

# **AWSGoat : A Damn Vulnerable AWS Infrastructure**

# About US

## Jeswin Mathai

- Chief Architect, Lab Platform @ INE
- Published Research at Black Hat US/Asia Arsenal, DEF CON USA/China Demolabs
- Gave research talk at DEF CON China and Rootcon Philippines
- Co-Trainer in Training: Black Hat Asia, HITB AMS, GSEC NZ OWASP day, Rootcon 13

# About Us

## Shantanu Kale

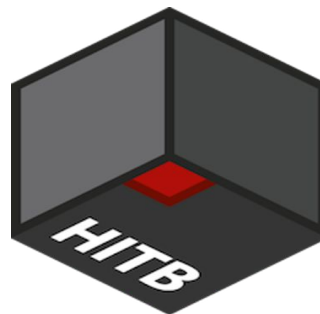
- Cloud Developer @ INE
- Published Research at Black Hat US/Asia Arsenal and DEFCON 30 DemoLabs
- Co-trainer in training at Seaside's Goa
- Strong roots in cloud and network penetration testing, vulnerability scanning, and Open Source Intelligence Techniques

## Sanjeev Mahunta

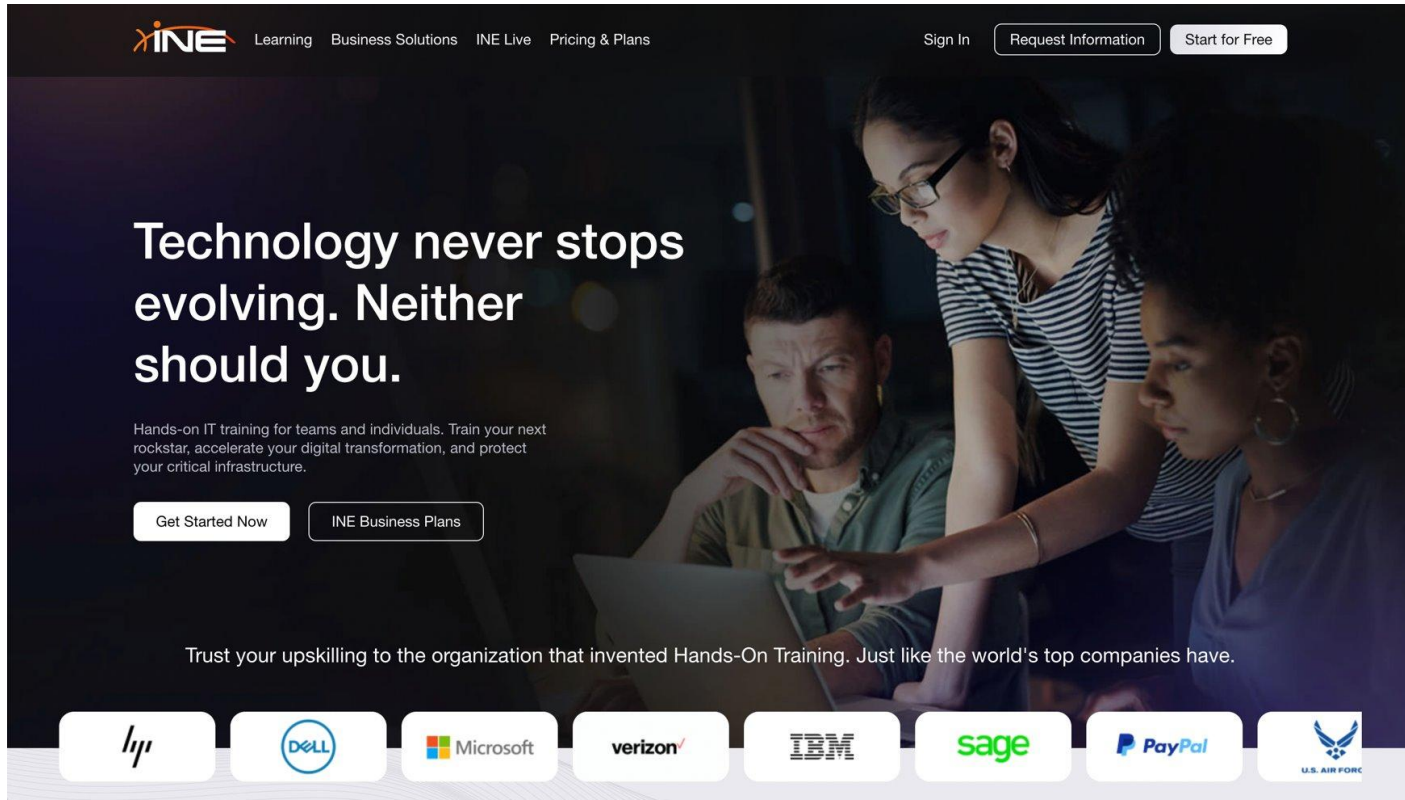
- Cloud Software Engineer @ INE
- Published research at Blackhat US Arsenal, DEF CON USA Demolabs.
- 2+ years of experience building front-end applications for the web and implementing ERP solutions
- Interned at Defence Research and Development Organisation (DRDO)



