AWS Systems Manager Operational Capabilities

Effectively manage operational tasks using AWS Systems Manager

December 2020
Notices

Customers are responsible for making their own independent assessment of the information in this document. This document: (a) is for informational purposes only, (b) represents current AWS product offerings and practices, which are subject to change without notice, and (c) does not create any commitments or assurances from AWS and its affiliates, suppliers or licensors. AWS products or services are provided “as is” without warranties, representations, or conditions of any kind, whether express or implied. The responsibilities and liabilities of AWS to its customers are controlled by AWS agreements, and this document is not part of, nor does it modify, any agreement between AWS and its customers.

© 2020 Amazon Web Services, Inc. or its affiliates. All rights reserved.
# Contents

Introduction ................................................................................................................. 1  
What is AWS Systems Manager? .................................................................................. 1  
Who can use Systems Manager? ................................................................................. 1  
AWS Systems Manager Components Deep Dive ....................................................... 3  

Operations Management ............................................................................................ 3  
  Systems Manager OpsCenter ...................................................................................... 4  
  Systems Manager Explorer ......................................................................................... 5  
  Amazon CloudWatch dashboards ............................................................................... 7  
  Trusted Advisor and Personal Health dashboards .................................................... 8  

Application Management ............................................................................................ 9  
  Resource Group ........................................................................................................ 9  
  AppConfig .................................................................................................................. 9  
  Parameter Store ........................................................................................................ 10  

Actions and Change .................................................................................................... 11  
  Automation ............................................................................................................... 12  
  Maintenance Windows .............................................................................................. 13  
  Change Calendar ....................................................................................................... 14  

Instances and Nodes .................................................................................................... 15  
  Configuration Compliance ......................................................................................... 16  
  Inventory .................................................................................................................. 16  
  Managed Instance ..................................................................................................... 17  
  Hybrid Activations .................................................................................................... 18  
  Session Manager ...................................................................................................... 18  
  Run Command .......................................................................................................... 19  
  State Manager .......................................................................................................... 19  
  Patch Manager .......................................................................................................... 20
Abstract

With the number of growing Amazon Web Services (AWS) services, developers, DevOps leads, and system administrators should be focusing on operational integration. This whitepaper provides an overview of several capabilities of AWS Systems Manager and explains how you can effectively use this service to meet your operational needs. The paper also helps you organize and manage your AWS services or your hybrid environments (on-premise servers or virtual machines) from one central place. We will cover the broad categorization of various Systems Manager offerings, and dive into details on each of the individual components within these categories.
Introduction

What is AWS Systems Manager?

AWS Systems Manager is an AWS service that you can use to view, manage, and control your infrastructure on multiple AWS services. Since its launch, Systems Manager has evolved at such a rapid pace that you can not only view and perform operational tasks but also automate operations on multiple AWS services. Systems Manager enables visibility and control of your cloud and on-premises infrastructure. It simplifies resource and application management, shortens the time to detect and resolve operational problems, and makes it easier to operate and manage your infrastructure securely at scale.

Who can use Systems Manager?

Systems Manager is a service designed to make it simple for multiple roles to perform various operations on managed resources such as Amazon Elastic Compute Cloud (Amazon EC2) instances. This service can be used by system administrators, software developers, security architects, cloud architects, and IT professionals who would like to manage AWS resources.

The capabilities of Systems Manager can be categorized into five key areas:

- Operations Management
- Actions and Change
- Instances and Nodes
- Application Management
- Shared Resources
Figure 1 - AWS Systems Manager Overview
AWS Systems Manager Components Deep Dive

In the following sections, we discuss each of the capabilities in brief, covering a few examples for some of them. Before we get into the each of these capabilities, here are some of the AWS Systems Manager features you should keep in mind as you highlight best practices for your business needs:

- To help improve your security posture, you can use AWS Systems Manager through AWS Privatelink. This enables you to privately access services hosted on AWS, without requiring the traffic to traverse the internet.

- Using AWS Systems Manager Quick Setup, you can enable operational actions like patch compliance scanning and instance inventory collection across all instances and accounts for your AWS Organization with a single click. Quick Setup helps you configure the required security roles (IAM role) and other important Systems Manager capabilities on your managed instances, such as scanning for missing patches, updating SSM Agent, installing Amazon CloudWatch Agent, and collecting inventory data.

