



AWS Management & Governance Blog

Preventing blacklisted applications with AWS Systems Manager and AWS Config

by Tanu Mutreja | on 26 APR 2018 | in [AWS Config](#), [AWS Systems Manager](#), [Management Tools](#) | [Permalink](#) | [Share](#)

[AWS Systems Manager](#) Inventory collects metadata from Amazon EC2 instances and on-premises instances. AWS Systems Manager Inventory integrates with [AWS Config](#) to record inventory data for historical views, change tracking, or auditing. When you use AWS Config recording for systems inventory data you can enable scenarios such as tracking newly installed or removed software applications, assessing security risks, troubleshooting, and tracking license usage. Additionally, you can create [AWS Config Rules](#) to define compliance rules based on inventory data (such as, detecting a blacklisted application) and take remediation action (such as sending email notifications or running an AWS Lambda function to uninstall the application) automatically.

In this blog post, we'll walk you through an example that shows how to use AWS Systems Manager Inventory with AWS Config to detect and track changes in applications installed on an instance, and with AWS Config and Config rules to detect prohibited (aka blacklisted) applications installed on your managed instances and report non-compliance.

Requirements

This blog requires a managed instance, that is an EC2 instance or on-premises instance that has AWS Systems Manager Agent (also called SSM Agent) installed and has IAM role with *AmazonEC2RoleforSSM* policy attached to it. Here are the instructions on how to convert an un-managed instance to a managed instance:

1. Create an IAM role. For this blog post we'll call it *MyAmazonEC2RoleforSSM*. Attach the *AmazonEC2RoleforSSM* policy to the role. To do this in the AWS Systems Manager console, from *AWS Service Roles*, select *Amazon EC2 Role for Simple Systems Manager*.
2. Attach the IAM role *MyAmazonEC2RoleforSSM* to a new or existing EC2 instance.
3. Install SSM Agent: AWS Systems Manager Agent (SSM Agent) is Amazon software that runs on your Amazon EC2 instances and your hybrid instances that are configured for Systems Manager. SSM Agent is installed, by default, on Amazon Linux base AMIs dated 2017.09 and later. SSM Agent is installed by default on Windows Server 2016 instances and instances created from Windows Server 2003-2012 R2 AMIs published in November 2016 or later. You must manually install SSM Agent on other versions of

Linux, including non-base images like *Amazon ECS-Optimized AMIs*. More details are available [here](#)

Using AWS Config and AWS Config Rules with AWS Systems Manager

Step 1: Collect Inventory from managed instances

In the AWS Management Console, go to the AWS Systems Manager console and choose **Managed Instances** on the left navigation pane. This should list all EC2 instances or on-premises managed instances in your account. Choose **Setup Inventory** and select the EC2 instance you want to collect inventory from. (For the example in this blog post I'm collecting inventory on all default types and using default values for inventory collection schedule.) Choose **Setup Inventory** to complete the action. Verify that the instance has collected an inventory of applications installed on the instance. When you choose the instance and then choose **Inventory**, a list of applications displays that is similar to the following one:

The screenshot shows the AWS Systems Manager console. On the left, the 'Managed Instances' link is highlighted under the 'Shared Resources' section. The main console area shows the 'Inventory' tab for a specific instance (ID: i-05...). Below the 'Inventory type' dropdown (set to 'AWS:Application'), there is a table of installed applications.

Name	Version	Publisher	Application type	Installed time (UTC)	Architecture	URL
aci	2.2.49	Amazon.com	System Environment/Base	Mon, 15 Jan 2018 18:43:05 GMT	x86_64	http://aci.bestbits.at/
acpid	2.0.19	Amazon.com	System Environment/Daemons	Mon, 15 Jan 2018 18:43:03 GMT	x86_64	http://sourceforge.net/projects/acpid2/
alsa-lib	1.0.22	Amazon.com	System Environment/Libraries	Mon, 15 Jan 2018 18:42:55 GMT	x86_64	http://www.alsa-project.org/
amazon-ssm-agent	2.2.120.0	Amazon.com	Amazon/Tools	Mon, 15 Jan 2018 18:43:06 GMT	x86_64	http://docs.aws.amazon.com/ssm/latest/APIReference/Welcome.html
at	3.1.10	Amazon.com	System Environment/Daemons	Mon, 15 Jan 2018 18:43:03 GMT	x86_64	http://ftp.debian.org/debian/pool/main/a/at
attr	2.4.46	Amazon.com	System Environment/Base	Mon, 15 Jan 2018 18:43:05 GMT	x86_64	http://aci.bestbits.at/
audit	2.6.5	Amazon.com	System Environment/Daemons	Mon, 15 Jan 2018 18:43:05 GMT	x86_64	http://people.redhat.com/sgrubb/audit/
audit-libs	2.6.5	Amazon.com	Development/Libraries	Mon, 15 Jan 2018 18:42:57 GMT	x86_64	http://people.redhat.com/sgrubb/audit/
authconfig	6.2.8	Amazon.com	System Environment/Base	Mon, 15 Jan 2018 18:43:03 GMT	x86_64	https://fedorahosted.org/authconfig

