Production-Grade AgentCore: Deploying Multi-Agent Systems at Scale

- **Session Hosts & Presenters:** Hosted by Midhat and Akanksha; presented by Rishabh Nagar, Bhanuendra Singh, and Pankaj Fuller.
- **Core Topic & Concept:** Demonstrating how to transition from traditional, manual root cause analysis (RCA) during production outages to automated, production-grade agentic workflows using AWS Amazon Bedrock Agent Core, custom tools, and strict cedar policy guardrails.

The Problem: Production Downtime & Traditional RCA

- **Engineering Pain Point:** When critical production deployments fail (e.g., checkout services failing during online shopping operations), engineers usually spend hours manually digging through CloudWatch logs, metric dashboards, and status codes to determine root causes under high pressure.

- **Objective:** Replace guesswork and manual log scrolling with an AI Investigator agent capable of querying live environment state, analyzing telemetry data, identifying structural bugs, and compiling formal Root Cause Analysis (RCA) reports for stakeholders in minutes.

Core Building Blocks of Amazon Bedrock Agent Core

1. **Agent Core Runtime:** A serverless environment hosted by AWS that provides execution runtimes, gateway management, identity policies, and observability for autonomous agents.
2. **Bedrock Model Catalog (LLM Engine):** Serves as the central intelligence (demonstrated using Claude 3.5 Sonnet) via a unified API framework for intent recognition and tool routing.
3. **Governed Tool Contracts:** Encapsulated tools that allow the agent to safely read logs, check infrastructure health, and pull trace metrics without granting unrestricted database or execution permissions.

Live Demonstration: Investigating Checkout Failure

- **Scenario:** An online shopping checkout service fails following a deployment by a cloud engineer.
- **Execution Steps:**

1. **Health Verification:** The agent queries container state using a Service Health Tool, confirming the underlying ECS containers are active and healthy.
2. **Log & Trace Analysis:** Using an Error Logs Tool and Recent Deployment Tool, the agent identifies a synthetic timeout caused by an invalid currency configuration value (an unsupported currency pushed in the code release instead of approved USD, EUR, or INR).
3. **Runbook Resolution:** The agent cross-references internal runbooks to outline exact remediation steps.
4. **Automated RCA Generation:** The agent drafts a complete, executive-ready Root Cause Analysis document detailing the incident timeline, cause, and remediation steps for direct customer communication.

- **Cedar Policies:** Implemented directly at the Agent Core Gateway layer to evaluate every inbound request, requiring explicit permission matches before invoking downstream tools.
- **IAM Task Roles:** Enforces strict principal boundaries so that agents cannot modify or delete production resources beyond read-only diagnostic actions.

- **AI DLC (AI-Driven Development Life Cycle):** Structured across three distinct execution phases:

    - **Inception:** Requirement gathering, guardrail definition, and success criteria setup.
    - **Construction:** Agent build, automated tool generation, and mandatory human-in-the-loop approval.
    - **Operation:** Deployment, live telemetry monitoring, and controlled execution.