

The background is a vibrant, multi-colored gradient. It features a diagonal split between a blue-purple gradient on the left and an orange-yellow gradient on the right. The text 'AWS re:Invent' is positioned on the left side, overlapping the blue-purple area.

AWS
re:Invent

CMP214-R

EC2 Image Builder: Building virtual machine images made easy

Samartha Chandrashekar

Senior Product Manager
Amazon Web Services

Agenda

Importance of “golden” VM images

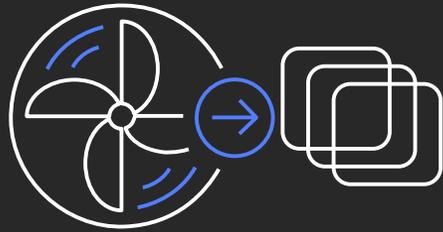
Pain points of building “golden” VM images

How EC2 Image Builder helps

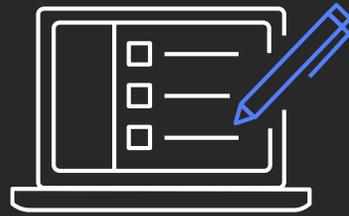
Usage workflows

Best practices

Golden VM images



Template server image.
Saves time & ensure
consistency



Pre-installed & pre-
configured with custom
software & settings



Hardened to meet IT
standards

Nearly every organization has to build golden VM images

However...



Producing “golden” VM images is challenging due to the high-skill bar to build automation



Customers using existing automation/scripting tools need to host & maintain deployments



Many manually build and test new images every time software updates are available



Enterprises encounter service disruptions from not catching issues before production use

Customers asked for a one-stop shop to build golden images

Customers have asked to be able to...



Quickly and easily build automation to create golden images without writing code



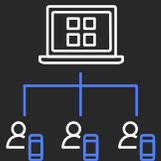
Easily test images with AWS-provided and custom tests before deploying to production



Secure images with AWS-provided & custom settings to meet internal/industry standards



Distribute and share images easily across accounts & regions with centralized enforcement

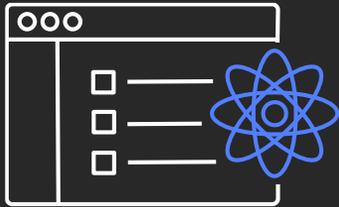


Build images for use on AWS and on-premises

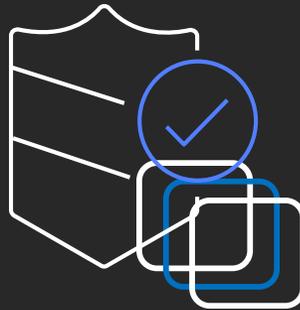
Introducing EC2 Image Builder...

EC2 Image Builder – benefits

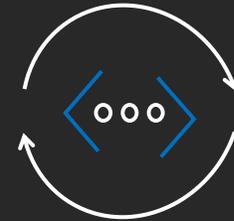
Quickly and easily automate the creation, management, and deployment of up-to-date and compliant “golden” VM images



