

Building a **Golden AMI Pipeline with AWS Marketplace, AWS Systems Manager,
Amazon Inspector, AWS Config, and AWS Service Catalog
(Guide)**

March 2018



© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

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.

Table of Contents

Table of Contents	3
Overview	4
Document Overview	5
Pre-requisites	6
Support	7
Step-by-Step Instructions	8
Step 1 (optional): Subscribe to the AWS marketplace product you want to distribute	8
Step 2: Create a cross-account role in the child account	9
Step 3: Set up the golden AMI pipeline environment	10
Step 4 (optional): Set up a compliance check in the child account(s)	13
Step 5: Create a golden AMI	14
Step 6: Approve the golden AMI	17
Step 7: Review the golden AMI metadata	19
Step 8 (optional): Manually trigger continuous vulnerability assessment of golden AMIs	19
Step 9: Distribute the golden AMI to child account	22
Step 10 (optional): Distribute the AWS Service Catalog product to the child account	24
Step 11 (optional): Deploy the golden AMI using Service Catalog Portfolio	27
Step 12: Decommission the golden AMI	28
Conclusion	30
References	31

Overview

As your organization moves more and more of your workloads to Amazon Web Services (AWS), your IT Team needs to ensure that they can meet the security requirements defined by your internal Information Security team. The [Amazon Machine Images](#) (AMIs) used by different customer business units must be hardened, patched, and scanned for vulnerabilities regularly. Like most companies, your organization is probably looking for ways to reduce the time required to provide approved AMIs.

Often evidence of compliance and approval is required before you can use AMIs in your production environments. It can be difficult for your development teams to determine which AMIs are approved, and how to integrate AMIs into their own applications. Organization-wide cloud teams need to ensure compliance and enforce that development teams use the hardened AMIs and not just any off-the-shelf AMI. It is not uncommon for an organization to build fragile, internal toolchains. Those are often dependent on one or two skilled people whose departure introduces risk.

This guide presents a golden AMI pipeline to address the challenges faced by customer cloud teams. It describes a method for providing a repeatable, scalable, and approved application stack factory that increases innovation velocity, reduces effort, and increases the chief information security officer's (CISO) confidence that teams are compliant.

In a typical enterprise scenario, a cloud team (often called Cloud Center of Excellence-CCOE team) is responsible for providing the core infrastructure services. This team owns providing the appropriate AWS environment for the many development teams and approved AMIs that include the latest OS updates, hardening requirements, and required third-party software agents. They need to provide these approved images to teams across the organization in a seamless way. In a more decentralized model, organizations typically use this same method.

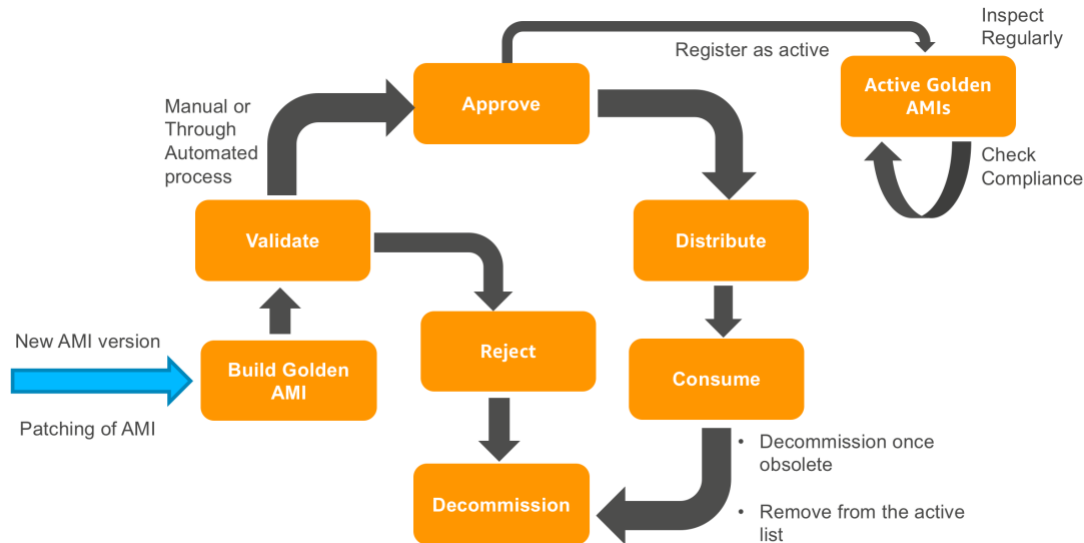
As a member of the CCOE team of your organization, you would:

1. Create an approved golden AMI that meets your InfoSec requirements.
2. Distribute the golden AMI to development teams deploying applications in one or more child AWS accounts.
3. Patch the golden AMI.
4. Set up a continuous health check of active golden AMIs.
5. Decommission an obsolete golden AMI.

Once you have shared the base golden AMI with development teams, they would want to consume the latest golden AMI in the simplest way possible, often through automation. They would want to customize the OS specific golden AMIs with the required software components, but also ensure that the AMIs continue to meet your organization's InfoSec requirements.

The development teams can repeat the above process. Each team can use the golden OS AMI shared by the CCOE team and can add their own software and produce a new golden AMI that is scanned, distributed, and consumed as necessary.

Here the workflow diagram of the golden AMI pipeline.



Document Overview

This document is a step-by-step guide to set up a pipeline that addresses creation, distribution, continuous health assessment, and decommissioning of a golden AMI. You can follow this guide to set up a multi-account-based golden AMI pipeline. Whenever you need to build a new build of a golden AMI, you would trigger the golden AMI automation document (process described in **Step 5**) which would create a golden AMI. You would then validate the golden AMI and approve or deny the golden AMI (process described in **Step 6**). Once you approve, you would distribute the golden AMI to one or more child accounts (process described in **Step 9**). If you have governance and compliance requirements around the consumption of the golden AMI, you can distribute the golden AMI through AWS Service Catalog (process described in **Step 10**). You may also want to perform a continuous vulnerability assessment on all your active golden AMIs (process described in **Step 8**). Finally, you would decommission the golden AMI once it becomes obsolete (process described in **Step 12**). As part of this step-by-step guide, you would set up the golden AMI pipeline and then test it.

To set up the golden AMI infrastructure:

- You would create a cross-account role in the child AWS account(s) that golden AMI pipeline will assume for creating metadata in the child account (process described in **Step 2**).
- You would set up the golden AMI pipeline in the master AWS account in a specific region (process described in **Step 3**). This need not be the region from which child account(s) would launch an EC2 instance of the golden AMI.
- To ensure that non-golden AMI launches are flagged as non-compliance, you will also set up an AWS Config rule that will flag any EC2 instances that are not launched from the active golden AMI list (process described in **Step 4**).
- To distribute the golden AMI in a governed manner, you would set up portfolios using AWS Service Catalog (process described in **Step 10**).
- If you need to patch an AMI, you can simply decommission the AMI that needs patching and generate a new one. This guide assumes that golden AMIs are immutable.

To test different aspects of the golden AMI pipeline:

- You can create golden AMI and then distribute the same to a child account.
- You can manually perform a continuous vulnerability assessment of your active golden AMI.
- You can deploy an instance of a golden AMI in governed manner.
- Finally, you can decommission the golden AMI.

This guide describes a step-by-step process for standardizing the AWS Marketplace-based **Deep Learning AMI (Amazon Linux) AMI**, however, you can use it to standardize your own private Linux-based EBS-backed AMI.

Pre-requisites

Before you begin, ensure that you have following details.

1. Master account details:

- Sign-in link.
- Account-ID.
- IAM user credentials (must have Administrator access and **AmazonSSMAutomationApproverAccess**).
- The region in which the pipeline would be set up.

Note

If you are using consolidated billing through AWS Organizations, this is not the master-payee account. It is a best practice not to launch any resources in the master-payee account. The master account referred to in this guide is an account that you, as a CCOE team, has identified as the central AWS account which you will use for standardizing the products before you distribute them to different accounts in your organization.

2. Child account details:

- Sign-in link.
- Account-ID.
- The region in which application team will deploy the golden AMI.
- Admin IAM user credentials (must have the **AWSServiceCatalogAdminFullAccess** managed policy associated with it).
- End-user IAM credentials (must have the **AWSServiceCatalogEndUserFullAccess** managed policy associated with it).

3. Cost-Center number - You can use any four-digit number if you are setting this up in your personal account.

4. VPC Information

- CIDR – a valid CIDR available in the master account’s region for creation of a new VPC.
- cidrPrivateSubnet – Must be a subset of VPC CIDR and should have the capacity to create instances of all golden AMIs (and more if you are going to simultaneously create multiple versions of a single product (or multiple products)).
- cidrPublicSubnet – CIDR to create a public subnet in VPC. This must be a subset of VPC CIDR.

