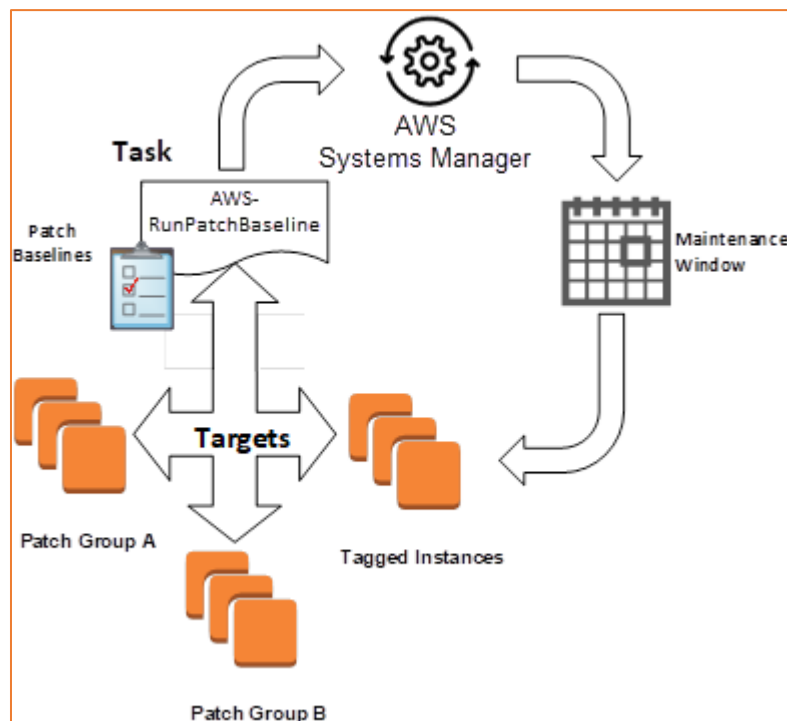


Patching your Windows EC2 instances using AWS Systems Manager Patch Manager

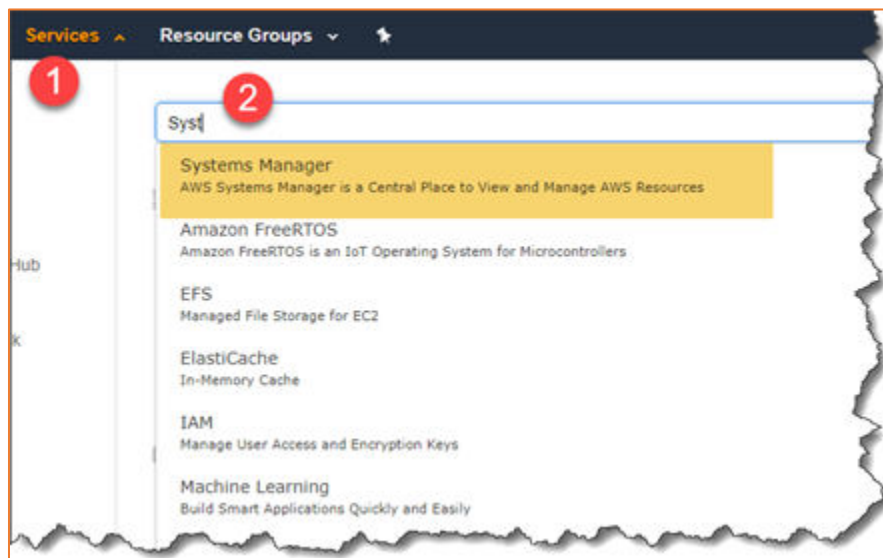
Patch Manager automates the process of patching Windows and Linux managed instances. Use this feature of AWS Systems Manager to scan your instances for missing patches or scan and install missing patches. You can install patches individually or to large groups of instances by using Amazon EC2 tags.

In this blog post, I show you how to use patch baselines to include rules for auto-approving patches within days of their release, as well to see a list of approved and rejected patches. I'll demonstrate how to leverage patch groups to organize instances for patching. For example, you can create patch groups for different environments/tagged instances such as development, test, and production. I'll show you how you can install patches on a regular basis by scheduling patching to run as a Maintenance Windows task. Don't worry, by the end of this blog you will have a good understanding of what is shown in the following diagram.



