3eac998dbc",
      "IamRole": "service-role/AmazonEC2RunCommandRoleForManagedInstances",
      "RegistrationDate": "2021-07-06T22:33:01.407000+12:00",
      "ResourceType": "ManagedInstance",
      "IPAddress": "10.1.0.4",
      "ComputerName": "aws-ssm.WORKGROUP",
      "AssociationStatus": "Success",
      "LastAssociationExecutionDate": "2021-07-06T22:34:25.094000+12:00",
      "LastSuccessfulAssociationExecutionDate": "2021-07-06T22:34:25.094000+12:00",
      "AssociationOverview": {
        "InstanceAssociationStatusAggregatedCount": {
          "Success": 1
        }
      }
    }
  ]
}

```

### Step 3 – Create an AWS SSM Distributor Package.

Distributor, a capability of AWS Systems Manager, helps us to package our own software to install on AWS Systems Manager managed instances. Distributor publishes resources, such as software packages, to Systems Manager managed instances.

For this demo, we have packaged our own software, Windows chrome, We can download the package and manifest file from the git [repository](#). You can learn [here](#) how to create your own custom package.

To create a distributor package, we need to copy the 1. Zip File (software package) and 2. Manifest.json (metadata) to S3 Bucket giving read permissions to AWS SSM service.

Navigate to AWS SSM service -> Distributor -> Create Package

- As we are providing our own manifest file, install and uninstall script, select the Advanced option (as shown in the image)
- Pick a name for the distributor package.
- Provide the S3 bucket name where we have copied the software package and manifest file.
- Click on “view manifest file”, it will populate the content of our manifest file from the S3 bucket.
- Create a package and wait for package creation.

## Create package

☐ Simple

Create a package, and have Distributor write your package manifest and your installation and uninstallation scripts for you.

☒ Advanced

Create a package, and provide your own installation and uninstallation scripts, and your own package manifest.

### Details

Specify a package name and version name. [Learn more](#)

Name

ChromeWindows10

Package names cannot contain special characters or spaces, and can be a maximum of 128 characters.

Version name - optional

Version names cannot contain special characters or spaces, and can be a maximum of 512 characters.

### Location

Specify the name of your bucket

☒ Choose a bucket name from a list

☐ Enter an S3 bucket URL

S3 bucket name

Choose a bucket name from a list

leaven-control-tower-customization-bucket

S3 key prefix

Specify the subfolder name

ssm

### Manifest

The package manifest provides information about the software that you are installing and which installers to use on different operating systems. [Learn more](#)

☒ Extract from package

Extract manifest from the package located in the S3 bucket provided above.

☐ New manifest

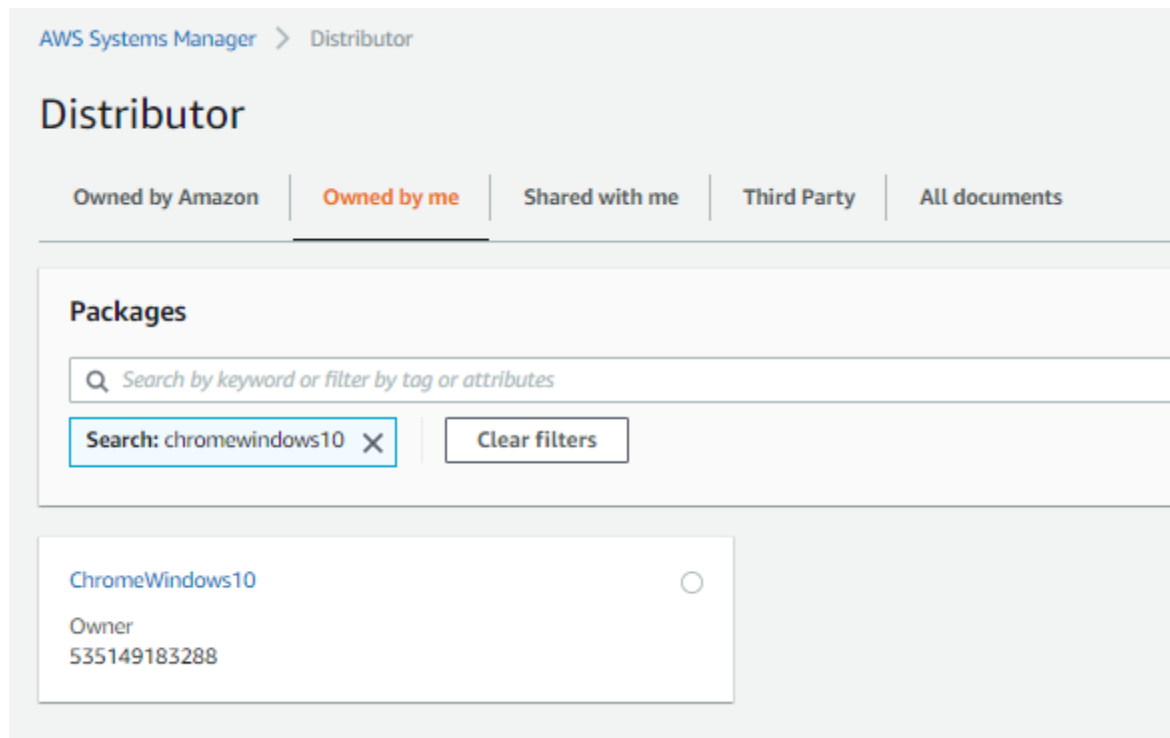
Create new manifest using the content editor.