Note

While choosing regions in the master account as well as the child account, ensure that all services used by the solution support the region you choose. Check the list of [supported regions](#). This guide has been tested Amazon

Linux AMI-based golden AMIs with the master region as us-east-1/us-east-2/us-west-2 and child region as us-east-1/us-east-2/us-west-1/us-west-2.

Disclaimer

This work extends architectures described in the following content.

- [Building a Secure, Approved AMI Factory Process Using Amazon EC2 Systems Manager \(SSM\), AWS Marketplace, and AWS Service Catalog](#)
- [How to Set Up Continuous Golden AMI Vulnerability Assessments with Amazon Inspector](#)

This guide assumes that the reader is familiar with following AWS services.

- [AWS Systems Manager](#)
- [Amazon EC2](#)
- [Amazon Inspector](#)
- [AWS Config](#)
- [AWS Marketplace](#)
- [AWS Service Catalog](#)

The golden AMI pipeline creates AWS resources not supported in the free tier limit and will incur charges.

Support

The pipeline does not support following:

- Encrypted AMI distribution.
- Instance-Store based AMIs.
- Windows-based golden AMIs.
- EC2-classic or hybrid VPC environments (containing EC2-VPC as well as the EC2-classic environment).
- Source AMIs with volumes which do not have **DeleteOnTermination** flag set.
- Re-distribution of a golden AMI.

Only Linux flavors supported by Systems Manager and Amazon Inspector service are supported by the golden AMI pipeline. Also, it only supports AMI based AWS Marketplace products.

Step-by-Step Instructions

Step 1 (optional): Subscribe to the AWS marketplace product you want to distribute

To successfully create a golden AMI of an [AWS Marketplace](#) AMI, your AWS account must be subscribed to the product. For this exercise, your procurement team can subscribe to the **Deep Learning AMI (Amazon Linux)** product from AWS Marketplace console in both, the master as well as the child account(s). You can skip this step if you do not want to test the golden AMI pipeline on an AWS Marketplace-based AMI product.

To subscribe to the **Deep Learning AMI (Amazon Linux)** product:

- a) Sign-in to the AWS Management console using the master account credentials and open the [AWS Marketplace listing for the product](#).
- b) Choose **Continue to Subscribe**.
- c) Select **Manual launch tab** and choose **Accept Software Terms**. You should not do this without a formal approval from your legal/procurement team. If this button does not appear, it indicates that either the AMI is provided by Amazon or that your account may already have a subscription to this product. Skip to **Step f)**. You can view subscriptions by choosing **Your marketplace Software** from the menu.
- d) A confirmation page would appear.
- e) Choose **Return to launch page**.
- f) Choose **6.0, released 03/26/2018** as the version and Note the **AMI-ID** for the master account's region. You would specify this AMI-ID as the source AMI while generating a golden AMI. Note that Amazon continuously releases new versions of AMIs. If **6.0, released 03/26/2018** is not available, choose the latest available version.

Usage Instructions		
Launch		
AMI IDs		
Region	ID	
US East (N. Virginia)	ami- ami-0262d900	Launch with EC2 Console
US East (Ohio)	ami- ami-0262d900	Launch with EC2 Console
US West (N. California)	ami- ami-0262d900	Launch with EC2 Console
US West (Oregon)	ami- ami-0262d900	Launch with EC2 Console
Canada (Central)	ami- ami-0262d900	Launch with EC2 Console
EU (Frankfurt)	ami- ami-0262d900	Launch with EC2 Console
EU (Ireland)	ami- ami-0262d900	Launch with EC2 Console
EU (London)	ami- ami-0262d900	Launch with EC2 Console

Next, you need to subscribe to the marketplace product from the child account. To subscribe to the product from the child account:

1. Sign-in to the AWS Management console using the child account credentials and open the [AWS Marketplace listing for the product](#).

2. Next, perform **steps b) to f)**. You need not note the AMI-ID of the child account/region.

As part of golden AMI creation process, automation would launch an instance of the AMI in the master account and once golden AMI has been distributed to the child account, you would launch the software from the child account.

Step 2: Create a cross-account role in the child account

In this step, you will sign-in to the child account and execute a [CloudFormation template](#). The CloudFormation template will set up a cross-account role. To know more about the cross-account role, see documentation on [Tutorial: Delegate Access Across AWS Accounts Using IAM Roles](#). The golden AMI pipeline will assume this cross-account role to create golden AMI metadata in the child account. This guide assumes that you are distributing to only one account, however, if you have multiple child accounts, you would need to do this step in each child account.

To create a cross-account role in the child account:

1. Open the following link, choose **Raw** and then download the JSON file to your computer:
<https://github.com/aws-samples/aws-golden-ami-pipeline-sample/blob/V-1.0/Golden-AMI-Cross-Account-Role.json>
2. Sign-in to the AWS Management console using child account's credentials and choose **CloudFormation** in the **Services** menu.
3. Ensure that you are in the correct region.
4. Choose **Create Stack**.
5. On the Select Template page, choose **Upload a template to Amazon S3**.
6. Choose **Choose File** and then choose the CloudFormation template you downloaded.
7. Choose **Next**.
8. On the Specify Details page specify following:
 - a. **Stack Name** as **Golden-AMI-Cross-AccountRole-Cost-Center**
 - b. **roleName** as **goldenAMICrossAccountRole-Cost-Center**
 - c. **parentAWSAccountID** as **Master-Account-ID**
9. Choose **Next**.
10. On the Options page, specify following key-value pairs as **Tags**:
 - a. Key as **Cost-Center** and corresponding value as **Cost-Center** provided to you.
 - b. Key as **Generated-By** and corresponding value as **Golden-AMI-Pipeline**.
11. Choose **Next**.
12. On the Review page, choose the **check-box** next to the following message:

"I acknowledge that AWS CloudFormation might create IAM resources with custom names."
13. Choose **Create**. The CloudFormation template creates an AWS Identity and Access Management (IAM) cross-account role.

Note

Ensure that the golden AMI cross-account role name (**roleName**) is identical in all child accounts as the pipeline does not let you specify account specific role names.

Step 3: Set up the golden AMI pipeline environment

To set up the golden AMI pipeline infrastructure in the master account:

1. Open the following link, choose **Raw** and then download the JSON file to your computer:
<https://github.com/aws-samples/aws-golden-ami-pipeline-sample/blob/V-1.0/Gold-AMi-Stack-CFT-Cl.json>
2. Sign-in to the AWS Management console using the master account's credentials and choose **CloudFormation** in the **Services** menu.
3. Ensure that you are in the correct region.
4. Choose **Create Stack**.
5. On the Select Template page, choose **Upload a template to Amazon S3**.
6. Choose **Choose File** and then choose the CloudFormation template you downloaded.
7. Choose **Next**.
8. On the Specify Details page, specify a Stack name and following parameters (values are case-sensitive):

Parameter	value	Description
ApproverUserIAMARN	arn:aws:iam:: Master-Account-ID :user/ Master_User_Name	This is the IAM ARN of the approver who can view Amazon Inspector findings and has AmazonSSMAutomationApproverAccess managed policy associated with it. Approver approves/denies the golden AMI.
cidrVPC cidrPrivateSubnet cidrPublicSubnet	Appropriate CIDRs	The golden AMI pipeline would create a VPC in which it would launch necessary instances to generate a golden AMI. The VPC would have a public subnet and a private subnet.
continuousInspection Frequency	rate(1 day)	This is the frequency at which vulnerability assessment of your active AMIs would take place. To know more valid values for this parameter, see documentation on Schedule Expressions Using Rate or Cron . If you do not want to perform a continuous assessment of your active golden AMIs, you can specify this as a large number and disable the rule once you have deployed the CloudFormation stack.
instanceType	Appropriate Instance-Type	This is the InstanceType that is compatible with all your golden AMIs. The golden AMI pipeline will use this InstanceType for launching an instance of each active golden AMI during continuous vulnerability assessment.