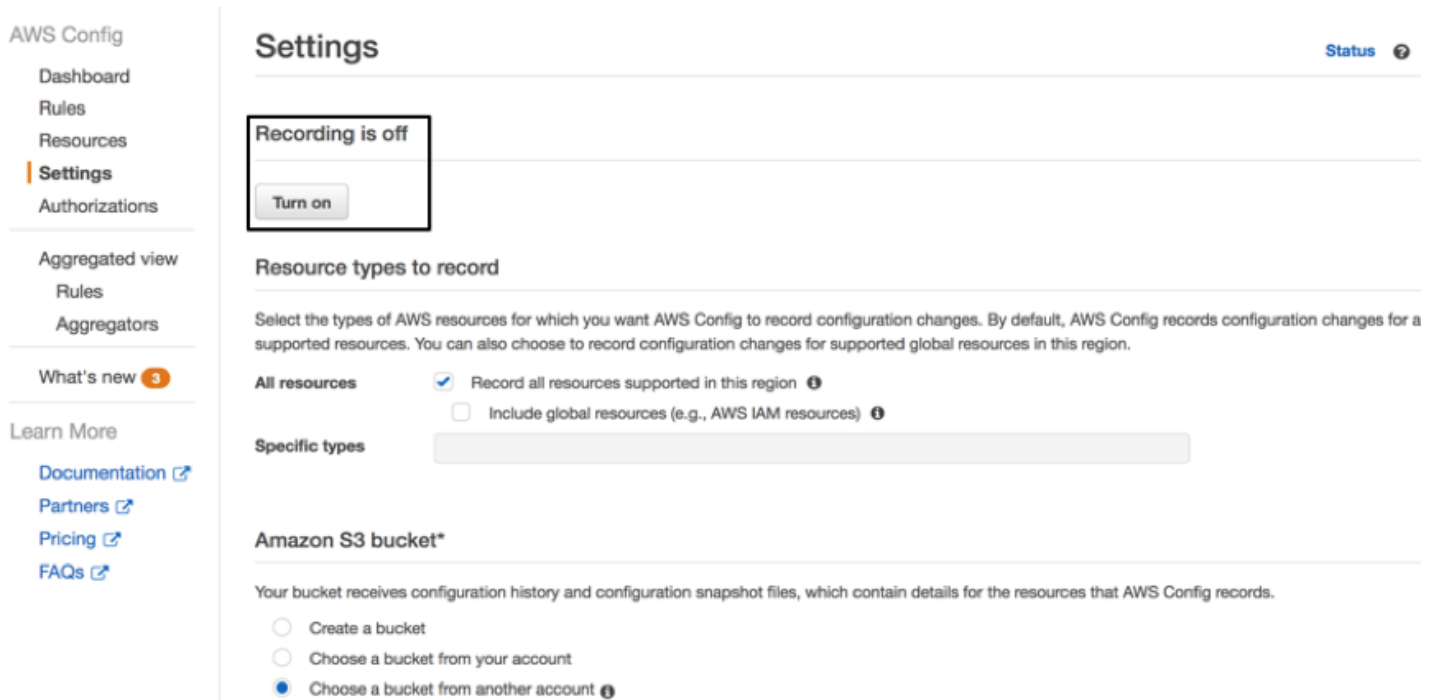
Step 2: Start recording AWS Systems Manager Inventory data to AWS Config

On the AWS Systems Manager console, choose Managed Instances, and then choose Edit AWS Config recording for the EC2 instance, as the following shows:

The screenshot shows the 'Managed instances' section of the AWS Systems Manager console. At the top, there are buttons for 'View details', 'Setup Inventory', 'Resource Data Syncs', and 'Actions'. Below these is a search bar and a table with columns 'Instance ID', 'Name', and 'Ping status'. The 'Actions' dropdown menu is open, showing several options: 'Create Association', 'Run Command', 'Execute Automation', 'Edit AWS Config recording' (which is highlighted), 'Change IAM role', and 'Deregister this managed instance'.

This should redirect to the AWS Config console as follows:

<https://aws.amazon.com/blogs/mt/preventing-blacklisted-applications-with-aws-systems-manager-and-aws-config/>



Choose recording to **Turn On**.