[View manifest file](#)

[Cancel](#)

[Create package](#)

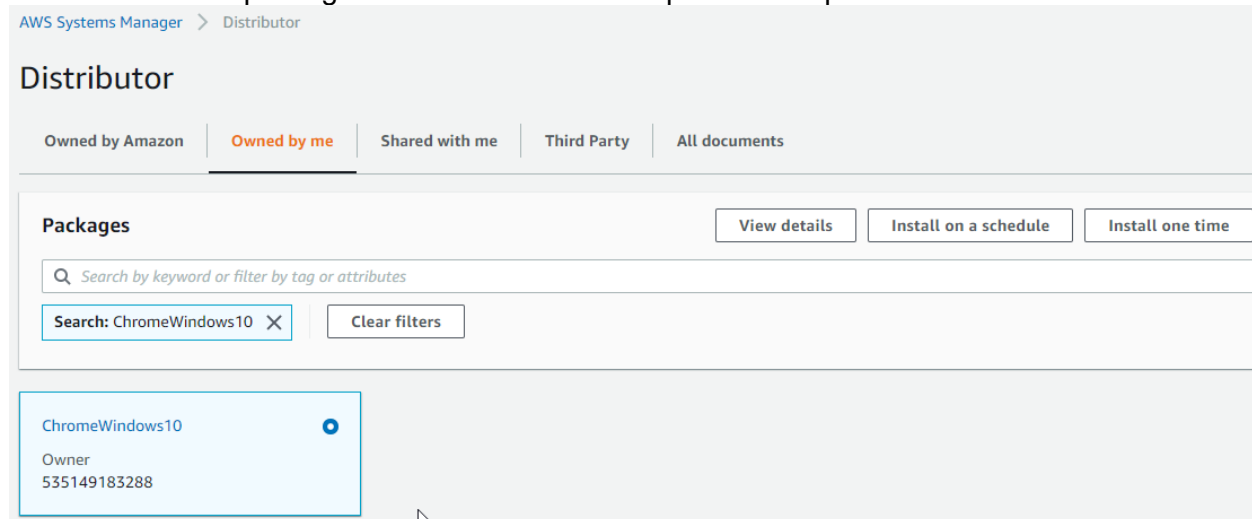
- We can view our package under the AWS SSM Distributor's "Owned by me" tab.



#### Step 4 – Install the Distributor package on Windows 10 Virtual Machine.

Navigate to AWS SSM service -> Distributor -> Owned by Me

- Select the package we have created in the previous step and click “Install one Time”.



- As we are managing a single instance for our use case, we will pick “Choose Instance Manually”. AWS recommends using tags for managing the fleet of EC2 instances and On-premise Virtual machines.



**Targets**

Targets  
Choose a method for selecting targets.

☐ Specify instance tags  
Specify one or more tag key-value pairs to select instances that share those tags.

☒ Choose instances manually  
Manually select the instances you want to register as targets.

☐ Choose a resource group  
Choose a resource group that includes the resources you want to target.

mi-0b0139cec2f9d9115 X

**Instances**

Q

<input checked="" type="checkbox"/>	Name	Instance ID	Instance state	Availability zone	Ping status	Last ping time	Agent version	Platform type
<input checked="" type="checkbox"/>	-	mi-0b0139cec2f9d9115	-	-	Online	7/7/2021 at 12:04:39 GMT+1200 (New Zealand Standard Time)	3.0.1295.0	Windows

- We have 2 options to log the command output either to S3 Bucket or to Cloudwatch logs. Here we are pushing command output logs to Cloudwtach for near real-time visibility. Press the “Run” command button.

### ▼ Output options

#### Write command output to an Amazon S3 bucket

Write all command output to an Amazon S3 bucket. Command output in the console is truncated after 2500 characters.

☐ Enable an S3 bucket

#### Send command output to Amazon CloudWatch logs

You can stream and encrypt log data for all commands in your account to a CloudWatch Logs log group in your account. [Learn more](#)

☒ Enable CloudWatch logs

#### Log group name - optional

You can specify the name of your log group. Log groups are groups of log streams that share the same retention, monitoring, and access control settings.

