**SSM Manager – Lab Instructions**

1. **Setup** Goal: set up your lab environment

Find the sample code repo in Github ([here](https://github.com/rjgleave/aws-ssm-immersion-day)) which contains the code needed for this lab. It may be convenient to download the repo to your local machine. It has a folder called *setup-lab* with all the Cloudformation templates needed to build your environment.

* 1. The table below shows the IAM resources you will need for this lab. These roles and profiles will be used by all lab participants. Use the IAM console to verify that they exist.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Role or Profile** | **ARN** | **Use this for…** |
| 1 | ManagedInstanceProfile | arn:aws:iam::<your-account>:instance-profile/ManagedInstanceProfile | Launching instances. Attach to each instance. |
| 2 | AutomationServiceRole | arn:aws:iam:: <your-account>::role/AutomationServiceRole | Running automations |

If they do not yet exist, follow the process below to create them.

* + 1. Find the ***create-ssm-roles.yaml*** template in the repo ([here](https://github.com/rjgleave/aws-ssm-immersion-day/blob/master/setup-lab/create-server-fleet.yaml)) and build the Cloudformation stack.
    2. When the process is complete, you will find the role names under the *Physical ID* column of the *Resources* drop-down section of the Cloudformation page.
    3. Take a few minutes to review the policies attached to the roles in the IAM console and read what they are used for below.
    4. Copy the ARNs of the resources above for the exercises below.
  1. Launch your sample fleet. It will create a number of foundational components, including VPC, subnet, NAT gateway, security group, etc. It will also launch a small fleet of 4 instances (Amazon Linux, Ubuntu, RHEL, Windows).
     1. Find the ***create-server-fleet.yaml*** cloudformation template in the immersion day repo ([here](https://github.com/rjgleave/aws-ssm-immersion-day/blob/master/setup-lab/create-server-fleet.yaml)) and launch it to create the stack.

|  |
| --- |
| Launch your sample fleet |
| 1. Open the AWS Cloudformation console ([here](https://github.com/rjgleave/aws-ssm-immersion-day/blob/master/setup-lab/create-server-fleet.yaml)). 2. Press the *Create Stack* button 3. Select *upload template to Amazon S3 button* and choose the *create-server-fleet.yaml* template you downloaded in step 1.3.1 above. 4. Press the *Next* button. 5. Enter a name for your stack. It needs to be unique from other lab participants, so one approach is to append your initials like the example below.      1. Enter a name for your fleet, such as: unicorns, tigers, penguines, etc. IT MUST BE LOWER CASE, SINGLE WORD, NO SPECIAL CHARACTERS. Don’t just accept the default name, because it will try to create an S3 bucket using this name, which must be unique. 2. At the *Create Sample Fleet*? prompt, answer *yes* (see above) 3. Press the *Next* button. 4. On the next two screens, you can leave the default values in place and complete the process to launch your stack. It should take about 5 minutes to build your environment. 5. Once the stack is built, validate that you have 4 running EC2 instances, as well as an S3 bucket named *<your fleet>-ssm-lab*. Notice that each of your instances has a *Fleet* tag with the value you entered. |

* 1. Set up your S3 bucket to hold inventory data and logs
     1. Go to the S3 console and find your bucket. It should be named: *<your fleet name>-ssm-lab*.
     2. Locate and review the bucket policy document, which was created by Cloudformation. You will find it under the *Permissions* tab. This policy is critical for allowing SSM inventory to sync to S3.

1. **SSM Inventory** Goal: set up your inventory system
   1. Select *Systems Manager* from the list of *Services* on the console.
   2. Select ***Inventory*** from the left navigation bar.
   3. Roll down to the bottom of the inventory page. You should see a list showing your 4 instances. Select one of your instances, then click the *Inventory* tab, and notice that there is no inventory displayed for that instance yet.
   4. Follow the steps below to create an inventory association.

|  |
| --- |
| Set up SSM Inventory |
| To configure inventory collection   1. In the left navigation bar, choose *Inventory*. 2. Choose *Setup Inventory*. 3. Enter a unique name for your inventory, such as: *Inventory-Association-<your fleet name>* 4. In the Targets section, identify the instances where you want to collect inventory. For this exercise we will choose the tag option.    * Click on the *Specifying a tag* option. This option lets you specify a single tag to identify instances in your account from which you want to collect inventory. If you use a tag, any instances created in the future with that same tag will be added automatically to your inventory.    * Go to the EC2 Instances tab and look at the tags on one of your running instances. You should see a tag *Key* called f*leet*, with the name of your fleet as the *Value*.    * Enter the tag key/value in the *Targets* section. 5. In the Schedule section, choose how often you want the system to collect inventory metadata from your instances (regardless of what you select, it will collect immediately the first time). 6. In the Parameters section, select the buttons to enable or disable different types of inventory collection. 7. Press the *Setup Inventory* button. Systems Manager creates a State Manager association and immediately collects Inventory from the instances. It should run quickly. You will see a green ‘Success’ message at the top of the panel when it is done. You can click on the *View Details* button to see information about the process. 8. If you want to view inventory data for a specific instance:    * Select *Managed Instances* in the navigation pane.    * Choose an instance (make sure it is from YOUR fleet), and then choose *View details*.    * On the instance details page, select the *Inventory* tab*.* Use the *Inventory type* dropdown list to practice filtering your inventory. |

* 1. Now archive your inventory history, by following the process below. It will create a resource sync to an S3 folder.

|  |
| --- |
| Set up Inventory Resource Sync |
| 1. In the left navigation bar, choose *Inventory*. 2. Press the *Resource Data Syncs* button (top right of page) 3. Press the *Create resource data* sync button. 4. Enter a name for your sync, such as: *<your fleet name> - resource-sync* 5. Enter the name of the S3 bucket created for your fleet in step 1. 6. Select the region where your bucket is located. 7. Refresh the screen. Once the sync is complete, you should see inventory information in your S3 bucket. |

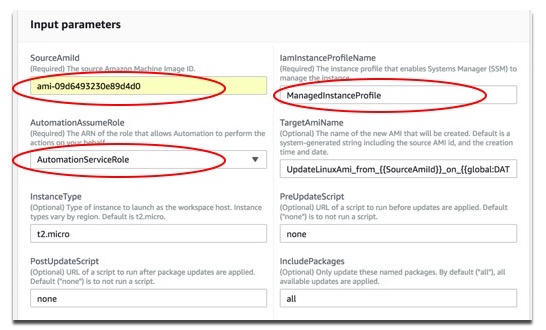
1. **SSM Automation** Goal: use Automation to create an AMI, patch it, and launch instances from it
   1. Select *Automation* from the left navigation bar.
   2. Launch an automation to stop your Red Hat instance (below).

|  |
| --- |
| Using automation to stop an instance |
| 1. In the navigation pane, choose *Automation*. 2. Choose *Execute automation*. 3. In the Automation document list, find and select *AWS-StopEC2Instance.* 4. In the Document details section, set the Document version to *Default version at runtime*. 5. In the Execution mode section, choose *Execute the entire automation at once*. 6. Leave the Targets and Rate Control option disabled. 7. In the Input parameters section, click the button to *Show interactive instance picker*. Select your RedHat instance (make sure it is from your fleet!) 8. Choose Execute automation. The console displays the status of the Automation execution. |