Step 3: Track changes to the applications installed on a managed instance

To see an example of changes that are tracked in software applications that are installed on a managed instance that collects inventory, let's install a Java Development Kit (JDK) on the managed EC2 instance.

- 1. Use SSH to connect to the managed EC2 instance.
- 2. Install the JDK using the following command.

```
[ec2-user@ip-172-xx-x-xxx ~]$ sudo yum install java-1.8.0-openjdk-devel
```

- 3. Make the default JDK the newly installed JDK 1.8.

```
[ec2-user@ip-172-xx-x-xxx ~]$ sudo alternatives --config java
There are 2 programs which provide 'java'.
  Selection    Command
-----
*+ 1          /usr/lib/jvm/jre-1.7.0-openjdk.x86_64/bin/java
    2          /usr/lib/jvm/jre-1.8.0-openjdk.x86_64/bin/java
Enter to keep the current selection[+], or type selection number: 2
```

- 4. Check to see if the default JDK is 1.8.

```
[ec2-user@ip-172-xx-x-xxx ~]$ java -version
openjdk version "1.8.0_161"
OpenJDK Runtime Environment (build 1.8.0_161-b14)
OpenJDK 64-Bit Server VM (build 25.161-b14, mixed mode)
[ec2-user@ip-172-xx-x-xxx ~]$ javac -version
javac 1.8.0_161
```

5. Wait for next scheduled Inventory collection on the instance. Then in the AWS Systems Manager console, on the left navigation pane, choose **Managed Instances**, then choose EC2 instance, and choose **Inventory** to ensure that the application is showing as installed on the system. Now choose EC2 instance and go to the AWS Config console. Note that it might take a few minutes for the changes to show up in the timeline of changes in AWS Config, based on the schedule we selected earlier.

SSM ManagedInstanceInventory
i-0[REDACTED]
on February 19, 2018 9:01:47 PM IST (UTC+05:30)

Manage resource ?

Timeline of changes:

- 19th February 2018 3:50:47 PM
- 19th February 2018 4:00:29 PM (3 Changes)
- 19th February 2018 9:00:06 PM (5 Changes)

Now

▼ Configuration Details [View Details](#)

Amazon Resource Name	arn:aws:ssm:us-west-2:[REDACTED]:managed-instance-inventory/i-[REDACTED]	Computer name	[REDACTED]-west-2.compute.internal
Resource type	AWS::SSM::ManagedInstanceInventory	Platform name	Amazon Linux AMI
Resource ID	[REDACTED]	Platform type	Linux
Resource name	null	Agent type	amazon-ssm-agent
Availability zone	null	Agent version	2.2.120.0
Created on	Not available	IP address	[REDACTED]
Tags (0)		Installed applications	View installed applications
		AWS Components	View AWS components

6. Choosing Changes gives you details of the applications that have been added/changed.

Configuration Changes **5**

Field	From	To
Configuration.AWS:Application.Content.lksctp-tools		▼ Object ApplicationType: "System Environment/Libraries" InstalledTime: "2018-02-19T15:11:36Z" Architecture: "x86_64" Version: "1.0.10" Summary: "User-space access to Linux Kernel SCTP" PackageId: "lksctp-tools-1.0.10-7.7.amzn1.src.rpm" Publisher: "Amazon.com" URL: " http://lksctp.sourceforge.net " Name: "lksctp-tools"
Configuration.AWS:Application.Content.java-1.8.0-openjdk		▼ Object ApplicationType: "Development/Languages" InstalledTime: "2018-02-19T15:11:37Z" Architecture: "x86_64" Version: "1.8.0.161" Summary: "OpenJDK Runtime Environment" PackageId: "java-1.8.0-openjdk-1.8.0.161-0.b14.36.amzn1.src.rpm" Publisher: "Amazon.com" URL: " http://openjdk.java.net/ " Name: "java-1.8.0-openjdk"
Configuration.AWS:Application.Content.copy-jdk-configs		▼ Object ApplicationType: "Unspecified" InstalledTime: "2018-02-19T15:11:36Z" Architecture: "noarch" Version: "1.2" Summary: "JDKs configuration files copier" PackageId: "copy-jdk-configs-1.2-1.2.amzn1.src.rpm" Publisher: "Amazon.com" URL: " https://hg.fedorahosted.org/hg/copy_jdk_configs " Name: "copy-jdk-configs"
Configuration.AWS:Application.Content.java-1.8.0-		▼ Object

Step 4: Apply AWS Config rules to detect prohibited or blacklisted applications

Today AWS Config has four built-in rules for AWS Systems Manager (more details on AWS Config rules can be found [here](#)).

