# Use AWS Config to Monitor License Compliance on Amazon EC2 Dedicated Hosts

**This paper has been archived.**

For the latest technical guidance about Amazon EC2, see the AWS Whitepapers & Guides page: https://aws.amazon.com/whitepapers/



*April 2016*

# Notices

This document is provided for informational purposes only. It represents AWS's current product offerings and practices as of the date of issue of this document, which are subject to change without notice. Customers are responsible for making their own independent assessment of the information in this document and any use of AWS's products or services, each of which is provided "as is" without warranty of any kind, whether express or implied. This document does not create any warranties, representations, contractual commitments, conditions or assurances from AWS, its affiliates, suppliers or licensors. 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.

# Contents

# Abstract

Amazon Elastic Compute Cloud (EC2) Dedicated Hosts can help enterprises reduce costs by allowing the use of existing server-bound licenses. Many customers can also use Dedicated Hosts to address corporate compliance and regulatory requirements. Oftentimes, customers using Dedicated Hosts want to continuously record and evaluate changes to their infrastructure to stay compliant with license terms and regulatory requirements.

This paper outlines the ways in which you can leverage AWS Config and AWS Config Rules to monitor license compliance on Amazon EC2 Dedicated Hosts.

# Introduction

This paper discusses how you can set up AWS Config to record configuration changes to Amazon EC2 Dedicated Hosts and EC2 instances in order to ascertain your licensing compliance posture. You'll learn how to create AWS Config Rules to govern the way your server-bound licenses are used on Amazon Web Services (AWS). We'll create a sample rule that checks whether all instances in an account created from an Amazon Machine Image (AMI) called MyWindowsImage are launched onto a specific Dedicated Host. We'll also describe other checks that can be employed to monitor compliance with common licensing restrictions and to govern your Dedicated Host resources.

An Amazon EC2 Dedicated Host is a physical server with EC2 instance capacity fully dedicated for your use. You get complete visibility into the number of sockets and physical cores that support your instances on a Dedicated Host. Dedicated Hosts allow you to place your instances on a specific, physical server. This level of visibility and control in turn allows you to use your existing per-socket, per-core, or per-virtual machine (VM) software licenses (e.g., Microsoft Windows Server) to save costs and meet compliance and regulatory requirements.

To track the history of instances that are launched, stopped, or terminated on a Dedicated Host, you can use AWS Config. AWS Config pairs this information with host- and instance-level information relevant to software licensing, such as

the host ID, AMI IDs, and number of sockets and physical cores per host. You can then use this data to verify usage against your licensing metrics.

You can use AWS Config Rules to choose from a set of pre-built rules based on common AWS best practices or define custom rules. You can set up rules that check the validity of changes made to resources tracked by AWS Config against policies and guidelines defined by you. You can set these AWS Config Rules to evaluate each change to the configuration of a resource, or you can execute them at a set frequency. You can also author your own custom rules by creating AWS Lambda functions in any supported language.

# Setting Up AWS Config to Track Dedicated Hosts and EC2 Instances

Open the AWS Management Console and go to the EC2 console. On the EC2 Dedicated Hosts page, notice the **Edit Config Recording** button at the top. The icon in **red** indicates that AWS Config is not currently set up to record configuration changes to Dedicated Hosts and instances.

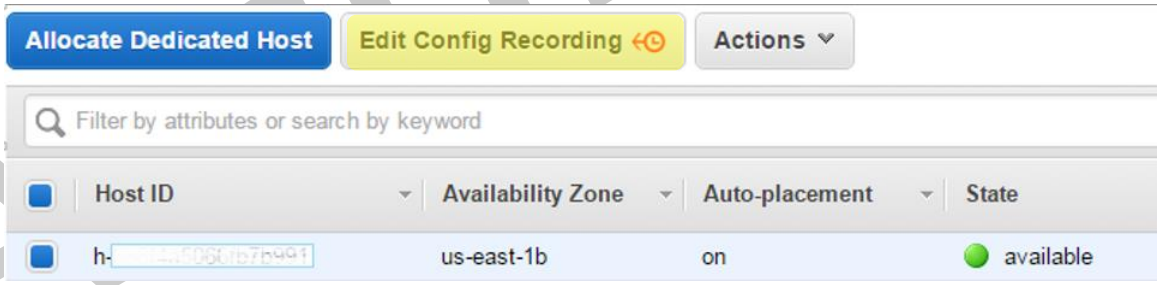


**Figure 1: Edit Config Recording Button with the Red Icon on Dedicated Host Console**

Getting started with AWS Config is simple. Click the **Edit Config Recording** button to open the AWS Config settings page. On this page, check **Record all resources supported in this region.**