- AWS Systems Manager offers a wide variety of integrations with other AWS services across various areas like Compute, Storage, Security, and Analytics.

- AWS Systems Manager is also available in GovCloud regions for regulated customers to take advantage of various features that can help automatically collect software inventory, apply OS patches, create system images, and configure Windows and Linux operating systems.

- Application configuration and deployment without code deployments with AWS AppConfig and Parameter Store.

Operations Management

Operations Management is a suite of capabilities that helps you keep track of your AWS resources across AWS Regions and accounts. These capabilities can help you effectively manage your AWS resources.

Operations Management offers four main capabilities:

- AWS Systems Manager OpsCenter
- Systems Manager Explorer
• Amazon CloudWatch dashboards
• Trusted Advisor and Personal Health Dashboards

**Systems Manager OpsCenter**

**OpsCenter** is the central location for operations engineers and system administrators where they can view, track, investigate, and resolve operational work items (OpsItems) related to AWS resources. This Systems Manager capability aggregates and standardizes OpsItems across services while providing contextual investigation data about each OpsItem, related OpsItems, and related resources. OpsCenter also provides Systems Manager automation documents (runbooks) that you can use to quickly resolve issues.

Here are some examples of Systems Manager OpsItems that can be automatically created through Amazon CloudWatch or Amazon EventBridge events:

• Security issues, such as alerts from [AWS Security Hub](https://aws.amazon.com/securityhub)
• Performance issues, such as a throttling event for [Amazon DynamoDB](https://aws.amazon.com/dynamodb) or degraded [Amazon Elastic Block Store](https://aws.amazon.com/ebs) (Amazon EBS) volume performance
• Failures, such as an [Amazon EC2 Auto Scaling](https://aws.amazon.com/ec2autoscaling) group failure to launch an instance or a Systems Manager automation execution failure
• Health alerts, such as an [AWS Health](https://aws.amazon.com/health) alert for scheduled maintenance
• State changes, such as an Amazon EC2 instance state change from *running* to *stopped*

The following screen shows a summary of OpsCenter that can track all of the *open* and *in progress* items at one place. Clicking on the second tab shows the details of OpsItems that you can open and take actions upon, such as executing an automation document to address a specific issue.
OpsCenter integrates with Amazon EventBridge (EventBridge was formerly called Amazon CloudWatch Events) by enabling you to automatically create OpsItems to address a number of issues. You can also manually create OpsItems, and OpsCenter provides automation runbooks for quickly remediating those issues.

**Systems Manager Explorer**

AWS Systems Manager Explorer is a customizable operations dashboard that reports information about your AWS resources. It gives you an aggregated view of operations data (OpsData) for all your AWS accounts and across AWS Regions. In Explorer, OpsData includes metadata about your managed instances, patch compliance details, and operational work items (OpsItems) as well as your desired state of compliance, summary of Premium support cases opened, Trusted Advisor checks, and other operational details.

While there is some overlap in functionalities between Explorer and OpsCenter, these two features also relate to each other. Systems Manager OpsCenter provides a central...
location where operations engineers can view, investigate, and resolve OpsItems related to AWS resources. Explorer is a report hub where DevOps managers can view aggregated summaries of their operations data, including OpsItems. Data across AWS Regions and accounts can be aggregated in Explorer and OpsCenter is now integrated with Explorer. If you have already set up OpsCenter, Explorer automatically displays operations data, including aggregated information about OpsItems. If you have not set up OpsCenter, you can use Explorer setup to get started with both capabilities.

The following screenshots show some of the pre-built widgets Explorer has to offer. Widgets offer the ability to drill down to gain more insight. You can also filter information based on AWS account, Region, OpsItem source, and tag.

![Instance count](image1)
![Managed instances](image2)
![Instance by AMI](image3)

![Open OpsItem summary](image4)
![OpsItems over time](image5)

**Figure 3 – Systems Manager Explorer (1)**
Amazon CloudWatch dashboards are customizable home pages in the CloudWatch console. They let you access and view data from Systems Manager to centrally monitor your resources, including resources in different Regions. You can use CloudWatch dashboards to create customized views of the metrics and alarms for your AWS resources.

The image below shows that from a single overview, you are able to access metrics and health information for various services and applications across one or more Regions.