EmailID	Administrator's Email ID	This is the Email ID of the administrator responsible for approving the golden AMI and validating the continuous assessment results. The pipeline would subscribe this email ID to receive continuous assessment result as well as AMI-pending-for-approval notifications. Once CloudFormation stack is created, two SNS topic subscription confirmation emails will be sent to this email-ID.
roleName	goldenAMICrossAccountRole- Cost-Center	This is a cross-account role for managing golden AMI metadata parameters in the child account(s). This is the roleName that you specified in Step 2 . The golden AMI pipeline will expect each child account to have a cross-account role with the exact same name configured using Step 2 .
MetadataJSON		<p>This is the default metadata for distributing the golden AMI. You will be able to override this when you are distributing a golden AMI build. It allows you to specify multiple accounts and multiple regions for each account. Please find the sample format below. If you do not wish to distribute the golden AMI to another account but another region in your account, then you can specify the Master account ID.</p> <p>Sample Format</p> <pre>{\"Destination-Account-1\": \"us-west-1,us-west-2\", \"Destination-Account-2\": \"us-east-1,us-east-2,us-west-2\"}</pre>
productName	DeepLearningAMI-5.0	This is the name of the golden AMI product along with major/minor version number. The syntax of this parameter is ProductName-ProductVersion . If you are creating golden AMI of the product you subscribed to in Step 1 , you can specify this as DeepLearningAMI-5.0. This is a default value, you can override this later when you create a golden AMI.
productOSAndVersion	AmazonLinux-2017.09	The syntax of this parameter is OSName-OSVersion . This is the default OS and version number of the OS. You can override this later when you trigger the golden AMI creation workflow. The parameter is associated as the metadata with the golden AMI and not used for any OS specific operations.
buildVersion	1	This is a default version that corresponds to your product. You will override this value when you trigger golden AMI creation/distribution/ decommissioning automation workflow.

9. Choose **Next**.

10. On the Options page, specify following key-value pairs as Tags:

- i. Key as **Cost-Center** and corresponding value as **Cost-Center** provided to you.
 - ii. Key as **Generated-By** and corresponding value as **Golden-AMI-Pipeline**.
11. Choose **Next**.
 12. On the Review page, choose the **check-box** next to the following message:

“I acknowledge that AWS CloudFormation might create IAM resources.”
 13. Choose **Create**.
 14. After CloudFormation creates the stack, choose the **check-box** next to your stack and then choose **Outputs** tab.

You will see following parameters in the output. Note the value of the parameters.

Parameter	Description of the value
GoldenAMIAutomationDoc	The name of the SSM automation document you can execute for generating a golden AMI.
CopyAndShareAMIAutomationDoc	The name of the SSM automation document you can execute for distributing the golden AMI.
DecommissionAMIVersionDoc	The name of the SSM automation document you can execute for decommissioning a golden AMI.
ContinuousInspectionScheduledRule	The CloudWatch Events rule that executes continuous vulnerability assessment at the frequency you specified.
BucketName	The name of the Amazon S3 bucket in which you need to upload CloudFormation Template for launching your golden AMI.
ContinuousAssessmentResultsTopic	The SNS topic on which continuous assessment results will be published.
SetupContinuousAssessmentLambdaFunction	This is the AWS Lambda function you need to execute if you want to manually trigger the continuous vulnerability assessment for all active golden AMIs.

Open administrator mailbox and confirm the subscription requests. Administrator will receive two subscription requests. One will enable you to receive golden AMI creation completion notification and the another to receive continuous vulnerability assessment results.

Next, you need to upload the CloudFormation Template that users can use to launch an instance of the golden AMI to an **Amazon S3** bucket specified in the **BucketName** field of the CloudFormation output of this step. Here is a sample CloudFormation template you can use to launch an EC2 instance of a golden AMI.

<https://github.com/aws-samples/aws-golden-ami-pipeline-sample/blob/V-1.0/simpleEC2-SSMParamInput.json>

Alternately, you can use your own custom JSON based CloudFormation template that does more than simply create an instance of the golden AMI. However, ensure that the CloudFormation template you upload has an AMI-ID parameter with **AMI_ID_TO_REPLACE** as the **default value** and **type** as

AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>. The distribution process will replace the placeholder with the golden AMI-ID's SSM parameter path and release it as a product version through AWS Service Catalog.

Step 4 (optional): Set up a compliance check in the child account(s)

You need to set up [AWS Config](#) in child account's region if it is not set up. To know more about how to set up AWS Config, see documentation on [Setting up AWS Config with the Console](#). While setting up the AWS Config for this guide, you do not need to select any of the existing rules.

Next, you will run a CloudFormation template to create an [AWS Config rule](#) to flag any EC2 instances not created from the golden AMIs as non-compliant.

Note

If you are distributing your golden AMI to multiple accounts/regions, you will need to run the CloudFormation Template in each child account in each region in which an EC2 instance of a golden AMI will be launched. [AWS CloudFormation Stacksets](#) provide an elegant way to execute a single CloudFormation template in multiple accounts in multiple regions simultaneously. To know more about CloudFormation Stacksets, see documentation on [Working with AWS CloudFormation StackSets](#). Before you execute the above CloudFormation Template using Stacksets, ensure that you have performed [stacksets account setup](#) in parent as well as all child accounts.

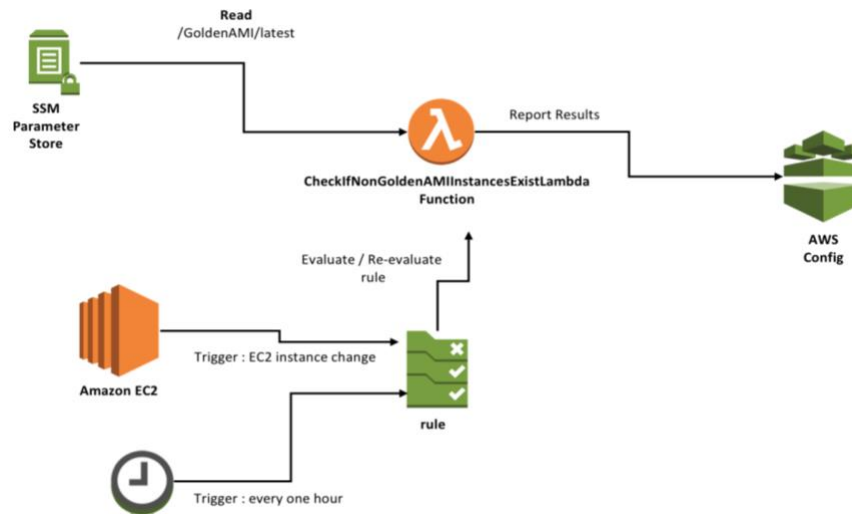
This guide assumes that you will distribute the golden AMI to only one account and hence run the CloudFormation template only once. To set up the rule:

1. Open the following link, choose **Raw** and then download the JSON file to your computer.
<https://github.com/aws-samples/aws-golden-ami-pipeline-sample/blob/V-1.0/Golden-AMI-Compliance-CFT.json>
2. Sign-in to AWS Management console using child account's credentials and choose **CloudFormation** in the **Services** menu.
3. Ensure that you are in the correct region (the region in which end-user will deploy an EC2 instance of the golden AMI).
4. Choose **Create Stack**.
5. On the Select Template page, choose **Upload a template to Amazon S3**.
6. Choose **Choose File** and then choose the CloudFormation template you downloaded.
7. Choose **Next**.
8. On the Specify Details page, specify:
 - a. **Stack Name** as **Compliance-Cost-Center**.
 - b. Specify **PathToSSMParameter** as **/GoldenAMI/latest**. This is the SSM parameter path on which comma-separated list of active golden AMIs will become available.
9. Choose **Next**.

10. On the Options page, specify following key-value pairs as **Tags**:
 - a. Key as **Cost-Center** and corresponding value as **Cost-Center** provided to you.
 - b. Key as **Generated-By** and corresponding value as **Golden-AMI-Pipeline**.
11. Choose **Next**.
12. On the Review page, choose the **check-box** next to the following message:

“I acknowledge that AWS CloudFormation might create IAM resources.”
13. Choose **Create**.

Here is an architecture diagram of the AWS Config rule.



The rule triggers an AWS Lambda function which reads the list of active AMIs from a parameter in the SSM parameter store and marks any instance that has an AMI-ID that does not match with any of the active golden AMIs, as non-compliant. The AWS Config rule gets evaluated on following occasions:

1. Whenever there is any configuration change detected in any EC2 instance in that region.
2. Every one hour. This is necessary to update compliance results once the list of active golden AMIs has changed (due to creation/decommissioning of a golden AMI)

Step 5: Create a golden AMI

To create a golden AMI:

1. Sign-in to the AWS Management Console using master AWS account credentials and then navigate to **Systems Manager** service.
2. Ensure that you are in the correct region.
3. In the navigation panel, choose **Automation** under **Actions** drop-down.
4. Choose **Execute Automation**.
5. Filter automations visible by choosing **owned by me** filter option.