Generate automation to build VM images with a GUI

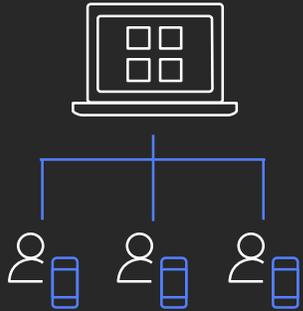


Reduce cost of building secure, compliant & up-to-date images

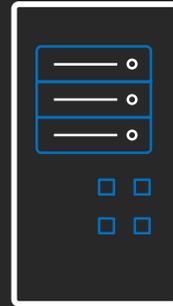


Improve service uptime by testing images before use in production

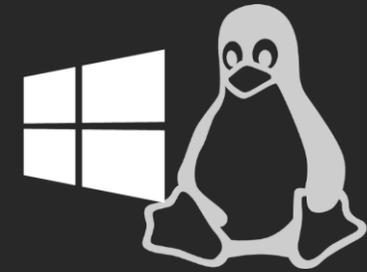
EC2 Image Builder – benefits



Enforce policies on VM image usage across AWS accounts

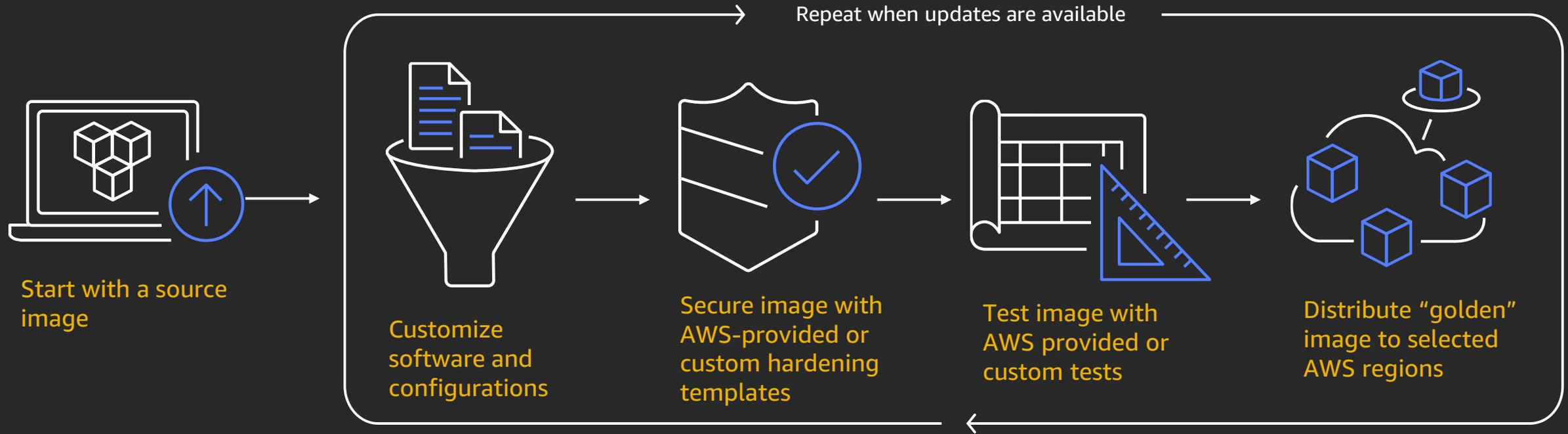


Build golden VM images for use on AWS and on-premises



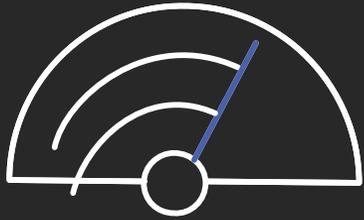
Works for both Windows and Linux

EC2 Image Builder – how it works

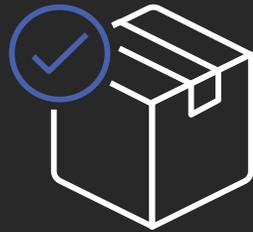


All EC2 Image Builder operations run in your AWS account

AWS-provided build components and tests



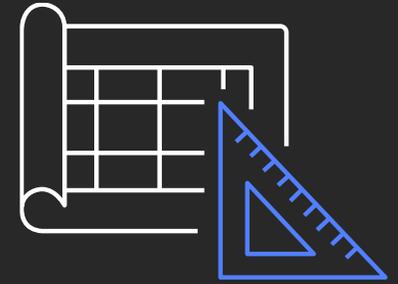
Get started quickly
with AWS provided
components



Commonly used
software provided as
"managed
components"