* 1. Use the ***AWS-StartEC2Instance*** automation document to restart the instance you stopped above. However, this time try to execute it one step at a time.
     1. Tip #1: you must change the dropdown from *Show Managed Instances* to *Show ALL Instances* to see the stopped instance in your list of targets.
     2. Tip #2: many automations require elevated privileges. Use the automation service role created in 1.2.4 as needed.
  2. Create an AMI from one of the instances in your fleet.
     1. Use the ***AWS-CreateImage*** automation to do this.
     2. Notice the parameters for the document. Some are required and others are not.
     3. Go to the AMI page of the EC2 console ([here](https://us-west-1.console.aws.amazon.com/ec2/v2/home?region=us-west-1#Images:sort=name)) to find your new image. Copy the image name to help you complete the steps below.
  3. Now create a process to automatically patch the AMI you created in step 3.4 above. For Linux images, follow the steps defined below. If you created a Windows AMI, follow the patching steps indicated [here](https://docs.aws.amazon.com/systems-manager/latest/userguide/automation-createdoc.html).

|  |
| --- |
| About Patching AMIs |
| This process will walk you through how to use the AWS-UpdateLinuxAmi document to automatically patch a Linux AMI with the latest versions of packages that you specify.   |  | | --- | | Note: when you run the *AWS-UpdateLinuxAmi* document, SSM Automation performs the following tasks**.**   * Launches a temporary Amazon EC2 instance from a Linux AMI. The instance is configured with a User Data script that installs SSM Agent. SSM Agent runs scripts sent remotely from Systems Manager Run Command. * Updates the Instance by performing the following actions:   1. Invokes a user-provided pre-update script on the instance (optional).   2. Updates AWS tools on the instance, if any tools are present.   3. Updates distribution packages on the instance by using the native package manager.   4. Invokes a user-provided post-update script on the instance (optional). * Stops the temporary instance. * Creates a new AMI from the stopped instance. * Terminates the instance. |   After Automation successfully completes this workflow, the new AMI is available on your AMIs page. |

* + 1. Open the AWS Systems Manager
    2. In the navigation pane, choose Automation.
    3. Choose Execute automation.
    4. In the Automation document list, find and choose AWS-UpdateLinuxAmi.
    5. In the Document details section, select Default version at runtime.
    6. In the Execution mode section, choose Execute the entire automation at once.
    7. Leave the Targets and Rate Control option disabled.
    8. In the Input parameters section please note the required parameters that you MUST enter. Also, pay special attention to the roles indicated below, which must have the proper rights, as highlighted below (more info [here](https://docs.aws.amazon.com/systems-manager/latest/userguide/automation-setup-user.html)).



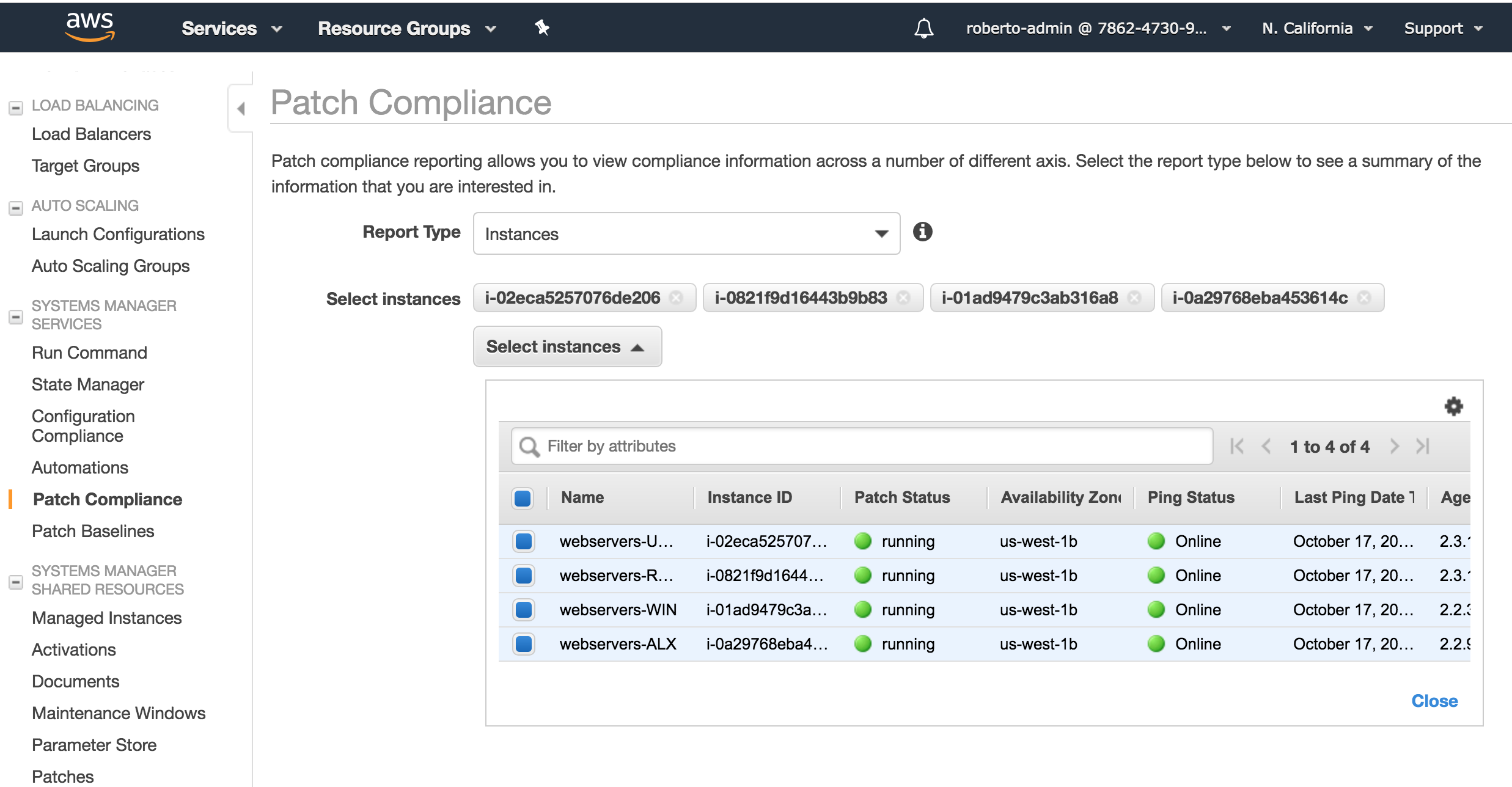
* + 1. Choose Execute automation. The console displays the status of the Automation execution. This may take a long time to complete. Move on to the next task. When the automation finishes, you may select AMIs from the EC2 menu to see your updated AMI.

