Building a Secure, Approved AMI Factory Process Using Amazon EC2 Systems Manager (SSM), AWS Marketplace, and AWS Service Catalog

November 2017

© 2017, 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.

Contents

Abstract

Customers require that AMIs used in AWS meet general and customer-specific security standards. Customers may also need to install software agents such as logging or antimalware agents. To meet this requirement, customers often build approved AMIs, that are then shared across the many teams. The responsibility of building and maintaining these can fall to a central cloud or security team, or to the individual development teams.

This paper outlines a process using the best practices for building and maintaining Approved AMIs through Amazon EC2 Systems Manager and delivering them to your teams using AWS Service Catalog.

Introduction

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 isn't uncommon for organization to build fragile, internal tool chains. Those are often dependent on one or two skilled people whose departure introduces risk.

This whitepaper presents 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 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 operating system 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.

Development teams want to consume the latest approved AMI in the simplest way possible, often through automation. They want to customize these approved AMIs with the required software components, but also ensure that the images continue to meet your organization's InfoSec requirements.

This solution uses Amazon EC2 Systems Manager Automation to drive the workflow. Automation defines a sequence of steps and is composable. The solution is broken down into a set of logical building blocks where the master workflow invokes the following individual components:

1. Build the AMI
2. Validate the AMI
3. Publish the AMI to AWS Service Catalog

The master Automation invokes all the steps, as illustrated in the following figure.

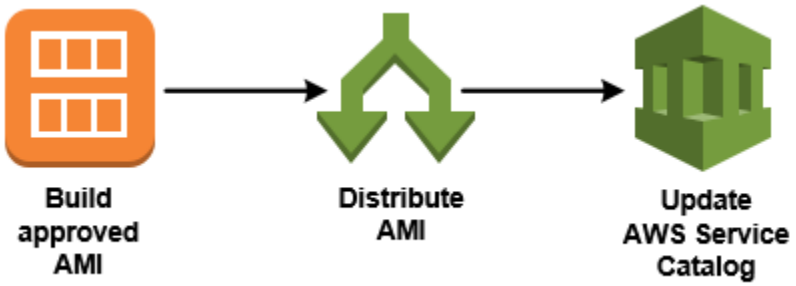

Figure 1: Solution overview

The development teams can repeat this process. Each team can add their own software and produce a new AMI that is scanned, distributed, and consumed as necessary. The extended flow across the teams is as follows:

- Central cloud engineering team is responsible for the following:
  - Setting policy on the specified operating systems, the variants, and the frequency of change policy.
  - Building the approved AMIs that include the latest operating system updates, hardening requirements, and approved software agents.
  - Running AWS EC2 Systems Manager Automation to build approved AMI.
  - Making the AMI available to teams for further automation with EC2 Systems Manager and making the product available through Service Catalog.

  - Optional: Setting up AWS EC2 Systems Manager to automate scheduled scanning of approved AMIs for vulnerabilities using Amazon Inspector.

- Development Teams are responsible for the following:
  - Building the application stacks used in production, and meeting any hardening requirements. You can use AWS EC2 Systems Manager or AWS Code Pipeline to build the required AMIs or AWS CloudFormation stacks.
  - Optional: Completing any steps that require authorized approval.
  - Optional: Provide the resulting approved application stack for deployment via automation or AWS Service Catalog.

The solution uses the following AWS Services:

- AWS Service Catalog[1]
- Amazon EC2 Systems Manager[2]
- Amazon Inspector[3]
- AWS Marketplace[4]
- AWS CodePipeline[5]
- AWS CodeCommit[6]

Building the Approved AMI

The key to the entire process is generating an AMI that meets all your hardening requirements. The following diagram illustrates the high-level process.