6. Choose the **GoldenAMIAutomationDoc** document name that you noted in the output tab of CloudFormation stack, in **Step 3**.
7. Choose following values:
 - a. **Document version** as the **latest version at runtime**.
 - b. Leave **execution mode** as it is.
 - c. Most input parameters will be prepopulated except the **sourceAMId** and **AMIVersion**.
 - d. Specify appropriate values for following parameters (values are case-sensitive):

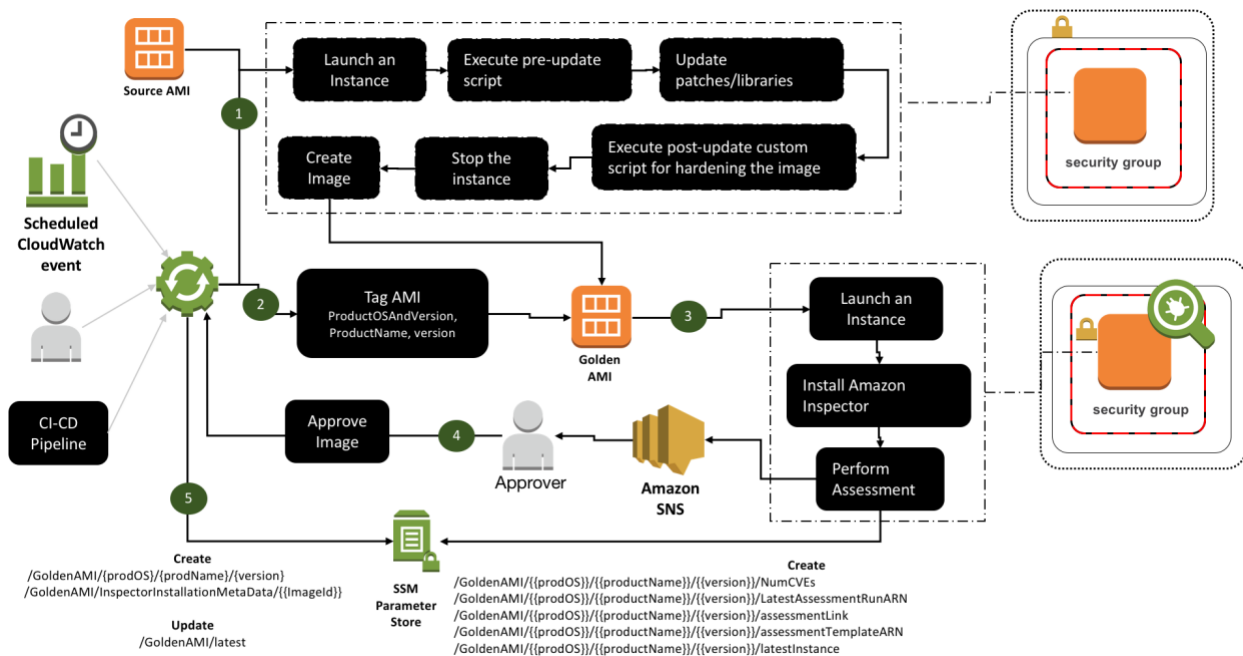
Parameter	Description of the value
sourceAMId	<p>Specify the AMI-ID you noted in Step 1. If you are not distributing the Marketplace product, specify the AMI-ID of the AMI you want to use as an input for the golden AMI creation process and is available in the pipeline's region. This can be an AMI owned by you in which you have already installed all your agents or an OS AMI provided by AWS/CCOE team. If later, you can deploy code/agents installation script through PostUpdateScript and PreUpdateScript parameters. Ensure that the AMI exists in the region in which the golden AMI pipeline has been set up.</p> <p>Note The pipeline currently does not have support for source AMIs with volumes which do not have DeleteOnTermination flag set. If you must use such an AMI as an input, you will need to manually decommission EBS volumes not decommissioned by the pipeline, yourself.</p>
productName	Name of the product for which you are generating the golden AMI. The syntax of this parameter is productName-productversion
productOSAndVersion	The OS details of the product. The syntax of this parameter is OSName-OS-version .
AMIVersion	The version number of the golden AMI to be created. This maps to the build version that you deploy. Typically, you will increment this number every time you create a new version for an existing product.
PostUpdateScript PreUpdateScript	<p>You can specify custom scripts you wish to execute on the AMI using these options. You can specify the AMI hardening script that meets your organization standards using one of the two. This script executes before/after package updates have been applied. The script must be accessible and compatible with the OS of the underlying AMI.</p> <p>Note This guide does not share any specific hardening script as it needs to be specific to your organization based on your organization's security guideline/requirements.</p>
IncludePackages Excludepackages	You can specify packages you want to include or exclude during the package update process.

ApproverUserIAMARN	Review the ARN of the user who will approve/deny the golden AMI. Ensure that the ARN has the AmazonSSMAutomationApproverAccess managed policy associated with it.
instanceType	Review whether the instance-type is compatible with the source AMI.

8. You can leave remaining parameters as it is. Automation document launches instances in a private subnet, with a security group that has no inbound access for launching instances.
9. Next, choose **Execute Automation**.

You have successfully triggered creation of a golden AMI for a product. The output product can be a standardized OS that you want to distribute to your teams or it can be an application specific AMI that has agents, code, and necessary tools built in. Every time you create a new version of the golden AMI, ensure that you have specified the consistent product-name as well as the operating system.

Here is the Architecture diagram for golden AMI creation process.



You can trigger the golden AMI creation process in multiple ways. The process may take approximately 30 minutes to 1.5 hours depending upon the available number of updates and scripts executed on the AMI. You can either schedule it to run periodically OR trigger it manually OR trigger it at the end of your CI-CD pipeline.

Here is how the process works:

- 1) The process takes a source AMI, creates an instance, updates packages, and executes your custom code for hardening the instance and installing required agents.
- 2) Next, it stops the instance and creates an image from the stopped instance.
- 3) Since the security posture of an AMI can be assessed based on the security posture of an instance, automation creates an EC2 instance from the newly created golden AMI.
- 4) It then installs an Amazon Inspector agent on the EC2 instance and triggers an Amazon Inspector assessment. Once the assessment is complete, Amazon Inspector records findings and publishes an SNS notification.
- 5) Since your email-ID was subscribed to the SNS topic, you will receive a notification email containing information like shown below.

Please check contents of SSM Parameter : /GoldenAMI/~~XXXXXXXXXX~~/5/NumCVEs , and confirm if the build is approved or not.

-- Approval Details --

Approval Step Name: approve

Region: ap-southeast-2

Automation Execution Id: ~~XXXXXXXXXX~~

Approval Expires At: 2018-03-21 01:06 AM UTC

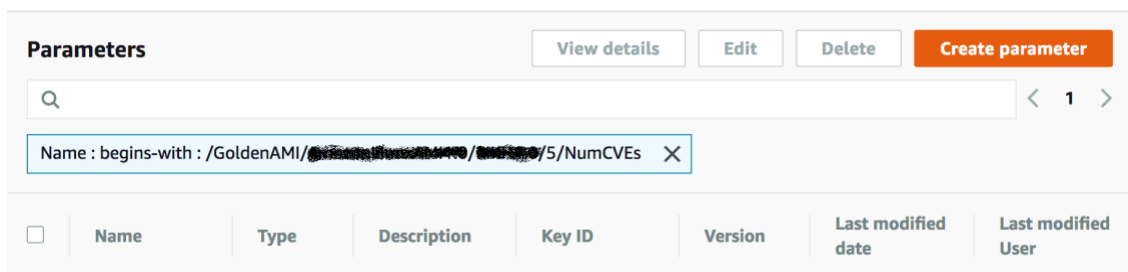
Once the AMI is available for approval, the administrator can approve the AMI and mark it as an active golden AMI.

Step 6: Approve the golden AMI

Next, you need to review the golden AMI assessment result and decide whether to approve or deny the AMI for distribution.

To review:

1. Sign-in to the master AWS account using approver's credentials and navigate to **Systems Manager** service.
2. Ensure that you are in the correct region.
3. In the navigation panel, choose **Parameter Store** under **Shared resources** drop-down.
4. Filter parameters by using a **Name : begins-with** filter and specify the path specified in the notification.



The screenshot shows the AWS Systems Manager Parameter Store console. At the top, there are buttons for 'View details', 'Edit', 'Delete', and 'Create parameter'. Below these is a search bar with a magnifying glass icon. A filter is applied: 'Name : begins-with : /GoldenAMI/~~XXXXXXXXXX~~/5/NumCVEs'. Below the search bar is a table with columns: Name, Type, Description, Key ID, Version, Last modified date, and Last modified User. The table is currently empty.

5. Choose the result.
6. Review the **Value**. The following **value** suggests that there were security findings found in the AMI.

Overview
History
Tags

Name
/GoldenAMI/[REDACTED]/[REDACTED]/NumCVEs

Last Modified date
Wed, 21 Mar 2018 21:49:23 GMT

Description
-

Value
Inspector findings found: High[1], Medium[1], Low[0], Info[2]

- To review findings, open the **Inspector** service console.
- The dashboard will display **Recent assessment runs**. Choose the **assessment run** corresponding to your golden AMI.

Amazon Inspector - Assessment Runs

An assessment run is the process of discovering potential security issues through the analysis of your assessment target's behavior against selected rules packages. [Learn more](#).