Figure 4 - Systems Manager Explorer (2)
Trusted Advisor and Personal Health dashboards

These tools help you monitor different aspects of the health of your resources. Trusted Advisor is an online tool that provides real-time guidance to help you provision your resources following AWS best practices. The AWS Personal Health dashboard provides information about AWS Health events that can affect your account. The information is presented in two ways: a dashboard that shows recent and upcoming events organized by category, and a full event log that shows all events from the past 90 days.
Application Management

Application Management offers capabilities that help you manage and run your applications efficiently in AWS, including Resource Groups, Parameter Store, and AppConfig.

Resource Group

A Resource Group is a collection of AWS resources that are all in the same AWS Region, and that match criteria based on either tags or AWS CloudFormation stacks. This feature is used for grouping various resources such as a set of managed instances or a set of Route 53 hosted zones. You can then use Automations or Run Command to take actions on this set of resources. For information on what AWS Services are supported, see supported resources in the AWS Resource groups user guide.

In this blog post, you can see how Resource Groups can assist in day-to-day tasks. If you manage a large number of related resources such as EC2, you can perform bulk actions on these resources.

AppConfig

AWS AppConfig is used to create, manage, and quickly deploy application configurations with reduced errors (without an application restart) and which are separated from the application code. AppConfig can perform controlled deployments to applications of any size that are hosted on Amazon EC2 instances, AWS Lambda, containers, mobile applications, or IoT devices.

Common use cases for AppConfig:

- Application tuning
- Feature toggle for ability to switch off and on new application features post-deployment
- Isolate operational issues for application dependencies
- User membership list for premium services or features

AppConfig offers a feature called validators that can be used to validate your application configuration. Validation can be utilized to ensure configurations are syntactically and
semantically correct. For syntactic validation, a JSON schema can be utilized to ensure configuration changes adhere to the application requirements. For semantic validation, an AWS Lambda function can be invoked prior to the configuration’s deployment.

To set up an application configuration, perform the following steps:

1. Define the application name.
2. Define the environments where the configuration is deployed.
3. Create the configuration to be used, along with a configuration profile.
4. Create the deployment strategy for configuration rollout to determine how the configuration will be rolled out to the targets.

AppConfig can monitor the configuration rollout with the ability to trigger a rollback in case of errors using CloudWatch alarm. See Creating a configuration and a configuration profile in the AWS documentation for a simple example of creating an access list configuration that can be stored in Amazon S3, Systems Manager Documents, or Systems Manager Parameter Store.

**Parameter Store**

Parameter Store is a hierarchical storage for secrets management and configuration data management. Parameter Store can be used to look up centralized configuration for CloudFormation templating or application configuration.

Parameter Store includes standard tier parameters and advanced tier parameters. You can individually configure parameters to use either the standard-parameter tier (the default tier) or the advanced-parameter tier. A standard parameter can be transitioned to an advanced parameter at any time, but you can’t revert an advanced parameter to a standard parameter. Reverting an advanced parameter to a standard parameter would result in data loss because the system would truncate the size of the parameter. If you have unknown or changing patterns of parameter count, value size, or parameter policies, you can also use the intelligent-tiering setting to allow Parameter Store to select the standard or advanced tier for you. For more information about tiering as per your requirements, see managing parameter tiers.

Parameter Store has a default limit for requests per second. If there is a need for higher throughput requirements to Parameter Store, you can enable the higher throughput limit from the Parameter Store Settings tab or AWS CLI. Once the higher throughput is enabled for your account, charges will be incurred per API interaction. A Parameter
Store API interaction is defined as an interaction between an API request and an individual parameter. For more information, see AWS Systems Manager Pricing.

Parameter Store has a number of parameter types to support multiple use cases. String and StringList can be used for configuration data and are not meant for storing sensitive data. SecureString is a parameter type for sensitive data and is encrypted using AWS KMS.