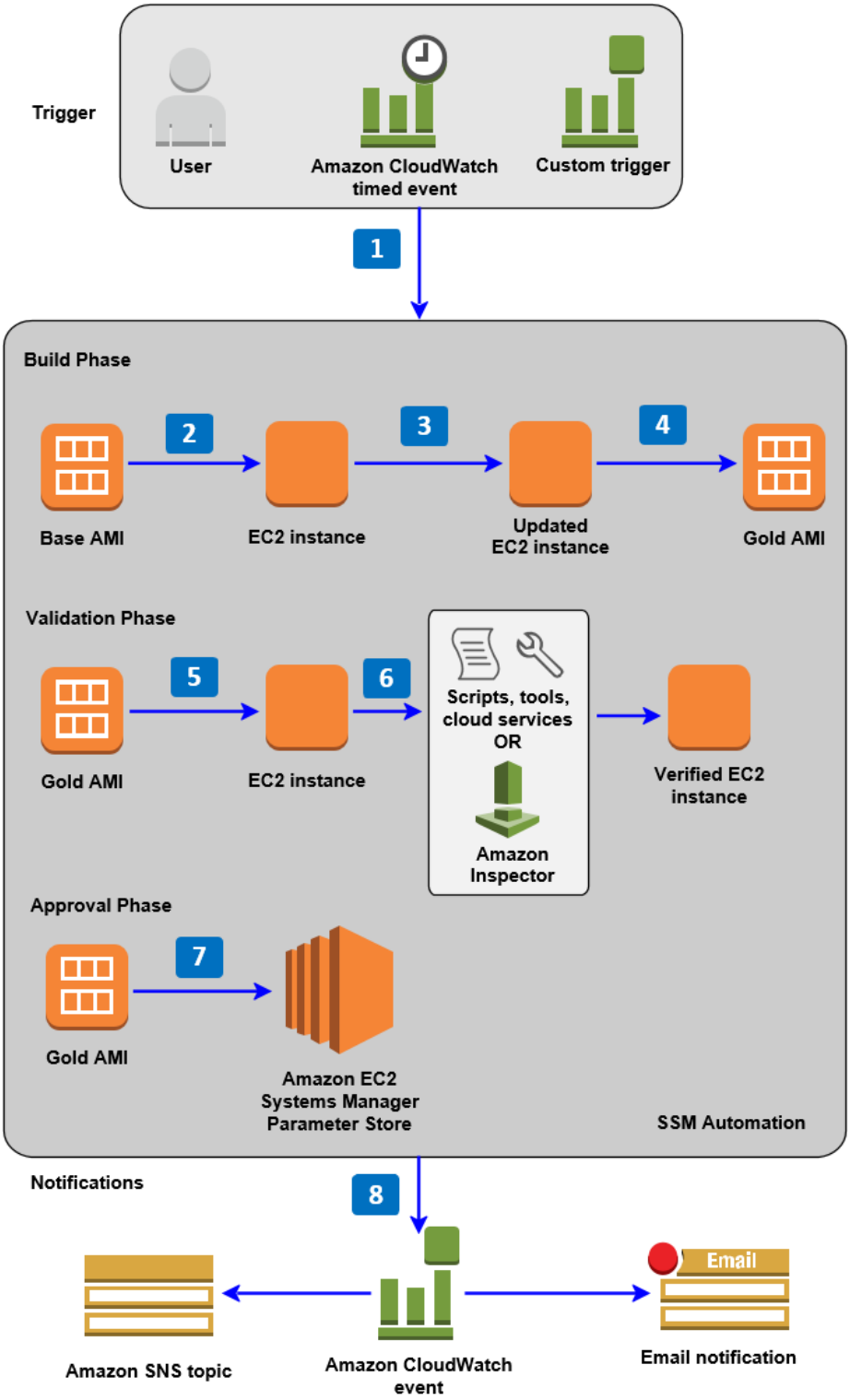

Figure 2: AMI hardening process

Phase	Description
Automation Trigger	You can configure Amazon EC2 Systems Manager Automation to be triggered by a user or an event.
1	You can set up an event using Amazon CloudWatch (for example, a monthly timed event) or some other customer event (for example, when code is checked into AWS CodeCommit.
Build Phase	The build phase takes a source AMI as the input and generates a hardened AMI ready for testing.
2	**Create instance** – An instance is created from the latest available base AMI. This could be an Amazon, AWS Marketplace or customer-provided AMI. As part of the instance launch, you install Amazon EC2 Systems Manager (SSM) Agent using userdata.
3	**Run command** –When the instance is up and running, packages and scripts are securely downloaded from an Amazon S3 bucket and executed. This could include operating system updates, operating system hardening scripts, and the installation of new software and configuration changes. These packages and scripts could be anything from custom bash scripts to Ansible playbooks.
4	**Build AMI** – After the instance has been updated, a new hardened AMI is created.
Validation Phase	Depending on your requirements, you can use custom scripts, third-party security software, or Amazon Inspector to verify that your instances meet your security requirements. Regardless of your choice, the process is the same. If you have implemented a custom scanning solution.
5	**Create instance** – A new instance is created from the hardened AMI.
6	**Run command** – When the instance is up and running, validation scripts and tools can be securely download from an S3 bucket, and then executed to validate the instance, or you can use Qualys, Nessus, or Amazon Inspector to validate the AMI.