# Conferences



# About INE











**Technology never stops evolving. Neither should you.**

Hands-on IT training for teams and individuals. Train your next rockstar, accelerate your digital transformation, and protect your critical infrastructure.

[Get Started Now](#) [INE Business Plans](#)

Trust your upskilling to the organization that invented Hands-On Training. Just like the world's top companies have.

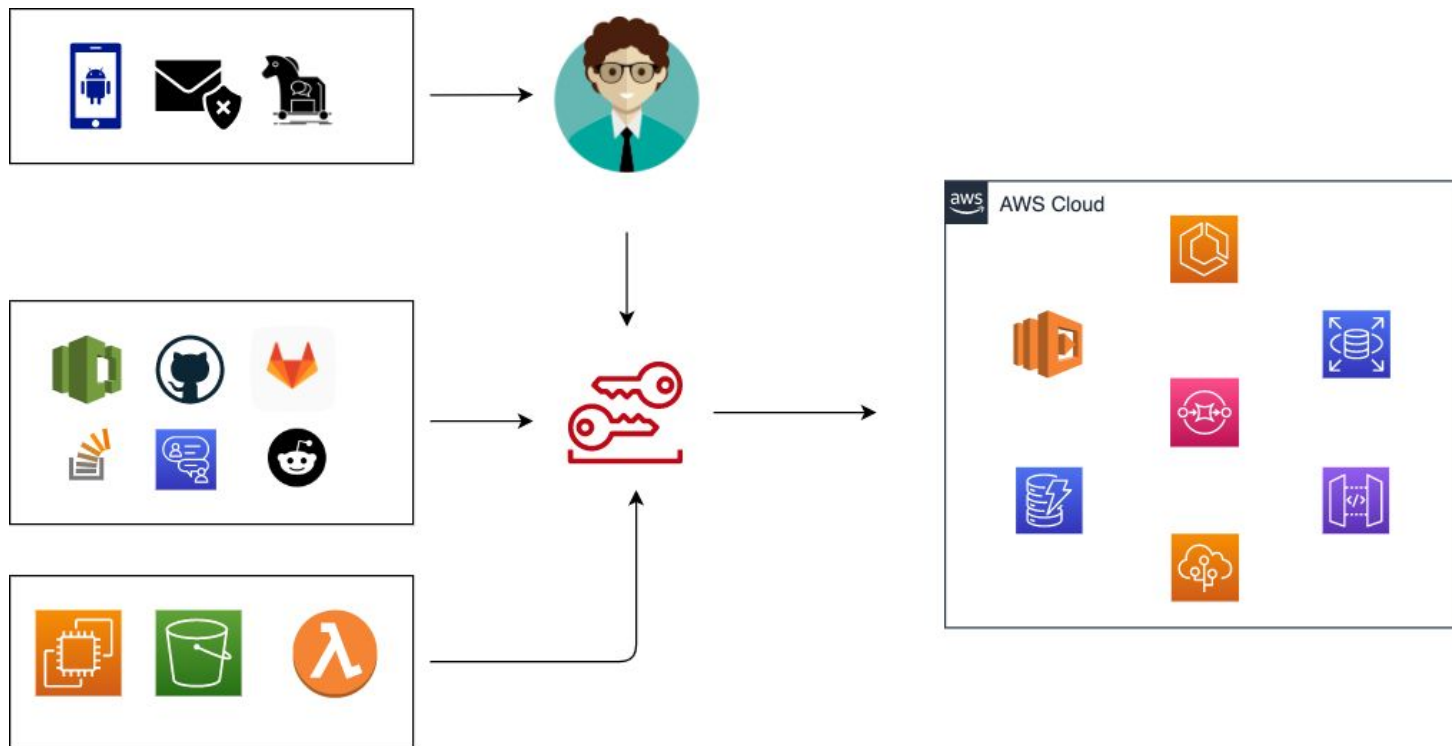




# Threatscape



# Threatscape



# The Motivation

- Training Needs
  - Basics and Fundamentals
  - Enumeration techniques
  - Abusing IAM, S3, API Gateway Misconfigurations
