Amazon Nova Act

AWS AI Service Cards

AWS AI Service Cards: Amazon Nova Act

Table of Contents

Amazon Nova Act

An AWS AI Service Card explains the use cases for which the service is intended, how machine learning (ML) is used by the service, and key considerations in the responsible design and use of the service. A Service Card will evolve as AWS receives customer feedback, and as the service progresses through its lifecycle. AWS recommends that customers assess the performance of any AI service on their own content for each use case they need to solve. For more information, please see AWS Responsible Use of AI Guide and the references at the end. Please also be sure to review the AWS Responsible AI Policy, AWS Acceptable Use Policy, and AWS Service Terms for the services you plan to use.

This Service Card applies to the Amazon Nova Act service available on AWS as of December 2, 2025.

Overview

Amazon Nova Act is an artificial intelligence (AI)-powered agentic system trained to perform actions on behalf of a human user within a web browser and extended beyond through API calls. Amazon Nova Act is designed for enterprise browser-based workflows focused on the following primary key use cases: (i) form-filling, (ii) search and extract, (iii) shopping and booking, and (iv) quality assurance testing. Amazon Nova Act generates automated workflows in response to natural language commands, executing tasks through browser interactions, API calls, and human escalation when needed. Not all of the content in this Service Card is applicable to the Amazon Nova Act developer tools on https://nova.amazon.com/act.

The success of an Amazon Nova Act workflow is measured by three criteria:

- The workflow is completed as specified in the natural language command;
- The task is free from errors requiring manual intervention; and
- The process adheres to safety, fairness, and reliability standards.

Customers should evaluate reliability using human judgment or automated benchmarks tailored to their workflows. They are best positioned to define and measure reliability because (i) they know what outcome best represents success their workflows, and therefore, should be included in an evaluation dataset and (ii) different workflows may respond differently to the same prompt, requiring tuning of the prompt and/or evaluation mechanism.

Like all AI systems, Amazon Nova Act is designed to understand relevant differences in commands (such as distinguishing between "fill out a job application form" and "submit an expense reimbursement request") while ignoring irrelevant variations (such as minor differences in phrasing like "click the submit button" versus "please click on the submit button").

The full set of variations encountered in natural language commands include: linguistic variations (synonyms like "complete" versus "fill out"), formatting differences (bullet points versus paragraph instructions), specificity levels (high-level goal, like "book a flight to Seattle," versus detailed step-by-step, like "navigate to airline website, select departure city as San Francisco and destination as Seattle, choose dates, and complete booking"), contextual assumptions (implicit versus explicit requirements, such as "use my saved payment method" when no method is specified), and temporal references (relative time like "next Monday" versus absolute dates).

Since different Amazon Nova Act deployment environments may interpret workflow contexts in varying ways, customers should experiment to understand how best to structure their natural language commands and validate workflow execution to achieve reliable automation results.

Intended Use Cases and Limitations

Amazon Nova Act serves a wide range of automation use cases and offers the following core capabilities:

- Natural language workflow creation: Translates intuitive commands (for example, "renew a business license") into executable workflows.
- Browser automation: Drives browser interactions (for example, form filling, navigation, and data extraction).
- Human-in-the-loop escalation: Automatically escalates to humans for unresolved scenarios or when requested.
- Tool use outside of browser (preview): Calls external tools (for example, PDF extraction and payment processing).

For technical specifications, see the Amazon Nova Act User Guide.

When assessing Amazon Nova Act for a use case, define: the business use case (for example, reducing manual processes), stakeholders (for example, developers, end users), workflow steps (for example, command input → execution → escalation), input/output types (for example, natural

language, browser actions, structured JSON), variations (for example, command phrasing), and error impacts (for example, failed renewals).

Example Use Case: Automating healthcare enrollment workflows for a benefits provider.