Approval Phase	After the scanning is complete, you can inspect the reports before approving the new hardened AMI.
7	You can store the new hardened AMI ID in a data store, such as the SSM Parameter Store, which can be used by other automations later in the pipeline.
Notifications	After the Automation job is complete, you can notify your teams.
8	You can use CloudWatch Events to generate email alerts to teams and Amazon Simple Notification Service (Amazon SNS) notifications to trigger other automations.

Considerations for AWS Marketplace AMIs

AWS Marketplace AMIs have a Marketplace product code attached to the AMI. When you create your version of the AMI, this product code is copied across to

the new AMI. You need to confirm that any changes you make to the AMI don't affect the stability or performance of the product.

Some Marketplace offers come with vendor designed Cloud Formation templates, to reduce effort on establishing clusters and HA configurations. If the product can only be launched from AWS Marketplace using an AWS CloudFormation template, you must update the AMI ID in the template to customize and harden the instance to create a new AMI. You can download and change the template from the AWS Marketplace product page. If the template launch requires any scripting, test the template to ensure that these scripts work as expected.

Distributing the Approved AMI

After you have an approved AMI, you can distribute the AMI across AWS Regions, and then share it with any other AWS accounts. To do this, you use an Amazon EC2 Systems Manager Automation document that uses an AWS Lambda function to copy the AMIs across a specified list of regions, and then another Lambda function to share this copied AMI with the other accounts. The resulting AMI IDs can be stored in the SSM Parameter Store or Amazon DynamoDB for later consumption.