Security hardening
primitives for STIG
available as inbox
components



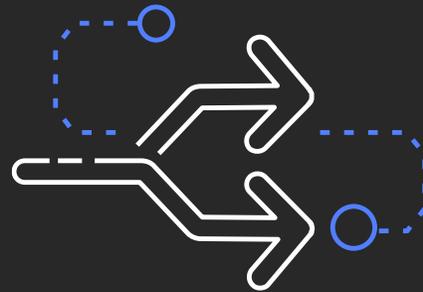
AWS provides
tests to validate
your images
after they are
built

Declarative document to define custom components



Phases

build & validate, test



Each step in each phase uses action modules & returns an exit code

ExecuteBinary, ExecuteBash, ExecutePowerShell
Reboot, UpdateOS
S3Upload, S3Download
SetRegistry (Windows-only)



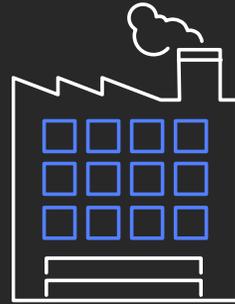
Files for each execution

detailedOutput.json
document.yaml
console.log
application.log

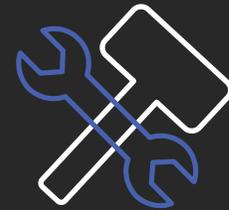
Integration with AWS License Manager



Associate license configurations to AMIs

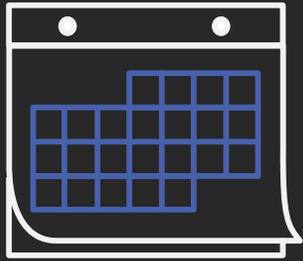


Use AWS License Manager to centrally manage licenses and their usage



Create and manage license configurations with AWS License Manager

Keeping images patched and up-to-date



Rebuild images with latest changes on a schedule



You can also manually trigger rebuilds



Include latest updates to the OS, build components, and tests based on semver based triggers

Shared responsibility for security

Ensuring the security of custom AMIs is a shared responsibility between AWS and the customer



You own the security posture of images produced

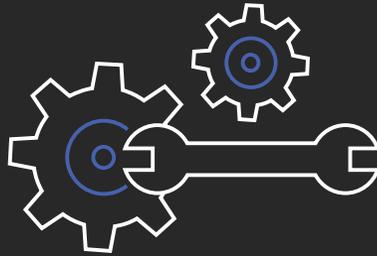
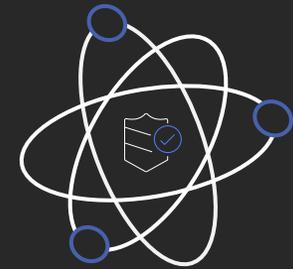


Image Builder enforces cleanup after building images (Linux: cleanup script run; Windows: Sysprep)



We recommend that you validate the security posture (AWS Inspector can help)

AWS does not guarantee images produced with Image Builder to be compliant with regulatory criteria

Identity and Access Management



Service Linked Role (SLR)

Grants permissions to EC2 Image Builder on your behalf. When you create your first Image Builder resource, an SLR is created for you.



IAM permissions to Instance role

IAM Role attached to the EC2 instance used to install build components, run tests, and write troubleshooting logs to Amazon S3 needs requisite permissions



Just enough permissions

EC2 Image Builder concepts

Pipeline

Recipe

Build component

Test

Build schedule

Managed Image

EC2 Image Builder workflow & demonstration

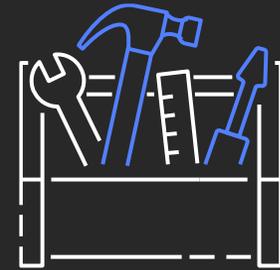
Debugging and troubleshooting



EC2 Image Builder tracks and displays progress for each step



Logs exported to Amazon S3



Run arbitrary commands & scripts with AWS Systems Manager RunCommand

Example of a log file for debugging

```
2019-11-29 23:31:57 Info Document TOE_2019-11-29_23-31-57.UTC-0_6f164c8e-1300-11ea-83b0-0e7e644ebf61/windows-throw-exception_1237940039285380274899124225.ps1
```

```
2019-11-29 23:31:57 Info Phase build
```

```
2019-11-29 23:31:57 Info Step HappyStep
```

```
2019-11-29 23:31:59 Info Command execution completed successfully
```

```
2019-11-29 23:31:59 Info Stdout: Directory: C:\
```