1. **SSM Run Command** Goal: practice using the *RUN* command
   1. Execute a simple remote command
      1. Select ***Run Command*** from the left navigation bar.
      2. Press the *Run Command* button.
      3. Use the search functionality to find the *AWS-RunShellScript* command and select it.
      4. Select the target instances to run the command (for example, select all). (Trivia Question: why do only 3 instances show up on the target list?)
      5. In the ‘command’ section, enter the linux command ‘ls’, to list files on the instance.
      6. Refresh the screen until you see ‘success’.
      7. See the results of your run command:
         1. On the *Run Command* menu, select *Command History*.
         2. Click on the Command ID to see the results of the command
         3. Click on one of the instance IDs to see results.
         4. View the Output of the command. You should see a list of files from each instance.
   2. Install the Clamav antivirus program on your Amazon Linux instance. This will be used in a later exercise.
      1. Select ***Run Command*** from the left navigation bar again.
      2. Press the *Run Command* button.
      3. Use the search functionality to find the *AWS-RunShellScript* command and select it.
      4. Select you Amazon Linux instance as the target.
      5. In the ‘command’ section, enter the following:

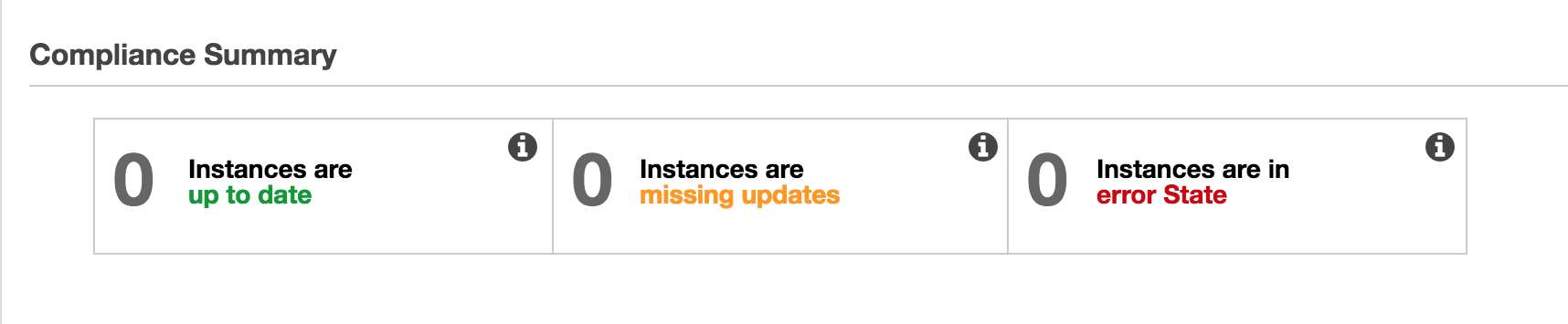
|  |
| --- |
| yes | sudo yum install clamav |

* + 1. On the output section, select to have the output routed to your S3 bucket.
    2. Refresh the screen until you see ‘success’.
    3. Open your S3 bucket and find the output of your run command. You should see that the anti-virus package was installed correctly.

1. **SSM Session Manager** Goal: learn how to log into an instance without using a bastion host
   1. Select ***Session Manager*** from the left navigation bar.
   2. Click the *Configure Preferences* button to setup your session
      1. Configure logging of session activity to the S3 folder you created in 1.4.2.3 above.
      2. Enter a log group name to configure Cloudwatch logging
   3. Click the *Start Session* button to start a session
   4. Select one of the Linux instances from the target list.
   5. Enter some sample commands on the remote instance command line
   6. End the session. There are two ways to do this (try both):
      1. Type ‘exit’ on the command line of the instance
      2. Click the *Terminate* button)
   7. History – review your activity on the *Session History* tab.
   8. Now select the Windows instance.
   9. Try to start a session on the instance. (NOTE: You may get an error saying that Systems Manager is not supported by the version of the agent on that instance. If so, press the button to update the version of the agent and try again once the update completes)
2. **SSM Patch Manager** Goal: learn how to configure patching in multiple ways and view compliance.
   1. View patch compliance using Patch Compliance Dashboard.
      1. Go to the EC2 Menu and find Patch Compliance on the left navigation bar.
      2. Select *Instances* from the dropdown and check mark all the instances



* + 1. You should see a compliance dashboard that looks like this:

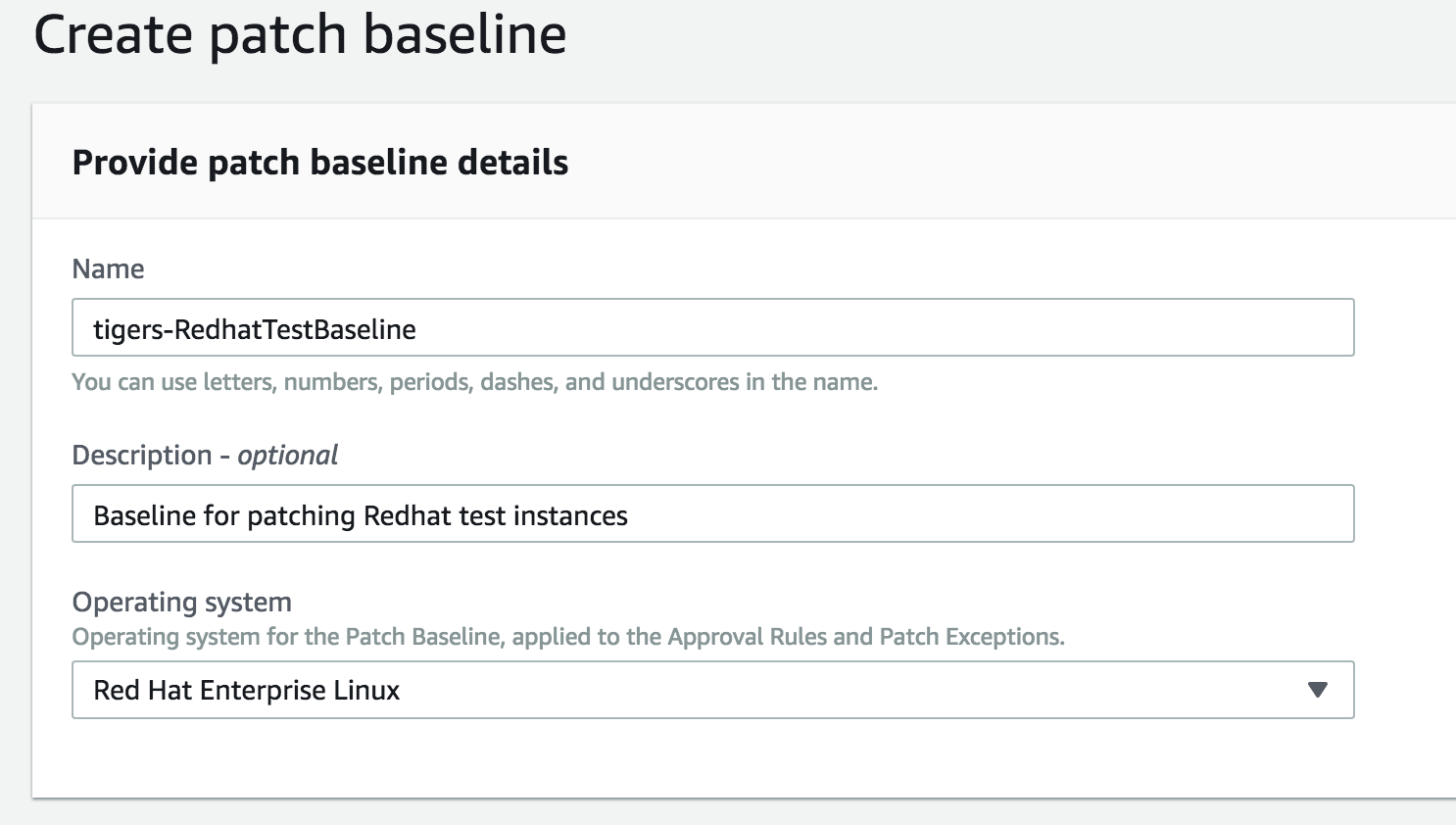


|  |
| --- |
| **Tip: Nothing registers on this dashboard until you associate instances with baselines (below)** |

