

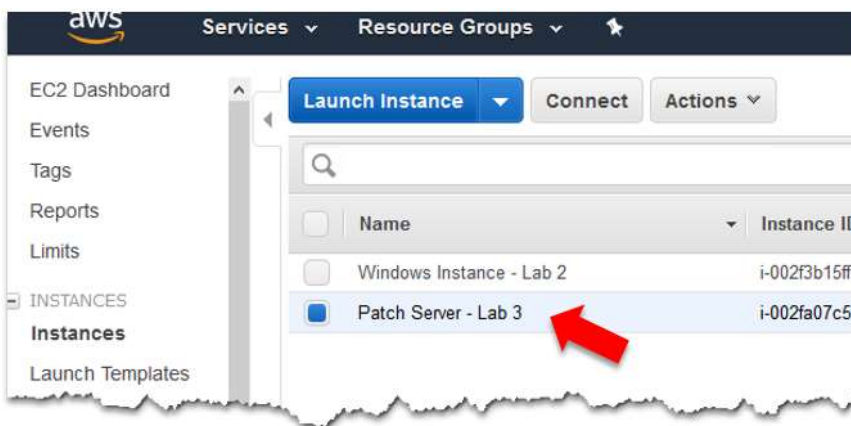
## Lab 3 – Image Operations

**AWS Systems Manager** – After resources are provisioned, keeping them secure and compliant can be a seemingly large task for administrators and often takes many hours to scan, stage, and remediate patches for persistent instances in an environment. In addition, third-party tools are often expensive and complicated to manage to help ensure workloads are being patched as required by an established security policy. As a free tool provided by AWS, Systems Manager provides a centralized interface to accomplish a variety of tasks, including automation of tasks, viewing infrastructure performance and configuration and even application management. AWS Systems Manager can also be used to patch instances throughout your AWS infrastructure with a single interface.

**Goal** – Using an EC2 instance deployed in the initial lab, use AWS Systems Manager to create a maintenance window, assign the instance a baseline for patches, and scan the instance for compliance against the patch baseline.

**Prerequisites** - Access to the AWS VPC console with an IAM user with appropriate permissions for EC2 and AWS Systems Manager.

1. Navigate the **EC2** console and select Instances from the left menu pane. Select the **Patch Server – Lab 3** instance:



2. From the **Actions** menu, choose **Instance Settings** → **Add/Edit Tags**.

**Note:** You will see several tags attached to the instance already, some that were created in the CloudFormation template and some that CloudFormation will automatically append to any resource it creates.

3. Click the **Create Tag** button and add the following tag:
  - **Key:** Patch Group (must include a space)
  - **Value:** Group1

**Add/Edit Tags** [X]

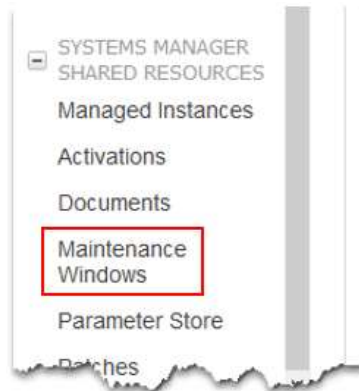
Apply tags to your resources to help organize and identify them.

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver. [Learn more](#) about tagging your Amazon EC2 resources.

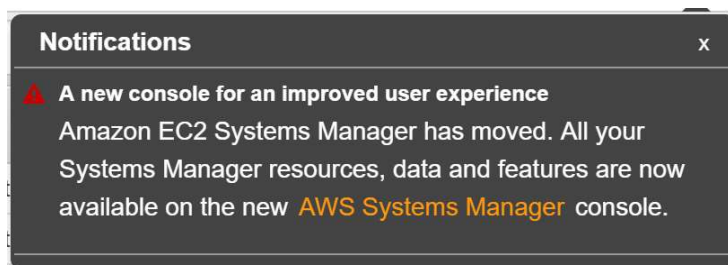
Key	Value	
Application	Windows Server	[X] Show Column
Environment	AHEAD Innovation Day	[X] Show Column
Name	Patch Server - Lab 3	[X] Hide Column
aws:cloudformation:log	PatchServer	[X] Show Column
aws:cloudformation:sta	arn:aws:cloudformation	[X] Show Column
aws:cloudformation:sta	ces-1QQV1SXE12HQ8	[X] Show Column
Patch Group	Group1	[X]

[Create Tag] [Cancel] [Save]

4. Click the **Save** button to apply the new tag.
5. Navigate to the EC2 console and select **Maintenance Windows** under the Systems Manager Shared Resources section (located at the bottom left of the navigation pane).