  - Attack vectors on Lambda and EC2
  - What Next?
- Lack of Real World AWS Pentesting Environment
- Contribution from the open source community and security professionals
- Release of OWASP Top 10: 2021



# Introducing AWSGoat!



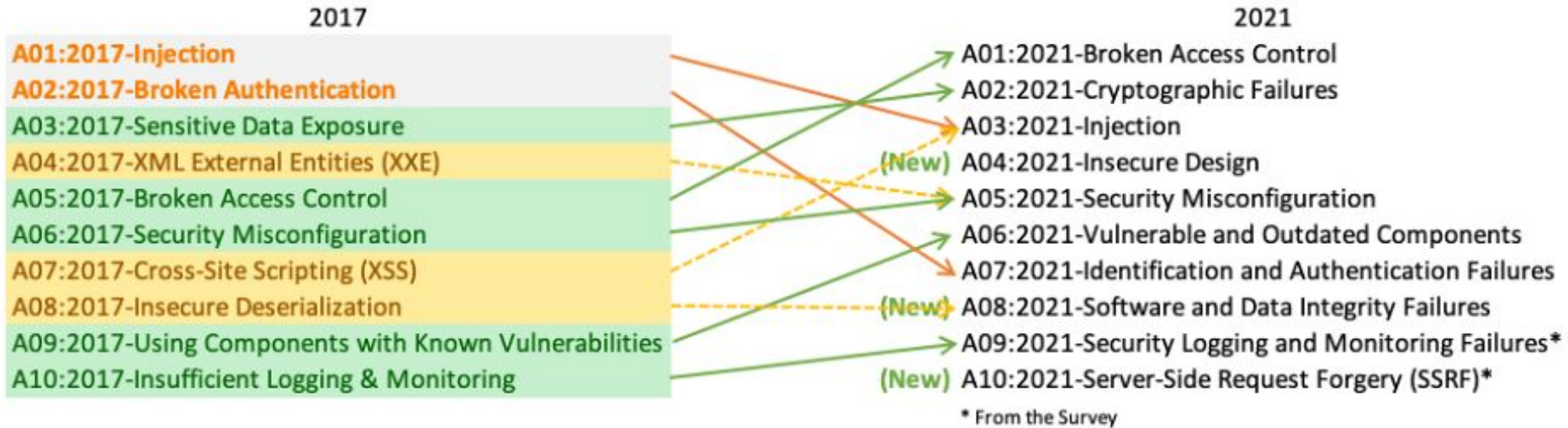
**AWS**Goat

# AWSGoat : A Damn Vulnerable AWS Infrastructure

- Mimics real-world infrastructure but with added vulnerabilities
- Multiple application stacks - Multiple exploitation/escalation paths
- Features OWASP Top 10: 2021
- Focused on Black-box approach
- Still in early stage
  - Module 1 : Blog Application
  - Module 2 : HR Application
- Co-exist with other projects