- Business goal: Reduce manual effort in processing health benefits applications.
- Stakeholders:
  - Developers create workflows using natural language.
  - HR teams use automated results to finalize enrollments.
- Workflow:
  1. Developer creates a script to run in Amazon Nova Act, including URL information and natural language instructions such as "Submit [Employee Name]'s health benefits application."
  2. Amazon Nova Act navigates the enrollment portal, fills out forms, extracts PDF data, and escalates if data is missing.
  3. HR reviews escalated cases.
- Inputs: Employee data, eligibility criteria, portal URLs.
- Outputs: Completed applications, error logs, escalation flags.
- Input variations: Command phrasing, portal UI changes, data formats.
- Error impacts: Incorrect submissions (high), delays (medium).

Amazon Nova Act is not intended to support prohibited practices under the EU AI Act or any other relevant law. Amazon Nova Act can be integrated into systems like enterprise automation platforms, customer service tools, or compliance management systems. Customers may not use Amazon Nova Act for surveillance, unlawful decision-making, manipulative practices, access restricted or unauthorized content, or practices against any other relevant law. For integration guidance, see the Amazon Nova Act User Guide. Amazon Nova Act use cases must comply with the AWS Acceptable Use Policy.

Amazon Nova Act has the following limitations requiring careful consideration:

(i) Appropriateness for Use

Customers should evaluate outputs for accuracy and appropriateness for their use case, especially if outputs will be directly surfaced to end users. If Amazon Nova Act is used in high-risk customer workflows (for example, healthcare and finance) that produce consequential decisions, customers must evaluate the potential risks of their use case and implement appropriate human oversight,

testing, and other use case-specific safeguards to mitigate such risks. For more information, see the AWS Responsible AI Policy.

(ii) Input/Output Constraints

Amazon Nova Act accepts natural language prompts up to approximately 10,000 characters to describe the desired action. The model supports a maximum of 100 sequential steps per task execution. Browser sessions are limited to 30 minutes, and API payloads must be under 5MB. For optimal performance, use specific, clear directions with appropriate element references where available.

(iii) Safety Filters

Amazon Nova Act includes built-in safety filtering to reject harmful prompts. Amazon Nova Act is evaluated on datasets containing unsafe prompts (for example, fraud and weapons creation) designed to train the model to reject harmful requests. Customers are responsible for end-to-end testing and validation against their specific use cases and safety requirements.

Amazon Nova Act implements multi-layered security controls, such as input filtering and training with proprietary datasets, to protect against prompt injection and other attack vectors. Browser-use agents, such as Amazon Nova Act, that operate using a visual understanding of content displayed on screen are susceptible to malicious actors using techniques such as prompt injection attacks. While we have trained Amazon Nova Act to deflect these attacks, we cannot guarantee all prompt injections attacks will be deflected. To further mitigate this risk, we recommend developers implement tooling below to further reduce their risk wherever possible and appropriate:

1. Restrict domains using a defined allow/block list of URLs via the Amazon Nova Act SDK, or as a natural language instruction within the act() statement, such as "to complete this task, do not access domains outside of example.company.com. If you find yourself on any other domains, immediately terminate the workflow and raise an error."
2. Minimize attack surfaces by only registering tools relevant for a given workflow, such as file upload/download. Note that file upload is blocked by default in the Amazon Nova Act SDK settings. Developers are recommended to allow this capability only selectively when necessary.
3. Restrict access to file:// path access unless necessary for a specific workflow. This is blocked by default in the Amazon Nova SDK so developers are recommended to allow this only selectively when necessary.

(iv) Supported Languages

Amazon Nova Act is currently optimized for English-language commands.

(v) Customization Limits

Amazon Nova Act's model is pre-trained and fine-tuned to support a variety of browser-based workflows; customers cannot fine-tune it directly.

(vi) Limited Prior Knowledge

Amazon Nova Act is trained to understand and interact with common web interfaces and UI patterns. It does not have explicit knowledge of every possible website layout, custom UI component, or proprietary application interface. The service does not maintain a comprehensive database of all web applications, their specific workflows, or real-time changes to website structures and designs. Amazon Nova Act also does not have inherent knowledge of organization-specific processes, internal tools, or custom enterprise applications that may have unique navigation patterns or specialized interfaces.