In this blog post, I'll show you how to run Patch Manager in the console, and then follow with how to use CLI commands to do the same thing. To use the AWS Systems Manager console, you need an AWS Account so you can leverage the available AWS services.

In the AWS Management Console, open the AWS Systems Manager console by choosing Services on the top menu (1), and then starting to type the service console you want in the search bar (2).



Using patch baselines

A patch baseline defines which patches should and shouldn't be installed on your instances. You can individually specify approved or rejected patches, or you can use auto-approval rules to specify that certain types of updates (for example, critical updates), should automatically be approved for patching.

Patch Manager has a pre-defined patch baseline that approves all patches classified as critical updates or security updates with a severity of Critical or Important. These patches are automatically approved by this baseline seven days after they are released by Microsoft.

AWS Systems Manager > Patch Manager

Patch Baselines [View Windows patches](#) [View details](#) [Edit](#) [Delete](#) [Actions](#) [Create patch baseline](#)

Search: < 1 >

	Baseline ID	Baseline name	Description	Operating system	Default baseline
<input type="radio"/>	pb-03e3f588ec25344c	AWS-CentOSDefaultPatchBaseline	Default Patch Baseline for CentOS Provided by AWS.	CentOS	✓ Yes
<input type="radio"/>	pb-07d8884178197b66b	AWS-SuseDefaultPatchBaseline	Default Patch Baseline for Suse Provided by AWS.	SUSE	✓ Yes
<input checked="" type="radio"/>	pb-09ca3fb51f0412ec3	AWS-DefaultPatchBaseline	Default Patch Baseline Provided by AWS.	Windows	✗ No
<input type="radio"/>	pb-0c10e657807c7a700	AWS-AmazonLinuxDefaultPatchBaseline	Default Patch Baseline for Amazon Linux Provided by AWS.	Amazon Linux	✓ Yes
<input type="radio"/>	pb-0c10e657807c7a700	AWS-UbuntuDefaultPatchBaseline	Default Patch Baseline for Ubuntu Provided by AWS.	Ubuntu	✓ Yes

Baseline ID: pb-09ca3fb51f0412ec3 [Edit](#) [Delete](#) [Actions](#)

Description	Approval rules	Patch exceptions
Baseline ID arn:aws:ssm:us-east-1:123456789012:patchbaseline/pb-09ca3fb51f0412ec3	Baseline name AWS-DefaultPatchBaseline	
Description Default Patch Baseline Provided by AWS.	Operating system Windows	
Default baseline No	Patch groups -	
Created date (UTC) Tue, 06 Jun 2017 17:07:17 GMT	Modified date (UTC) Tue, 06 Jun 2017 17:07:17 GMT	

Creating a patch baseline

You can create your own custom patch baselines, here you can choose which patches to auto-approve by using the following categories.

- Operating system: Windows, Amazon Linux, Ubuntu Server, etc.
- Product name: For example, RHEL 6.5, Amazon Linux 2014.09, Windows Server 2012, Windows Server 2012 R2, etc.
- Classification: For example, critical updates, security updates, etc.
- Severity: For example, critical, important, etc.

For each auto-approval rule that you create, you can specify an auto-approval delay. This delay is the number of days to wait after the patch was released, before the patch is automatically

approved for patching. For example, if you create a rule using the Critical Updates classification and configure it for five days auto-approval delay, then a new critical patch released on January 1 will automatically be approved on January 6.

Create patch baseline

Provide patch baseline details

Name

You can use letters, numbers, periods, dashes, and underscores in the name.

Description - *optional*

Operating system
Operating system for the Patch Baseline, applied to the Approval Rules and Patch Exceptions.

Approval rules

Product	Classification	Severity	Auto approval delay	Compliance level - <i>optional</i>
<input type="text" value="Select products"/>	<input type="text" value="Select classific..."/>	<input type="text" value="Select severi..."/>	<input type="text" value="5"/> days	<input type="text" value="Unspecified"/>
<input type="button" value="X"/> WindowsServer2016	<input type="button" value="X"/> CriticalUpdates	<input type="button" value="X"/> Critical		<input type="button" value="X"/>
<input type="button" value="Add another rule"/> 9 remaining				

AWS Systems Manager

>

Patch Manager

>

Baseline ID: pb-0d7e05e5964ca987c

Baseline ID: pb-0d7e05e5964ca987c

Edit

Delete

Actions ▾

Description	Approval rules	Patch exceptions

Setting a baseline as a default

By default, the pre-defined patch baseline that ships with Patch Manager is designated as the default patch baseline. However, you can specify your own patch baseline as the default.