Filters: {"assessmentRunArns":["arn:aws:inspector:ap-southeast-2:[REDACTED]:target/[REDACTED]/template/[REDACTED]/run/[REDACTED]"]}

Run
Cancel
Delete

Last updated on March 21, 2018 4:59:28 PM (0m ago)

Filter

Viewing 1-1 of 1

	Start time	Status	Template name	Findings	Findings by sev...	Reports
<input type="checkbox"/>	Today at 4:32 PM...	Analysis complete	[REDACTED]	4	High Medium L...	Download report

- You can either choose **Download report** to see details of the finding or you can review the information by choosing the number under the **Findings** column. Based on what you see, you can choose to **Approve** or **Deny** the AMI.

To approve or deny the golden AMI:

- Sign-in to the master AWS account using credentials with approver permissions and navigate to **Systems Manager** service.
- Ensure that you are in the correct region.
- In the navigation panel, choose **Automation** under **Actions** section.
- You will see a list of automations, the automation that is ready for approval will have status as **"Waiting"**.
- Choose the result corresponding to automation execution and then from **Actions** drop-down, choose **Approve/Deny**.
- For this exercise, you can choose **your decision** on Approve/Deny page as **approve**.
- Choose **Submit**.

Note

If you choose to deny the AMI, you will need to trigger the decommissioning of the golden AMI version manually (process described in **Step 12**).

Step 7: Review the golden AMI metadata

If you approved the golden AMI in **Step 6**, you will see a new private golden AMI registered under your AMIs in the Amazon EC2 console. You will also see metadata created for your golden AMI in **Parameter Store**.

To view parameters created for your golden AMI:

1. Sign-in to the master AWS account and navigate to **Systems Manager** service.
2. Ensure that you are in the correct region.
3. In the navigation panel, choose **Parameter Store** under **Shared resources** drop-down.
4. Filter parameters by using a **Name : begins-with** filter. Create a filter using parameters you provided while creating a golden AMI,

Format - **/GoldenAMI/ProductOS/ProductName/Version**

Step 8 (optional): Manually trigger continuous vulnerability assessment of golden AMIs

For this exercise, you can manually trigger continuous vulnerability assessment of all active golden AMIs. However, you do not need to trigger continuous vulnerability assessment process manually as the pipeline has set up a Cloudwatch Events rule that will trigger the assessment routinely. While deploying the golden AMI pipeline creation infrastructure in **Step 3**, you had specified **continuousInspectionFrequency** as **rate(1 day)**. This means that continuous vulnerability assessment would happen every day.

To manually trigger the continuous vulnerability assessment:

1. Sign-in to the master AWS account's AWS Management console and choose **CloudWatch** from the **Services** menu.
2. Choose **Rules** from the navigation panel.
3. In the rules pane, you would see the schedule rule you noted in the output of **Step 3**.
4. Open the result by choosing the rule. You will see the summary screen in which you will find the details of the Lambda function and an input.

The screenshot shows the AWS CloudWatch Rules console. The breadcrumb navigation is 'Rules > golden4-ScheduledRule-...'. There is an 'Actions' button in the top right corner. The 'Summary' section displays the following information:

- ARN**: `arn:aws:events:ap-southeast-2:123456789012:rule/golden4-ScheduledRule-...`
- Schedule**: Fixed rate of 1 days
- Status**: Enabled
- Description**: ScheduledRule
- Monitoring**: [Show metrics for the rule](#)

The 'Targets' section shows a table with one target:

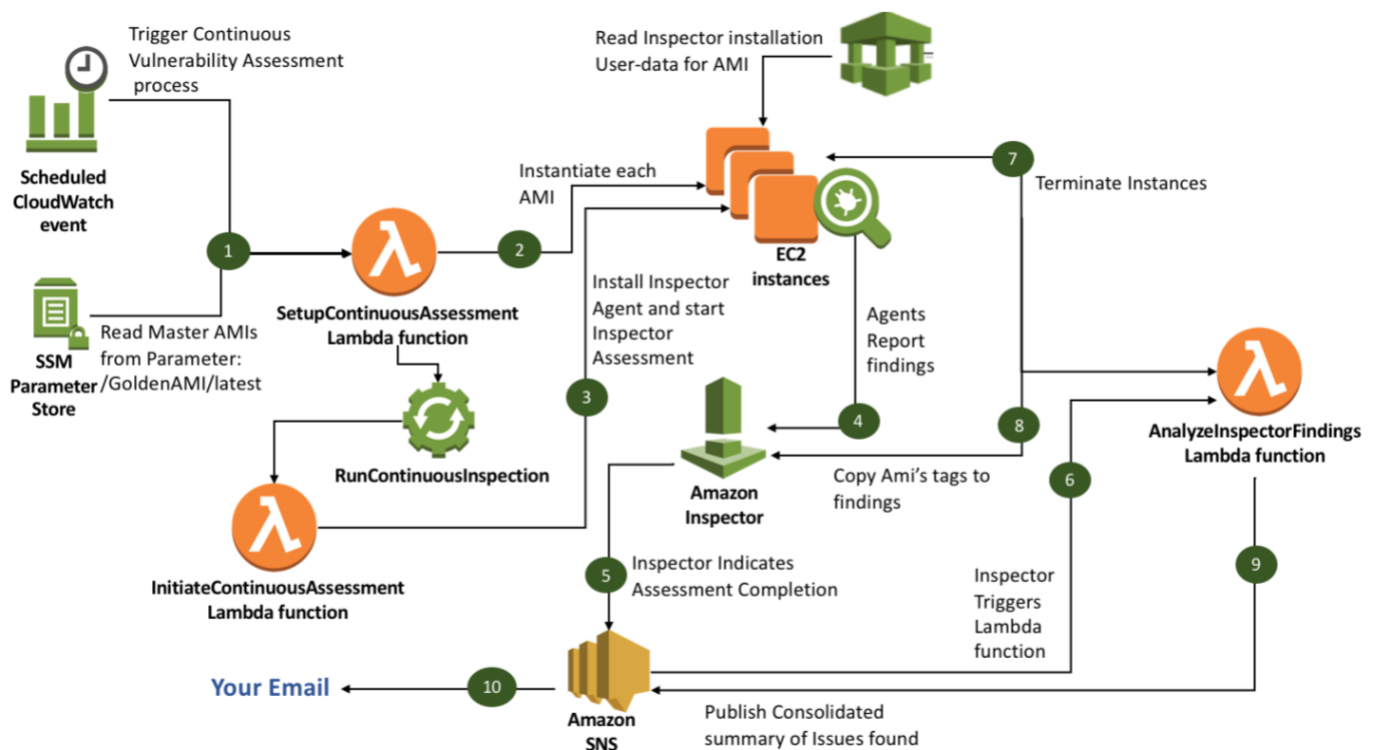
Type	Resource name	Input	Role
Lambda function	golden4-SetupContinuousAssessmentLambdaFunction-...	Constant: {"AMIsParamName":"/GoldenAMI/latest", "instanceType":"t2.large"}	

Note

If you want to change the frequency at which continuous assessment would execute, choose **Actions**, and then **Edit**, and then specify an appropriate schedule. Next, you can choose **configure details** and later, **update rule**.

- Copy the **Input Constant** value present in the **Input column** and choose the value in the resource column to open the lambda function.
- Choose **Select a test event** and then **configure test events** to create a test event.
- Paste the text you copied from Input column earlier in the text area, specify an appropriate event name and then choose **Create**.
- Choose **Test** to execute the lambda function.

Once continuous vulnerability assessment executes (takes approximately ~80 minutes), you will receive a notification. The process is set up in the region in which your golden AMI pipeline infrastructure is set up. Here is the architecture of the continuous vulnerability assessment process.



A scheduled CloudWatch Events event triggers the SetupContinuousAssessment [AWS Lambda](#) function, which starts the security assessment of your golden AMIs. The SetupContinuousAssessment Lambda function reads a parameter with the key as `/GoldenAMI/latest` stored in the [AWS Systems Manager](#) (Systems Manager) Parameter Store. This parameter contains the comma-separated list of active golden AMIs.

For each AMI specified in the JSON parameter, the Lambda function creates an EC2 instance. The Lambda function then copies each golden AMI's [tags](#) to the corresponding EC2 instance. It then triggers an automation that waits for instances to come up and later starts InitiateContinuousAssessment lambda function.

When each instance starts, the lambda function installs the Amazon Inspector agent. The function also adds a tag with the key of **continuous-assessment-instance** and value as **true**. This tag identifies EC2 instances that require regular security assessments.

The Amazon Inspector assessment setup evaluates following rules packages:

1. [Common Vulnerabilities and Exposures \(CVEs\)](#)
2. [Center for Internet Security \(CIS\) Benchmarks](#)
3. [AWS Security Best Practices](#)
4. [Runtime Behavior Analysis](#)

The Amazon Inspector agents installed on each EC2 instance collect behavior and configuration data and pass it to Amazon Inspector. Amazon Inspector analyzes the data and generates [Amazon Inspector findings](#), which are possible security findings you may need to address.