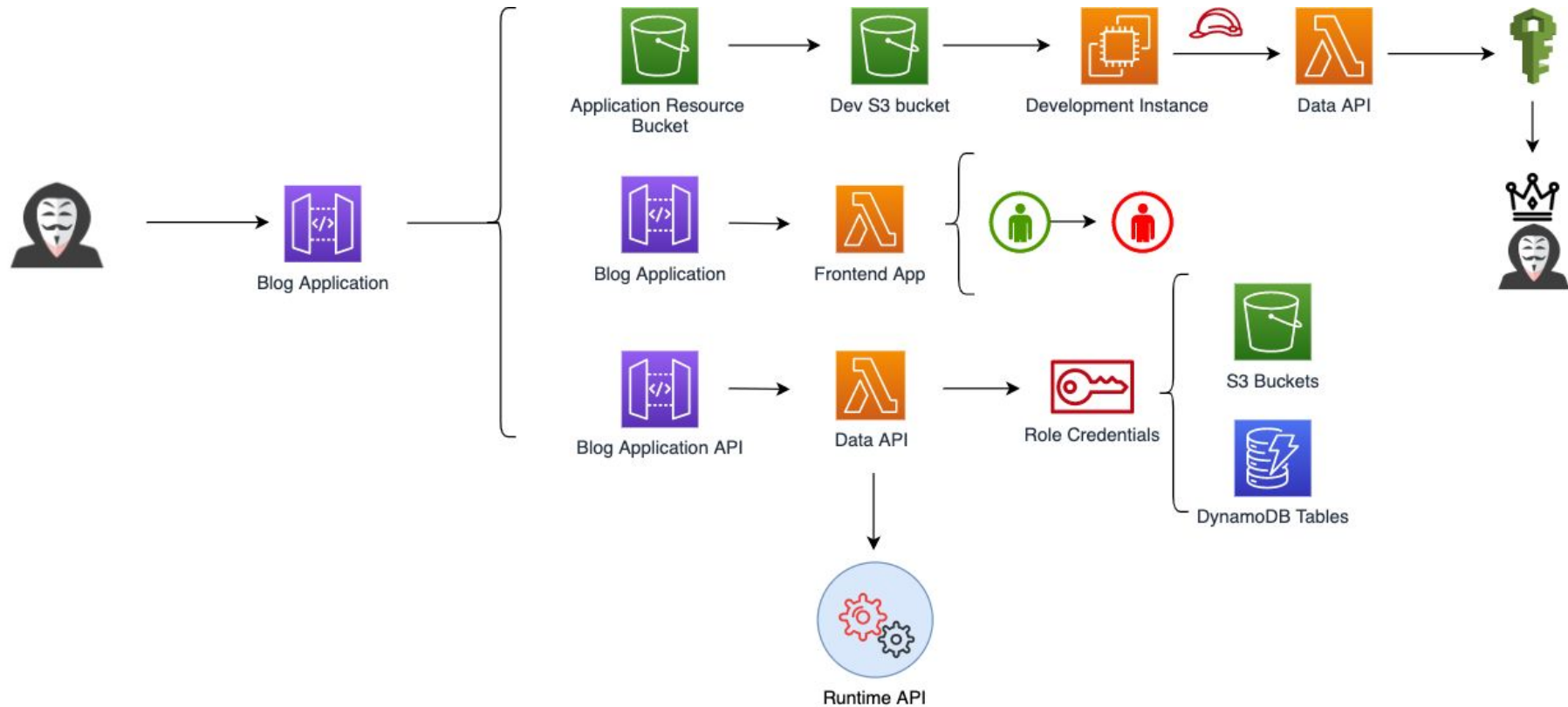
# OWASP Top 2021



# AWSGoat : Module 1 (Blog Application)

- A01: Broken Access Control
- A02: Cryptographic Failure
- A03: Injection
- A04: Insecure Design
- A05: Security Misconfiguration
- A07: Identification and Authentication Failures
- A10: Server Side Request Forgery

# AWSGoat : Module 1 (Blog Application)



# Building Realistic Insecure Application : Challenges

- Security Professional vs Seasoned Developers
- Mimicking Development Process
- Multiple Developer Environments
- Fast paced development.
- Lack of secure code practices

# Project Family



# Installation

- Repository: <https://github.com/ine-labs/AWSGoat>
- Using GitHub Actions
  - Configure Credentials in GitHub Secrets
  - Run the “terraform apply” workflow
- Manual Installation (Linux Machine)
  - Requirements
    - AWS CLI
    - Terraform
    - Python
    - Git
  - Commands:
    - aws configure
    - git clone <https://github.com/ine-labs/AWSGoat>
    - terraform init
    - terraform apply



