AI-Driven SDLC (AIDLC): Orchestrating Autonomous Software Development

- **Session Hosts & Presenters:** Hosted by Akanksha and Midhat; presented by Rajat Khandelwal and Nitesh (Cloud for Good).
- **Core Topic & Concept:** Orchestrating an autonomous AI-driven Software Development Life Cycle (SDLC) using custom Python agents, Claude API, Jira, GitHub, and Salesforce, integrated with explicit human review gates.

Industry Pain Points & Solution Overview

- **Architect Overhead:** Senior system and solution architects spend disproportionate time on repetitive component creation rather than high-level architecture design.
- **Backlog & Quality Variance:** Development backlogs continuously grow under tight deadlines, while code quality fluctuates based on individual developer habits and mindsets.
- **Custom Orchestrator Solution:** Centered around a custom Python application using Claude as the intelligence engine. It interfaces between Jira requirements, GitHub repositories, and target deployment environments (e.g., Salesforce orgs).

End-to-End AIDLC Workflow & Live Demonstration

1. **Jira Ingestion & Design Generation:** The AI agent monitors Jira tickets transitioned from "To-Do" to "Ready for AI", extracts requirement specifications, reviews existing codebase patterns to prevent hallucinations, and generates a detailed system architecture design.

2. **Human Gate 1 (Design Approval):** A reviewer evaluates the AI-generated design document. Upon shifting the ticket status to "Design Approved", the agent moves to execution.

3. **Automated Building & Testing:** The AI agent implements code changes, deploys them to the target environment (e.g., creating Lightning Web Components in Salesforce), conducts unit tests, and attaches test reports directly to Jira.

4. **Human Gate 2 (Test Approval):** A reviewer verifies test execution and coverage. Setting the status to "Test Approved" triggers pull request creation.

5. **PR Creation & Merging:** The agent uses GitHub APIs to automatically submit a Pull Request complete with Jira cross-references, test summaries, and code diffs for final developer review.

Impact & Takeaways

- **Faster Time-to-Value:** Reduces feature delivery cycles from 2 –– 3 working days down to minutes per phase.

- **Governed Autonomy:** Ensures strict human control at key decision points (design and post-test gates) while automating repetitive labor.

- **Scalable Senior Talent:** Senior engineers transition into strategic decision-makers and gatekeepers rather than manual coders.

- **Language & Tech Agnostic:** Built to support cross-platform development beyond Salesforce across diverse tech stacks.