After the Lambda function completes the assessment, Amazon Inspector publishes an assessment-completion notification message to an [Amazon SNS](#) topic called **ContinuousAssessmentCompleteTopic**. SNS uses *topics*, which are communication channels for sending messages and subscribing to notifications.

The notification message triggers the AnalyzeInspectorFindings Lambda function, which performs the following actions:

1. Associates the tags of each EC2 instance with security findings found for that EC2 instance. This enables you to identify the security findings using the app-name tag you specified for your golden AMIs. You can use the information provided in the findings to patch your golden AMIs.
2. Terminates all instances associated with the continuous-assessment-instance=true tag.
3. Aggregates the number of findings found for each EC2 instance by severity and then publishes a consolidated result to an SNS topic called ContinuousAssessmentResultsTopic. Since your email was subscribed to the SNS topic, once the assessment has finished, you will receive an email containing aggregate information about the result.

You can also review the details of assessment findings in **Amazon Inspector** console. To review findings:

- Open AWS Management console of the master account and then open **Inspector** using services menu.
- The dashboard will display **Recent assessment runs**. Choose the assessment run corresponding **ContinuousAssessment** template.

Amazon Inspector - Assessment Runs



An assessment run is the process of discovering potential security issues through the analysis of your assessment target's behavior against selected rules packages. [Learn more.](#)

✖ Filters: {"assessmentRunArns":["arn:aws:inspector:ap-southeast-2:██████████:target/██████████/template/██████████/run/██████████"]}

Run
Cancel
Delete

Last updated on March 21, 2018 4:59:28 PM (0m ago)
 ↺
⬇
⚙

Viewing 1-1 of 1

<input type="checkbox"/>	Start time	Status	Template name	Findings	Findings by sev...	Reports
<input type="checkbox"/>	Today at 4:32 PM...	Analysis complete	██████████	4	High Medium L...	Download report

- You can either choose **Download report** to see details of the finding or you can review the information by choosing the value under **Findings** column. After you analyze security findings, you can patch your golden AMIs by decommissioning them and creating a new version containing fixes for the security findings.

Step 9: Distribute the golden AMI to child account

To distribute the golden AMI build:

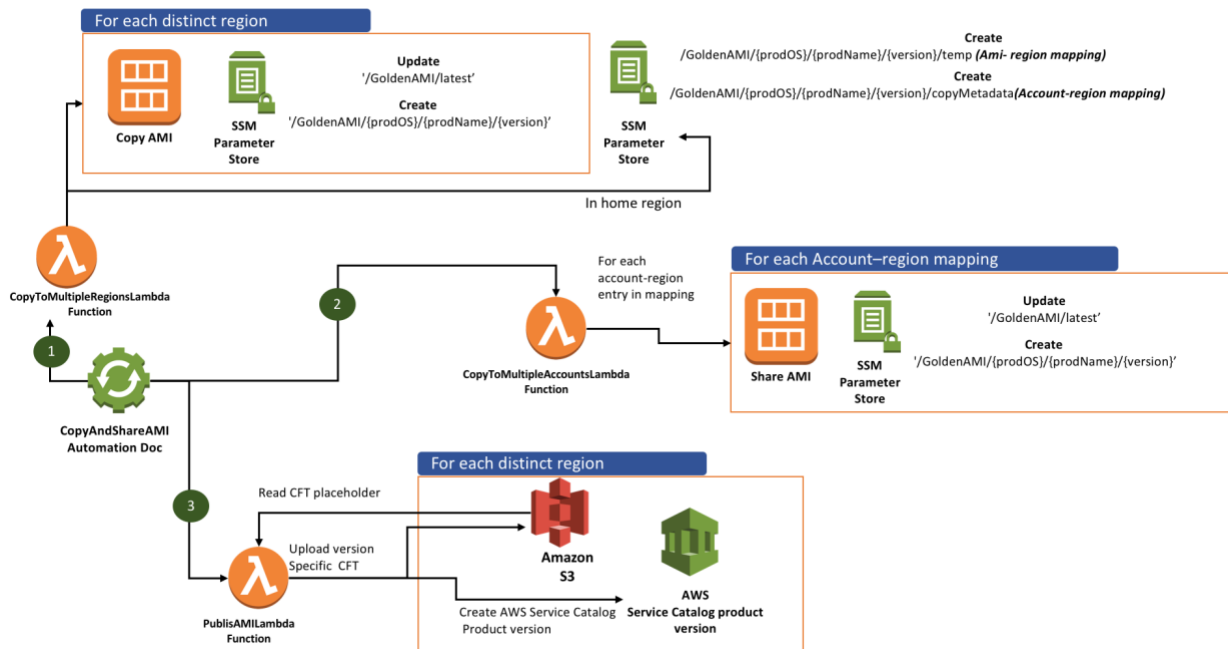
- Sign-in to the master AWS account and navigate to **Systems Manager** service.
- Ensure that you are in the correct region.
- In the navigation panel, choose **Automation** under **Actions** drop-down.
- Choose **Execute Automation**.
- Filter automations visible by choosing **owned by me** filter option.
- Choose the **CopyAndShareAMI** automation document name you noted in the output tab of CloudFormation stack, in **Step 3**.
- Choose following values:
 - Document version** as the **latest version at runtime**.
 - Leave **execution mode** as it is.
 - Under **Input parameters**, all parameters would be prepopulated. Review following parameters. Ensure that these parameters have correct values specified (values are case-sensitive).

Parameter	Description of the value
productName	Name of the product of the golden AMI. The syntax of this parameter is productName-productversion .
productOSAndVersion	The OSName-OS-version of the golden AMI to be distributed.
buildVersion	The build version of the golden AMI you want to distribute.
bucketName	This needs to match exactly with the name of the bucket in which you uploaded the CFT for launching golden AMI, in Step 3 . The distribution process creates and uploads the version specific template to the bucket.

templateFileName	This needs to match exactly with the file-name of the CFT you uploaded for launching a golden AMI, in Step 3 . The distribution process creates and uploads the version specific template to the bucket.
MetadataJSON	<p>This is the metadata for distributing the golden AMI. It allows you to specify multiple accounts and multiple regions for each account. Please find the sample format below. Ensure that quotes are escaped. If you do not wish to distribute the golden AMI to another account but another region in your account, then you can specify the Master account ID.</p> <p>Sample Format</p> <pre>{\"Destination-Account-1\": \"us-west-1,us-west-2\", \"Destination-Account-2\": \"us-east-1,us-east-2,us-west-2\"}</pre>

8. Choose **Execute Automation**.

Here is an architecture diagram of the distribution process.

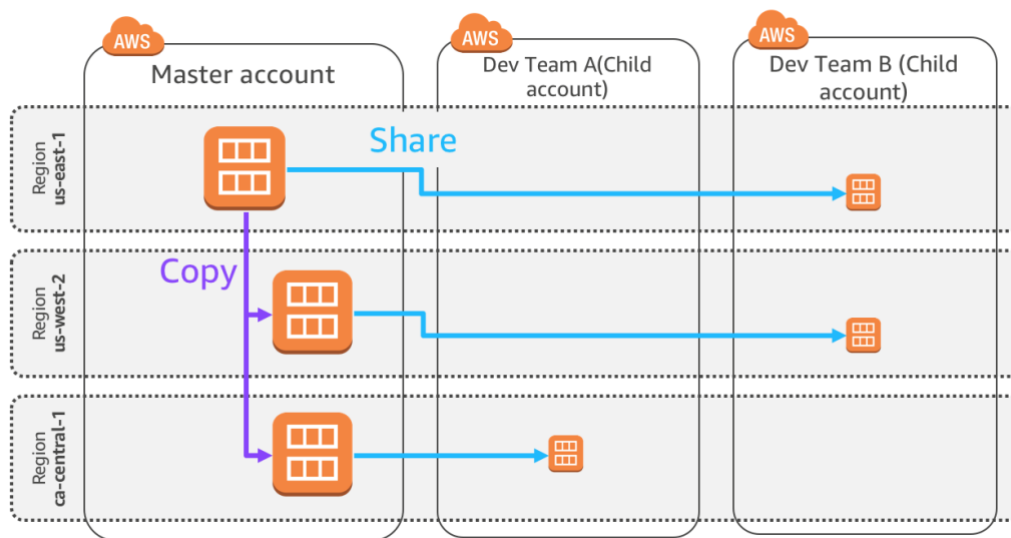


The above process distributes the golden AMI in the following manner:

1. It creates a copy of the golden AMI in each unique region specified in the **metadataJSON**. It then creates metadata in parameter store to maintain the AMI-ID of the version as well as the list of active golden AMIs for that region.
2. It maintains an account-region mapping as well as the AMI-region mapping in the parameter store in the master account's region where the pipeline is set up.