AWS Systems Manager > Patch Manager > Baseline ID: pb-09ca3fb51f0412ec3

Baseline ID: pb-09ca3fb51f0412ec3 Edit Delete Actions ▼

- Set default patch baseline
- Modify patch groups

Description	Approval rules	Patch exceptions
Baseline ID arn:aws:ssm:us-east-1:123456789012:patchbaseline/pb-09ca3fb51f0412ec3	Baseline name AWS-DefaultPatchBaseline	
Description Default Patch Baseline Provided by AWS.	Operating system Windows	
Default baseline No	Patch groups -	
Created date (UTC) Tue, 06 Jun 2017 17:07:17 GMT	Modified date (UTC) Tue, 06 Jun 2017 17:07:17 GMT	

Multi-patching approaches

By using multiple patch baselines with different auto-approval delays, you can deploy patches at different rates to different instances. For example, you can create separate patch baselines and auto-approval delays for development and production environments. This enables you to test patches in your development environment before they get deployed in your production

```
aws ssm create-patch-baseline
```

```
--name "Prod-WindowsServer2016-Critical-5Day"
```

```
--approval-rules
```

```
"PatchRules=[{PatchFilterGroup={PatchFilters=[{Key=MSRC_SEVERITY,Values=[Critical]},{Key=CLASSIFICATION,Values=[CriticalUpdates]}}},ApproveAfterDays=5}]"
```

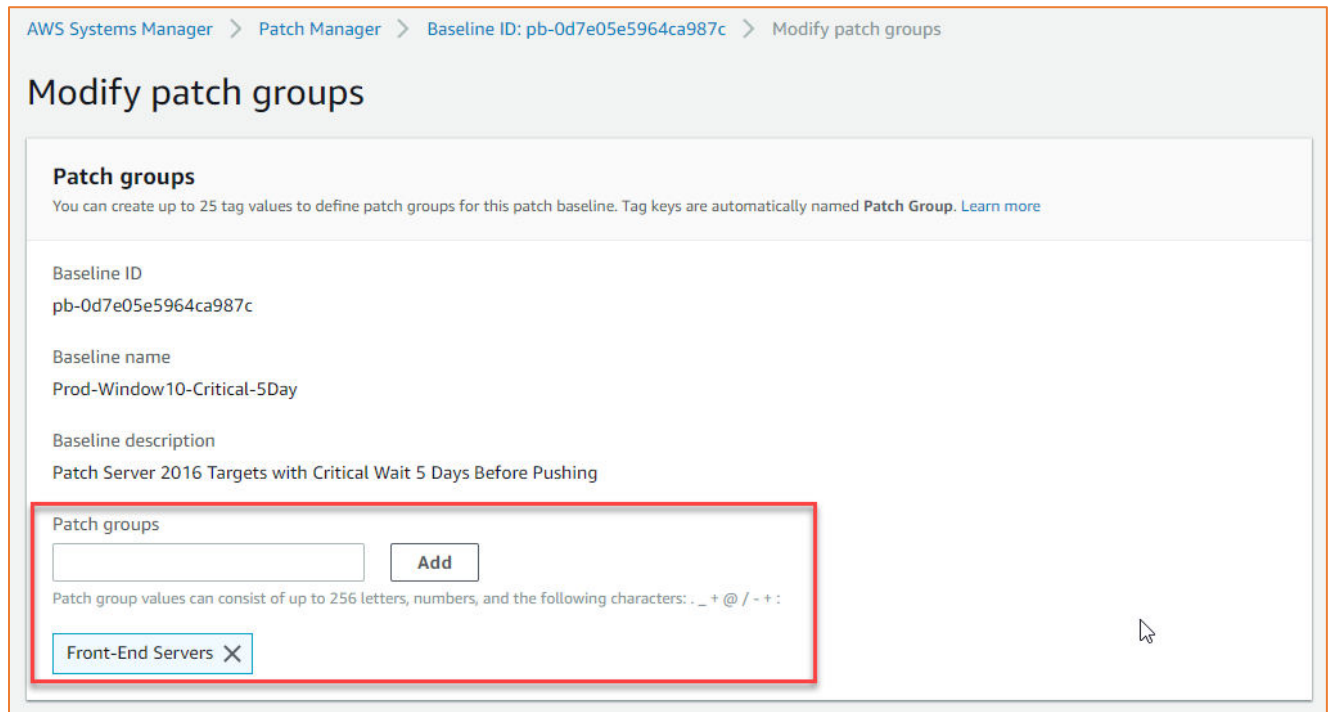
```
--description "Baseline containing all updates approved for production svstems. wait 5
```