## Exploring AWSGoat

# Attacking the Application

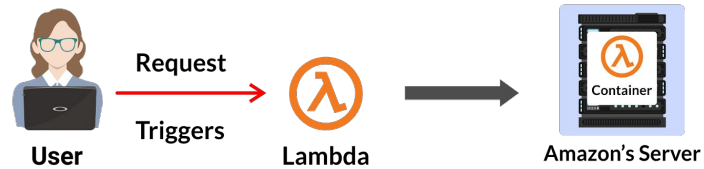
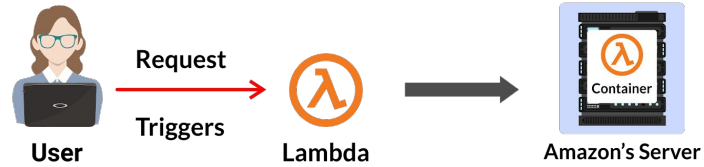
- XSS
- SQL Injection
- Insecure Direct Object Reference
- Server Side Request Forgery
- Sensitive Data Exposure and Password Reset
- S3 Misconfiguration
- IAM Privilege Escalation

# Lambda Environment : Overview

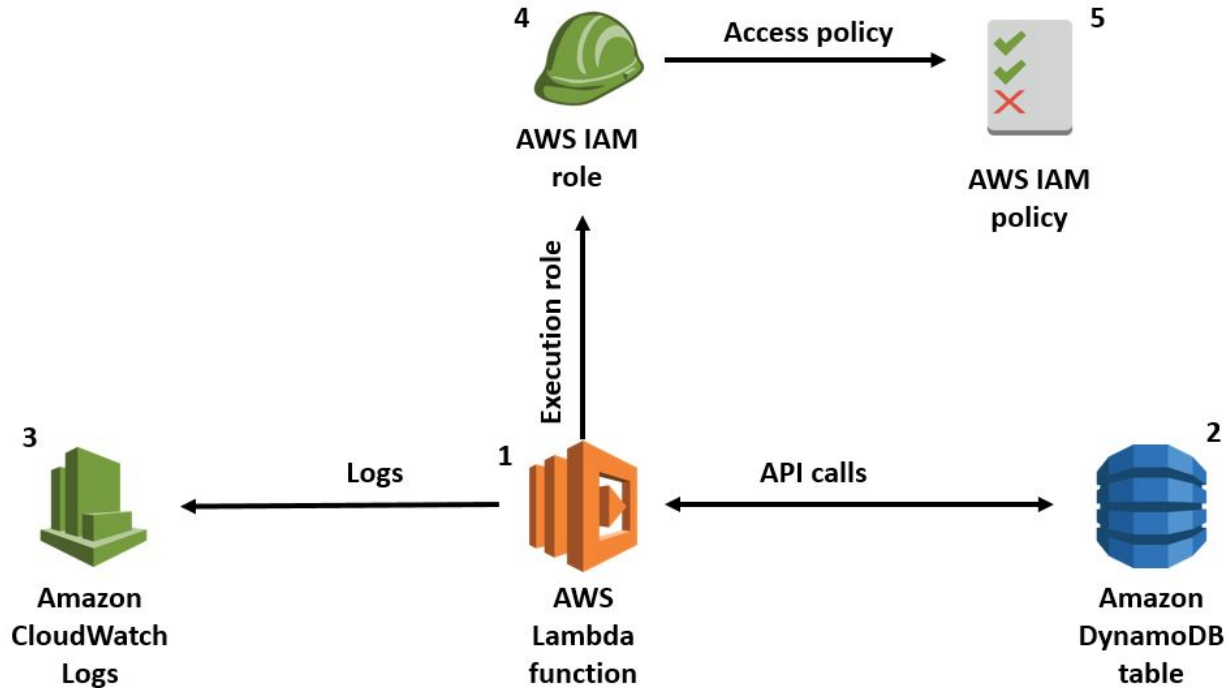
- Function Code
- Highly Scalable
- Underlying servers are managed by AWS



# Lambda Environment : Overview



# Lambda Environment : Role



# Server Side Request Forgery

- Interacting with the Lambda Runtime API
- Reading the source code of the application
- Reading the environment variables
  - Enumerate and attack other AWS Resources
  - Escalate Privileges
- Enumerate other applications/instances in the VPC