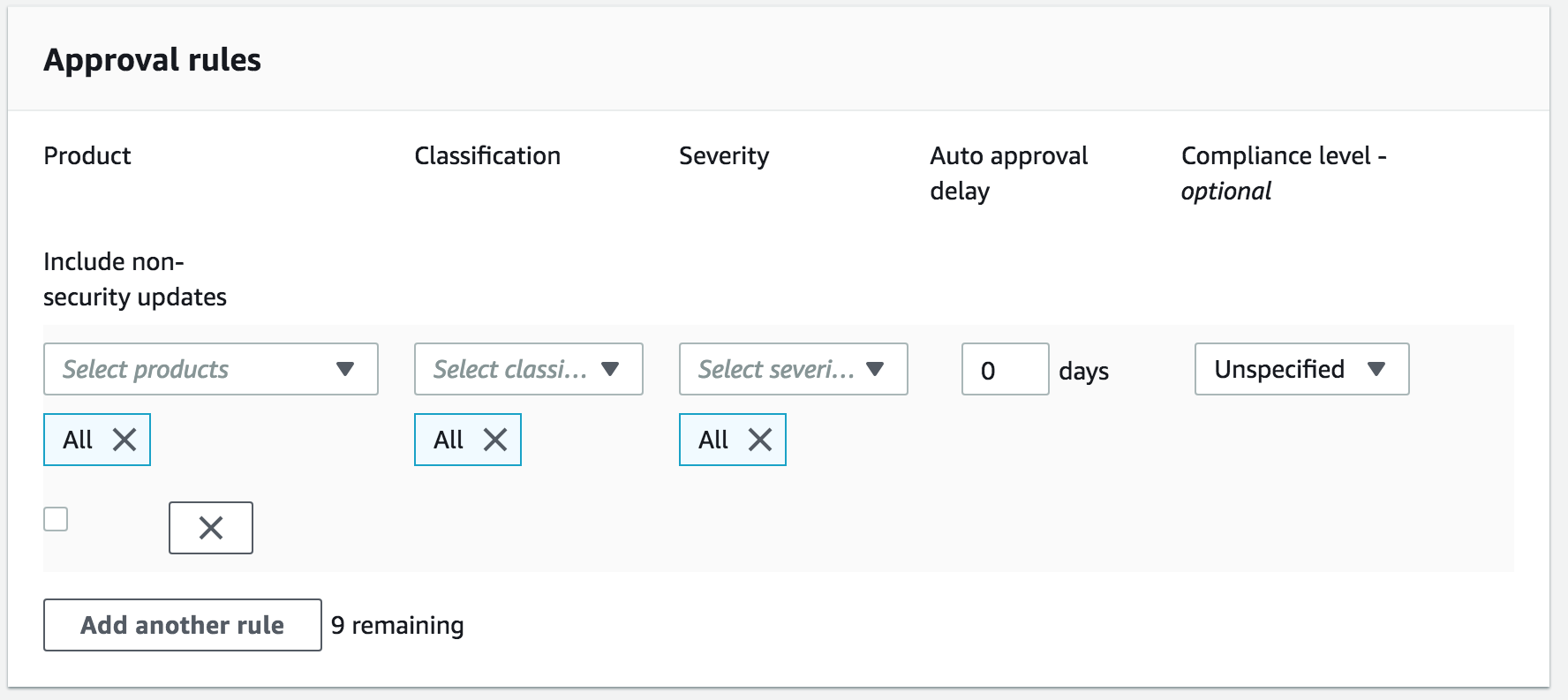
* 1. Work with your patch baselines. Select ***Patch Manager*** from the left navigation bar.
     1. Look for the *default patch baselines* link on the Patch Manager Screen as shown below.



* + 1. Select one of the baselines and review each of the tabs - *Description*, *Approval Rules* and *Patch Exceptions* - to familiarize yourself with what baselines contain.
    2. Create a custom baseline
       1. On the *Patch Baselines* screen, press the *Create patch baseline* button
       2. Enter *<your fleet>-RedhatTestingBaseline* for the name, then enter a description and choose the operating system, as shown below.



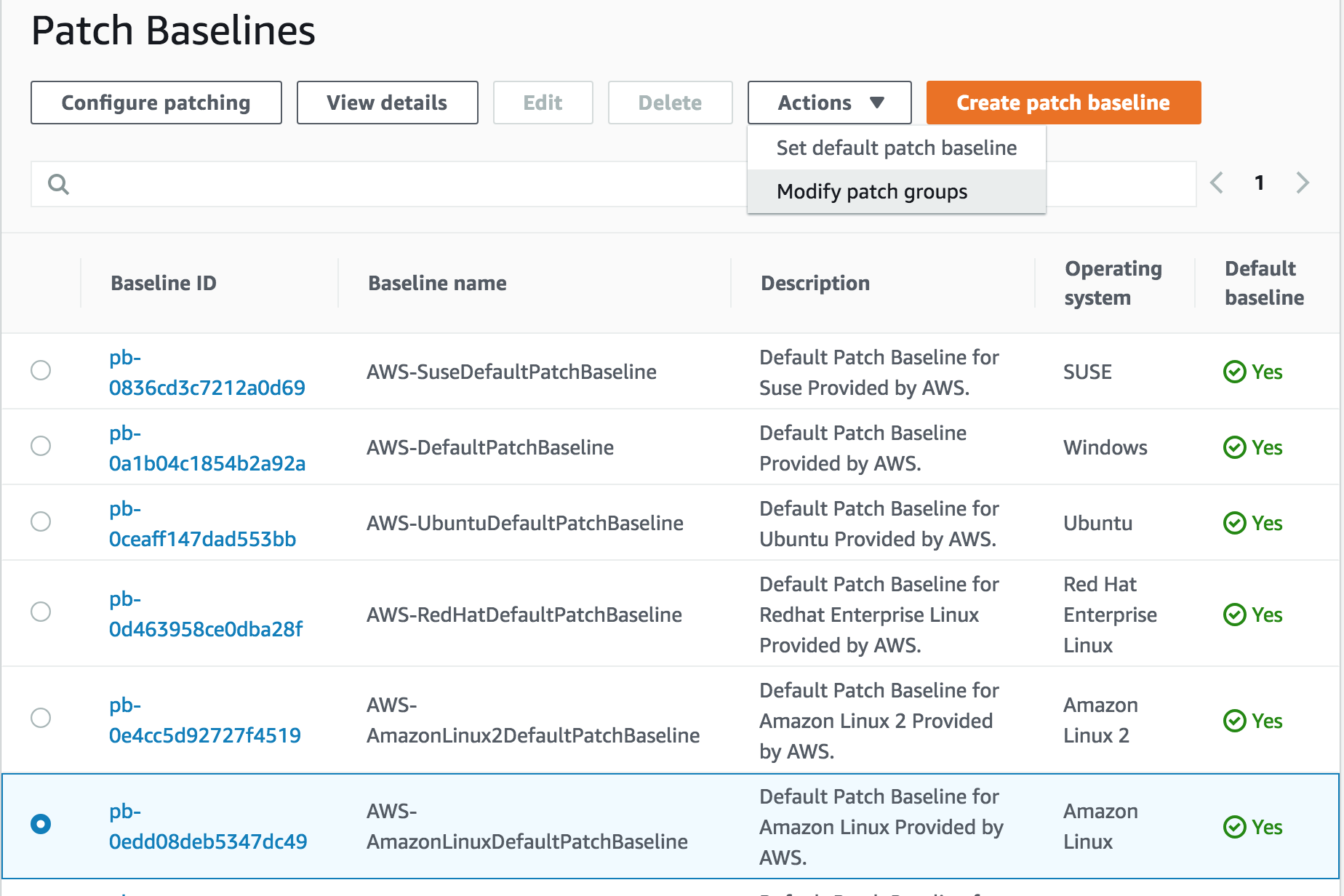
* + - 1. Enter rules for your custom baseline. Leave them all blank, with zero days delay so that all patches apply immediately to the dev and test instances.