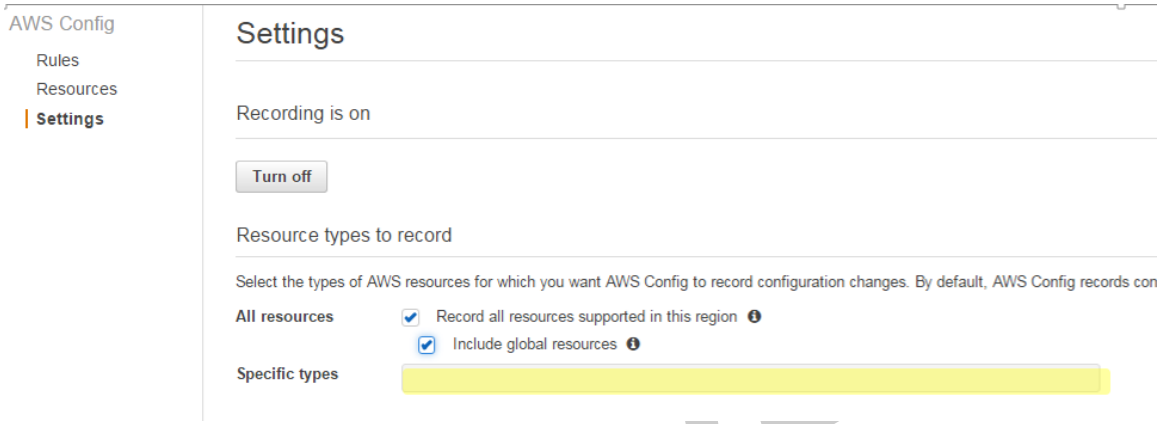


**Figure 2: Selecting Resource Types to Record on the AWS Config Settings Page**

You can choose to only enable recording for Dedicated Hosts and instances by selecting these resources in **Specific types**. If you are setting up AWS Config for the first time, you must specify an Amazon S3 bucket into which AWS Config can deliver configuration history and snapshot files. Optionally, you can also provide an Amazon Simple Notification Service (SNS) topic to which change and compliance notifications will be delivered. Finally, you'll be asked to grant appropriate permissions to AWS Config and save the settings. For more details on setting up AWS Config using the AWS Management Console or the CLI, see the Getting Started with AWS Config documentation.

After the AWS Config setup is complete, you'll notice that the icon on the EC2 console page for Dedicated Hosts has turned **green**. This indicates that AWS Config is recording configuration changes to all EC2 instances and Dedicated Hosts.

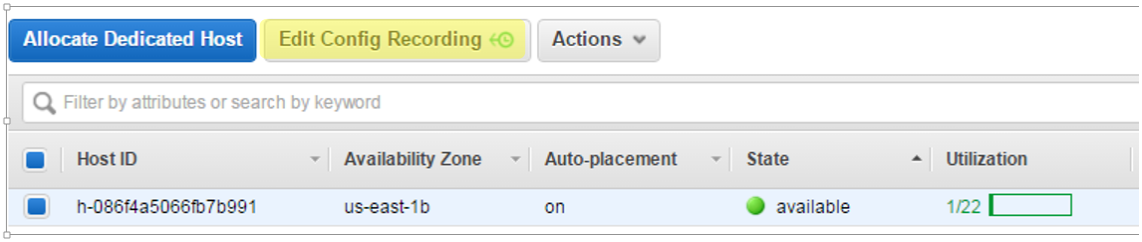


**Figure 3: The Edit Config Recording Button with Green Icon**

# Creating a Custom Rule to Check that Launched Instances Are on a Dedicated Host

Now that you have set up AWS Config to start recording configuration changes to Dedicated Hosts and EC2 instances, you can start writing rules to evaluate the license compliance state of all instances in the account. To get started, you will write a rule that checks whether all instances launched from the MyWindowsImage AMI are placed onto a specific Dedicated Host. For this sample, assume that MyWindowsImage is the name of an AMI you have imported and is the machine image of a Microsoft Server license you own.