Mode	LastWriteTime	Length	Name
d----	11/14/2018 6:43 AM		EFI
d----	11/29/2019 11:31 PM		imagebuilder
d----	9/15/2018 7:12 AM		PerfLogs
d-r---	11/14/2018 4:08 PM		Program Files
d----	11/29/2019 11:28 PM		TOE_2019-11-29_23-28-10.UTC-0_e7ca00c4-12ff-11ea-a640-0e7e644ebf61
d----	11/29/2019 11:31 PM		TOE_2019-11-29_23-31-57.UTC-0_6f164c8e-1300-11ea-83b0-0e7e644ebf61

```
2019-11-29 23:31:59 Info Stderr:
```

```
2019-11-29 23:31:59 Info ExitCode 0
```

```
2019-11-29 23:31:59 Info Step UnhappyStep
```

```
2019-11-29 23:32:02 Info Command execution resulted in an error
```

```
2019-11-29 23:32:02 Info Stdout: This is standard output from Write-Host
```

```
2019-11-29 23:32:02 Info Stderr: This is an exception from a PowerShell throw command
```

```
At C:\Windows\TEMP\AWSTOE836058097\script-702966940.ps1:2 char:1
```

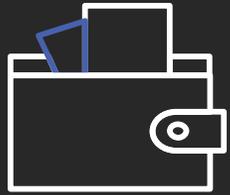
```
+ throw 'This is an exception from a PowerShell throw command'
```

```
+ CategoryInfo          : OperationStopped: (This is an exce...l throw command:String) [], RuntimeException
```

```
+ FullyQualifiedErrorId : This is an exception from a PowerShell throw command
```

```
2019-11-29 23:32:02 Info ExitCode 1
```

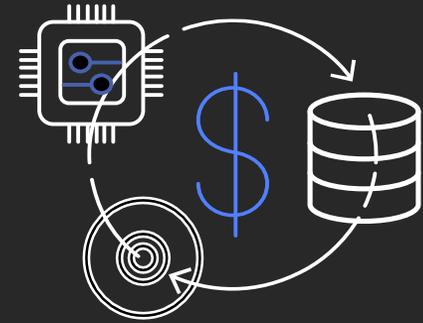
Pricing



No cost

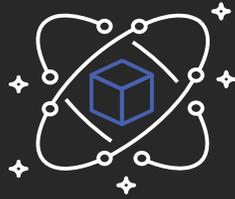


All operations run in your
AWS account



Pay for the resources used
in your account (e.g. EC2 instance
usage, Amazon S3 usage, Systems Manager
Advance, AWS Inspector, etc.)

Summary



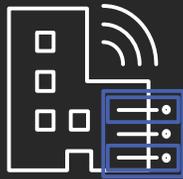
Produce automation to build images with ease

No need to write and maintain code to build automation
GUI wizard to create image building pipelines



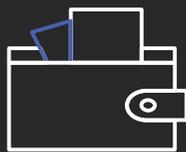
Improve security and uptime

Keep images secure and up-to-date
Capture and reuse security settings
Run tests to catch issues before deploying to production



Hybrid use cases

Produce AMIs for use on AWS
Generate on-prem VM images



No cost

Runs in customer account
Pay for resources used in your account

Thank you!

Samartha Chandrashekar

samarthc@amazon.com

EC2 Image Builder workflow

1) Start with a known good VM image

Amazon Image Factory > Image pipelines > Create pipeline

Step 1
Define recipe

Step 2
Configure pipeline

Step 3
Configure additional settings - optional

Step 4
Review and create

Define recipe
A recipe includes the source image and the software assets of a build. A recipe cannot be modified on recipe or recipe version.