Common use cases for String and StringList Parameters:

- Store configuration data about an Amazon EC2 instance IDs
- Store data on the data on the latest AMI’s
- Store license codes for third party software
- Store non-sensitive centralized configuration data

Common use cases for SecureString Parameters:

- Ability to use data/parameters across AWS services without exposing the values as plain text in commands, functions, agent logs, or AWS CloudTrail logs
- Control who has access to sensitive data
- Ability to audit when sensitive data is accessed via AWS CloudTrail
- Ability to encrypt your sensitive data and bring your own encryption keys to manage access

There are AWS Service teams that publish artifacts as public parameters. Public parameters can be utilized to get the latest Amazon Linux AMI IDs and latest Window AMI IDs as well as AWS services, Regions, endpoints, availability zones, and local zones.

Note: Parameter Store can be used with Secrets Manager for the ability to access secrets. This will allow for Secrets Manager to rotate your secrets and access them via Parameter Store.

Actions and Change

Actions and Change covers capabilities that help you manage actions or change the configuration of your AWS resources in bulk-based groups; for example, on tags such
as application, environment, Region, project, campaign, business unit, or software lifecycle.

**Automation**

Automating repeatable tasks is important for removing undifferentiated heavy lifting so your teams can focus on business development. This feature simplifies the automation of common maintenance and deployment tasks of Amazon EC2 instances and other AWS resources, enabling you to do the following tasks:

- Automate common IT tasks like stopping or restarting multiple servers with approval
- Automate workloads for AWS Multi-Account or AWS Multi-Region
- Simplify complex tasks like creating golden Amazon Machines Images (AMIs) and recovering unreachable EC2 instances
- Enhance operations security using delegated administration to allow a particular user to execute such automation documents through IAM permissions
- Run automation as an Amazon EventBridge target to perform a task-based operation on the event, such as scheduling, infrastructure state changes, or completion of another task
- Monitor automation progress and execution details by using the AWS Systems Manager console
- Centralize configuration for application and AWS services

A Systems Manager Automation document defines the actions that Systems Manager performs on your managed instances and other AWS resources when an automation execution runs. A document contains one or more steps that run in sequential order or dynamically branch based on the results of the previous step. Each step is built around a single action. Output from one step can be used as input in a later step. The process of running these actions and their steps is called the automation workflow. For more information about lists of all the supported automated actions that can be used in your automation documents and workflows to either run custom Python scripts, PowerShell scripts, or multiple other use cases, see Systems Manager action reference.
• We recommend that you take time in reviewing the list of Systems Manager automation documents from AWS and AWS Support. These cover a number of common use cases and provide best practices in areas such as security, patching, remediation, resource and cost management, data backups, and more.

**Maintenance Windows**

**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. Maintenance Windows also let you schedule actions on numerous AWS resource types, including Amazon Simple Storage Service (Amazon S3) buckets, Amazon Simple Queue Service (Amazon SQS) queues, AWS Key Management Service (AWS KMS) keys, and many more.

Maintenance Windows consist of a schedule, a maximum duration, a set of registered targets (the instances or other AWS resources that are acted upon), and a set of registered tasks. Maintenance Windows can be specific dates that the maintenance should not run before or after, and you can specify the international time zone on which to base the maintenance window schedule.

Maintenance Windows support running the following tasks:

• AWS Lambda functions
• AWS Step Functions tasks
• Automation workflows
• Run Command tasks

![Figure 7 - Systems Manager Maintenance Windows](image-url)
Common use cases for Maintenance Windows:

- Install or update applications
- Apply patches
- Install or update SSM Agent
- Run PowerShell commands and Linux shell scripts using a Systems Manager Run Command task
- Build Amazon Machine Images (AMIs), boot-strap software, and configure instances using a Systems Manager Automation task
- Run AWS Lambda functions that trigger additional actions, such as scanning your instances for patch updates
- Run AWS Step Functions state machines to perform tasks such as removing an instance from an Elastic Load Balancing environment, patching the instance, and then adding the instance back to the Elastic Load Balancing environment
- Target instances that are offline by specifying an AWS resource group as the target

**Note:** Maintenance Windows supports scheduling of maintenance tasks on an offset from a specific day in a specific week of the month. For example: Microsoft’s patches are currently released on the second Tuesday of the month. To apply these patches, add the offset for the chosen day following Microsoft’s patch Tuesday.

**Change Calendar**

*Change Calendar* lets you set up date and time ranges when actions you specify, such as executing Systems Manager Automation documents, may or may not be performed in your AWS account. These ranges are called events.