Before creating the rule, first inspect the instances and Dedicated Hosts on your account: Look up **EC2 Instance** and **EC2 Host** resource types. In Figure 4, you can see one Dedicated Host and a number of instances.

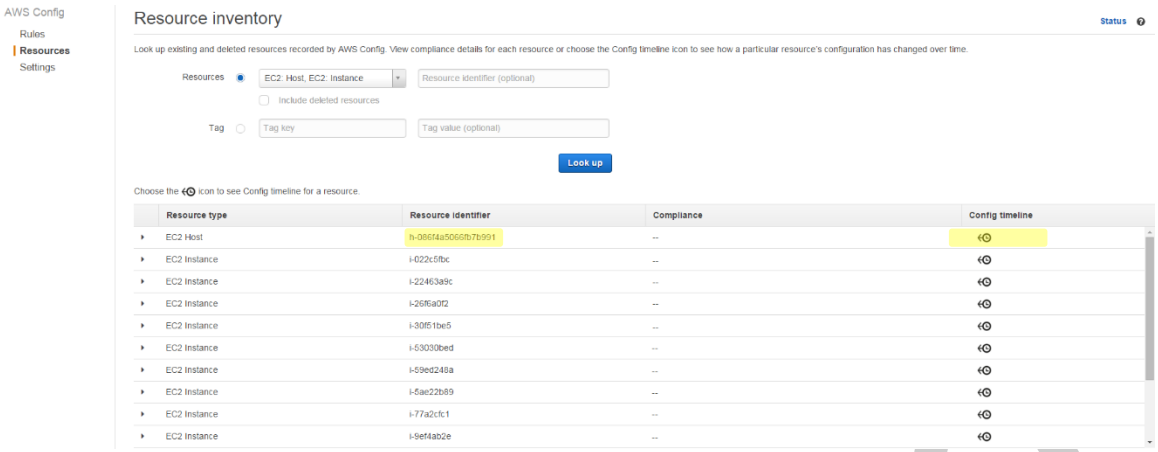


**Figure 4: Review the Resource Inventory**

Click the icon for the Dedicated Host to go to the Config Timeline to see the configuration of the Dedicated Host including the sockets, cores, total vCPUs, and available vCPUs. You can also see all the instances that are currently running on the host. Traversing the timeline provides all historical configurations of the Dedicated Host, including the instances that were launched onto the Dedicated Host in the past. You also can look into the Config timeline of each of those instances.