Design of Amazon Nova Act

Machine Learning

Amazon Nova Act performs browser-based task automation using machine learning, specifically, a multi-modal large language model (LLM) fine-tuned for reasoning and action generation. The core service processes natural language instructions with visual context from User Interface (UI) screenshots, generating a sequence of reasoning steps and corresponding browser actions through a ReAct (Reasoning and Action) framework. The model has been fine-tuned using human-annotated data to optimize agentic use cases, enabling it to break down complex tasks into logical steps and corresponding browser interactions. Below is the runtime service architecture for Amazon Nova Act:

1. **Initial Setup:** Users establish the target webpage/UI for automation using the Amazon Nova Act SDK;
2. **Input Reception:** The Amazon Nova Act SDK receives the user's natural language prompt describing the desired task;
3. **Context Processing:** The Amazon Nova Act SDK captures the current UI screenshot and forwards both the visual context and the user prompt to the AWS Nova Act service via invokeStep API calls;

4. **Reasoning Loop:** The Amazon Nova Act service initiates a multi-step ReAct loop. In this step, the model observes the current state (UI screenshot). The model reasons the required actions and generates specific browser automation instructions;
5. **Action Execution:** Each step output undergoes guardrail validation before being sent back to the Amazon Nova Act SDK, which translates the model's instructions into concrete browser actions using Playwright; and
6. **Task Completion:** Steps (iii)-(v) iterate continuously until the model determines the task has been successfully completed.

This architecture enables Amazon Nova Act to handle complex, multi-step browser automation tasks while maintaining safety and reliability through continuous validation and controlled execution. More information on how to use Amazon Nova Act can be found in the Nova Act Cookbook Github repo.

Controllability

Amazon Nova Act model exhibits a particular "behavior" when it generates the same kind of Think and Act outputs for the same kinds of action prompts and UI contexts. For a given model architecture, Amazon Nova Act's primary control levers over the behaviors are: (i) the training data corpus of the core reasoning model, (ii) high-quality training data generated by specialized human annotator teams, (iii) different parameters such as seed, and (iv) filters and guardrails applied to pre-process prompts and post-process outputs.

Our development process exercises the following control levers: (i) we pre-train the underlying foundation model using curated data from a variety of sources, including licensed or proprietary data, open-source datasets, and publicly available data where appropriate, (ii) we adjust model weights via supervised fine tuning (SFT) optimized for agentic use-cases to ensure the model can break down complex tasks into logical steps, generate appropriate browser automation instructions, and maintain task coherence across multiple steps, and (iii) we include safety data designed to block or evade potentially harmful prompts and outputs to further increase alignment with our design goals.

In addition to the Amazon Nova Act model, we include standard Amazon Nova model safety filters, specialized guardrails designed specifically for browser automation scenarios, and runtime safety checks for automated actions. The combination of these controls ensures that Amazon Nova Act maintains consistent, safe, and reliable behavior while performing automated browser interactions.

Performance Expectations

Generally, we expect implementations of similar browser automation use cases by different customers to vary in their inputs, their configuration parameters, and in how overall effectiveness is measured. Consider two applications A and B, each automating a customer service workflow, but deployed by different companies. Each application will face similar challenges (for example, the variations in website structures, the specificity of user prompts, and the complexity of tasks to be automated). These factors can lead to differences in (i) language used to describe tasks, (ii) the level of granularity required in step-by-step instructions, (iii) the ability to handle unexpected website changes or errors, and (iv) the speed and accuracy of task completion. These variations will result in different interaction patterns with Amazon Nova Act, leading to differing performance statistics. As a result, the overall utility of Amazon Nova Act will depend both on the model's capabilities and the specific workflow. We recommend customers test Amazon Nova Act on their own websites and use cases, considering factors such as (i) complexity and variability of their target websites, (ii) diversity of tasks they aim to automate, (iii) level of human oversight required in their processes, and (iv) desired balance between task completion speed and accuracy. By testing with different workflows and prompt styles, customers can optimize their use of Amazon Nova Act for their specific needs. This approach allows for fine-tuning of the automation process and helps set realistic performance expectations for each unique implementation.