**Note:** If you receive a message regarding the new console for AWS Systems Manager, you can ignore it for now. Just click the X in the top right corner of the message.



6. Choose **Create a Maintenance Window**.
7. In the Name field, type **Window1**.
8. In the schedule section, make sure that the Window starts **every 30 minutes**. Enter **1** in the **Duration** field and **0** into the **Stop initiating tasks** field.

The screenshot shows the 'Create maintenance window' form. It has two main sections: 'Provide maintenance window details' and 'Specify schedule'. In the first section, the 'Name' field contains 'Window1'. In the second section, 'Specify with' has 'Cron schedule builder' selected. Under 'Window starts', 'Every 30 Minutes' is selected. The 'Duration' field is set to '1' hours. The 'Stop initiating tasks' field is set to '0' hour before the window closes. Red arrows point to these four fields.

9. Click the **Create Maintenance Window** button to finish.
10. Select the Maintenance Window that was just created (**Window1**). From the Actions menu, select **Register Targets**.
11. Under the Targets section, use the drop-down and select the tag **Patch Group** that was created earlier. For the Tag value, select **Group1**.

The screenshot shows the 'Targets' section. It has a 'Select Targets by' section with 'Specifying Tags' selected. Below it, there's a table with 'Tag Name' and 'Tag Value' columns. The 'Tag Name' column has 'PatchGroup' selected, and the 'Tag Value' column has 'Group1' selected. Red arrows point to these two selections. There is also an 'Add Tag' button and a note '4 remaining'.

12. Click **Register Targets**. Click **Close** on the acknowledgment.
13. Select the Maintenance Window and choose **Register run command task** from the Actions menu.

14. Type **RunPatchBaseline** under the Name field. Select **AWS-RunPatchBaseline** from the list of command document section.

Maintenance windows > Register run command task

### Register run command task

A maintenance window tasks define what actions will be executed in the maintenance window. In order to create a task select a document

Name: RunPatchBaseline

Description:

Command Document

Document\*

Owned by Me or Amazon	
Name	Owner
<input type="radio"/> AWS-InstallPowerShellModule	Amazon
<input type="radio"/> AWS-InstallApplication	Amazon
<input type="radio"/> AWS-JoinDirectoryServiceDomain	Amazon
<input checked="" type="radio"/> AWS-RunPatchBaseline	Amazon
<input type="radio"/> AWS-InstallSpecificWindowsUpdates	Amazon
<input type="radio"/> AWS-RunShellScript	Amazon

15. Under **Targets**, ensure that a target is selected.

**Note:** If you have multiple Maintenance Windows, you might need to identify the correct one by navigating back to the Maintenance Window console and making note of the Window Target ID.

16. Under **Role\***, select the **MaintenanceWindowRole** that was created as part of the initial CloudFormation template.
17. In the **Execute on\*** field, enter **100** and ensure **Percent** is selected from the drop-down.
18. In the **Stop after\*** field, type **0**.
19. Modify the Operation field and change to **Install**.

Your selections should like the screenshot below:

**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

**Parameters**

Role\*

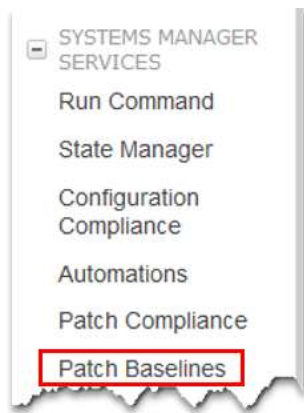
Execute on\*

Stop after\*  errors

Operation\*

Snapshot Id

20. Click **Register Task**. Click **Close** on the acknowledgment.
21. In the EC2 console, select **Patch Baselines** under the Systems Manager Services group:



22. Select the **AWS-DefaultPatchBaseline** from the list of default baselines. It should have **Windows** listed as the Operating System.
23. From the Actions menu, select **Modify Patch Groups**.
24. In the Patch Groups field, type **Group1**, be sure to **click the check mark** to the right of the text box.
25. Click **Close**.
26. In the EC2 console, click on **Run Command** on the navigation pane under the System Manager Services section.

**Note:** For lab purposes, you will be manually executing a Scan operation. In a production environment, this would be configured to run on a recurring schedule.