1. *ec2-managedinstance-applications-blacklisted* – Checks that none of the specified applications are installed on the instance. Optionally, specify the version. Newer versions will not be blacklisted. Optionally, specify the platform to apply the rule only to instances running that platform.
2. *ec2-managedinstance-applications-required* – Checks whether all of the specified applications are installed on the instance. Optionally, specify the minimum acceptable version. Optionally, specify the platform to apply the rule only to instances running that platform.
3. *ec2-managedinstance-inventory-blacklisted* – Checks whether instances managed by AWS Systems Manager are configured to collect blacklisted inventory types.
4. *ec2-managedinstance-platform-check* – Checks whether EC2 managed instances have the desired configurations.

AWS Config

Dashboard

Rules

Resources

Settings

What's new

Learn More

[Documentation](#)[Partners](#)[Pricing](#)[FAQs](#)

Rules > Add rule

Add rule

Add rules to define the desired configuration settings of your AWS resources. Customize any of the following rules to suit your needs, or add a custom rule. To add a custom rule, you must create an AWS Lambda function for the rule.

[Add custom rule](#)

systems manager

<< < Viewing 1 - 4 of 4 AWS managed rules > >>

ec2-managedinstance-applications-bl...

Checks that none of the specified applications are installed on the instance. Optionally, specify the version. Newer versions will not be blacklisted. Optionally,

Systems Manager

ec2-managedinstance-applications-re...

Checks whether all of the specified applications are installed on the instance. Optionally, specify the minimum acceptable version. Optionally, specify the platform to

Systems Manager

ec2-managedinstance-inventory-blac...

Checks whether instances managed by Amazon EC2 Systems Manager are configured to collect blacklisted inventory types.

Systems Manager

ec2-managedinstance-platform-check

Checks whether EC2 managed instances have the desired configurations.

Systems Manager

For the purpose of this blog, let's select a built-in AWS Config rule for EC2 – *ec2-managedinstance-applications-blacklisted*. Using this rule, we will specify an application that we want to prohibit / blacklist in my organization and set a remediation action so that I get email notifications any time the application is installed on one or more managed instances in my fleet. In this example, we will evaluate if an old JDK like java-1.7.0-openjdk is installed on the managed EC2 instance.

Step A. Select the *ec2-managedinstance-applications-blacklisted* Config rule.

AWS Config

Dashboard

Rules

Resources

Settings

What's new

Learn More

[Documentation](#)[Partners](#)[Pricing](#)[FAQs](#)

Add rule

Add rules to define the desired configuration settings of your AWS resources. Customize any of the following rules to suit your needs, or add a custom rule. To add a custom rule, you must create an AWS Lambda function for the rule.

[Add custom rule](#)

ec2

<< < Viewing 1 - 9 of 18 AWS managed rules > >>

approved-amis-by-id

Checks whether running instances are using specified AMIs. Specify a list of approved AMI IDs. Running instances with AMIs that are not on this list are noncompliant.

EC2

approved-amis-by-tag

Checks whether running instances are using specified AMIs. Specify the tags that identify the AMIs. Running instances with AMIs that don't have at least one of the specified tags

EC2

cloudwatch-alarm-resource-check

Checks whether the specified resource type has a CloudWatch alarm for the specified metric. For resource type, you can specify EBS volumes, EC2 instances, RDS clusters,

CloudWatch

desired-instance-tenancy

Checks instances for specified tenancy. Specify AMI IDs to check instances that are launched from those AMIs or specify Host IDs to check whether instances are launched on

EC2

desired-instance-type

Checks whether your EC2 instances are of the specified instance types.

EC2

ebs-optimized-instance

Checks whether EBS optimization is enabled for your EC2 instances that can be EBS-optimized.

EC2

ec2-instance-detailed-monitoring-ena...

Checks whether detailed monitoring is enabled for EC2 instances.

EC2

ec2-instances-in-vpc

Checks whether your EC2 instances belong to a virtual private cloud (VPC). Optionally, you can specify the VPC ID to associate with your instances.

EC2

ec2-managedinstance-applications-bl...

Checks that none of the specified applications are installed on the instance. Optionally, specify the version. Newer versions will not be blacklisted. Optionally,

Systems Manager

Step B. Specify java-1.7.0-openjdk as the application to be prohibited

AWS Config

Dashboard

Rules

Resources

Settings

Authorizations

Aggregated view

Rules

Aggregators

What's new 2

Learn More

Documentation

Partners

Pricing

FAQs

Rules > Configure rule

Add AWS managed rule

AWS Config evaluates your AWS resources against this rule when it is triggered.

Name*

ec2-managedinstance-applications-blacklisted

A unique name for the rule. 64 characters max. No special characters or spaces.