Test Driven Methodology

Amazon Nova Act is evaluated using multiple datasets (both human-generated and synthetic) and human review. No single evaluation dataset is sufficient to completely capture performance because evaluation datasets vary based on use case, intrinsic and confounding variation, and other factors specific to browser automation scenarios. The development testing involves automated testing against publicly available and proprietary datasets of web interactions, benchmarking against proxies for anticipated customer use cases, human evaluation of outputs against proprietary datasets of complex web workflows, manual red teaming, and more. The development process examines Amazon Nova Act's performance through such testing, takes steps to improve the model and/or the suite of evaluation datasets, and then iterates.

Human Evaluation: Human evaluation is a critical step in evaluating the model's outputs. For example, the effectiveness criteria for automating an e-commerce checkout process could differ from the criteria for a content management workflow. An e-commerce site owner might care more about accurate product selection and cart management, while a content manager might focus on precise data entry and formatting. Using human judgment is critical for assessing the effectiveness of a browser automation model on more challenging tasks because only humans can fully understand the context, intent, and nuances of more complex prompts and the resulting browser interactions.

Independent Red Teaming Network: Consistent with our Frontier AI Safety Commitments on ensuring Safe, Secure, and Trustworthy AI, we partner with skilled evaluators to conduct red teaming against our AI models. Amazon Nova Act leverages red teaming firms to complement our in-house testing in areas such as safety, security, privacy, fairness, and veracity-related topics. We also work with specialized firms to red team our models for areas specific to browser automation, such as web application security and data privacy compliance. The goal of red teaming is to continuously explore more use cases and prompt variations to ensure Amazon Nova Act maintains high standards of safety and reliability.

Child Sexual Abuse Material (CSAM)

To help prevent potential misuse, Amazon Bedrock implements automated abuse detection mechanisms. These mechanisms are fully automated, so there is no human review of, or access to, user inputs or model outputs. To learn more, see Amazon Bedrock Abuse Detection in the Amazon Bedrock User Guide.

Amazon Nova Act utilizes Amazon Bedrock's Abuse Detection solution, which uses hash matching or classifiers to detect potential CSAM. If Amazon Bedrock detects apparent CSAM in Amazon Nova Act's input prompt, it will block the request, display an automated error message, and may also file a report with the National Center for Missing and Exploited Children (NCMEC) or a relevant authority. We take CSAM commitments seriously and will continue to update our detection, blocking, and reporting mechanisms.

Safety

Safety is a shared responsibility between AWS and our customers. Our goal for safety is to mitigate key risks of concern to our enterprise customers, and to society more broadly. We align the behaviors of our foundation models with internal design policies and our commitment to responsible AI development practices. Amazon Nova Act is designed to prevent the generation of harmful content, including content that may cause physical or emotional harm, and content that may harass, harm, or encourage harm to individuals or specific groups, especially children. Amazon is committed to producing generative AI services that keep child safety at the forefront of development, deployment and operation. We test and implement safeguards to prevent Amazon Nova Act from executing tasks that could generate, access, or distribute inappropriate content related to children. Amazon Nova Act is designed to block the agent from visiting malicious websites unknowingly or conducting dangerous tasks. For example, if Amazon Nova Act is asked to perform a task on a suspected phishing website, the system is designed to refuse to visit the website and display an error.

Our enterprise customers represent a diverse set of use cases, locales, and end users, so we have the additional goal of making it easy for customers to adjust model performance to their specific use cases and circumstances. AWS offers services and tools to help customers identify and mitigate safety risks, such as Amazon Bedrock Guardrails and Amazon Bedrock Model Evaluations. Customers are responsible for end-to-end testing of their applications on datasets representative of their use cases and any additional safety mitigations, and deciding if test results meet their specific expectations of safety, fairness, and other properties, as well as overall effectiveness.

We evaluate Amazon Nova Act's capability to reject potentially harmful prompts using multiple datasets. For example, on a proprietary dataset containing prompts to perform unsafe activities (for example, perform fraudulent activities or aid in the creation of harmful substances or weapons), Amazon Nova Act correctly blocks 96.4% of harmful prompts. We recommend Amazon Nova Act users (i) restrict domains using a defined allow/block list of URLs via the SDK, (ii) only register tools relevant for a given workflow, such as file upload/download, and (iii) restrict access to file:// path to only those necessary for a workflow.

