aws

COMMUNITY DAY

Security Guardrails for the Autonomous Development Era

Ana Shah | August 22, 2026

aws
COMMUNITY DAY

Ana Shah

Cybersecurity engineer
Previous careers in politics, finance, and education
Former high school and college cybersecurity teacher
Tulsa Remoter

ML | AI

Jim’s Gym

The Shadow AI Clause

Skipping the diagramming discipline doesn't just create risk - it can void your coverage

Shadow AI - AI tools employees use without IT or security approval, including AI features built into apps already sanctioned, like Slack, Notion, or Copilot.

40%

of cyber-insurance claims now denied

missing AI governance is a new reason why

55%

of employees use AI tools their org never approved

37%

of organizations have no policy to detect it

$670K

added to breach cost by unauthorized shadow AI

The fix pays for itself: documented AI governance -> 40-60% lower premiums

From Traditional to Autonomous Development

THE AUTONOMY SPECTRUM

Autocomplete
Suggests the next line

Chat-Assisted
"Vibe coding" - you steer every step

Agentic
Works a task, checks in

Fully Autonomous
Runs for hours, unsupervised

	PLAN	WRITE	TEST	REVIEW (PR)	DEPLOY
TRADITIONAL Human drives every step	**Reads ticket** By hand	**Writes code** File by file	**Writes tests** Manually	**Opens a PR** Peer reviews it	**Deploys** Manual/script
	Threat model	● **Secure code**	● **Dep scan**	● **Sec review**	● **Sec gate**
AUTONOMOUS Agent drives the workflow	**Parses task** On its own	**Writes code** Machine speed	**Runs tests** Fixes failures	**Opens a PR** Human reviews Gated	**Deploys** Still gated
	○ **No threat model**	○ **Not reviewed**	○ **Maybe scanned**	○ **No sign-off**	○ **Gate missing**

What Is a Threat Model?

The questions you answer before writing a line of code - not the scan you run after

THE FIVE QUESTIONS

1. **What are we protecting?**
   Data, systems, users
2. **Who might attack it, and why?**
   Opportunistic, targeted, or insider
3. **How could they get in?**
   Every entry point - APIs, uploads, integrations
4. **What's the worst case if they succeed?**
   Blast radius, not just point of entry
5. **What controls reduce that risk?**
   A design decision, not a patch applied later

STRIDE - A QUICK FRAMEWORK

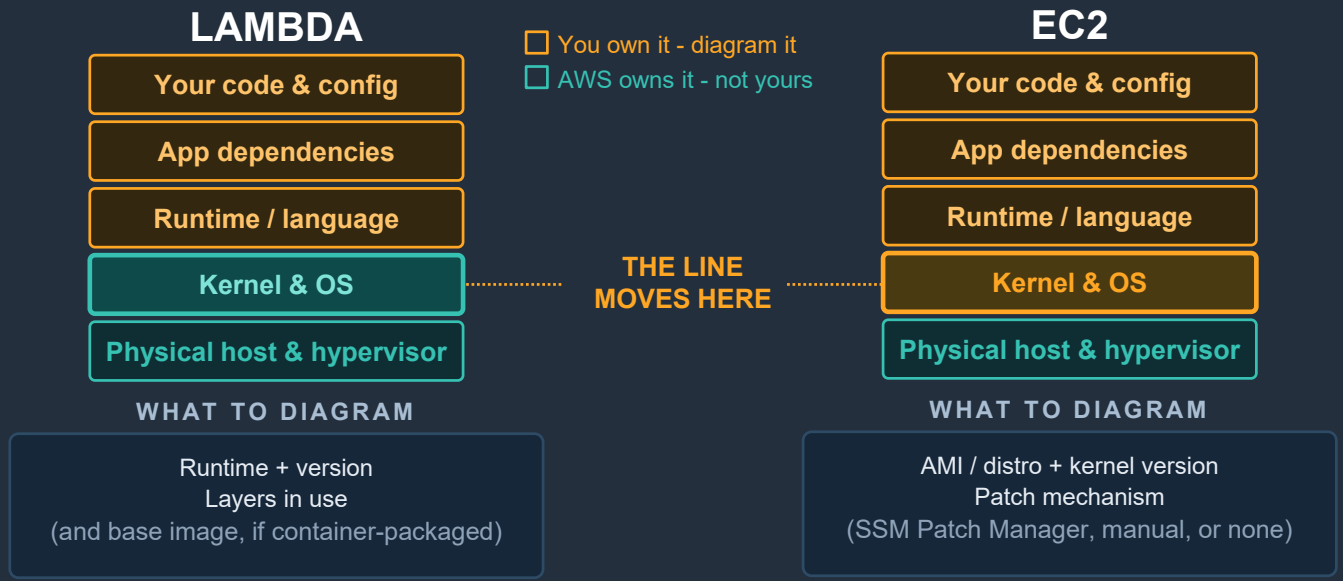
- **S** **Spoofing**
  Pretending to be someone else
- **T** **Tampering**
  Modifying data without authorization
- **R** **Repudiation**
  Denying an action, with no proof either way
- **I** **Information disclosure**
  Exposing data that shouldn't be exposed
- **D** **Denial of service**
  Making the system unavailable
- **E** **Elevation of privilege**
  Gaining more access than intended

Catch it at planning - that's a conversation.

Catch it after deployment - that's an incident.

What To Diagram: The AWS Shared Responsibility Model

Lambda vs. EC2 - where does responsibility lie?



If you can SSH into it or choose its base image - diagram it.

If AWS spins it up and tears it down for you - track the runtime version instead.

Sample Lambda diagram and location

Sample EC2 diagram and location

Where To Actually Build It

Two tools that get out of your way - pick based on what your team already uses

draw.io

drawio.com

- Free and browser-based - no account needed
- Built-in AWS architecture icon library
- Exports to PNG, SVG, or editable XML
- Good fit for solo builders and small teams

BEST FOR

Fast iteration, personal projects, and teams that don't want a license fee

Microsoft Visio

Part of Microsoft 365 / Office

- Deep AWS and Azure stencil libraries
- Native integration with Microsoft 365 and Teams
- Built-in version history and co-authoring
- Enterprise governance - permissions, audit trail

BEST FOR

Larger orgs already in the Microsoft ecosystem, and formal documentation needs

The tool doesn't matter as much as doing it.

Pick whichever one you'll keep updated.

The Diagram Is the inventory of your product.

Once something's on the diagram, here's where you go to check if an update exists.

OS (Operating System) / Kernel

Ubuntu Security Notices, Amazon Linux Security Bulletins

Runtime / language

Node.js, Python, and Java security release pages

App dependencies

npm audit, pip-audit, GitHub Dependabot alerts

General CVE (Common Vulnerabilities and Exposures) lookup

nvd.nist.gov, cve.org

AWS services

AWS Security Bulletins

Cybersecurity is intrusive –
not disruptive.

Diligent layering of
security is key.