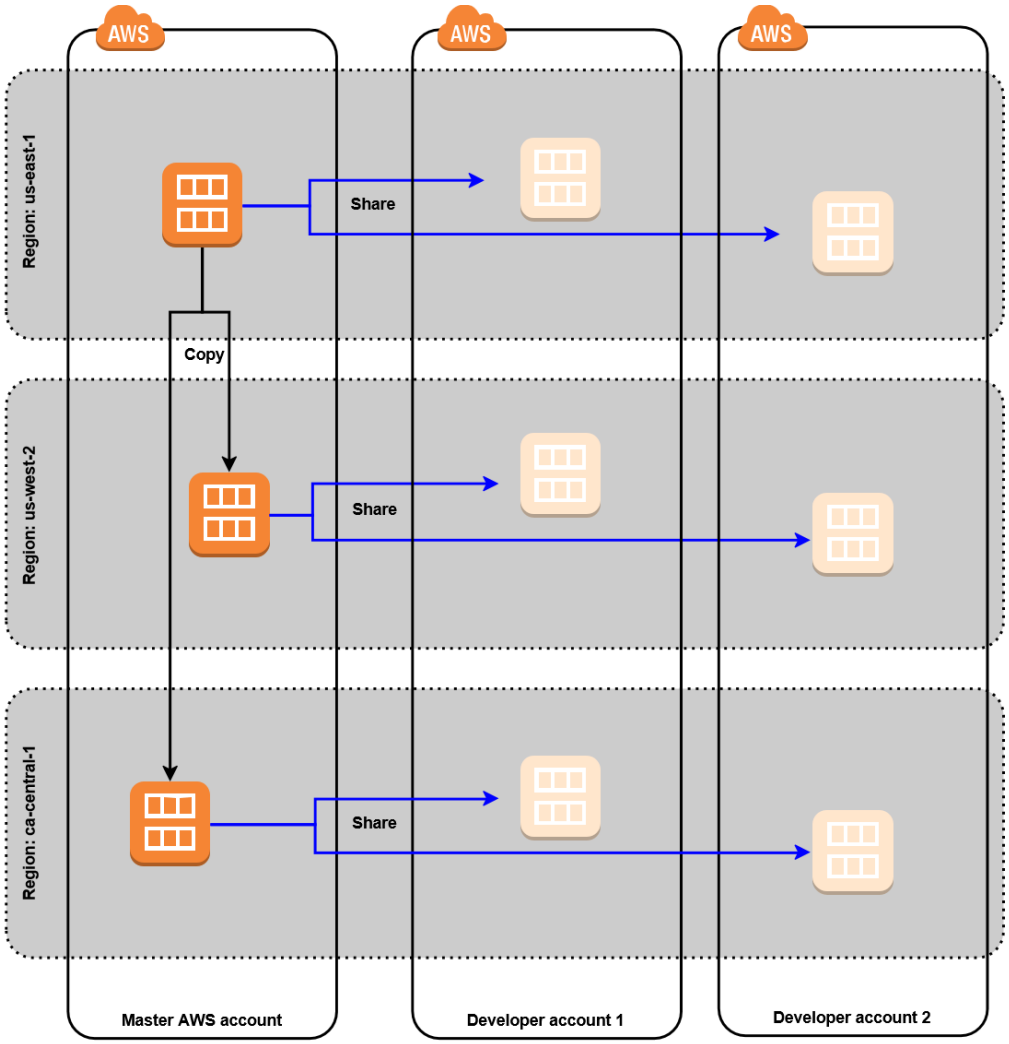

Figure 3: Copying and sharing across AWS Regions and accounts

After the AMI is shared with the specified accounts, you can trigger another notification using email or SNS which could start further automations.

If there is a requirement to encrypt the AMIs, the process is similar, except instead of sharing the AMI with accounts, the AMI must be copied to each account and then encrypted. This increases the number of AMIs to manage, but you can still automate it using the same process.

> **Note** If you have sourced the AMI from AWS Marketplace, make sure that any accounts you share this new AMI with subscribes to the product in Marketplace.

Distributing and Updating AWS Service Catalog

AWS Service Catalog has two important components: products and portfolios (a collection of products). Both components use JSON/YAML CloudFormation templates. You can apply constraints, tags, and policies to a product or portfolio. AWS Service Catalog supports up to 50 versions per product. AWS Service Catalog provides a TagOption library that enables you to create and apply repeatable and consistent tags to a product.

After you build and distribute the AMIs, you can update AWS Service Catalog portfolios across the AWS Regions and accounts.

When managing multiple AWS Service Catalog product portfolios across AWS Regions and your organization's AWS accounts, it is good practice to use a script to create portfolios and products. You can store portfolio definitions in a JSON or YAML file, and then create portfolios using scripts that target specific accounts and regions as shown in the following figure.