Fairness

Amazon Nova Act is designed to avoid performing activities that generate and proliferate content related to stereotypes or make bias comments about specific groups of people, roles, or behaviors. Amazon Nova Act is designed to deflect when asked to perform tasks such as generating or promoting content related to stereotypes. We test Amazon Nova Act's ability to refuse tasks that generate content associated with stereotypes with a proprietary dataset of text prompts (for example, leave inflammatory comments on social media, share content that promotes bias content). We find that Amazon Nova Act correctly blocks 99.5% of prompts.

Reliability

Amazon Nova Act optimizes for reliability (high success rate for a supported workflow) by using training data that capture different scenarios across a variety of web environments and the use of Reinforcement Learning (RL) to encourage our model to explore an environment and to recover from an unexpected trajectory. While customers can expect that the same prompt will perform similar actions, they should not expect the same prompt to always generate identical actions, as there are external factors that may cause the Agent to perform different actions (for example, pop-up screens appearing at a different step of the task due to latency variation) to accomplish the requested task. Customers should use prompt engineering to ensure the input to the model is clear and resilient to variations in the web environment. Amazon Nova Act evaluation scores are reported as the average of multiple runs (typically 5).

Privacy

Prompts and outputs are never shared between customers. Separate terms apply for the Amazon Nova Act developer tools using API key authentication on https://nova.amazon.com/act. AWS does not use inputs or outputs generated through the managed service to train or improve Amazon Nova Act. For more information, see Section 50.3 of the AWS Service Terms and the AWS Data Privacy FAQs. If a user is concerned that their personal information has been included in an Amazon Nova Act output, the user should contact us here.

Security

Amazon Nova Act comes with enterprise security that enables customers to build browser automation applications. Customers can use AWS Identity and Access Management (IAM) to securely control access to the Amazon Nova Act service. The Nova Act SDK can be configured with granular permissions for browser automation activities. Amazon Nova Act offers comprehensive monitoring and logging capabilities that can support customer governance and audit requirements specific to browser automation. For example, Amazon CloudWatch can help track automation metrics and interaction patterns required for audit purposes, and AWS CloudTrail can help monitor API activity and troubleshoot issues as Amazon Nova Act integrates with other AWS systems. Customers can also choose to store their automation metadata, prompts, and interaction logs in their own encrypted Amazon Simple Storage Service (Amazon S3) bucket. For more information, see Amazon Nova Act Security.

Intellectual Property

Amazon Nova Act is designed for performing UI-oriented tasks. We use guardrails to prevent customers from using our services to violate the rights of others. AWS offers uncapped intellectual property (IP) indemnity coverage for outputs of Indemnified Generative AI Services (see Section 50.10 of the AWS Service Terms). This means that customers are protected from third-party claims alleging IP infringement or misappropriation (including copyright claims) by the outputs generated by Amazon Nova Act. In addition, our standard IP indemnity for use of the Services protects customers from third-party claims alleging IP infringement (including copyright claims) by the Services (including Amazon Nova Act) and the data used to train them.

Output Control: Amazon Nova Act targets UI interaction instead of content creation. However, it leverages creative content control from the Amazon Nova family of models to control output that may violate intellectual property rules.

Transparency

Amazon Nova Act provides information to customers in the following locations: this Service Card, AWS documentation, AWS educational channels (for example, blogs and developer classes), AWS Console, and SDK developer documentation on Github. We accept feedback through customer support mechanisms such as account managers. Where appropriate for their use case, customers who incorporate Amazon Nova Act in their workflow should consider disclosing their use of ML to end users and other individuals impacted by the application, and customers should give their end users the ability to provide feedback to improve workflows. In their documentation, customers can also reference this Service Card.

Governance

We have rigorous methodologies to build our AWS AI services responsibly, including a working backwards product development process that incorporates Responsible AI at the design phase, design consultations, and implementation assessments by dedicated Responsible AI science and data experts, routine testing, reviews with customers, best practice development, dissemination, and training.