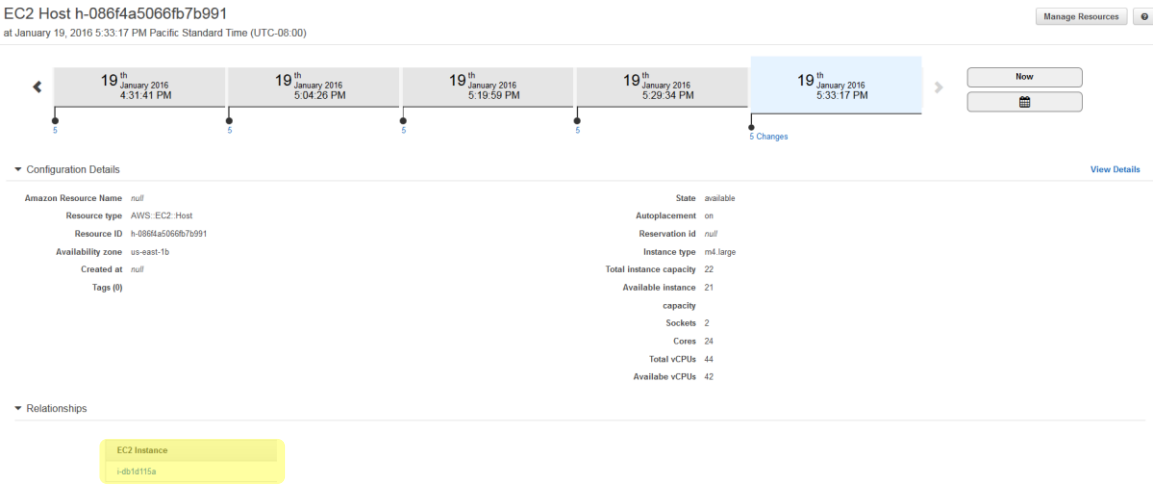


**Figure 5: The Config Resource Configuration History Timeline**

Next, you will set up the new rule in AWS Config and write the AWS Lambda function for the rule. To do this, click **Add rule** in the AWS Config console, and then click **Create AWS Lambda function** to set up the function you want to execute.

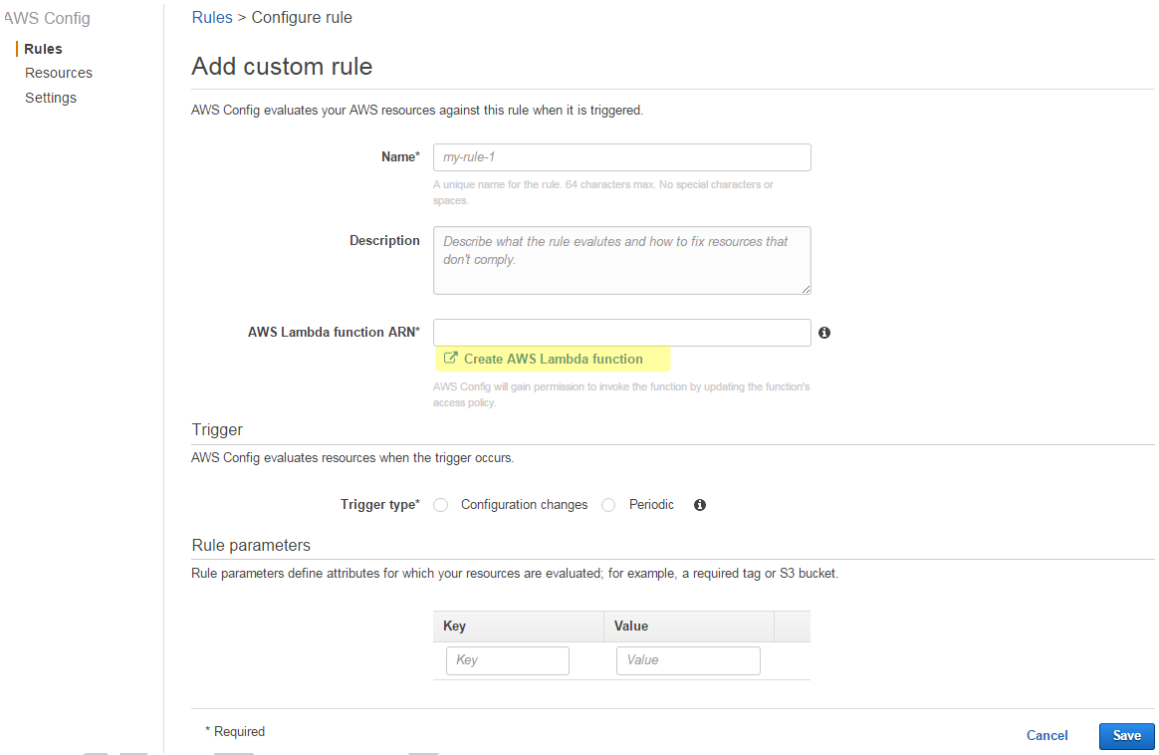


**Figure 6: AWS Config Rule Creation Page**

On the Lambda console, select the `config-rule-change-triggered` blueprint to get started.