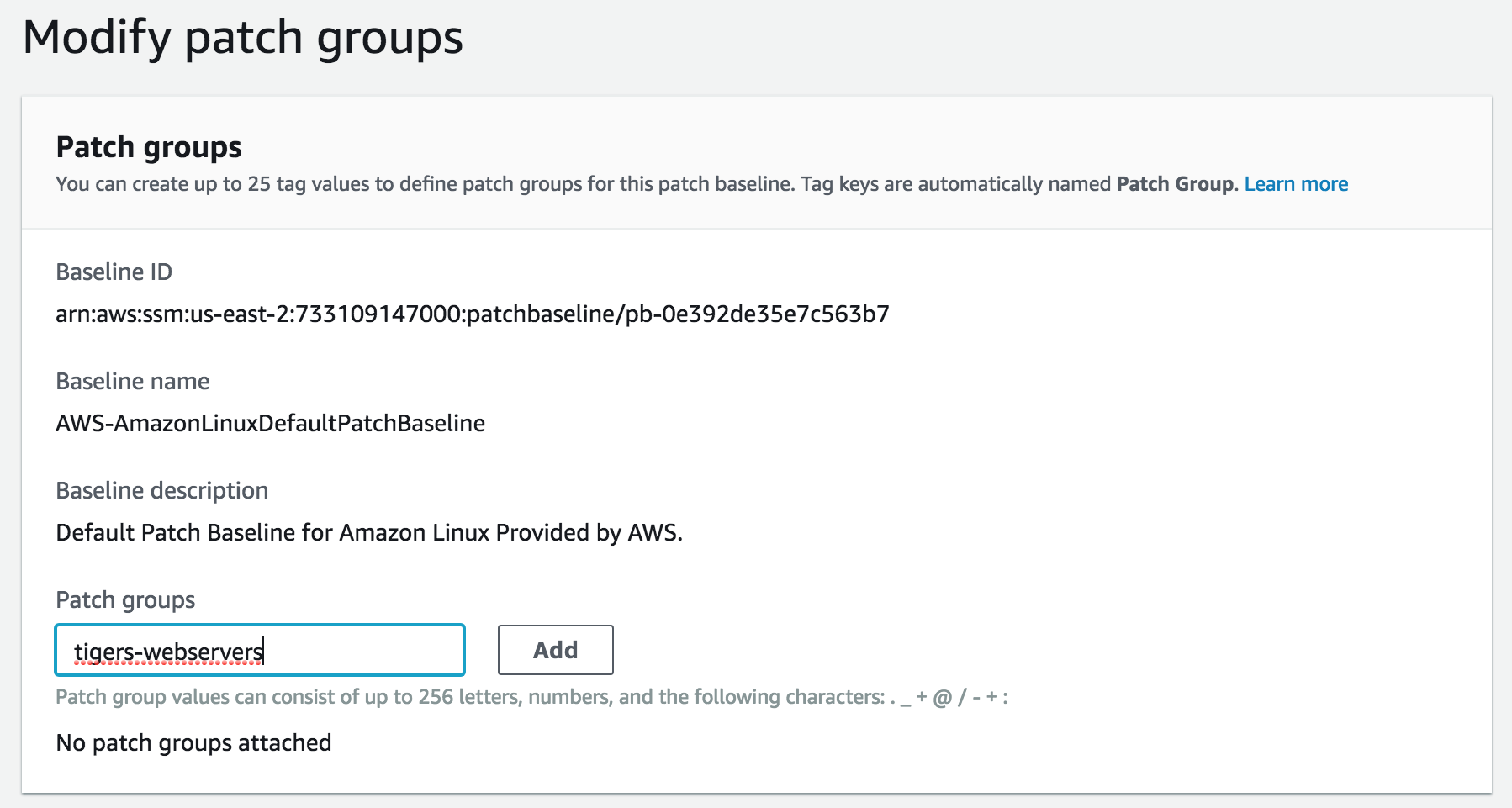
* + - 1. Press the *Create patch baseline* button
      2. You now have created a custom baseline!

|  |
| --- |
| What is a Patch Group? |
| A Patch Group is an optional means of organizing instances for patching. For example, you can create patch groups for different operating systems (Linux or Windows), different environments (Development, Test, and Production), or different server functions (web servers, file servers, databases).  Patch Groups can help you avoid deploying patches to the wrong set of instances. They can also help you avoid deploying patches before they have been adequately tested.  IMPORTANT: Patch Groups are very useful because they are the ONLY way to associate non-default baselines to instances.   |  | | --- | | The mechanism for creating patch groups is to tag instances with special EC2 tags. Patch groups require use of the special tag key name ***Patch Group***. You can specify any tag value you prefer, but the tag key must be **Patch Group**. Note that the key is case sensitive. |   When Patch Manager runs the AWS-RunPatchBaseline task on that instance, the service checks to see which patch baseline is registered with a given patch group. If a patch baseline is found, the system uses that baseline. If no patch baseline is registered, the system uses the default patch baseline.  For more information about patch groups, see [Organize Instances into Patch Groups](https://docs.aws.amazon.com/systems-manager/latest/userguide/sysman-patch-patchgroups.html). |