/aws/ssm/ChromeInstall

Wait for the AWS SSM command to execute on our on-premise instance. Google Chrome is installed successfully on our windows 10 machine

AWS Systems Manager > Run Command > Command ID: a70adc8a-0183-407b-b630-33b40f17a654

Command ID: a70adc8a-0183-407b-b630-33b40f17a654 Refresh Cancel command

**Command status**

Overall status	Detailed status	# targets	# completed	# error	# deliv
<span>In Progress</span>	<span>In Progress</span>	1	0	0	0

**Targets and outputs**

Q

	Instance ID	Instance name	Status	Detailed Status	Start time
<input type="radio"/>	mi-0b0139cec2f9d9115		<span>In Progress</span>	<span>In Progress</span>	

AWS Systems Manager > Run Command > Command ID: a70adc8a-0183-407b-b630-33b40f17a654

Command ID: a70adc8a-0183-407b-b630-33b40f17a654 Refresh Cancel command Rerun

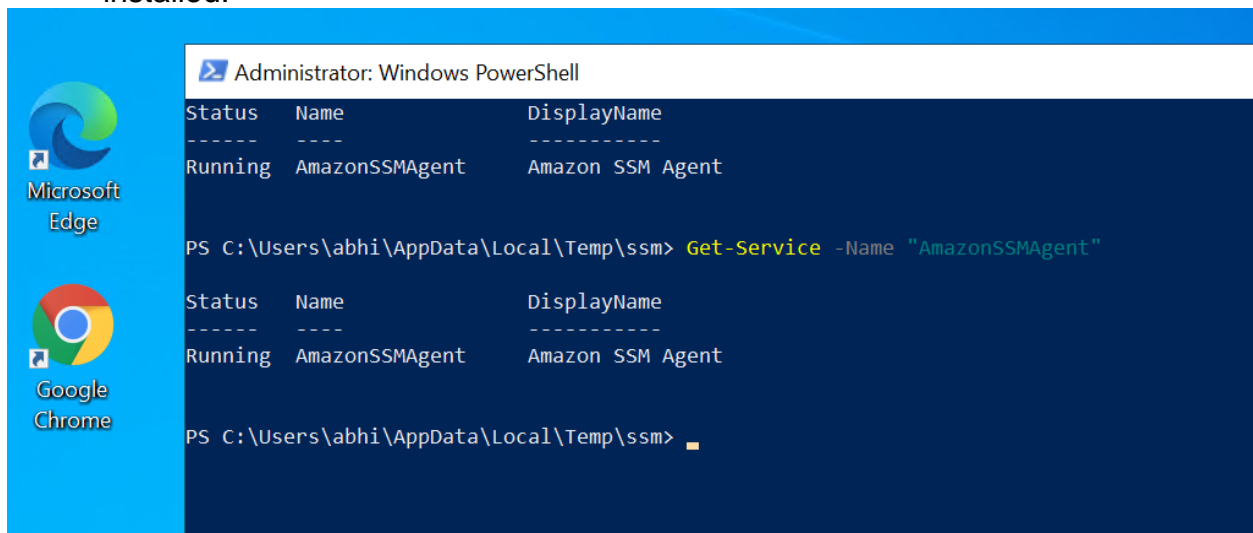
**Command status**

Overall status	Detailed status	# targets	# completed	# error	# delivery timed out
Success	Success	1	1	0	0

**Targets and outputs**

Instance ID	Instance name	Status	Detailed Status	Start time	Finish time
mi-0b0139cec2f9d9115		Success	Success	Wed, 07 Jul 2021 00:31:14 GMT	Wed, 07 Jul 2021 00:31:42 GMT

- Log on to the Windows virtual machine and validate if the chrome browser is installed.



- We can view the installation logs in the AWS Cloudwatch log group.

CloudWatch > Log groups > /aws/ssm/ChromeInstall > a70adc8a-0183-407b-b630-33b40f17a654/mi-0b0139cec2f9d9115/configurePackage/stdout

**Log events**

You can use the filter bar below to search for and match terms, phrases, or values in your log events. [Learn more about filter patterns](#)

Filter events

Timestamp	Message
	No older events at this moment. <a href="#">Retry</a>
2021-07-07T12:38:22.529+12:00	Initiating ChromeWindows10 2.0.0 install Plugin aws:runPowerShellScript ResultStatus Success install outp install output: Running install.ps1 Installing ExamplePackage on Windows... Installing ExamplePackage on Windows... Successfully installed ChromeWindows10 2.0.0
	No newer events at this moment. Auto <a href="#">retry</a> paused. <a href="#">Resume</a>

- To Uninstall the chrome package from the windows machine, run the command again, but this time with Command Parameters “Action” as Uninstall. (Run Command will read the uninstall.ps1 script from our Distributor package and take appropriate action.)

**Command parameters**

**Action**  
(Required) Specify whether or not to install or uninstall the package.

Uninstall ▼

**Name**  
(Required) The package to install/uninstall.

ChromeWindows10

**Version**  
(Optional) The version of the package to install or uninstall. If you don't specify a version, the system installs the latest published version by default. The system will only attempt to uninstall the version that is current system returns an error.

## **Summary**

AWS Systems Manager is a powerful tool, with AWS SSM we can manage AWS EC2 instances, on-premise servers, or Virtual machines at scale. AWS SSM Distributor package and Run Command improves operational efficiency and give greater control to manage the software installations on Linux and Windows Operating Systems.

Further utilizing services like AWS Codepipeline, CodeCommit, and Code Deploy, we can automate the Software installation and Uninstallation process for a fleet of cloud and on-premise servers.