Change Calendar entries help keep your environment stable during event times. For example, you can [create a Change Calendar entry](#) to block changes when you expect high demand on your resources, such as during a conference or a marketing promotion. A calendar entry can also block changes when you expect limited administrator support, for example during vacations or holidays. In the following screenshot, you see an
example of a Change Calendar event created for a month-end freeze to block any deployments during this period.

![Image of Change Calendar](image)

**Figure 8 - Systems Manager Change Calendar**

Change Calendar can effectively control your environments and avoid disruptions of all the business operations. Change Calendar helps you with reviewing planned changes, ensures execution of such changes only during appropriate times, and gets the current or upcoming state of the calendar.

**Note**: Calendars can be *shared* across AWS Accounts. This will provide a single source of truth of when events are allowed/disallowed.

## Instances and Nodes

Instances and Nodes provides multiple capabilities for managing Amazon EC2 instances, on-premises servers or virtual machines (VMs) in a hybrid environment, as well as other types of AWS resources. You can apply patches, manage inventory, manage sessions, etc. This category offers the following capabilities:

- **Configuration Compliance**
- **Inventory**
- **Managed Instances**
- **Hybrid Activations**
- **Session Manager**
- **Run Command**
• State Manager
• Patch Manager
• Distributor

**Configuration Compliance**

*Configuration Compliance* lets you scan your fleet of managed instances for patch compliance and configuration inconsistencies. You can collect and aggregate data from multiple AWS accounts and Regions to drill down into specific non-compliant resources and help you meet your compliance needs. You can also take remediation actions using Systems Manager Run Command, State Manager, Automation, or Amazon CloudWatch Events.

Other important benefits of Configuration Compliance:

- Use AWS Config to view compliance history and change tracking for Patch Manager patching data, State Manager associations, and Inventory data
- Customize Systems Manager compliance to create your own compliance types based on your IT or business requirements
- Port data to [Amazon Athena](https://aws.amazon.com/athena/) and [Amazon QuickSight](https://aws.amazon.com/quicksight/) to generate fleet-wide reports

**Note:** Systems Manager also integrates with [Chef InSpec](https://www.chef.io/inspec/). InSpec is an open-source, run-time framework that enables you to create human-readable profiles on GitHub or Amazon S3. You can then use Systems Manager to run compliance scans and view compliant and non-compliant instances.

**Inventory**

*Inventory* provides visibility into your Amazon EC2, on-premise, and other cloud computing environments, capturing all the metadata from managed instances across multiple AWS Regions and accounts. Inventory does not access proprietary information or data. You can store this metadata in a central Amazon S3 bucket, and then use built-in tools to query and analyze the data. You can use this data to quickly determine which instances are running the software and configurations required by your software policy, and which instances need to be updated.
In this **walkthrough**, you can see how to get your fleet data using AWS SSM Custom inventory types from across all of your accounts. You can also use your inventory data in combination with AWS Config to detect any prohibited applications installed on your managed instances as shown here. You can find more walkthroughs here to setup collection of inventory data.

If the pre-configured metadata types collected by Systems Manager Inventory don’t meet your specific IT or business needs, you can create custom inventory. Custom inventory is a JSON file with information that you provide and add to the managed instance in a specific directory. When Systems Manager Inventory collects data, it captures this custom inventory data.

### Managed Instance

A **managed instance** is any machine configured for AWS Systems Manager to execute different types of actions. You can configure Amazon EC2 instances, on-premises machines, and other cloud resources in a hybrid environment as managed instances. Systems Manager supports various distributions of Linux, including Raspberry Pi devices and Microsoft Windows Server. In addition, you can use Systems Manager to **restrict access to root-level commands** through SSM Agent, which can help tighten your security posture against unauthorized root-level commands on your managed instances.

AWS Systems Manager offers a standard-instances tier and an advanced-instances tier for servers and VMs in your hybrid environment.

### Notes:

Advanced instances tier also enable you to connect to your hybrid machines by using AWS Systems Manager Session Manager, which provides an interactive shell to your instances without requiring any open ports.

With the advanced instances tier, you can perform Microsoft application patching on your on-premise instances as well, which is otherwise available for EC2 instances.