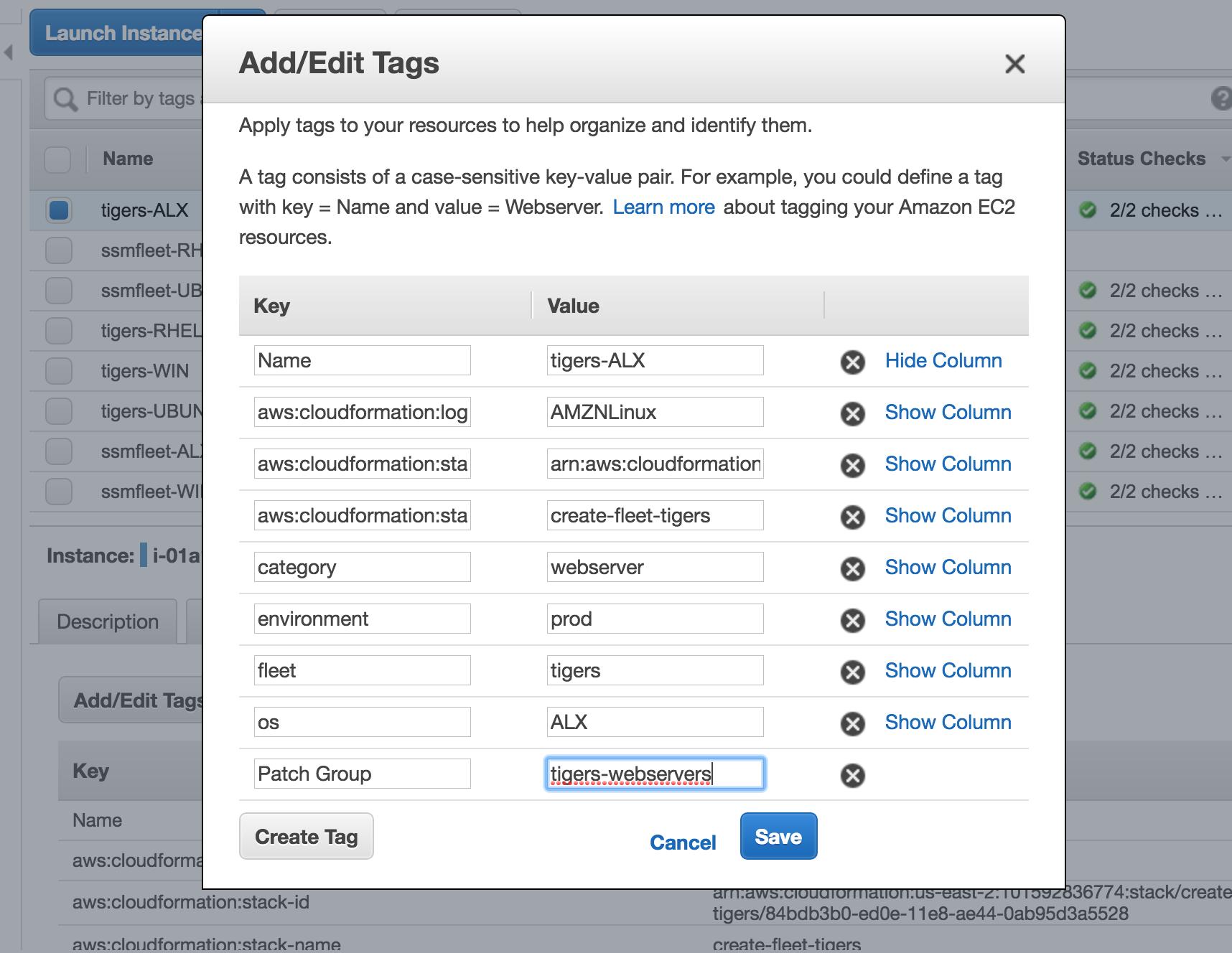
* + 1. Define a patch group and attach it to one of your default and custom baselines
       1. On the *Patch Baselines* screen, select the Amazon Linux baseline (see 6.2.1 above if you can’t find the screen)
       2. On the *Actions* dropdown, select *Modify Patch Groups* as shown below



* + - 1. Enter the name *<your fleet>-webservers* for this patch group and press the *Add* button to save it, as shown below.



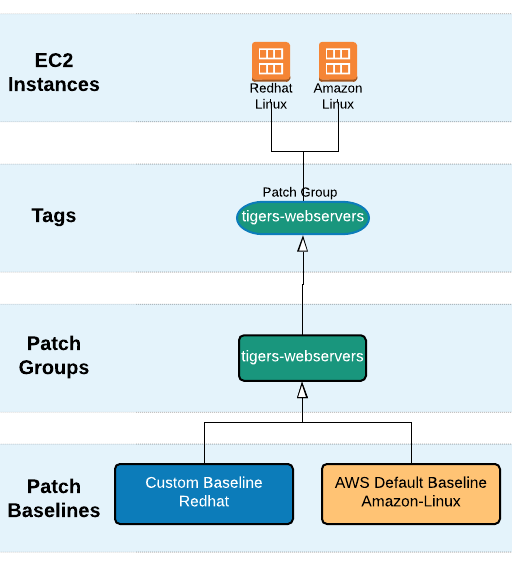
* + - 1. Press the *Close* button to return to the *Patch Baselines* screen.
      2. Select the CUSTOM baseline (*<your fleet>-RedhatTestingBaseline*) which you created in step 6.2.3.
      3. Repeat steps 6.2.4.2 through 6.2.4.4. to add the same *<your fleet>-webservers* patch group to this baseline.
      4. You have now created a patch group and associated it with two of your baselines.
    1. Now that you have created a patch group, you will link instances to it, by adding ***Patch Group*** tags to your Amazon Linux and Redhat instances.
       1. Open the [Amazon EC2 console](https://console.aws.amazon.com/ec2/), and choose *Instances* in the navigation pane.
       2. Choose your Amazon Linux instance.
       3. From the *Actions* menu, choose *Instance Settings*, *Add/Edit Tags*.
       4. Choose *Create Tag*.
       5. For the Key, type *Patch Group* (NOTE: must be typed exactly as shown below).
       6. In the Value field, type *<your fleet>-webservers* and press *Save*.



* + - 1. Repeat steps 6.2.5.2 through 6.2.5.6 to add the Redhat instance to this patch group.

|  |
| --- |
| NOTE: An instance can only be assigned to ONE patch group |

Congratulations! You have completed the definition of the patch group below.