- Next, it shares the AMI with the child account and assumes the cross-account role to create necessary metadata parameters in the child account. It also updates the list of active golden AMIs in the child account's SSM parameter store.
- Finally, it downloads the template you uploaded in **Step 3** and uploads a version specific copy of the template.
- It then creates an AWS Service Catalog product or if the product already exists, a provisioning artifact for the existing product. Later, end-users will launch this product in the child account.

The following diagram depicts the distribution of a golden AMI version to two accounts in three regions. Where **Dev Team B** needs to deploy the golden AMI in us-east-1 and us-west-2 regions whereas **Dev Team A** needs to deploy the AMI in ca-central-1 region.



You may have a requirement of distributing the golden AMI in more governed manner – especially if it is a paid marketplace AMI. You can use [AWS Service Catalog](#) to facilitate a governed distribution. You will not provide **RunInstances** permission to your users on paid golden AMIs. Instead, you can create an AWS Service Catalog Portfolio and allow them to deploy only specific golden AMIs through AWS Service Catalog. The distribution process creates an AWS Service Catalog product to facilitate the governed distribution. To know more about how AWS Service Catalog works, see [Getting Started](#) guide.

Step 10 (optional): Distribute the AWS Service Catalog product to the child account

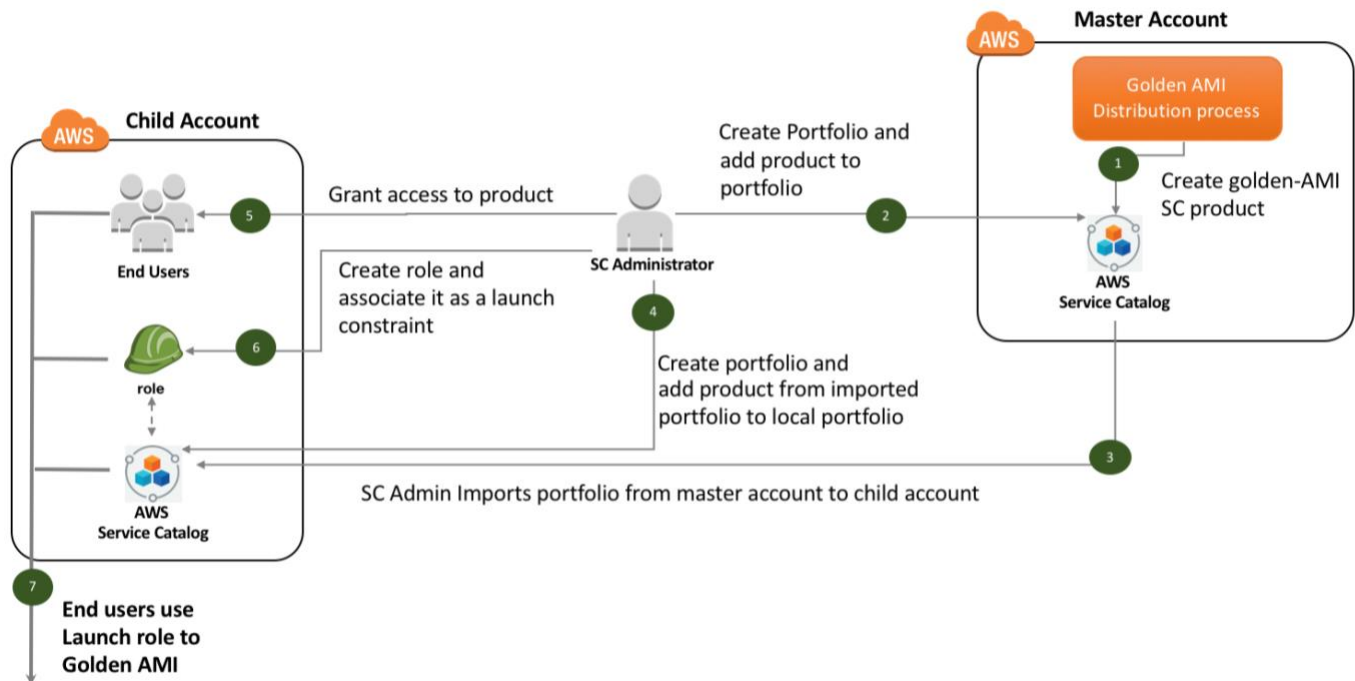
With AWS Service Catalog, you can allow users to create a stack using a specific CloudFormation template while assuming a specific IAM role. You can also enable governance by automatically assigning specific tags to resources created from a specific AWS Service Catalog product. AWS Service Catalog automatically associates the principalARN, portfolioARN, and productARN tag with resources provisioned through AWS Service Catalog portfolio. This information lets you track the user that deployed a specific resource through the AWS Service Catalog.

Before I get into how it works, let's first review a few key AWS Service Catalog concepts:

- A **product** is an IT service that you want to make available for deployment on AWS. You create a product by importing a **CloudFormation template**.
- A **provisioned product** is a **CloudFormation stack**. When an end user launches a product, the AWS Service Catalog provisions the product in form of a CloudFormation stack.
- A **portfolio** is a collection of products, together with the configuration information. You can use portfolios to manage the user access to specific products.
- **Constraints** control the way users can deploy a product. With launch constraints, you can specify a role that the AWS Service Catalog can assume to launch a product from the portfolio.

Next, you will set up the infrastructure for distribution of your golden AMI in a Hub-and-Spoke manner. For more information about the hub-spoke model see the Blog on [Hub-and-Spoke-model](#).

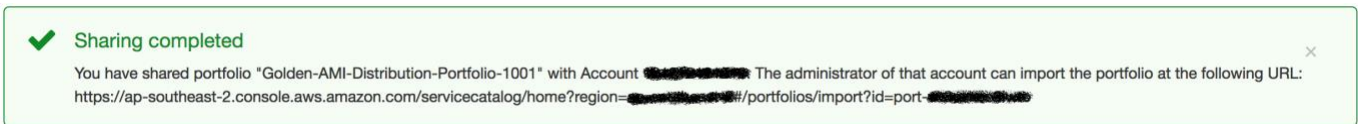
The golden AMI distribution process would create an **AWS Service Catalog product** in the master account and then you would distribute it to the child account(s). End-users will launch the product from the child account. Here is the architecture diagram of the process. However, you would need to set up portfolios (one-time process) for your users and would need to import the product to the portfolio every time a new Product-OS combination is used for creating a golden AMI build.



To set up the AWS Service Catalog portfolio:

- Sign-in** to the AWS Management Console using master account's credentials. Ensure that you are in the child account's region.
- Navigate to the **Service Catalog** service in the **Management Tools** section of the **Services** drop down.
- Choose **Create Portfolio**.
- Specify:
 - Portfolio Name:** **Golden-AMI-Distribution-Portfolio-Cost-Center**
 - Owner:** Specify your department's name

- e) Choose **Create**.
- f) Open the portfolio to view the details.
- g) Choose **Add Product**.
- h) Choose the product corresponding to the golden AMI. The product's name would be in **ProductName-ProductOS** format.
- i) Choose **Add Product to Portfolio**.
- j) Expand **Share with Other AWS Accounts** and choose **Add Account**.
- k) Provide Child-Account-ID.
- l) Choose **Share**.
- m) After a few seconds, a message like following will appear. Note the URL. Later, you will Sign-in to child account, browse to correct region, and launch the URL you noted.



- n) Next, Sign-in to the AWS Management console using child account's Admin credentials and choose **Service Catalog** from the **services** menu.
- o) Ensure that you are in the correct region.
- p) In the child account, launch the URL you noted in **Step m**).
- q) Choose **Import**.
- r) Follow **steps c) to e)**.
- s) Choose **Add Product**.
- t) From **Select Portfolio group**, choose **Imported Portfolio**.
- u) Choose **Select Portfolio** and then Golden-AMI-Distribution-Portfolio-**Cost-Center**.
- v) Choose product name you chose in **Step h**).
- w) Choose **Add product to Portfolio**.
- x) Expand **Users, groups, and Roles** and choose **Add User, group or role**. Choose **Users** tab and select the user that has the **AWSServiceCatalogEndUserFullAccess** policy associated with it. Choose **Add Access**.
- y) Expand **Tags** and add following key-value pairs:
 - a. Key as **Cost-Center** and corresponding value as **Cost-Center** provided to you.
 - b. Key as **Generated-By** and corresponding value as **Golden-AMI-Pipeline**.
- z) Next, create a launch constraint in the child account's Service Catalog portfolio and associate it with the golden AMI product. To know more about how to create a launch constraint, see documentation on [AWS Service Catalog Launch Constraints](#). Ensure that Launch role has following permission to create an instance of the golden AMI.
 - a. Run an EC2 instance and create any resource specified in the CloudFormation template
 - b. Get SSM parameter
 - c. Get Object from S3
 - d. CloudFormation permissions
 - e. Service Catalog permissions

Note

- If you are distributing the golden AMI to multiple regions, you need to do this for each region of the master as well as the child account's region in which product would be launched.
- You can also manage tags (cost-center, business-unit, etc) across portfolios using TagOptions feature to ensure that resources deployed from the portfolio have consistent tags associated with them. To know more about TagOptions, see documentation on [Managing TagOptions](#).