environment. Use the following CLI command:

Patch groups

A patch group is an optional means of defining which patch baseline should be used for what instances. For example, you can create patch groups for different environments such as development, test, and production. You can also create primary and secondary failover cluster groupings. Patch groups can be created based on server function, for example, web servers and databases. Patch groups can help you avoid deploying patches to the wrong set of instances.

After you've opened the AWS Systems Manager console select Patch Manager from the left menu. A patch group must be defined with the tag key Patch Group. In the example that follows an instance that we want to patch as a patch group has been tagged with Front-End Servers. A fleet of instances that have these tags can be patched using this approach.



AWS Systems Manager > Patch Manager > Baseline ID: pb-0d7e05e5964ca987c > Modify patch groups

Modify patch groups

Patch groups
You can create up to 25 tag values to define patch groups for this patch baseline. Tag keys are automatically named **Patch Group**. [Learn more](#)

Baseline ID
pb-0d7e05e5964ca987c

Baseline name
Prod-Windows10-Critical-5Day

Baseline description
Patch Server 2016 Targets with Critical Wait 5 Days Before Pushing

Patch groups

Add

Patch group values can consist of up to 256 letters, numbers, and the following characters: . _ + @ / ~ + :

Front-End Servers X

```
aws ssm register-patch-baseline-for-patch-group
```

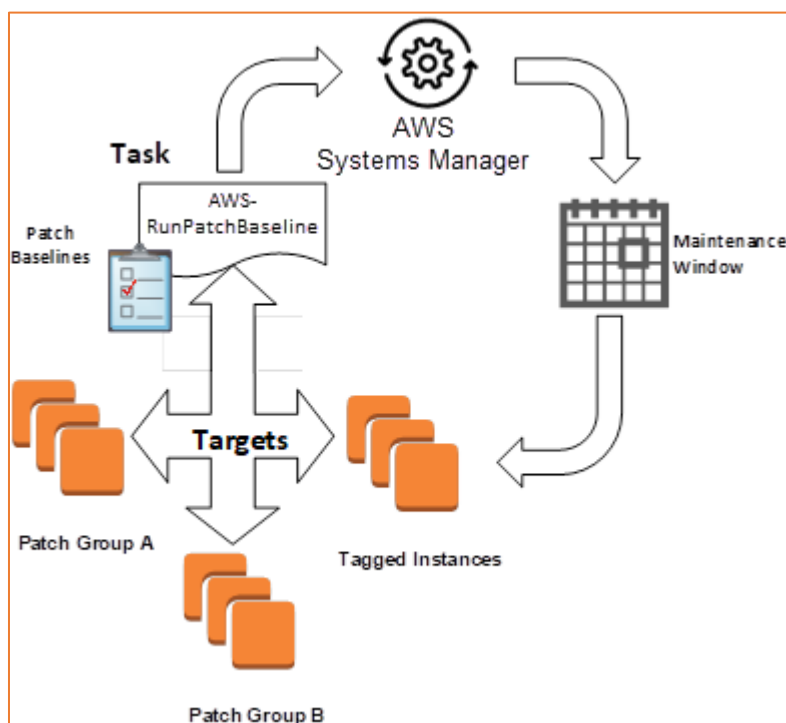
```
--baseline-id pb-0d7e05e5964ca987c
```

```
--patch-group "Front-End Servers"
```

CLI Command:

Maintenance Windows

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 (OS), updating drivers, or installing software. Each Maintenance Window has a schedule, a duration, a set of registered targets, and a set of registered tasks. Typically you want to apply your patches at a time when there is the least impact to your organization.