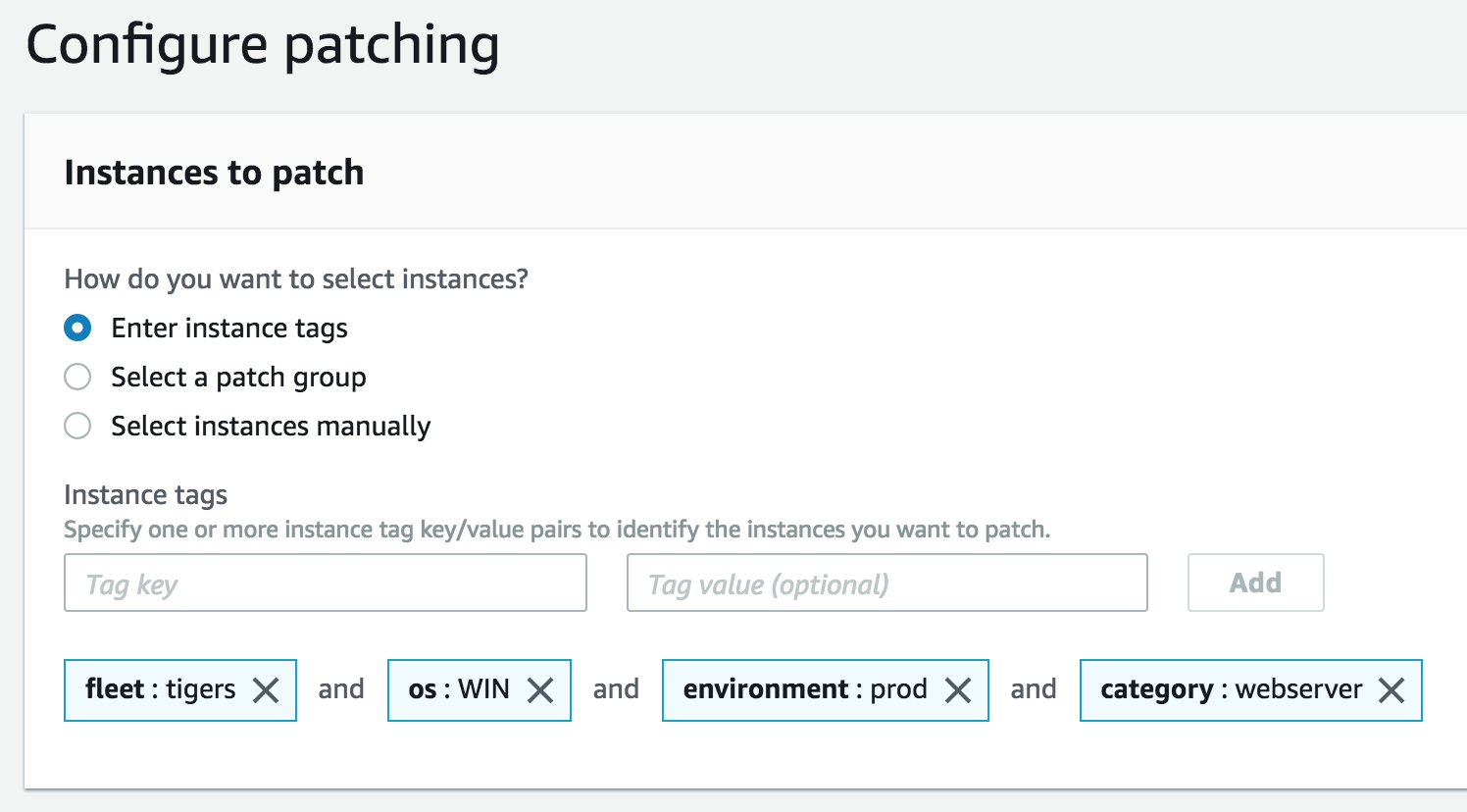
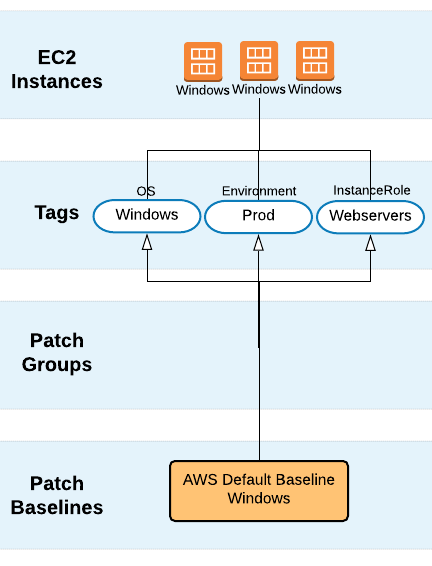


* 1. Configuring Patching – there are three ways to configure patching described below. Try as many as you can in the allotted time.

* + 1.  Option 1: Configure patching manually for individual instances
       1. Select ***Patch Manager*** from the left navigation bar.
       2. Press the *Configure Patching* button
       3. For instances to patch, choose *Select instances manually*
       4. On the instances list, select your UBUNTU server
       5. Select *Skip scheduling and patch instances now*
       6. For patching operation, select *Scan only*
       7. Press the *Configure Patching* button. This will execute a run command immediately to scan the instance and compare against the baseline.
       8. Return to the Patch Compliance Dashboard (6.1 above) and you should see that this instance is now registered for patching and being tracked.

|  |
| --- |
| NOTE: You can only associate DEFAULT baselines when you select individual instances |

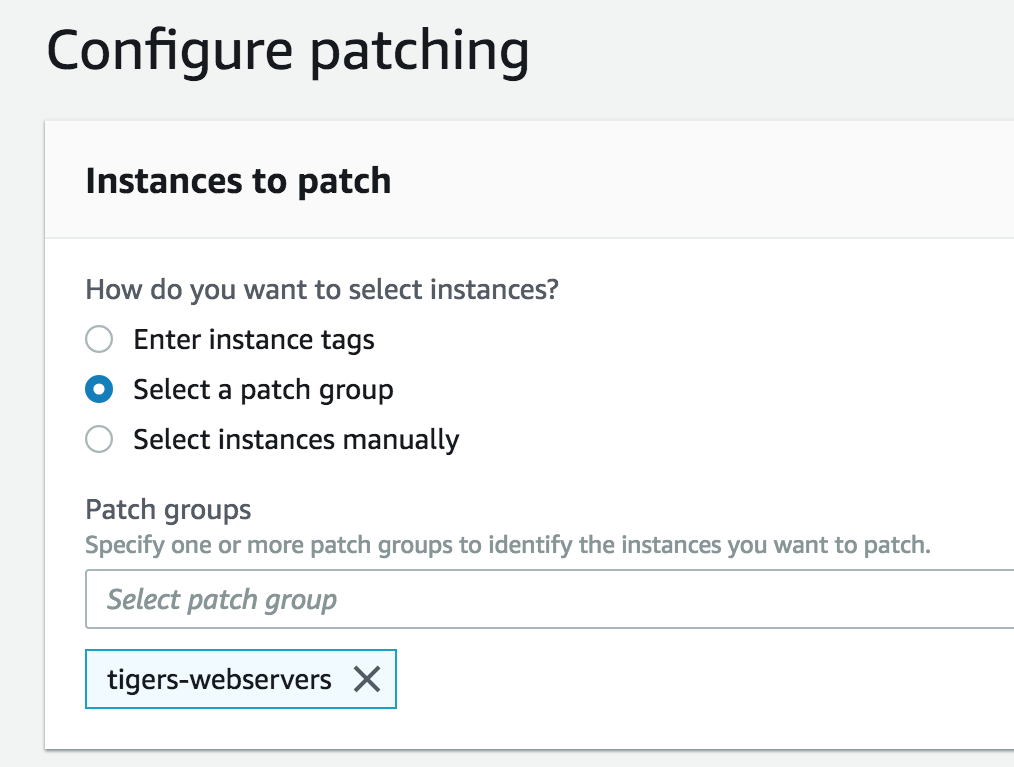
* + 1. Option 2: Configure patching using Instance Tags. We will patch all production Windows webservers in your fleet.
       1. Select ***Patch Manager*** from the left navigation bar.
       2. Press the *Configure Patching* button
       3. Choose *Enter instance tags*
       4. For *Instance Tags* specify the key/value pair for fleet=tigers and press the Add button.
       5. Repeat the process for the following key/value tag pairs: os=WIN, environment=prod, category=webservers as shown below.