The standard-instances tier enables you to register a maximum of 1,000 servers or VMs per AWS account per AWS Region. If you need to register more than 1,000 servers in a single account and Region, use the advanced-instances tier.
Hybrid Activations

To set up servers and virtual machines (VM) in your hybrid environment as managed instances, you can create a managed-instance hybrid activation. With the completion of the activation, you receive an activation code and activation ID. This code and ID combination functions like an Amazon EC2 access ID and secret key to provide secure access to the Systems Manager service from your managed instances.

Session Manager

Session Manager is a fully managed AWS Systems Manager capability that allows you to connect to and manage your Amazon EC2 instances or your hybrid instances (with Adv Tier) through an interactive one-click browser-based shell or through the AWS CLI. Session manager provides a secure shell connection right on your browser, eliminating the need to open inbound ports and multiple other benefits mentioned below:

- **Centralized access control to instances using IAM policies**: Administrators have a single place to grant and revoke access to instances. Using only AWS IAM policies, you can control which individual IAM users or groups in your organization can use Session Manager, and which instances they can access.

- **No open inbound ports and no need to manage bastion hosts or SSH keys**: Leaving inbound SSH ports and remote PowerShell ports open on your instances greatly increases the risk of entities running unauthorized or malicious commands on the instances. Session Manager helps improve your security posture by letting you close these inbound ports. It also frees you from managing SSH keys and certificates, bastion hosts, and jump boxes.

- **One-click access to instances from the console and AWS CLI**: Using the AWS Systems Manager console or Amazon EC2 console, you can start a session with a single click. Using the AWS CLI, you can also start a session that runs a single command or a sequence of commands. Because permissions to instances are provided through IAM policies instead of SSH keys or other mechanisms, the connection time is greatly reduced.

- **Port forwarding**: Redirect any port inside your remote instance to a local port on a client. Next, connect to the local port and access the server application running inside the instance.
• **Cross-platform support for both Windows and Linux:** Session Manager provides both Windows and Linux support from a single tool. For example, you don’t need to use an SSH client for Linux instances or an RDP connection for Windows instances.

• **Logging and auditing session activity:** To meet operational or security requirements in your organization, you may need to provide a record of the connections made to your instances and the commands that were run on them. You can also receive notifications when an IAM user in your organization starts or ends session activity.

You can replace all of the SSH accesses in your organization using this feature, as explained here. You can also configure session manager access for federated users using SAML session tags with your identity provider as explained in this blog post.

**Run Command**

*Run Command* enables you remotely and securely manage the configuration of your managed instances. *Run Command* also helps you automate common administrative tasks and perform configuration changes at scale. These tasks include installing or bootstrapping applications, building a deployment pipeline, and joining instances to a Windows domain. You can capture log files when an instance is terminated from an Auto Scaling group and use AWS CLI or PowerShell to execute *Run Command*. You can also re-execute a previous command exactly as before by using the rerun command feature: select the desired command from your command history and click either “rerun” or “copy-to-new,” which automatically copies all of the parameters and controls from the selected command into the new command. *These walkthroughs* will help you use this feature to execute AWS CLI commands on your managed instances remotely.

**State 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.

By performing the following four steps, you can use State Manager to get your managed instances to your desired state:
5. Determine the state you want to apply to your managed instances.

6. Determine whether a pre-configured Systems Manager document would help you create the State Manager association.

7. Create the association.


A State Manager association is a configuration assigned to your managed instances. The configuration defines the state that you want to maintain in your instances. For example, an association can specify that antivirus software must be installed and running on your instances, or that certain ports must be closed. The association specifies a schedule for when the configuration is reapplied. The association also specifies actions to take when applying the configuration. For example, an association for antivirus software might run once a day. If the software is not installed, then State Manager installs it. If the software is installed but the service is not running, the association might instruct State Manager to start the service.

Common use cases for State Manager:

- Deploy complex Ansible playbooks at scale
- Join Windows instances to a Windows domain
- Bootstrap instances with specific software at start-up
- Download and update agents on a defined schedule, including Systems Manager Agent
- Configure network settings

**Patch Manager**

AWS Systems Manager Patch Manager automates the process of patching managed instances with security and other types of updates. You can use Patch Manager to apply patches for both operating systems and applications. You can patch fleets of Amazon EC2 instances or your on-premises servers and VMs by operating system type, including supported versions of Windows Server, Ubuntu, Red Hat Enterprise Linux (RHEL), SUSE Linux Enterprise server (SLES), CentOS, Debian, Oracle Linux, Amazon Linux, and Amazon Linux 2 (on most of the latest versions). For more information, see Patch Manager prerequisites. You can scan instances to see a report of missing patches, or you can scan and automatically install all missing patches.
Note: On Windows Server, application support is limited to updates for Microsoft applications. Microsoft application patching is only available on EC2 instances and in the advanced-instances tier. To patch Microsoft applications on on-premises servers and VMs, you must enable the advanced-instances tier.