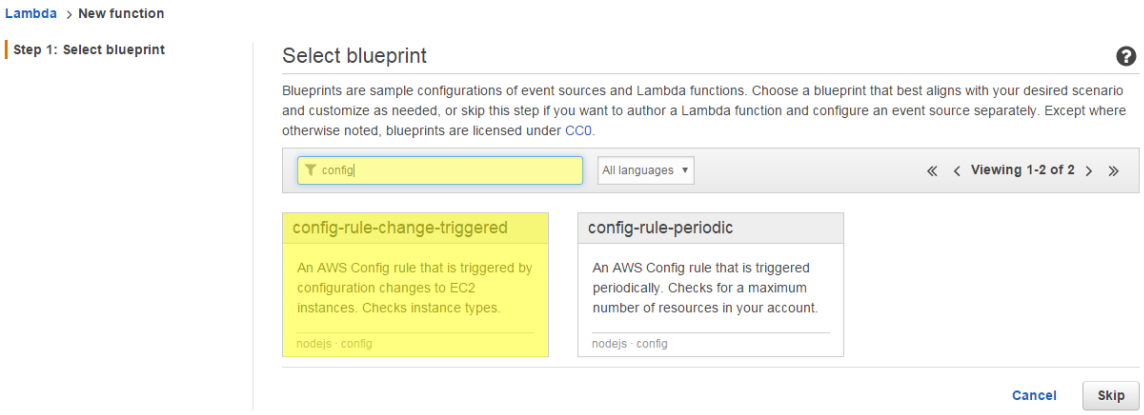


**Figure 7: The Lambda Select Blueprint Page**

You can annotate compliance states. To do this, first add a global variable called `annotation`.

```
var aws = require('aws-sdk');
var config = new aws.ConfigService();
var annotation;
```

You also need to modify the `evaluateCompliance` function and the handler invoked by AWS Lambda. The rest of the blueprint code can be left untouched.

```
function evaluateCompliance(configurationItem, ruleParameters, context) {
  checkDefined(configurationItem, "configurationItem");
  checkDefined(configurationItem.configuration,
"configurationItem.configuration");
  checkDefined(ruleParameters, "ruleParameters");

  if ('AWS::EC2::Instance' !== configurationItem.resourceType) {
    return 'NOT_APPLICABLE';
  }

  if (ruleParameters.imageId === configurationItem.configuration.imageId
```

```
&& ruleParameters.hostId !==
configurationItem.configuration.placement.hostId) {
    annotation = "Instance " + configurationItem.configuration.instanceId
                    + " launched from BYOL  AMI " +
    configurationItem.configuration.imageId
                    + " has not been placed on dedicated host " +
    ruleParameters.hostId;

     return 'NON_COMPLIANT';
  }
  else {
     return 'COMPLIANT';
  }
```

For this example function, `imageId` and `hostId` are parameters that are passed to the function by the AWS Config rule that will be created next. The `imageId` parameter will contain the AMI ID of MyWindowsImage. Use this to identify instances that are launched from this image. After you detect that an instance was launched from MyWindowsImage, you then can check whether the instance was launched onto the specified Dedicated Host identified by the `hostId` parameter. The instance is marked noncompliant if it is found to be not running on the host on which all instances launched from MyWindowsImage should be running.

You can annotate compliance states of a resource with additional information indicating why the resource was marked noncompliant. This sample elaborates the details of why the instance was marked noncompliant and assigns this text to the `annotation` global variable. Finally, changes are made to the handler to pass on the annotation along with the rest of the compliance information.

```
putEvaluationsRequest.Evaluations = [
  {
    ComplianceResourceType: configurationItem.resourceType,
    ComplianceResourceId: configurationItem.resourceId,
    ComplianceType: compliance,
    OrderingTimestamp: configurationItem.configurationItemCaptureTime,
    Annotation: annotation
  }
];
```

After changes are made to the AWS Lambda function, select the appropriate role and save the function. In our example, we also noted the Amazon Resource Name (ARN) of the function. After the function is created, go back to the AWS Config console and enter the ARN of the function that was just created.

