这份讲座由 HashiCorp 的开发者倡导者 **Rosemary Wang** 主讲，主题是**“平台工程师的 AI 自动化之旅” (A Platform Engineer’s Journey to AI Automation)**。

与前两个侧重于工具（LangFlow, Watsonx）的演示不同，Rosemary 从一个**基础设施和运维（Ops）**的角度，分享了她如何利用 AI 来自动化工作，以及作为平台工程师在构建 AI 平台时遇到的真实挑战。

以下是讲座的核心内容总结：

1. AI 自动化的两个视角

Rosemary 提出在企业中看待 AI 的两种方式：

- **AI 助力生产力 (AI for Productivity)**：个人开发者使用 AI 来加速任务（如生成代码、写脚本）。
- **AI 作为平台产品 (AI as a Platform Product)**：平台团队为整个组织提供标准化的 AI 能力。这不仅仅是给一个 Prompt，还涉及到安全性、模型托管、上下文管理（Context）和基础设施。

2. 实战案例：用 AI 写书（第二版）

为了演示这些概念，Rosemary 分享了她自己的项目：使用 AI Agent 帮她写书（《Infrastructure as Code》第二版）。

- **挑战**：出版商要求将代码示例从 Python/AWS CloudFormation 转换为 **Terraform**，但她没有时间，也没有预算使用昂贵的云端 LLM。
- **技术栈 (The Stack)**：
    - **编排工具**：LangFlow（低代码，用于构建 Agent 流）。
    - **模型**：Ollama（本地运行模型，节省成本）。
    - **数据处理**：Docling（用于处理第一版书的 PDF 文档）。
    - **向量数据库**：OpenSearch Serverless（用于 RAG）。
    - **工具集成**：**MCP (Model Context Protocol)** 服务器（连接 Terraform 和 GitHub）。
    - **基础设施管理**：Kubernetes (EKS) 和 Terraform。

3. 智能体工作流设计

她构建了一个双 Agent 系统：

1. **Agent 1 (Book Writer)**：负责生成章节的核心概念和解释性文本。
2. **Agent 2 (Terraform Refiner)**：负责将概念转化为具体的 Terraform 代码示例。它使用 Terraform MCP 工具来查找正确的资源定义，而不是依靠模型可能过时的训练数据。

4. “兔子洞”与教训 (The Rabbit Hole & Lessons)

Rosemary 强调，从简单的 Prompt 到构建企业级 AI 平台，会遇到许多意想不到的坑：

- 安全性与权限 **(Security)**：
  - 她发现默认的 IAM 策略往往过于宽泛（例如给了写权限，而实际上只需要读权限）。
  - **教训**：必须实施最小权限原则，并在基础设施层面（如 Kubernetes 和 AWS IAM）进行严格控制。
- 网络连接 **(Connectivity)**：
  - 连接私有 Kubernetes 集群和公共 Serverless 服务（如 OpenSearch）非常痛苦，需要复杂的网络配置（VPC Endpoints 等）。
- 人工干预 **(Human-in-the-loop)**：
  - AI 不是“设置后即不管”的。她发现 AI 会产生幻觉（例如混淆 Google Cloud 和 AWS 的配置）。
  - **核心观点**：**“自动化需要干预” (Automation with Intervention)**。必须有人类进行审查（Review）、微调（Refine）和反思（Rethink）。

5. 总结观点

- **自动化现有的自动化**：AI 自动化最适合用于那些已经标准化、文档化且可观测的系统。
- **AI 平台工程**：构建 AI 平台不仅是关于模型，更是关于如何安全、可靠地交付这些模型（Ops, Security, Platform 的交集）。
- **基础设施即代码 (IaC)**：她最终使用 Terraform 来部署整个 AI 栈（包括向量库、集群等），证明了传统运维工具在 AI 时代依然至关重要。

Rosemary 的演讲非常“接地气”，展示了平台工程师在面对 AI 浪潮时，如何利用现有的技能（如 Terraform, Kubernetes）来驾驭新技术，同时也坦诚地分享了过程中的挫折和调试经历。