Description

Checks that none of the specified applications are installed on the instance. Optionally, specify the version. Newer versions will not be blacklisted. Optionally, specify the platform to apply the rule only to instances running that platform.

Managed rule name

EC2_MANAGEDINSTANCE_APPLICATIONS_BLACKLISTED

Trigger

AWS Config evaluates resources when the trigger occurs.

Trigger type*

☒ Configuration changes

☐ Periodic

Scope of changes*

☒ Resources

☐ Tags

☐ All changes

Resources*

SSM: ManagedInstanceInventory

Resource identifier (optional)

This rule can be triggered only when recorded resources are created, changed, or deleted. Specify which resources are recorded on the Settings page.

Rule parameters

Rule parameters define attributes for which your resources are evaluated; for example, a required tag or S3 bucket.

Key	Value
applicationNames	java-1.7.0-openjdk
platformType	Value (optional)

* Required

Step C: Report compliance against the *ec2-managedinstance-applications-blacklisted* Config rule.

The AWS Config rule evaluates and shows the findings as compliant or as non-compliant. In this case, since **java-1.7.0-openjdk** was installed on the instance, the AWS Config rule will report the instance non-compliant

https://aws.amazon.com/blogs/mt/preventing-blacklisted-applications-with-aws-systems-manager-and-aws-config/

7/12

as the following indicates.

Rules > Rule details

ec2-managedinstance-applications-blacklisted

Description Checks that none of the specified applications are installed on the instance. Optionally, specify the version. Newer versions will not be blacklisted. Optionally, specify the platform to apply the rule only to instances running that platform.

Trigger type Configuration changes

Scope of changes Resources

Resource types SSM ManagedInstanceInventory

Config rule ARN arn:aws:config:us-west-2:123456789012:config-rule/config-rule-padbdt

Parameters applicationNames: java-1.7.0-openjdk platformType: null

Overall rule status Last successful invocation on February 19, 2018 at 9:40:20 PM Last successful evaluation on February 19, 2018 at 9:40:21 PM

Resources evaluated

Click on the icon to view configuration details for the resource when it was last evaluated with this rule.

Resource type	Config timeline	Compliance	Last successful invocation	Last successful evaluation	Manage resource
SSM ManagedInstanceInventory		Noncompliant	February 19, 2018 9:40:20 PM	February 19, 2018 9:40:29 PM	

Annotation

The EC2 instance has the following blacklisted applications: java-1.7.0-openjdk.

Step D: Take automated remediation action, such as sending email notification of non-compliance.

(Note: As part of remediation actions that you take using AWS Config Rule, you could also execute an AWS Lambda function that will automatically uninstall the blacklisted application.)

We can stream configuration changes and notifications to an Amazon Simple Notification Service (SNS) topic. In the AWS Config console, in the left navigation pane, choose **Settings**, and then choose the **Amazon SNS topic**.

AWS Config

- Dashboard
- Rules
- Resources
- Settings**
- What's new
- Learn More
 - Documentation
 - Partners
 - Pricing
 - FAQs

Your bucket receives configuration history and configuration snapshot files, which contain details for the resources that AWS Config records.

☐ Create a bucket
☒ Choose a bucket from your account
☐ Choose a bucket from another account

Bucket name* config-bucket-123456789012 / **Prefix (optional)** / AWSLogs-123456789012 / Config/us-west-2

