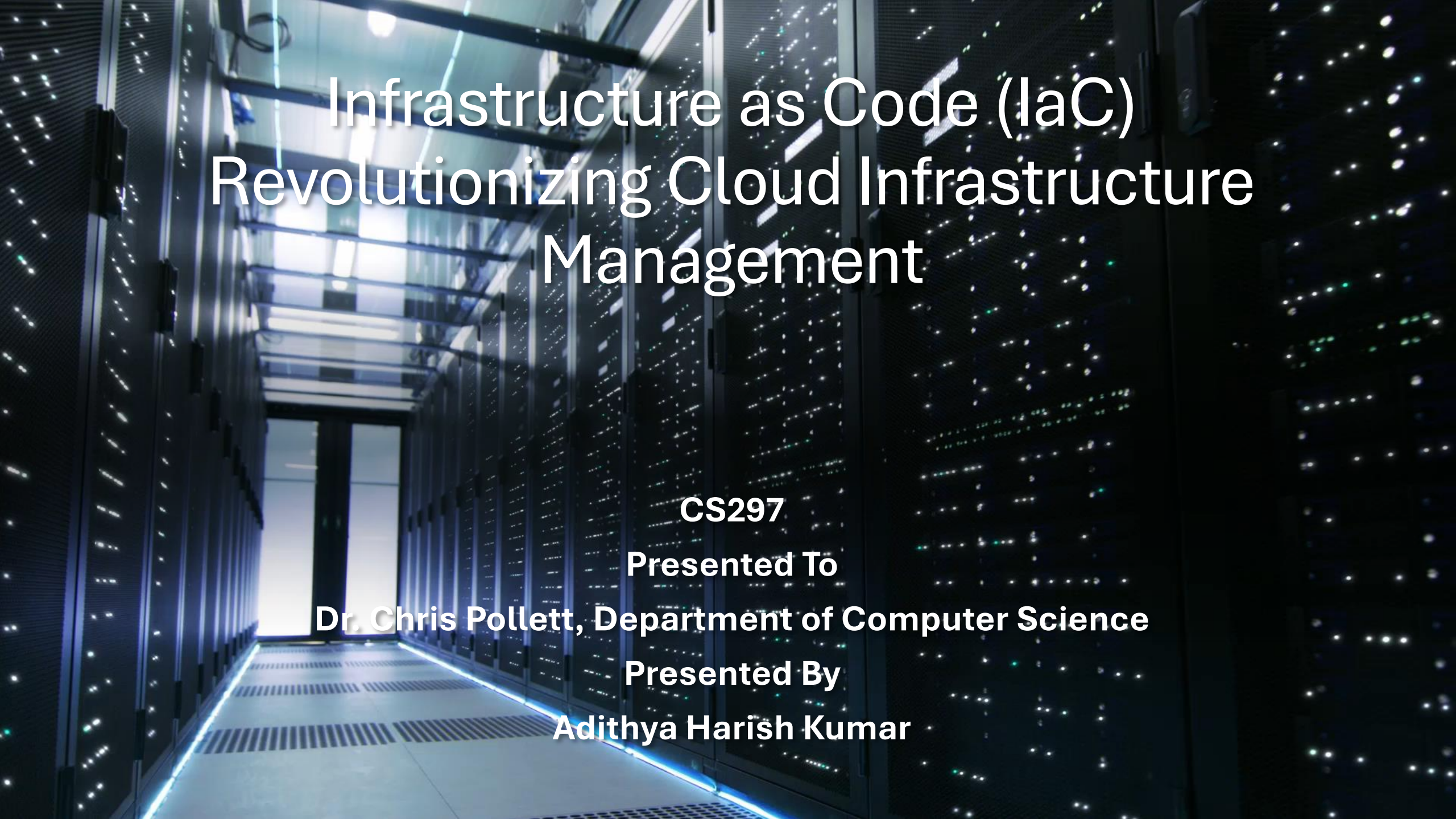


Infrastructure as Code (IaC) Revolutionizing Cloud Infrastructure Management

CS297

Presented To

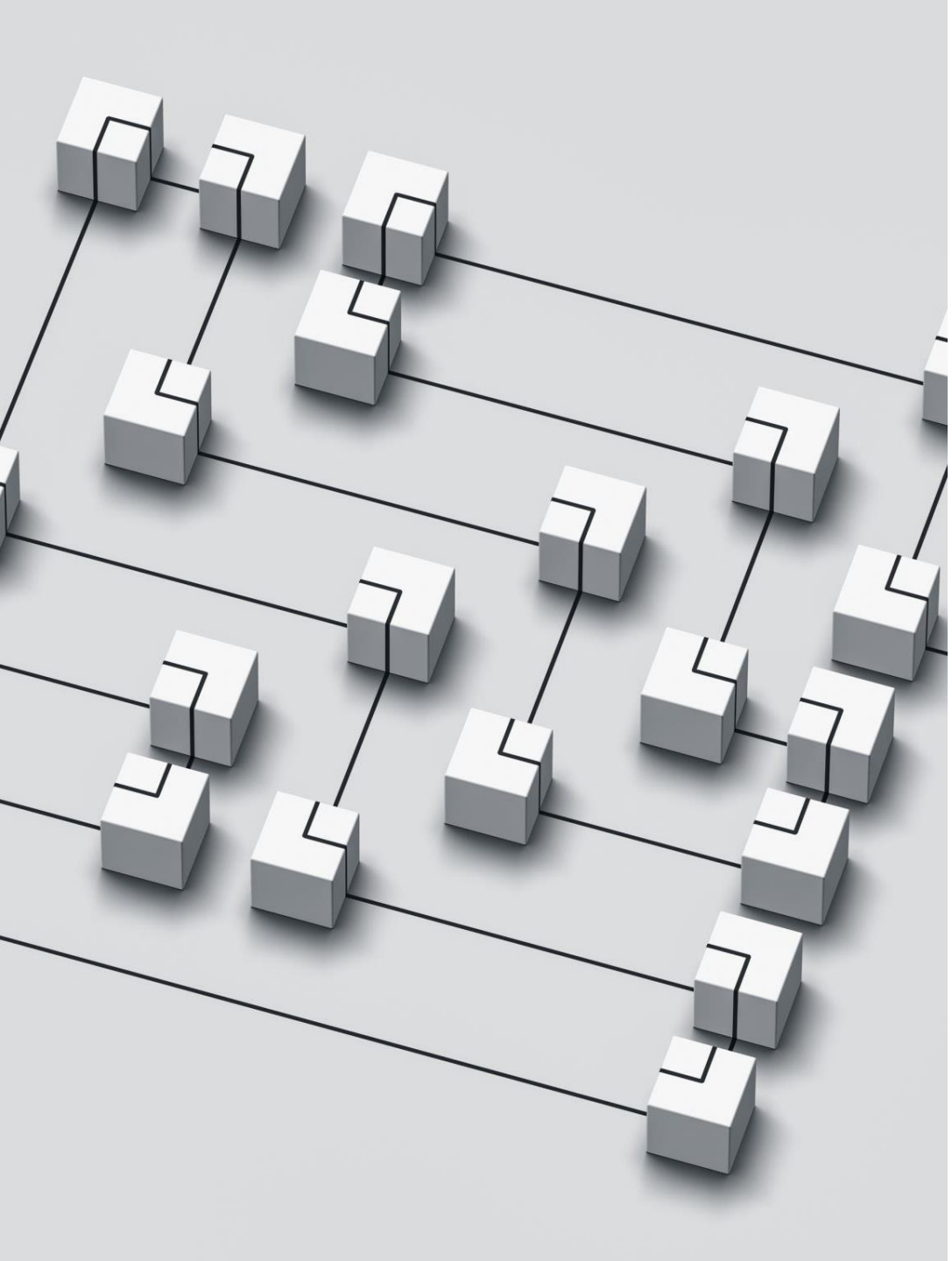
Dr. Chris Pollett, Department of Computer Science

Presented By

Adithya Harish Kumar

Introduction

- **Definition:** The process of managing and provisioning computing infrastructure through machine-readable configuration files, rather than physical hardware or interactive configuration tools
- **Purpose:** Simplifies infrastructure setup, increases consistency, and enables automation
- **Examples:** AWS CloudFormation, Terraform, Ansible
- **Idempotency:** Ensures that applying the configuration multiple times results in the same infrastructure state.



Imperative IaC:

- Describes **how** the infrastructure should be configured.
- Step-by-step instructions for the configuration of resources.
- Example tools: Ansible (can be imperative).
- **Analogy:** Like writing a script to follow specific steps.

Declarative IaC:

- Describes **what** the infrastructure should look like without specifying how to achieve it.
- The tool ensures the desired state is achieved.
- Example tools: Terraform, AWS CloudFormation.
- **Analogy:** Like writing a plan and letting the tool figure out the steps.

Aspect	Imperative	Declarative
How vs What	Describes how to do things	Describes what the end state should be
Granularity	Step-by-step execution	Desired final outcome
Flexibility	More flexible in execution	Tool decides how to achieve the state
Examples	Ansible (can be both), Bash scripts	Terraform, CloudFormation

Imperative vs Declarative

```ini
[server]
port = 8080
hostname = localhost

[database]
user = admin
password = password123
```

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: mypod
spec:
  containers:
    - name: mycontainer
      image: nginx
```

```json
    "aws": {
      "region": "us-west-2"
    }
  },
  "resource": {
    "aws_instance": {
      "my_instance": {
        "ami": "ami-0abcdef1234567890",
        "instance_type": "t2.micro"
      }
    }
```

Configuration Files

- Configuration files separate *logic* from *settings*, allowing applications to be flexible and customizable
- They control system variables, define how services should operate, or set parameters for applications, such as databases, networks, user preferences among others
- Common formats include:
  - **YAML (.yml/.yaml)**: Used in tools like Ansible, Docker Compose, and Kubernetes.
  - **JSON (.json)**: Common for APIs, web apps, and tools like Terraform.
  - **INI (.ini)**: Simple key-value pairs, often used for local application settings.