27. Select the **Run a command** button.

28. In the command document section, select the radio button next to **AWS-RunPatchBaseline**.
29. Under **Select Targets by\***, click the radio button next to **Specifying a Tag**.
30. Under Tag Name, select **Patch Group**. Select the value of **Group1** under the Tag Value.
31. In the Execute On text box, enter **100** and modify the concurrency box to **Percent**.
32. Enter **0** in the Stop after text box.
33. In the Operation\* box, select "Scan."

The screenshot shows the configuration for the **AWS-RunPatchBaseline** command. At the top, a list of commands is shown with radio buttons; **AWS-RunPatchBaseline** is selected. Below this, the **Description** states: "Scans for or installs patches from a patch baseline to a Linux or Windows operating system." Under **Select Targets by\***, the **Specifying a Tag** radio button is selected. The **Tag Name** dropdown is set to **Patch Group** and the **Tag Value** dropdown is set to **Group1**. The **Execute on** field is set to **100** with a **Percent** concurrency dropdown. The **Stop after** field is set to **0** errors. The **Operation\*** dropdown is set to **Scan**.

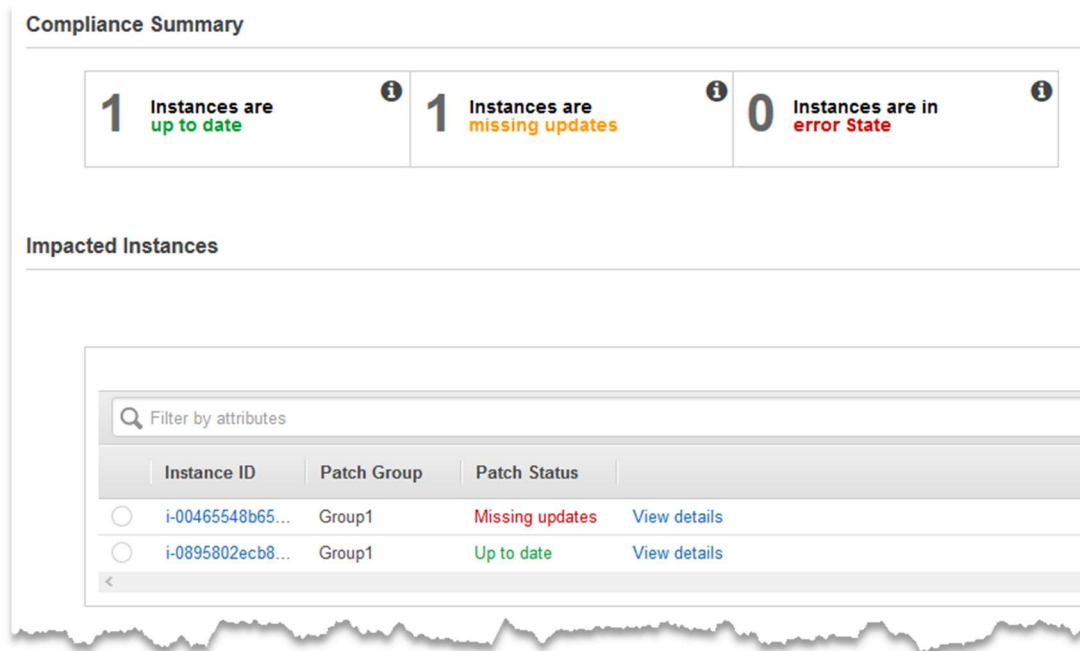
34. Click **Run**.
35. You should receive a success message. Click **View Result**.
36. Wait until the Run command status changes to **Success** (this takes about 10 minutes on average).

The screenshot shows the AWS Run Command results table. The table has columns: **Command ID**, **Instance ID**, **Document name**, **Status**, and **Requested date**. A single row is visible with the following data:

Command ID	Instance ID	Document name	Status	Requested date
4f5b3563-7862-4f97-...	i-00465548b658ec234	AWS-RunPatchBaseline	Success	March 12, 2018 at 1...

37. Navigate to the **Patch Compliance** console found in the navigation pane in the EC2 console under the Systems Manager Services section.

38. Is it reporting that your instances up to date?



**Note:** The screenshot above is an example. It's very likely that your instances are indeed up to date, as Amazon keeps its AMIs up to date with the latest Microsoft patches. However, workloads that have persisted longer than a month or so would likely require security patches.

39. Select the instance and click **View Details**.

40. Are there patches listed for the instance?

**That's it. You've completed this lab.**

**Conclusion:** In this lab, participants used AWS Systems Manager to enable patching compliance for instances running in EC2. You created a Maintenance Window to install patches and scanned the EC2 instances to ensure compliance against a patch baseline.