Amazon SNS topic

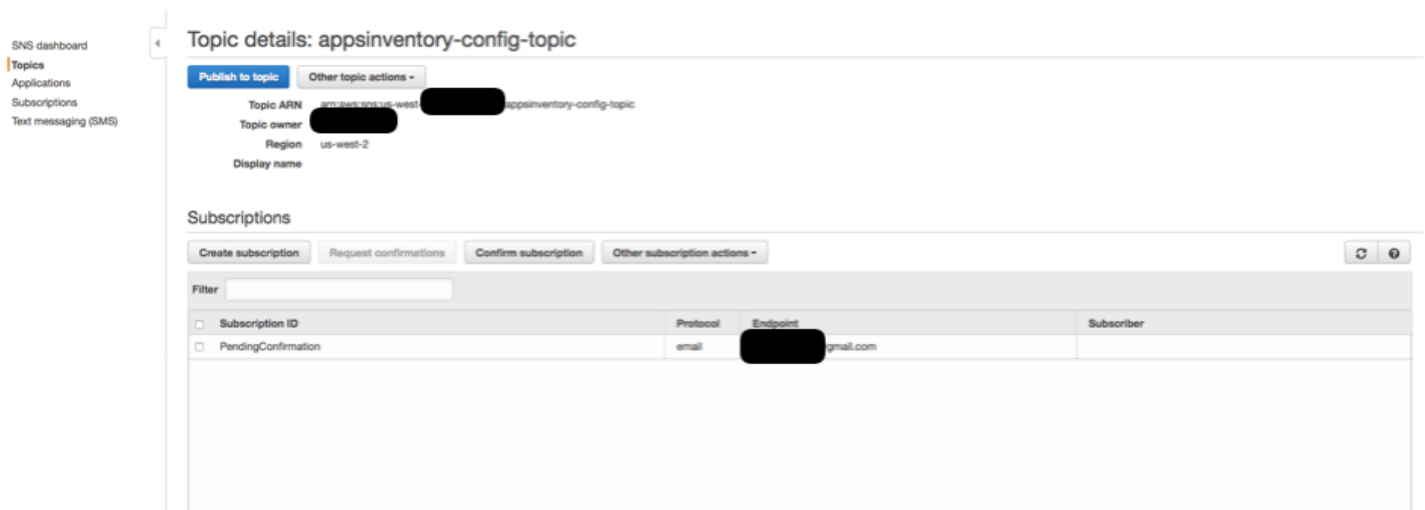
☒ Stream configuration changes and notifications to an Amazon SNS topic.

If you choose email as the notification endpoint for your SNS topic, this can cause a high volume of email. [Learn more.](#)

☒ Create a topic
☐ Choose a topic from your account
☐ Choose a topic from another account

Topic name* appsinventory-config-topic

Go to the dashboard in the SNS console and create an subscription to this topic to either send an email, send a SMS, call a Lambda function, or use other delivery options. We will just choose **Create subscription** to send an email.



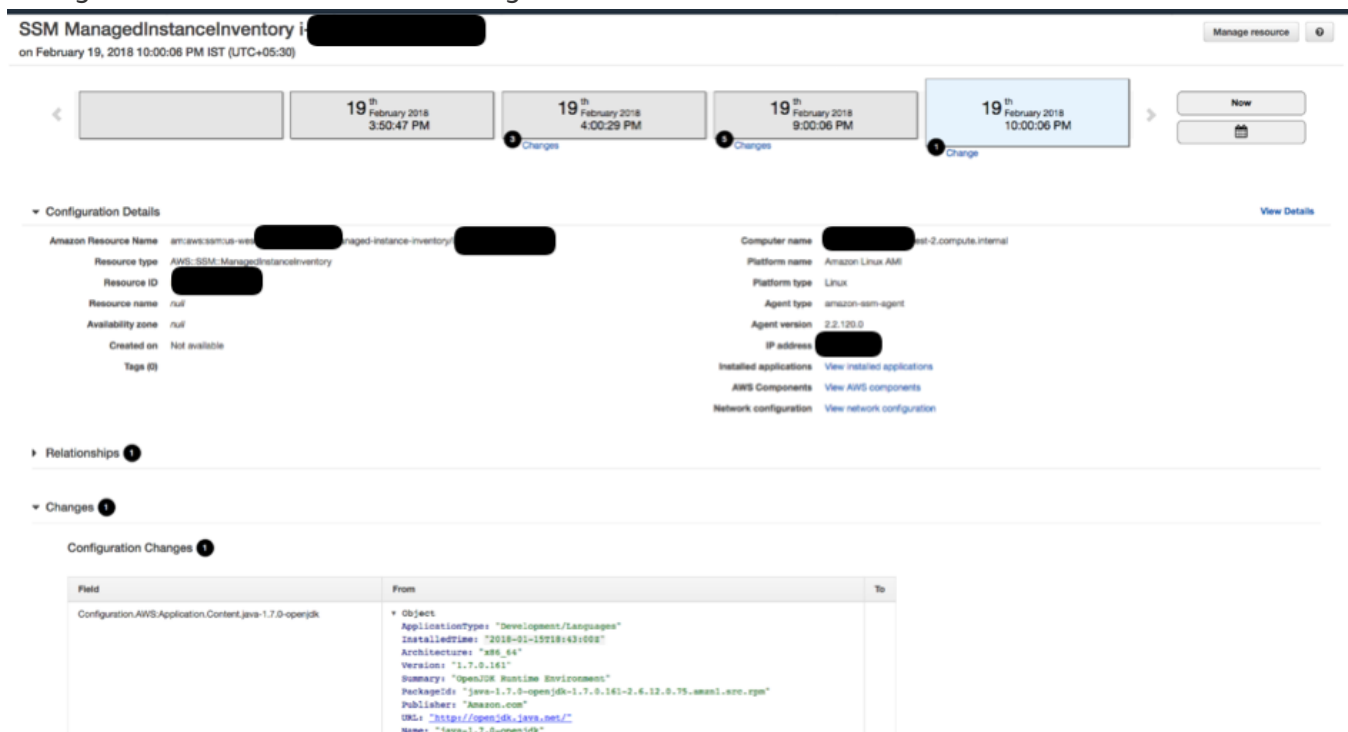
Now we'll choose Confirm subscription to send an email to the email address specified. We will go to the specified email inbox and confirm SNS subscription so that we're ready to start receiving SNS notifications.