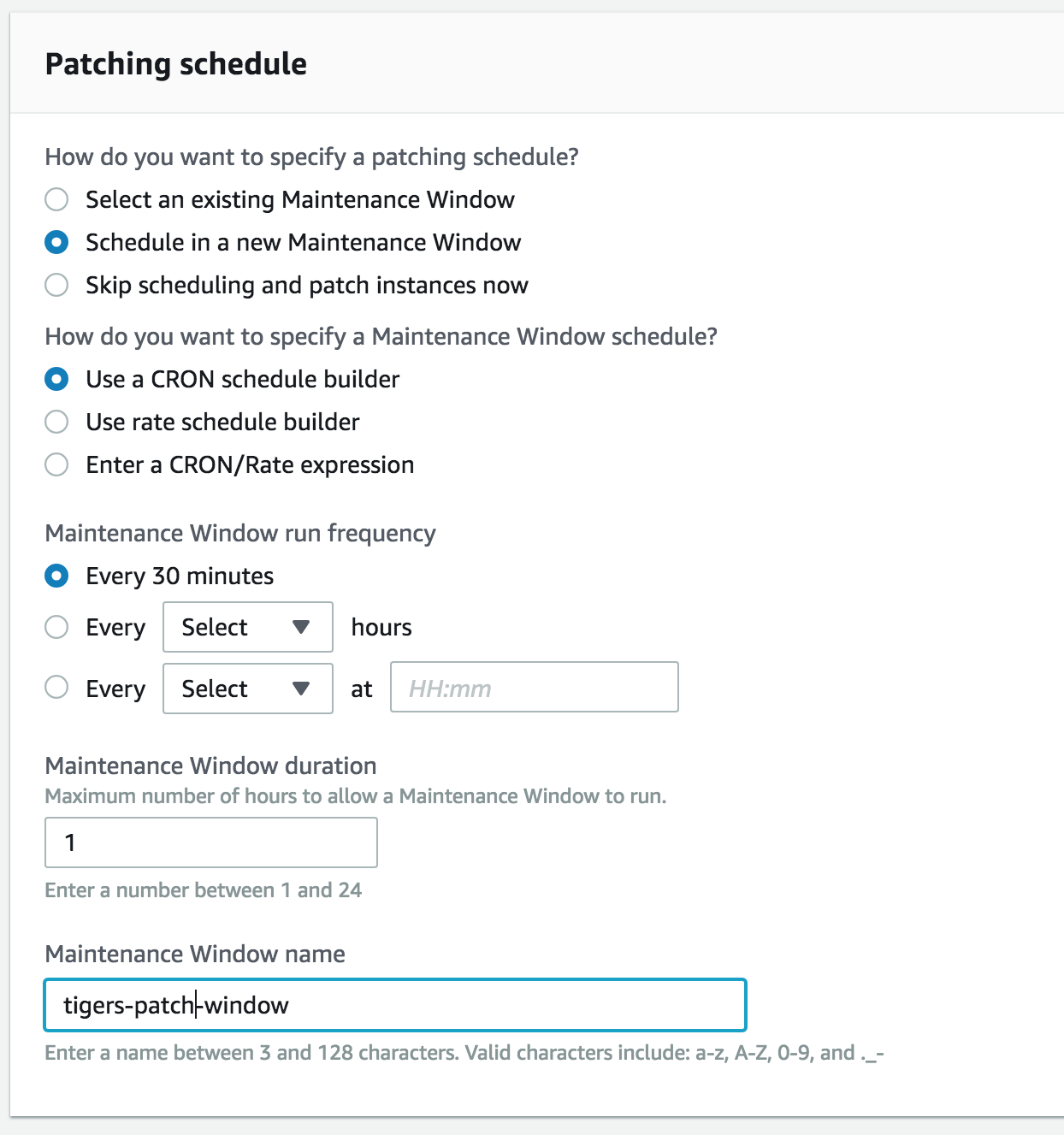
* + - 1. Select *Skip scheduling and patch instances now*
      2. This time, install patches on the instance. Select *Scan and install.*
      3. Press the *Configure Patching* button
      4. This will patch the instance immediately if it is out of compliance. You can monitor the status of this process on the *Command history* tab of the *Run Command* menu.

|  |
| --- |
| NOTE: Regular tags can only associate to DEFAULT baselines (no custom) |

* + 1. Option 3: Configure Patching using pre-existing Patch Groups
       1. Select ***Patch Manager*** from the left navigation bar
       2. Press the *Configure Patching* button
       3. Choose *Select a Patch Group*
       4. Select the *<your fleet>-webservers* patch group you created in section 6.2.4



* + - 1. For patching schedule, select *Schedule in a new maintenance window*
      2. Select the *CRON schedule builder* with a run frequency of 30 minutes, so that it runs soon
      3. Set the *Maintenance window duration* at 1 hour
      4. Enter a *Maintenance window name* such as *<your fleet>-patch-window*



* + - 1. For *patching operation*, select *Scan and install*
      2. Press the *Configure Patching* button
      3. Check the Patch Compliance dashboard later to see patching results

1. **SSM Maintenance Windows** Goal: define a maintenance window to perform automated tasks
   1. Select ***Maintenance Windows*** from the left navigation bar.

|  |
| --- |
| About Maintenance Windows |
| Maintenance windows merely establish the SCHEDULE component of an SSM association.  Associations link together targets, documents, schedules and configuration details to perform work, such as patching, configuration, etc.      If time allows, you may create a maintenance window by following the example below:      Once you save the maintenance window, it will be available for doing various kinds of work. |

1. **SSM State Manager** Goal: create a custom configuration state, then execute it

|  |
| --- |
| About State Manager |
| SSM State Manager will help you maintain your fleet in a desired state.  Configuration states are maintained by establishing an SSM Association, which links together targets, documents, schedules and configuration details to perform maintenance. |

* 1. Select ***State Manager*** from the left navigation bar
  2. Press the *Create association* button
  3. In the Name box, type *<your fleet>-virus-virus-scan-association*, then follow the example below to finish the association

|  |
| --- |
|  |

* 1. Press the *Create Association* button to complete the state definition.
  2. After a few minutes, view the association history (if you need full instructions here >>>[**Viewing Association Histories**](https://docs.aws.amazon.com/systems-manager/latest/userguide/sysman-state-assoc-history.html)

1. **Inventory Reporting** Goal: learn how to report on inventory data using Athena and Quicksight
   1. Follow the abbreviated steps below to export a small subset of your inventory data to Athena (complete inventory export process, plus Quicksight described ([here](https://docs.aws.amazon.com/systems-manager/latest/userguide/sysman-inventory-resource-data-sync.html))