Deployment and performance optimization best practices

We encourage customers to build and operate their applications responsibly, as described in AWS Responsible Use of AI Guide. This includes implementing Responsible AI practices to address key dimensions including controllability, safety, fairness, veracity, robustness, explainability, privacy, security, transparency, and governance.

Workflow Design

The performance of any application using Amazon Nova Act depends on the design of the customer workflow, including the factors discussed below:

- **Effectiveness Criteria:** Customers should define and enforce criteria for the kinds of automation use cases they will implement, and for each use case, further define criteria for the inputs and outputs permitted and how humans should monitor and validate automation results. These criteria should systematically address controllability, safety, privacy, and key dimensions listed above.

- **Configuration:** In addition to the required natural language instructions, Amazon Nova Act has various required and optional configuration parameters to help customers achieve optimal automation results. For more information, see Amazon Nova Act User Guide.
- **Prompt Engineering:** Prompt engineering refers to the practice of optimizing the natural language inputs needed to achieve browser automation tasks. High-quality prompts lead to more reliable automation outcomes. Some key aspects of prompt engineering include word choice, specificity, structure, and error handling guidance:
  - Prompt style and clarity: To get the best results, instructions should be specific and structured ('navigate to the product page and add the blue shirt in medium size to the cart'), not vague or conversational ('I want to buy a blue shirt'). Effective instructions should break down complex tasks into clear steps to increase reliability, specify expected outcomes, and include error handling preferences.
  - Task scoping: When crafting prompts, customers should experiment with a variety of prompts. Using a prompt which covers broader use cases can help the workflow be robust to changes in the environment (for example, website), but this may reduce reliability by creating a more ambiguous task for the model.
  - Error Handling: Amazon Nova Act performs better when instructions include explicit error handling guidance. For example, 'If the size is not available, select the next available size up and notify the user' instead of just 'select size medium'. This type of explicit error handling helps ensure graceful handling of edge cases and unexpected situations.
- **Base Model Customization:** We do not support customization of the underlying base model, but if you have any issues solving your use case with the model, please reach out to our team here.
- **Human Oversight:** If a customer's automation workflow involves high risk or sensitive operations (for example, financial transactions or data submission), human validation should be incorporated into appropriate checkpoint stages of the workflow where appropriate. This ensures proper oversight of critical operations while maintaining the efficiency benefits of automation. Amazon Nova Act is designed to trigger human oversight (via Human in the loop functionality) when requested.
- **Performance Monitoring:** Amazon Nova Act is robust to website changes, though there may be scenarios where website layout, content, or interaction patterns might affect Amazon Nova Act's performance. Regular monitoring of automation success rates and testing of critical paths via evaluation sets is essential. Customers should actively monitor automation failures and do the necessary prompt changes to bring the workflow back to a successful state.
- **Model Updates:** When we release new versions of Amazon Nova Act, we will notify customers and will provide customers with time to migrate from an old version to the new one. Customers

should test new versions in a staging environment, validate critical automation workflows, update prompts if needed, and review error handling procedures before deploying to production.

Further information

- For service documentation, see Amazon Nova Act User Guide.
- For details on privacy and other legal considerations, see AWS's Acceptable Use Policy, Responsible AI Policy, Legal, Compliance, Privacy.
- For help optimizing a workflow, see Generative AI Innovation Center, AWS Customer Support, AWS Professional Services, AWS Well-Architected.
- For other tools to help customers work with foundation models, see Amazon Bedrock, Amazon Bedrock Guardrails, and Amazon Bedrock Guardrails automated reasoning checks.
- If you have any questions or feedback about AWS AI Service Cards, please complete this form.

Glossary

Controllability: Steering system behavior to reflect system design goals

Privacy & Security: Appropriately obtaining, using and protecting data and models

Safety: Preventing harmful system output and misuse

Fairness: Considering impacts on different groups of stakeholders

Explainability: Understanding and evaluating system outputs

Veracity & Robustness: Achieving correct system outputs, even with unexpected or adversarial inputs

Transparency: Enabling stakeholders to make informed choices about their engagement with an AI system

Governance: Embedding best practices within the AI supply chain, including providers and deployers