1. In the AWS Systems Manager console, go to the Create a maintenance window page to create a maintenance window with a schedule for your patching operations.
 - 1) Duration: The duration of the Maintenance Window in hours.
 - 2) Stop initialing tasks (cutoff): The number of hours before the end of the Maintenance Window that Systems Manager stops scheduling new tasks for execution.

Create maintenance window

A maintenance windows lets you specify when a target set of managed instances should install updates or perform maintenance activities. Specify the details below to create a new maintenance window:

Provide maintenance window details

Name

Type a name for this maintenance window.

It has to be between 3 and 128 characters. Valid characters contain the following: a-z, A-Z, 0-9, and . _ -

Description - optional

Type description for this maintenance window.

It has to be between 1 and 128 characters.

Unregistered targets

Allow maintenance tasks scheduled for this maintenance window to run on targets that are not currently registered with this maintenance window.

☒ Allow unregistered targets

Schedule

Specify with

- ☒ Cron schedule builder
☐ Rate schedule builder
☐ CRON/Rate expression

Window starts

- ☐ Every 30 minutes
☐ Every hours
☒ Every at

Duration

Maintenance window duration

 hours

Value from 1 to 24.

Stop initiating tasks

Time to stop starting scheduled task before maintenance window ends

 hour before the window closes

Value from 0 to 23.

Cancel

Create maintenance window

```
aws ssm create-maintenance-window  
  
--name "Patch_Front-End-Servers"  
  
--schedule "cron(0 16 ? * TUE *)"  
  
--duration 4  
  
  
  
--cutoff 1
```

CLI Command:

2. On the Register targets page in the console, create register targets for your Maintenance window by specifying the Patch Group tag for the tag name, and any value for which you have defined EC2 tags, (in our example it's Front-End Servers). You don't have to target by patch group, instead can use any tags defined for your instances, completely independent of the Patch Group tag. A common use case is that you may have a patch baseline for publicly accessible web servers that you use for all internet facing web servers, but you have many different applications so you may want to create a target for the web servers for one of these applications.

Register target

Assign a set of instances to your maintenance window. You can choose to target by a tag group or managed instances.

Maintenance window target details

Maintenance window

mw-08167912695c6687c

Target name - *optional*

Front-End-Servers

It has to be between 3 and 128 characters. Valid characters contain the following: a-z, A-Z, 0-9, and _

Description - *optional*

Target Front-End-Servers

It has to be between 1 to 128 characters.

Owner information - *optional*

It has to be between 1 to 128 characters.

Targets

Targets are the instances you would like to register with maintenance window. You can choose to target by both managed instance and tag.

Select Targets by

☒ Specifying tags

☐ Manually selecting instances

Tags

Patch Group

Front-End-Servers

Remove

Select tag key

Select tag value

Remove

Cancel

Register target

```
aws ssm register-target-with-maintenance-window

--window-id mw-0c66948c711a3b5bd

--targets "Key=tag:Patch Group,Values=Front-End Servers"

--owner-information "Production servers"

--resource-type "INSTANCE"
```

CLI Command:

3. In the Register task pane in the console, do the following tasks.
 - 1) Maintenance Window task details: add, name and description.
 - 2) The Run Command document for this task is AWS-RunPatchBaseline. You can choose to either scan instances or scan and patch instances. If you choose to scan instances, then Patch Manager scans each instance and generates a list of missing patches for you to review. Note that AWS-RunPatchBaseline is cross-platform and works for both Windows and Linux.
 - 3) Fill in the Document Version to use and the Task Priority that determines the order in which tasks are run, when multiple tasks are registered.
 - 4) The Windows Target ID is the patch group defined earlier.
 - 5) For Rate Control:
 - i. Concurrency: The maximum number of targets allowed to run this task in parallel. You can specify a number, such as 10, or a percentage, such as 10%. The default value is 10.
 - ii. Error Threshold: The number of errors that are allowed before the system stops running the automation on additional targets. You can specify either an absolute number of errors, for example 10, or a percentage of the target set, for example 10%.