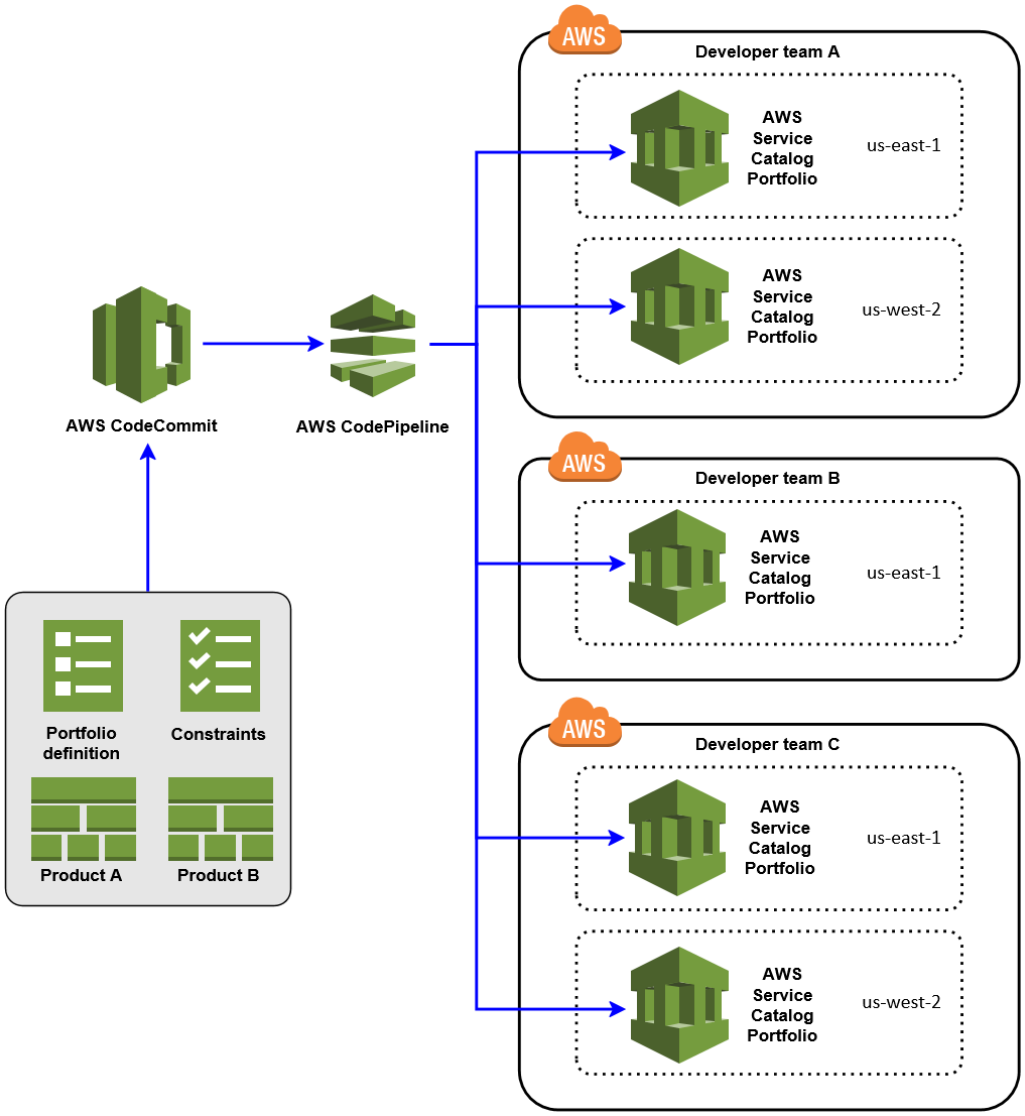

Figure 4: Distributing AWS Service Catalog portfolios and products

When the AMI is updated you can create a new version of an AWS Service Catalog product. To do this, you need to generate a new AWS CloudFormation template for the product containing the updated AMIs. You can handle AWS Regions using the standard CloudFormation mappings sections. You can standardize the template and use a parameter for the AMI ID. You can enforce the AMI ID by defining a template constraint. Regardless of how you choose to set it up, the process for deploying portfolios and products remains the same.

Continuously Scanning Published AMIs

You need to regularly scan approved AMIs to ensure that they don't contain any newly discovered Common Vulnerabilities and Exposures (CVEs). You can schedule daily inspections of the AMI, as shown in the following architecture diagram.

To kick start the continuous scanning process, you set up a CloudWatch Event that is triggered based on a schedule. The event starts a new Automation document execution as illustrated in the following figure.