Source image [Info](#)

Image Operating System (OS)
Image Factory currently supports Amazon Linux 2 and Windows Server 2012R2, 2016, and 2019.

Amazon Linux Windows

3) Select how frequently images are rebuilt with pending updates

Build schedule

Select a build preference for running your pipeline.

Custom
You choose when to run the pipeline.

Schedule builder
Automatically run the pipeline using a job scheduler.

CRON expression
Automatically run the pipeline using a syntax that specifies the time and intervals to run it.

Run pipeline every on at

2) Select components to install and run

Build components [Info](#)

Collection of software installations that are run during the build phase to create custom output images. Enter component name or browse to select from a list of Image Factory and custom components.

Enter or select components in the order you want them to run in the pipeline.

Description of the asset

Description of the asset

Description of the asset

4) Select from a list of AWS and custom tests to run

Test components [Info](#)

Collection of test software that are run to validate the output image before distribution. Enter component name or browse from a list of Image Factory components.

Enter or select components in the order you want them to run in the pipeline.

Description of the asset

Description of the asset

EC2 Image Builder workflow

5) Select license configurations to attach to images

Associate license configuration to AMI [Info](#)

Associate output AMI with existing license configurations created within AWS License Manager to track license usage.

License configuration ID
The unique ID of the license configuration created in AWS License Manager.

Choose a license configuration

License 659233704

License 659233704

6) Select AWS regions for distribution

AMI Distribution settings

Allow all AWS accounts (make public) or allow only specified accounts to launch the AMI.

AWS region(s) to distribute AMI

Choose region

Option 1 Tag value

Option 2 Tag value that is longer

Current region included by default.

Launch permissions

Add or remove AWS user account numbers to set launch permissions for the AMI. Launch permissions apply to the current region by default.

Private

Public

Enter account number

account 342344

account 758696

7) Review automated pipeline

Pipeline123-Recipe 01 was successfully created. [View details](#)

Image pipelines (3) [View details](#) [Actions](#) [Create pipeline](#)

Search for pipeline by name

<input type="checkbox"/>	Pipeline name	Description	Date created	Recipe name	Recipe version	Date of last run	Date of next run
<input type="checkbox"/>	Pipeline123	some description here	07/01/19	Pipeline123-recipe	1.0.3	07/11/19 + time stamp	07/11/19 + time stamp
<input type="checkbox"/>	Pipeline456	some description here	02/18/19	Pipeline456-recipe	1.0.0	08/11/19 + time stamp	07/11/19 + time stamp
<input type="checkbox"/>	mylinux1	some description here	02/18/19	mylinux1-recipe	1.0.0	08/11/19 + time stamp	07/11/19 + time stamp

Done! Images are produced by automated pipeline

Image build version (6)

Search by date created or status

Version	Date created	Owner	Status	Reason for failure
1.0.0/6	08/01/19 12.31PST	Amazon owned	Success	
1.0.0/5	08/01/19 12.31PST	Account # 6847698	Success	
1.0.0/4	08/01/19 12.31PST	Account # 6847698	Success	
1.0.0/3	08/01/19 12.31PST	Account # 6847698	Failed	Here is where the reason for failure is shown
1.1.3/2	08/01/19 12.31PST	Account # 6847698	Success	
1.0.0/1	08/01/19 12.31PST	Account # 6847698	Success	

Limits

Name	Description	Limit
Concurrent builds	Maximum concurrent builds that can be in progress in this account in current region	100 builds per account per region
Components	Maximum EC2 Image Builder components that you can create in an account in current region	1000 components per account per region
Component size	Maximum size of the data field of an EC2 Image Builder component	16 KiB
Image pipelines	Maximum EC2 Image Builder image pipelines that you can create in an account in current region	75 image pipelines per account per region
Image recipes	Maximum EC2 Image Builder image recipes that you can create in an account in current region	1000 image recipes per account per region
Build components & tests per recipe	Maximum build components that can be associated with a single image recipe	20 components per image
Infrastructure configurations	Maximum EC2 Image Builder infrastructure configurations in an account in current region.	1000 configurations per account per region
Distribution configurations	Maximum EC2 Image Builder distribution configurations in an account in the current region.	1000 configurations per account per region

