



Architecture diagrams for manufacturing

Identify Product Defects using Industrial Computer Vision



Identify Product Defects using Industrial Computer Vision:

Architecture diagrams for manufacturing

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

Table of Contents

Product defects computer vision **i**

Product defects computer vision architecture diagram 1

Further reading 2

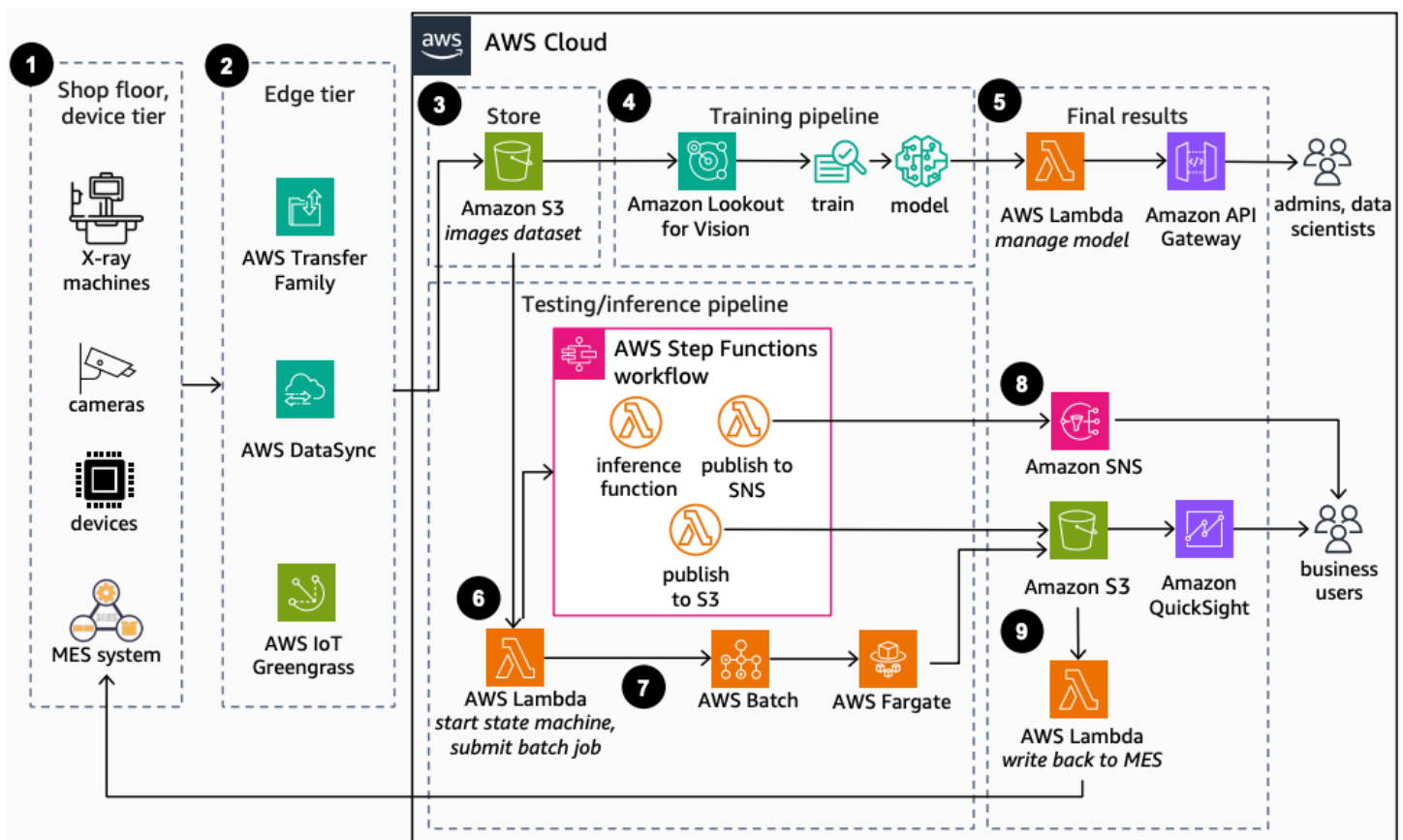
Diagram history 2

Identify Product Defects using Industrial Computer Vision

Publication date: **October 10, 2024** ([Diagram history](#))

With this architecture, you can detect anomalies such as casting metal defects, damage, and irregularities in X-ray images. You use [Amazon Lookout for Vision](#), [Amazon Simple Storage Service](#) (Amazon S3), and [AWS Lambda](#) for quality inspection in manufacturing.

Product defects computer vision architecture diagram



The following steps describe the architecture:

1. Capture images under consistent conditions with X-ray machines, cameras, and other devices.
2. Transfer product images to AWS with [AWS Transfer Family](#), [AWS DataSync](#), or [AWS IoT Greengrass](#) for edge devices.
3. Store product images in Amazon S3 separated into train and test datasets.

4. Use Amazon Lookout for Vision on the training dataset to label, train, tune, and deploy the defect detection model.
5. Expose the model through [Amazon API Gateway](#) and Lambda for admins and data scientists to manage.
6. For runtime inference, use Lambda to start the model. Use [AWS Step Functions](#) with Lambda to orchestrate a serverless workflow.
7. For batch anomaly detection, submit a batch job to [AWS Batch](#) with [AWS Fargate](#) for compute.
8. Notify users on confidence level through [Amazon Simple Notification Service](#) (Amazon SNS). If the minimum confidence threshold is not met, the user provides input to label undetected data.
9. Store final results in Amazon S3 for [Quick](#) visualization. Write results back to the manufacturing execution system (MES) on the shop floor.

Further reading

For additional information, see the following resources:

- [AWS Architecture Icons](#)
- [AWS Architecture Center](#)
- [AWS Well-Architected](#)

Diagram history

To be notified about updates to this reference architecture diagram, subscribe to the RSS feed.

Change	Description	Date
Initial publication	Reference architecture diagram first published.	October 10, 2024

RSS subscription

To subscribe to RSS updates, you must have an RSS plugin enabled for the browser that you are using.