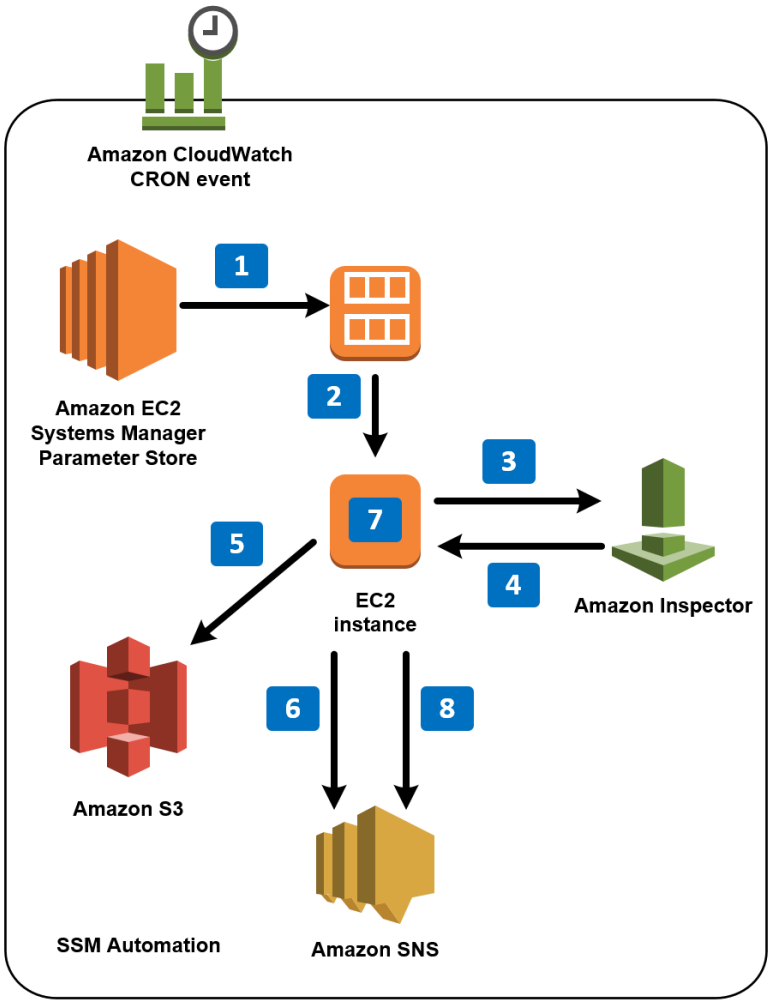

Figure 5: Continuous scanning architecture overview

Number	Description
1	**Read AMI ID** – The SSM Automation document reads the AMI IDs from parameter store.
2	**Launch AMI** – The SSM Automation document launches EC2 instances with a userdata script and installs the Amazon Inspector Agent.
3	**Trigger Amazon Inspector assessment** – The Automation document starts the Amazon Inspector assessment on the instance.
4	**Update assessment execution status** – The results of the Amazon Inspector is sent from the agent on the instance, back to Amazon Inspector.
5	**Update Amazon Inspector assessment result** – The Amazon Inspector results are stored in an S3 bucket for later retrieval.
6	**Notification of any high/medium/low CVEs** – A notification is sent via SNS if any CVE's are found.
7	**Terminate the instance** – The SSM Automation document terminates the instance.
8	**Send notification** – After the Amazon Inspector assessment is complete, a message containing the CVE details is published to an SNS topic.

You can also set up CloudWatch Events to identify Automation document execution failures.

Conclusion

Setting up an efficient tool chain 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 doesn't scale and usually doesn't 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 whitepaper, you can tag, catalog, apply policies, and distribute AMIs across your organization.

Document Revisions

Date	Description
November 2017	First publication

Notes

[1] https://aws.amazon.com/servicecatalog/

[2] https://aws.amazon.com/ec2/systems-manager/

[3] https://aws.amazon.com/inspector/

[4] https://aws.amazon.com/marketplace/

[5] https://aws.amazon.com/codepipeline/

[6] https://aws.amazon.com/codecommit/