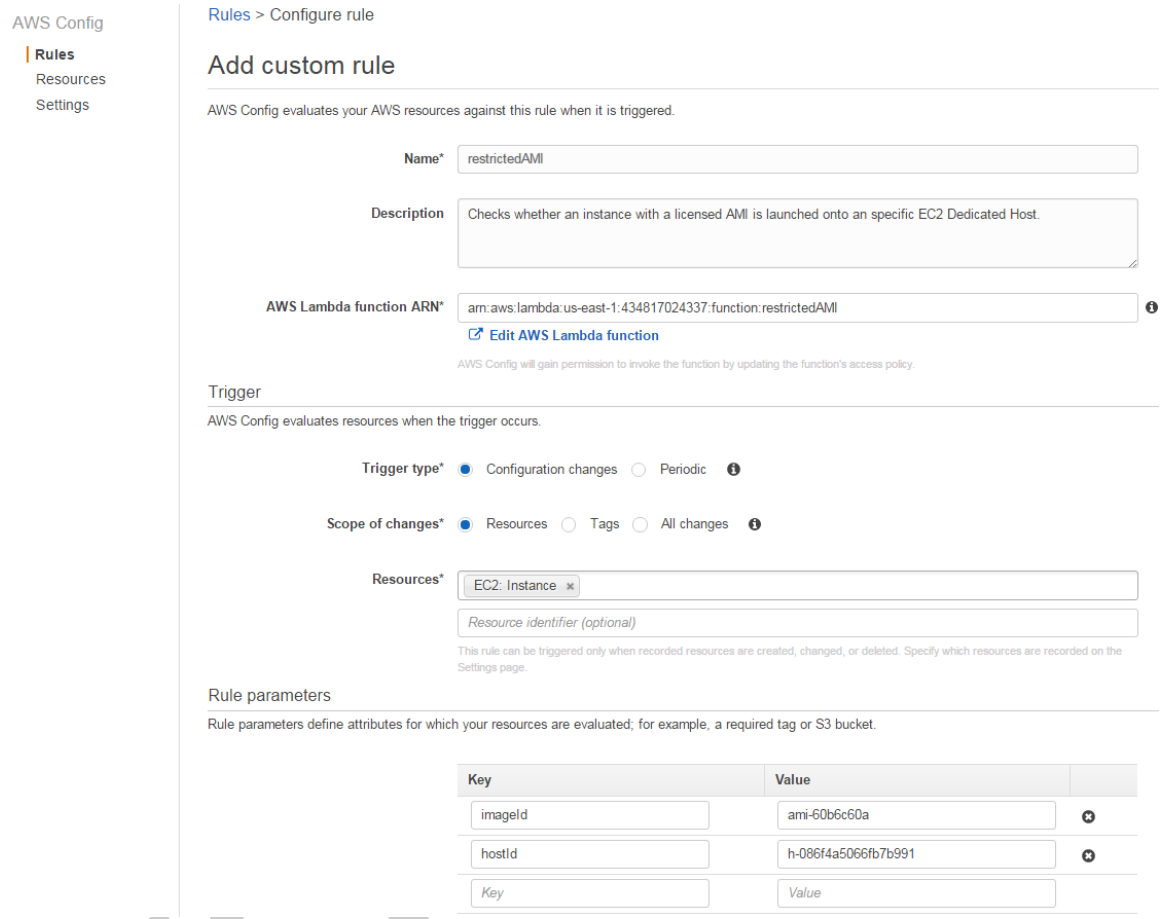


**Figure 8: Entering the AWS Lambda Function ARN on the AWS Config Rule Creation Page**

After specifying the appropriate settings for the rule, save it. The rule is evaluated once immediately after it is created and thereafter for any changes that are made to EC2 instances. In this example, two instances were launched from MyWindowsImage, out of which only one was launched onto the specified Dedicated Host. The AWS Config rule marks the other instance noncompliant.