You have successfully created a local portfolio in the master account and imported the same in the child account. You have also created a local portfolio in the child account for distribution. You do not need to perform this process for each product or golden AMI. Portfolio set up is a one-time process in which you identify all appropriate users/user-groups and then share the portfolio from the master and import portfolio into the local portfolio in the child account.

You need to import product to the master portfolio every time a new product-OS combination is specified during the golden AMI creation process i.e. **Step f) to i)**. You would also need to import the product to the child portfolio from the imported portfolio i.e. - **Step s) to w)**. However, if you are creating a new version of the golden AMI for an existing Product-OS combination, you do not need to do anything. it would appear in all child account's AWS Service Catalog automatically without having to re-import the product.

Step 11 (optional): Deploy the golden AMI using Service Catalog Portfolio

To verify if end-user can launch the golden AMI using AWS Service Catalog:

1. Sign-in to the AWS Management Console with the end-user's credentials. Next, open the **AWS Service Catalog** console at <https://console.aws.amazon.com/servicecatalog/>.
2. In the **Products** section of the console, choose product corresponding to the golden AMI.
3. Choose **Launch Product**.
4. On the Product version page, for **Name**, type **Golden-AMI-Instance**.
5. In the Version table, choose the version corresponding to the golden AMI you distributed.
6. Choose **Next**.
7. On the Parameters page, leave the AMI-ID as it is and specify appropriate parameters.
8. Choose **Next**.
9. On the **TagOptions** page, choose **Next**.
10. On the **Notifications** page, choose **Next**.
11. On the **Review page**, review the information you typed, and then choose **Launch** to launch the stack.

The initial status of the product will be **Under Change**. The status becomes **Available** once product launches successfully.

Next, Sign-in to the AWS management console using child account's admin credentials and open the AWS Config console. You will notice that the instance you launched is marked as **Compliant**.

Step 12: Decommission the golden AMI

The pipeline does not decommission active AWS Service Catalog products. This means that any product that has been associated with a portfolio (the master or the child portfolio) must be dis-associated with all portfolios before you trigger the decommissioning workflow for the golden AMI. To disassociate product from all accounts:

- a. Sign-in to child AWS management console using Admin credentials, browse to **Service Catalog** dashboard.
- b. Open the local portfolio in which you had added the product.
 - Remove all constraints associated with the product.
- c. Disassociate the product.
- d. Perform **Step a. to c.** for each child account as well as the the master account for each applicable region.

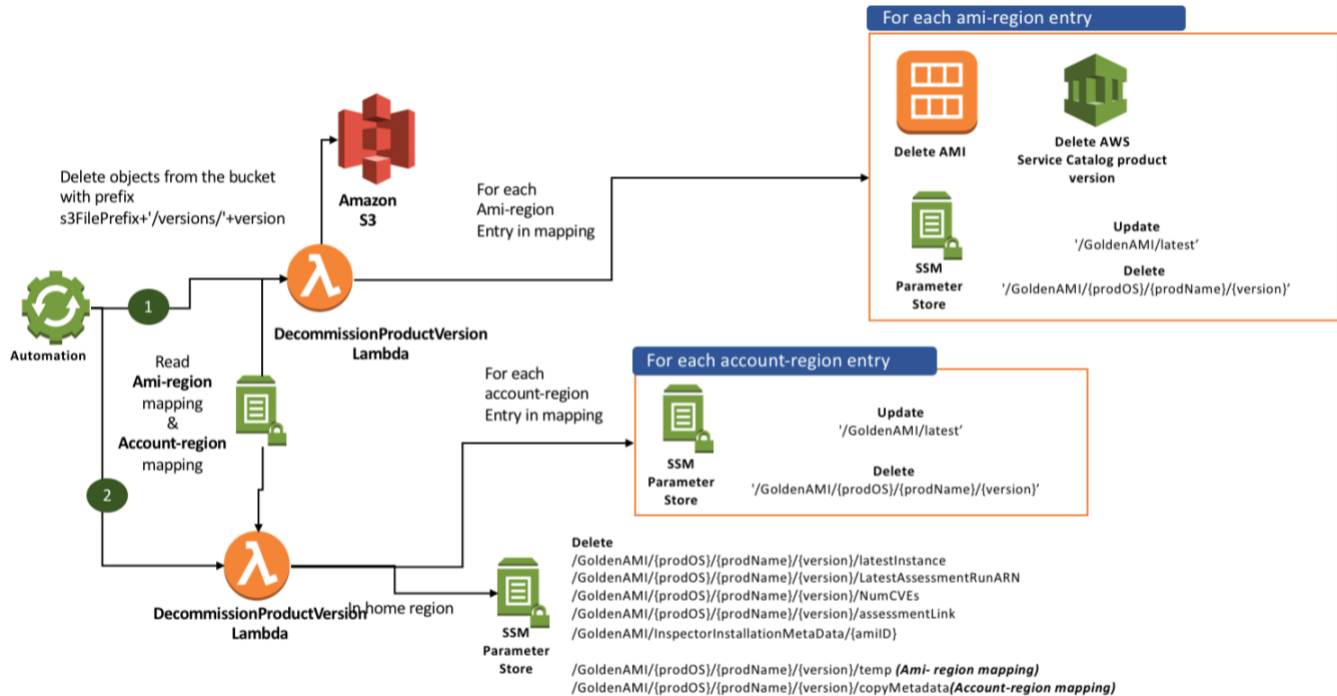
Proceed with instructions to trigger the workflow only after you have disassociated the product with all AWS Service Catalog portfolios. Next, you will decommission the golden AMI version. To decommission the golden AMI version:

1. Sign-in to AWS Management console using master AWS account credentials and navigate to **Systems Manager** service.
2. Ensure that you are in the correct region.
3. In the navigation panel, choose **Automation** under **Actions** drop-down.
4. Choose **Execute Automation**.
5. Filter automations visible by choosing **owned by me** filter option.
6. Choose the **DecommissionAMIVersion** automation document name you noted in output tab of CloudFormation stack, in **Step 3**.
7. Choose following values:
 - a. **Document version** as the **latest version at runtime**.
 - b. Leave **execution mode** as it is.
 - c. Under input parameters, all parameters will be prepopulated. Review following parameters. Ensure that the values belong to the golden AMI version you wish to decommission (values are case-sensitive).

Parameter	Description of the value
productName	The productName-productversion of the golden AMI build you want to decommission.
productOSAndVersion	The OSName-OS-version of the golden AMI build you want to decommission.
buildVersion	The build version of the golden AMI build you want to decommission.
bucketName	This must match exactly with the name of the bucket in which you uploaded the CFT for launching golden AMI, in Step 3 . The decommissioning process deletes the file and the Service Catalog product for the build.
templateFileName	This must match exactly with the name of the CFT you uploaded to S3, in Step 3 . The decommissioning process deletes the file and the Service Catalog product for the build.

8. Choose **Execute Automation**.

Here is an architecture diagram of the golden AMI version decommissioning workflow.



The automation will remove all traces of the specified version including the AMI, copies of the AMI in other regions, and all parameters created in the master as well as the child account. It also deletes the AWS Service Catalog product/provisioning artifact. The process does retain the Inspector assessment result for future reference.

Conclusion

Setting up an efficient toolchain for a large enterprise can require substantial effort, and often hinges on a few people in a big company. Many companies build internal tools and processes using code written by one or two developers. This approach creates problems as companies grow because it does not scale and usually does not include automation. AWS provides a consistent template model, which ensures consistency and reduces the risk of failure.

You can source many AMIs from the Amazon EC2 Console or AWS Marketplace. By building and verifying approved hardened AMIs using the solution described in this Read-me-guide, you can tag, catalog, apply policies, and distribute AMIs across your organization.

References

- [Building a Secure, Approved AMI Factory Process Using Amazon EC2 Systems Manager \(SSM\), AWS Marketplace, and AWS Service Catalog](#)
- [How to Set Up Continuous Golden AMI Vulnerability Assessments with Amazon Inspector](#)
- <https://aws.amazon.com/servicecatalog/>
- <https://aws.amazon.com/ec2/systems-manager/>
- <https://aws.amazon.com/inspector/>
- <https://aws.amazon.com/marketplace/>