# API Gateway

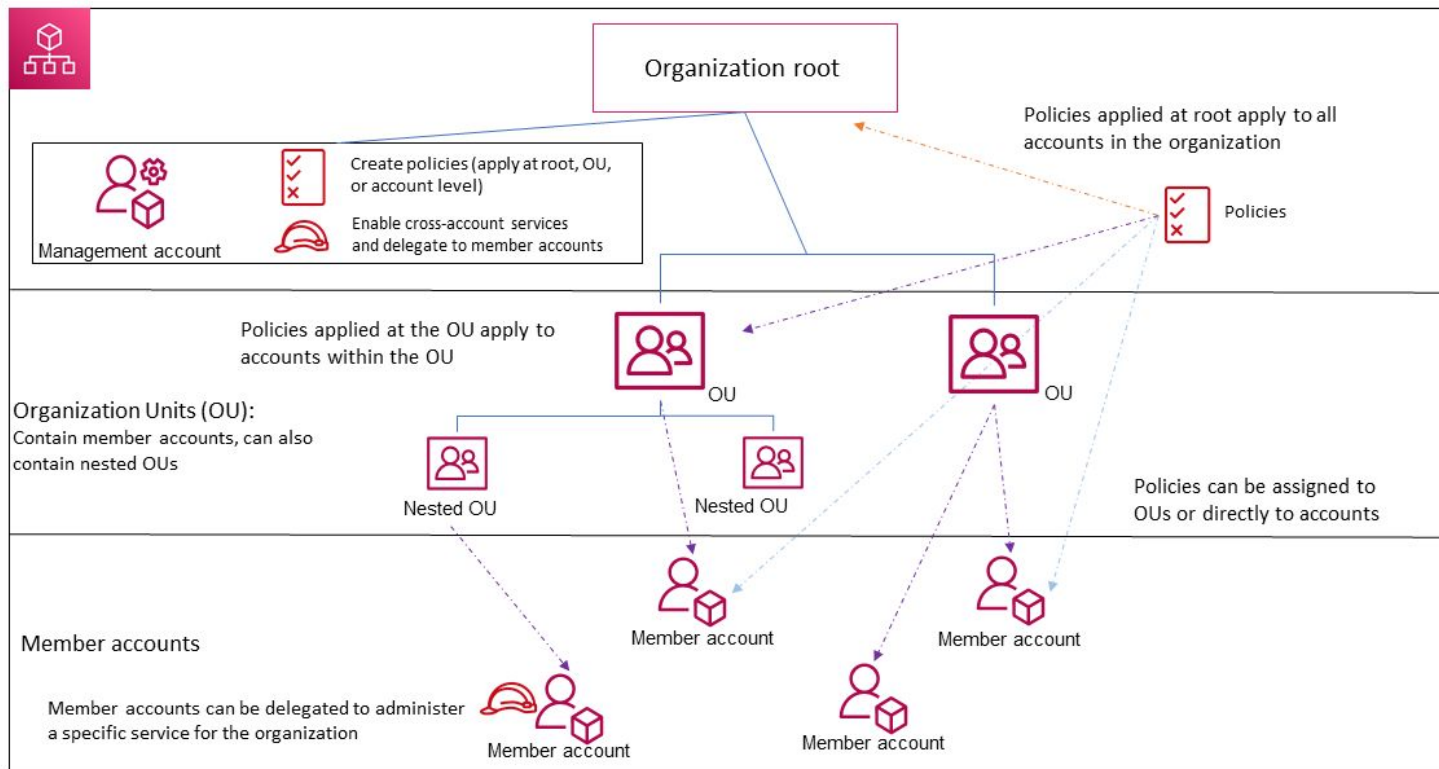
- Service Endpoints
  - protocol://service-code.region-code.amazonaws.com
  - e.g: <https://dynamodb.us-west-2.amazonaws.com/>
- `https://{restapi_id}.execute-api.{region}.amazonaws.com/{stage_name}/`
  - `https://0od87ivnul.execute-api.us-east-1.amazonaws.com/dev/`
- `https://{restapi_id}.execute-api.{region}.amazonaws.com/{stage_name}/{resource_name}/`
  - `https://0od87ivnul.execute-api.us-east-1.amazonaws.com/dev/list`

# Hunting S3 buckets

- Globally unique
- Company-wide naming practices
- Predictable names - based on departments/applications
- Misconfigured Policy - plethora of information
- Tool: <https://github.com/jordanpotti/AWSBucketDump>



# Future Plans: Multiple Applications across Multiple Accounts



# Future Plans

- More modules: EC2, EKS and Elastic Beanstalk
- Multi account infrastructure
- Working with the community
- IaC Misconfigurations
- Secure coding/deployment practices



**Thank you!**

[jmathai@ine.com](mailto:jmathai@ine.com)