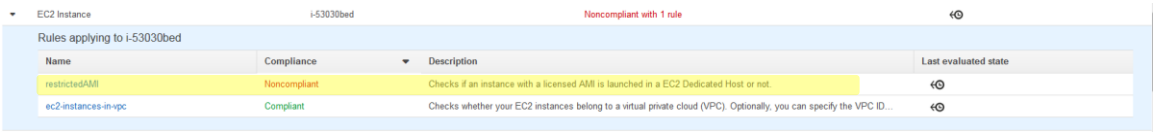


**Figure 9: Instance Marked as Noncompliant**

The **Compliant** or **Noncompliant** state for each rule is also sent as a notification via the Amazon SNS topic you created when you set up AWS Config. You can configure these notifications to send an email, trigger a corrective action, or log a ticket. The Amazon SNS notification contains details about the change in compliance state, including the annotation that elaborates the reason for noncompliance.


View the Timeline for this Resource in AWS Config Management Console:
https://console.aws.amazon.com/config/home?region=us-east-1#/timeline/AWS::EC2::Instance/i-a46d7125?time=2016-01-28T02:02:35.606Z

New Compliance Change Record:
---
```
{
 "awsAccountId": "434817024337",
 "configRuleName": "restrictedAMI",
 "configRuleARN": "arn:aws:config:us-east-1:434817024337:config-rule/config-rule-hz8yxz",
 "resourceType": "AWS::EC2::Instance",
 "resourceId": "i-a46d7125",
 "awsRegion": "us-east-1",
 "newEvaluationResult": {
  "evaluationResultIdentifier": {
   "evaluationResultQualifier": {
    "configRuleName": "restrictedAMI",
    "resourceType": "AWS::EC2::Instance",
    "resourceId": "i-a46d7125"
   },
   "orderingTimestamp": "2016-01-28T02:02:35.606Z"
  },
  "complianceType": "NON_COMPLIANT",
  "resultRecordedTime": "2016-01-28T02:02:41.417Z",
  "configRuleInvokedTime": "2016-01-28T02:02:40.396Z",
  "annotation": "Instance i-a46d7125 launched from BYOL AMI ami-60b6c60a has not been placed on dedicated host h-086f4a5066fb7b991",
  "resultToken": null
 },
 "oldEvaluationResult": {
  "evaluationResultIdentifier": {
   "evaluationResultQualifier": {
    "configRuleName": "restrictedAMI",
    "resourceType": "AWS::EC2::Instance",
    "resourceId": "i-a46d7125"
   },
   "orderingTimestamp": "2016-01-28T01:44:54.553Z"
```

```
  },
  "complianceType": "COMPLIANT",
  "resultRecordedTime": "2016-01-28T01:45:03.438Z",
  "configRuleInvokedTime": "2016-01-28T01:45:01.298Z",
  "annotation": null,
  "resultToken": null
 },
 "notificationCreationTime": "2016-01-28T02:02:42.317Z",
 "messageType": "ComplianceChangeNotification",
 "recordVersion": "1.0"
}
```


# Addressing Other Bring Your Own License (BYOL) Compliance Requirements with AWS Config Rules

The AWS Config rule created in the example above checks one of the several compliance requirements you may have associated with BYOL server-bound licenses. This rule can be further extended to check other license-specific restrictions such as the following:

- Host affinity of the instances
- Number of sockets or number of cores of the Dedicated Host onto which the instances are launched
- Duration for which an instance needs to be on a specified Dedicated Host

In addition, you can also monitor the utilization of Dedicated Hosts you own and mark them noncompliant if their usage drops below a threshold. This can help you optimize your fleet of Dedicated Hosts.

# Conclusion

In this paper, you learned how you can use AWS Config in conjunction with AWS Config rules to ascertain your license compliance posture on Amazon EC2 Dedicated Hosts. AWS Config can be more broadly used to monitor and govern all your resources. For more information, see Further Reading, below.

# Contributors

The following individuals and organizations contributed to this document:

- Chayan Biswas, Senior Product Manager, AWS Config

# Further Reading

For additional help, please consult the following sources:

- Documentation on what AWS Config supports: Supported Resources, Configuration Items, and Relationships
- Blog post: How to Record and Govern your IAM Resource Configurations Using AWS Config
- AWS Config product page: AWS Config