Step E: Maintain compliance by uninstalling the blacklisted application.

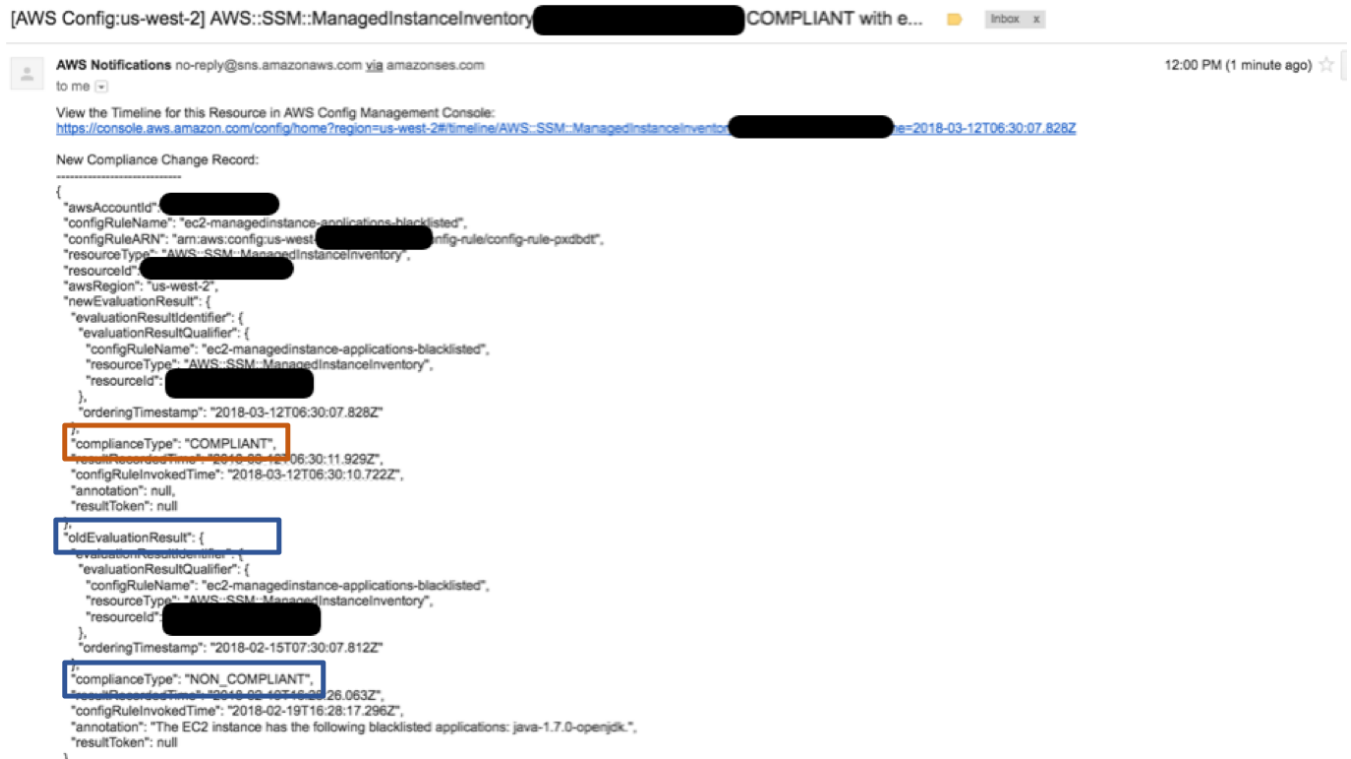
1. Let's uninstall JDK 1.7 and see whether the EC2 instance becomes compliant.
2. Using SSH, connect to the managed EC2 instance and uninstall JDK1.7

```
ec2-user@ip-172-xx-x-xxx ~]$ sudo yum remove java-1.7.0-openjdk
```

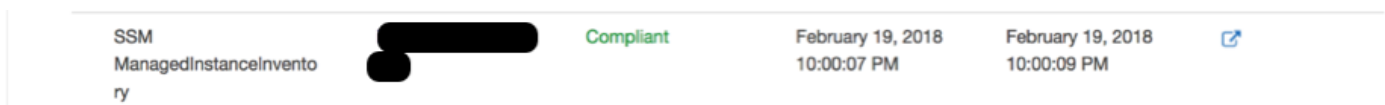
3. After the next Inventory collection (by default inventory collection happens every 30 minutes), AWS Config timeline will show that the changes have been recorded