6) IAM Role that has the AmazonSSMMaintenanceWindowRole policy attached to it, to perform the patching on the registered targets. Please refer to the following [link](#) for guidance on how to configure the role for Maintenance Windows.

7) Input Parameters: These are the parameters that are passed to the chosen Run Command document in step 2.

Register Run command task

Maintenance window tasks define what actions will be executed in the maintenance window. In order to create a task select a document and specify the document parameters for the task.

Maintenance window task details

Maintenance window

mw-032bae396e23c6f92

1

Name - optional

Automation-Task-Patch-Front-End-Servers

It has to be between 3 and 128 characters. Valid characters contain the following: a-z, A-Z, 0-9, and ._-

Description - optional

It has to be between 1 and 128 characters.

<input type="radio"/>	AWS-RunAnsiblePlaybook	Amazon	Linux
<input type="radio"/>	AWS-RunDockerAction	Amazon	Windows,Linux
<input type="radio"/>	AWS-RunDocument	Amazon	Windows,Linux
<input type="radio"/>	AWS-RunInspecChecks	Amazon	Windows,Linux
<input checked="" type="radio"/>	AWS-RunPatchBaseline	Amazon	Windows,Linux
<input type="radio"/>	AWS-RunPowerShellScript	Amazon	Windows,Linux

2

Document description

Scans for or installs patches from a patch baseline to a Linux or Windows operating system.

Task priority

1

Minimum value of 0.

3

Targets

Targets are the instances you would like to associate with this document. You can choose to target by both managed instance and tag.

Strict Targets

06ca7a94-d452-4a3f-8ea8-816a2821ca7f X

< 1 >



Window target ID

4

Name

Owner information



06ca7a94-d452-4a3f-8ea8-816a2821ca7f

Patch-Front-End-Servers

-

Rate control

5

Concurrency

Specify the number or percentage of targets on which to execute the task at the same time



targets



25

percentage

Error threshold

Stop the task after the task fails on the specified number or percentage of targets



1

errors



percentage

Role

IAM Role

Maintenance Window service role `arn`, will be used by Maintenance Window to execute the task. [Add new custom role](#)

`arn:aws:iam::[redacted]:role/SSMManagedInstanceProfileRole` ▼

6

Output options

Write to S3

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

☐ Enable writing to S3

Parameters

7

Operation

(Required) The update or configuration to perform on the instance. The system checks if patches specified in the patch baseline are installed on the instance. The install operation installs patches missing from the baseline.

Scan ▼

Snapshot Id

(Optional) The snapshot ID to use to retrieve a patch baseline snapshot.

Comment - optional

Type a note about the command.

Timeout (seconds)

Specify a timeout for the command in seconds.