YAML-based Document for build components & test

3 possible phases - build, validate, test

Each step in each phase uses **action modules** that return an exit code

ExecuteBinary

ExecuteBash

ExecutePowerShell

Reboot

UpdateOS

S3Upload

S3Download

SetRegistry (Windows only)

Output files for each execution

detailedOutput.json describes details on the orchestration

document.yaml is sent to the application for the execution

console.log contains stdout and stderr information captured during the execution

application.log contains logs generated by debugging executions

Actions performed by on-instance agent

ExecuteBinary to execute binaries with command line arguments

ExecuteBash to run bash scripts with inline shell code/commands

ExecutePowerShell to run PowerShell scripts with inline shell code/commands

Reboot to reboot the instance with a configurable delay

UpdateOS to install Windows and Linux updates

S3Upload to upload a file from a source file/folder to an Amazon S3 location

S3Download to download an S3 object or **KeyPrefix** to a local destination path

SetRegistry (Windows only) accepts inputs and sets a specified registry key

Start with a known good VM image

Select OS

Select image source

(either Image Builder managed image kept up-to date OR existing AMI)

Source image [Info](#)

Image operating system (OS)
Image Factory currently supports Amazon Linux 2 and Windows Server 2012R2, 2016, and 2019

Amazon Linux Windows

Select image [Info](#)
Select the image to configure. You can select images from Image Factory catalog, import an on-premises image, or use a custom AMI.

Select managed images
Image Factory images created by you, shared with you and public images.

Enter custom AMI ID

Browse to select an image

Select software, hardening scripts, and configurations to install, run, and apply

Select components to be installed and applied

View selected components here

Build components [Info](#)

Collection of software installations that are run during the build phase to create custom output images. Enter component ARN or browse to select from a list of Image Factory and custom components.

[Browse build components](#)

- update-linux, Version: 1.0.0
arn:aws:imagefactory:us-east-1:aws:component/update-linux/1.0.0/1
OS: Linux
- amazon-corretto-8-jre, Version: 1.0.0
arn:aws:imagefactory:us-east-1:aws:component/amazon-corretto-8-jre/1.0.0/1
OS: Linux

Select tests to run

Select AWS provided tests to be run

View selected tests here

Tests [Info](#)

Collection of test software that are run to validate the output image before distribution. Enter component ARN or browse to select from a list of Image Factory components

- simple-boot-test-linux, Version: 1.0.0
arn:aws:imagefactory:us-east-1:aws:component/simple-boot-test-linux/1.0.0/1
OS: Linux
- inspector-test-linux, Version: 1.0.0
arn:aws:imagefactory:us-east-1:aws:component/inspector-test-linux/1.0.0/1
OS: Linux

Select how frequently images are rebuilt with pending updates

Set up a schedule to automatically produce patched and up-to-date images

Build cadence

Build schedule
Select a build preference for running your pipeline.

- Manual**
Manually run the pipeline by clicking "Run pipeline" on pipeline detail page.
- Schedule builder**
Automatically run the pipeline using a job scheduler.
- CRON expression**
Automatically run the pipeline using a syntax that specifies the time and intervals to run it.

The screenshot shows a configuration interface for pipeline build cadence. It features three radio button options. The 'Manual' option is selected and highlighted with a blue border. A blue arrow points from the text 'Set up a schedule to automatically produce patched and up-to-date images' to the 'Manual' option.

Select AWS Regions for distribution and permissions

Select AWS Regions to distribute images

Select account IDs that can launch images

AMI distribution settings

Allow all AWS accounts (make public) or allow only specified accounts to launch the AMI.

Select AWS region(s) to distribute the AMI

Choose region ▼

ap-southeast-1 ✕ us-east-1 ✕ us-west-1 ✕

Your current region is included by default.