The primary focus of Patch Manager is on installing operating systems security-related updates on instances. By default, Patch Manager doesn’t install all available patches, but rather a smaller set of patches focused on security. In addition, Patch Manager integrates with AWS IAM, AWS CloudTrail, Amazon EventBridge, and AWS Security Hub to provide a secure patching experience that includes event notifications and the ability to audit usage.

The blog post for Windows and Linux workloads walks you through the setup of Patch Manager including the use of Maintenance Windows. Note that Patch Manager allows you to either only scan or scan and immediately install patches to keep your systems updated. You can either patch your instances on a schedule or on-demand by creating a patching configuration.

Note: AWS Systems Manager now supports on-demand patching with just two clicks.

With Patch Manager, you have the option to defer rebooting your instance after patch installation to a later time to avoid disruptions to your applications or jobs running on the instance.

**Distributor**

Distributor helps customers who want to create new, or deploy existing, software packages, including AWS-published packages, to multiple Systems Manager managed instances at one time. This tool is useful for administrators who are responsible for keeping managed instances current with the most up-to-date software packages according to their organizations’ standards. When you choose simple package creation in the Distributor console, Distributor generates the installation and uninstallation scripts, file hashes, and the JSON package manifest for you, based on the software-executable file name and target platforms and architectures.
Distributor lets you create packages for a number of operating systems including both Windows and Linux. You can choose to deploy packages one time, on a regular schedule, or whenever the default package version is changed to a different version. To install a new package version, you can completely uninstall the current version and install a new one in its place, or only update the current version with new and updated components, according to an update script that you provide.

Here is an example walkthrough of how you can use Distributor to package and distribute a monitoring agent called Datadog.

Shared Resources

Shared Resources cover capabilities AWS Systems Manager documents that can be shared across accounts and AWS Regions for managing and configuring your AWS resources. As the name suggests, these help with sharing the most commonly used resources across AWS services.

Documents

Documents can be used to define the actions to be performed by Systems Manager in executing one of the aforementioned tasks, such as executing specific Run Command tasks on a managed instance. You could also use it to run an Automation document to perform automation of certain scripts in a group of managed instances. Documents enable you to share these automations across accounts, make them public, or use them privately within a specific account.

Systems Manager Document Types

The following are Systems Manager document types that can be used as a shareable resource within Systems Manager:

- **Command document** can be used with Run Command, State Manager, and Maintenance Windows on one or more targets at a lifecycle of an instance to apply configuration.

- **Automation document** can be used with Automation, State Manager, and Maintenance Windows to perform common maintenance and deployment tasks.

- **Package document** can be used with Distributor and includes attached ZIP archive files that contain software or assets to install on managed instances.
• **Session document** can be used with [Session Manager](#) containing type of session, port forwarding session, SSH tunnel, or run as an interactive command.

• **Policy document** can be used with [State Manager](#) and can be used in conjunction with Systems Manager inventory to collect inventory data from a managed instance.

• **Change Calendar document** can be used with [Change Calendar](#) to store a calendar entry and associated events that can allow or prevent Automation actions from changing your environment.

**Conclusion**

Since its launch, AWS Systems Manager has evolved tremendously to serve not just system administrators, but also developers and other IT roles. You can use it to automate various types of tasks and help keep all of your applications and managed instances healthy, secure, and updated.

**Contributors**

Contributors to this document include:

• Umesh Kumar Ramesh, Sr. Cloud Infrastructure Architect, AWS ProServe

• Jared Sutherland, Cloud Application Architect, AWS ProServe

**Document Revisions**

<table>
<thead>
<tr>
<th>Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>December</td>
<td>First Publication</td>
</tr>
<tr>
<td>2020</td>
<td></td>
</tr>
</tbody>
</table>