600

CLI Command:

```
aws ssm register-task-with-maintenance-window

--window-id mw-0c66948c711a3b5bd

--targets "Key=tag:Patch Group,Values=Front-End Servers"

--task-arn "AWS-RunPatchBaseline"

--service-role-arn "arn:aws:iam::12345678:role/MW-Role"

--task-type "RUN_COMMAND"

--max-concurrency 2

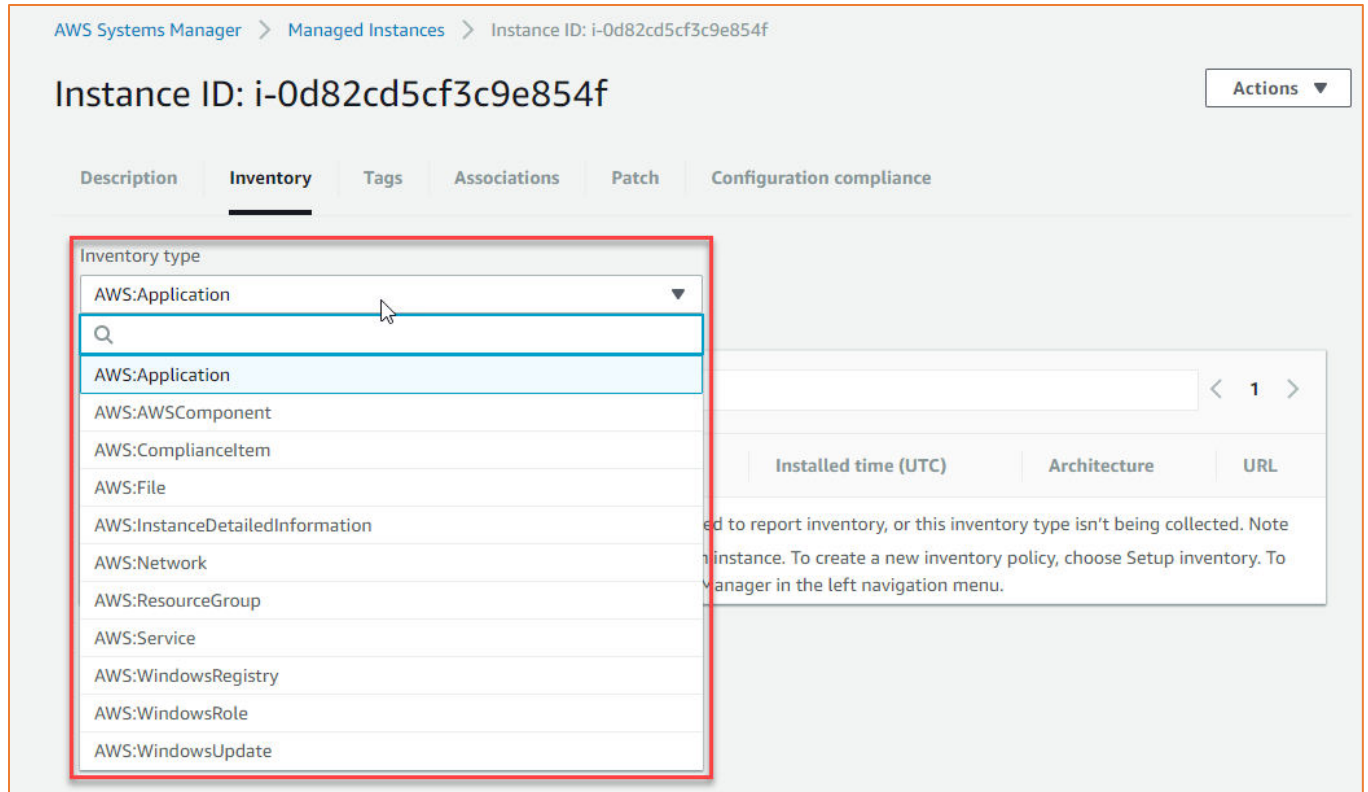
--max-errors 1

--priority 1

--task-parameters '{"Operation\":"Values\":[\"Scan\"]}'
```

Monitoring patch compliance

You can view results and patch compliance details on the Managed Instances page by selecting the Inventory tab and filtering by the AWS: PatchSummary and AWS: PatchCompliance. Note that the Configuration compliance section gives an aggregated high-level view of patch compliance from where you can dive deeper into details of compliance. You can also review a specific instance by choosing the instance and then choosing the Patch tab.



AWS Systems Manager > Managed Instances > Instance ID: i-0d82cd5cf3c9e854f

Instance ID: i-0d82cd5cf3c9e854f Actions ▼

Description **Inventory** Tags Associations Patch Configuration compliance

Inventory type

- AWS:Application
- AWS:AWSComponent
- AWS:ComplianceItem
- AWS:File
- AWS:InstanceDetailedInformation
- AWS:Network
- AWS:ResourceGroup
- AWS:Service
- AWS:WindowsRegistry
- AWS:WindowsRole
- AWS:WindowsUpdate

Installed time (UTC) Architecture URL

ed to report inventory, or this inventory type isn't being collected. Note
n instance. To create a new inventory policy, choose Setup inventory. To
v manager in the left navigation menu.

Conclusion

In this blog post I covered the key aspects of using AWS Systems Manager Patch Manager. I started by showing you how to create a patch baseline that defines which patches should and shouldn't be installed on your instances. After you define your baselines, I showed you how to control the deployment of these baselines to patch groups for different environments such as development, test, and production. Finally, I showed you how to minimize the impact to your organization by using the Maintenance Window and scheduling the rollout of the patches.

I encourage you to use what you have learned here and consider leveraging AWS Systems Manager resources in your own organization. You might begin by using Patch Manager in a small part of the organization to demonstrate a proof of concept to your peers. I welcome your comments or questions.