Launch permissions

Add/remove AWS user account number to set launch permissions for the AMI. Launch permissions apply to the current region by default.

Private
 Public

Enter account number Add

234234234234 ✕ 092340912348 ✕

Select license configurations to attach to images

Select and apply license configurations

Associate license configuration to AMI [Info](#)

Associate output AMI with existing license configurations created within AWS License Manager to track license usage. These settings are only applied to the output AMI in the current region.

Associate license configuration to AMI
The unique ID of the license configuration created in AWS License Manager

[Create new License Configuration](#) 

Review pipeline details

Review and create

Step 1: Image recipe Edit

Recipe details

Recipe name (default) test112343-recipe	Recipe version (default) 1.0.0	Image OS type Linux	Parent image amzn2-ami-hvm-arm64-gp2
Build Assets arn:aws:imagefactory:us-east-1:aws:component/update-linux/1.0.0/1 arn:aws:imagefactory:us-east-1:aws:component/amazon-corretto-8-jre/1.0.0/1	Test Assets arn:aws:imagefactory:us-east-1:aws:component/simple-boot-test-linux/1.0.0/1 arn:aws:imagefactory:us-east-1:aws:component/inspector-test-linux/1.0.0/1		

Step 2: Pipeline configuration Edit

Configuration details

Pipeline name test112343	Description -	IAM role AwsImageFactoryInstanceProfileFromCCIS	Build cadence manual
Infrastructure			
Instance type	Terminate instance on failure Enabled	VPC ID	Subnet

Create pipeline

Step 3: Additional settings Edit

Additional settings

Output AMI name: - Tags: -

AMI distribution settings

Region	Launch permissions	Export format	Export location	Associated license configuration
us-east-1	Users: [234234234234, 092340912348]	AMI	-	-
us-west-1	Users: [234234234234, 092340912348]	AMI	-	-
ap-southeast-1	Users: [234234234234, 092340912348]	AMI	-	-

Cancel Previous Create Pipeline

Create pipeline after reviewing settings

Pipeline is now running!

✔ Image pipeline test112343 was successfully created
Image pipeline ARN: arn:aws:imagefactory:us-east-1:275321894251:image-pipeline/test112343 [View details](#)

AWS Image Factory > Image pipelines

Image pipelines

Represents the configurations to automate and build secure operating system (OS) images on AWS and on premises.

[View details](#) [Actions](#) [Create image pipeline](#)

🔍 Find pipeline < 1 > ⚙️

<input type="checkbox"/>	Pipeline name	Date created	Version	Status	Date of last run	ARN
<input type="checkbox"/>	test112343	-	1.0.0	✔ Enabled	-	arn:aws:imagefactory:us-east-1:275321894251:image-pipeline/test112343

Pipeline is producing images

Output images				
<input type="text" value="Filter images"/>				
Image name	Version	Platform	Date created	ARN
amzn2-ami-hvm-arm64-gp2	2019.10.31	Linux	2019-11-07 23:51:07.636Z	arn:aws:imagefactory:us-east-1:aws:image/amzn2-ami-hvm-arm64-gp2/2019.10.31
amzn2-ami-hvm-x86_64-gp2	2019.10.31	Linux	2019-11-07 23:51:10.660Z	arn:aws:imagefactory:us-east-1:aws:image/amzn2-ami-hvm-x86-64-gp2/2019.10.31
amzn2-ami-minimal-hvm-arm64-ebs	2019.10.31	Linux	2019-11-07 23:51:13.356Z	arn:aws:imagefactory:us-east-1:aws:image/amzn2-ami-minimal-hvm-arm64-ebs/2019.10.31
amzn2-ami-minimal-hvm-x86_64-ebs	2019.10.31	Linux	2019-11-07 23:51:15.353Z	arn:aws:imagefactory:us-east-1:aws:image/amzn2-ami-minimal-hvm-x86-64-ebs/2019.10.31