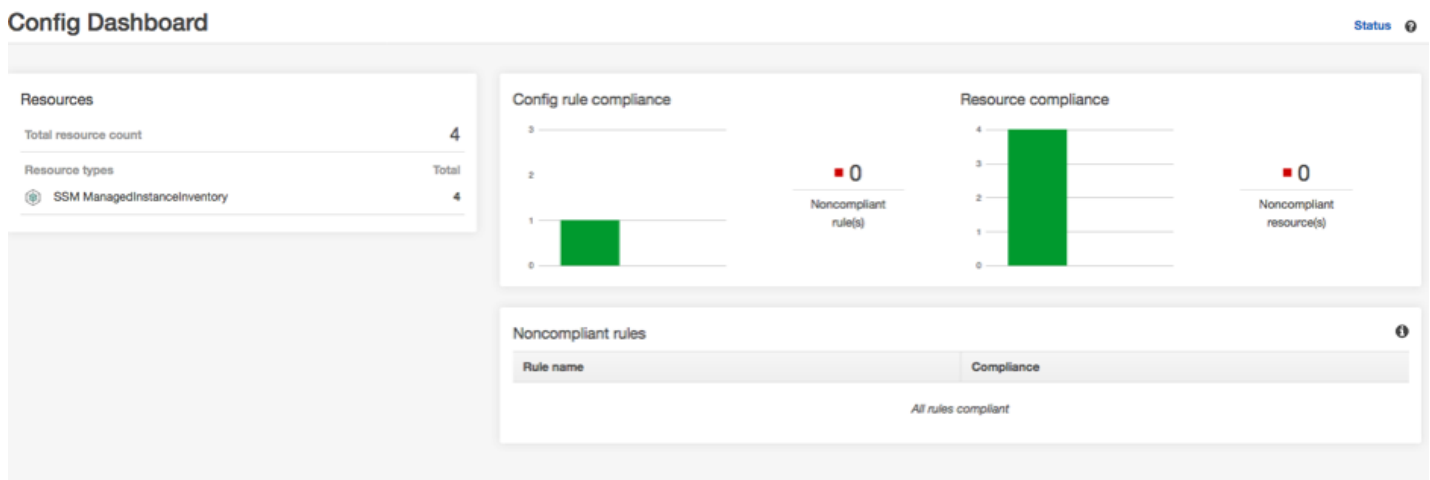
4. You will also get an email notification that says that the EC2 instance is now compliant.



5. Go to the AWS Config console Rules section and the EC2 instance will now show that it is compliant.



The AWS Config Dashboard will also show the overall compliance across all managed instances.



Conclusion

In this blog post we showed you how easy it is to set up an inventory of applications using AWS Systems Manager and record inventory data with AWS Config. We showed you how to set rules, remediation actions, and report compliance or non-compliance on inventory data based on AWS Config rules.

About the Authors



Mani Chandrasekaran is an AWS Solution Architect (SA) based in Bangalore, India.



Tanu Mutreja is a Senior Product Manager for AWS Systems Manager. She loves to work on innovative products and solve customer problems with simple end to end experiences.

TAGS: [application compliance](#), [application inventory](#), [application management](#), [AWS Config](#), [AWS Systems Manager](#), [AWS Systems Manager Inventory](#), [blacklisted applications](#)

Resources

[AWS Config](#)
[AWS CloudTrail](#)
[AWS OpsWorks](#)
[AWS Systems Manager](#)
[AWS Service Catalog](#)
[AWS CloudFormation](#)
[AWS Management Tools](#)

Follow

[Twitter](#)
[Facebook](#)
[LinkedIn](#)
[Twitch](#)



Amazon Managed Service for Grafana

Create unified, interactive data visualizations

[Learn more »](#)

Related Posts

[Viewing permission issues with service-linked roles](#)

[DevSecOps for auto healing PCI DSS 3.2.1 violations in AWS using custom AWS Config conformance packs, AWS Systems Manager and AWS CodePipeline](#)

[Automate FedRAMP controls in your AWS environment using AWS Config conformance packs](#)

[Automate Amazon S3 versioning using AWS Config rules](#)

[How data recipients can implement Open Banking on AWS](#)

[Audit your SAP systems with AWS Config – Part II](#)

[Audit your SAP systems with AWS Config – Part I](#)

[AWS Management and Governance at Re:Invent 2020](#)