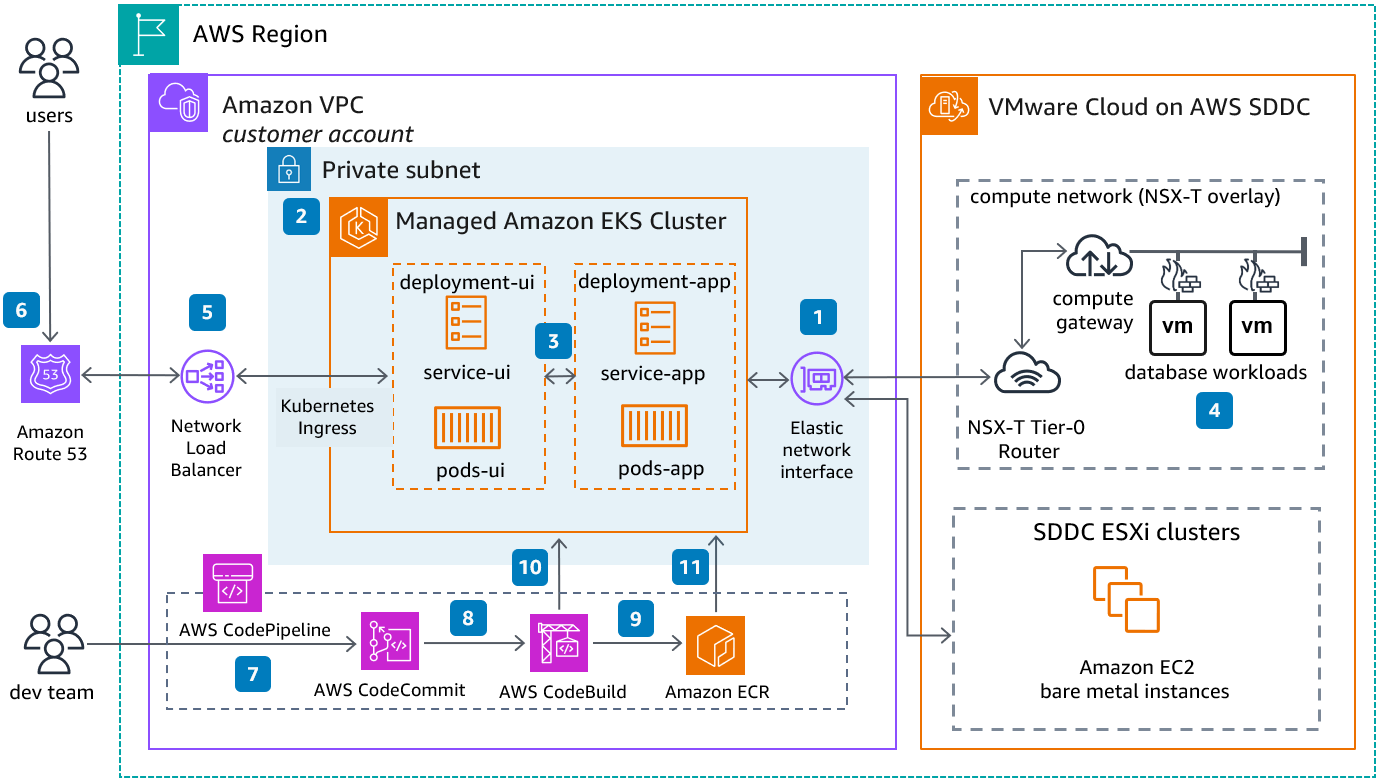
# AWS architecture diagram (from the [AWS documentation](https://docs.aws.amazon.com/architecture-diagrams/latest/modernize-applications-with-microservices-using-amazon-eks/modernize-applications-with-microservices-using-amazon-eks.html)) as the **basis for a STRIDE-based Threat Modeling Lab Exercise** using **OWASP Threat Dragon**.



**✅ Lab Objective**

Create a threat model using Threat Dragon based on an architecture that modernizes applications using microservices with **Amazon EKS**, **VMware Cloud on AWS**, and **CI/CD pipelines**.

**🧑‍🏫 Lab Setup – STRIDE Threat Modeling Using AWS Microservices Architecture**

**📌 Part 1: Use Case Summary**

This architecture includes:

* **Amazon EKS Cluster** hosting microservices
* **Elastic Network Interface** (ENI) communication with **VMware Cloud**
* **Kubernetes ingress** with **Route 53** and **Load Balancer**
* **CI/CD** via AWS CodePipeline, CodeCommit, CodeBuild, and Amazon ECR

We’ll model **external users**, **DevOps pipelines**, **Kubernetes services**, and **database connections**.

**🧩 Part 2: Build in Threat Dragon**

**🛠 Project Metadata**

| **Field** | **Value** |
| --- | --- |
| **Model Name** | AWS EKS Microservices Architecture |
| **Description** | Threat model for a microservice-based cloud-native application on EKS with CI/CD integration and VMware Cloud extension. |
| **Contributors** | Usama Wahab Khan, [students] |
| **Reviewers** | Instructor, Peers |

**➕ Add Diagram**

| **Field** | **Value** |
| --- | --- |
| **Diagram Name** | EKS-based Microservices Threat Model |
| **Diagram Description** | Visual representation of an EKS-based app environment with ingress, deployment units, CI/CD flow, and database interaction using STRIDE methodology. |

**🧱 Part 3: Diagram Components to Add**

| **Threat Dragon Element** | **Name** | **Description** |
| --- | --- | --- |
| **External Entity** | User | End-users accessing app via DNS |
| **External Entity** | Dev Team | CI/CD contributors |
| **Process** | Amazon Route 53 | DNS resolution |
| **Process** | Network Load Balancer | Entry point into EKS |
| **Process** | Ingress Controller | Routes to services |
| **Process** | service-ui | Frontend microservice |
| **Process** | service-app | Backend microservice |
| **Data Flow** | User Credentials | Flow from UI to backend |
| **Process** | Auth Service | Handles user sessions |
| **Data Store** | User DB (VMware) | Database in SDDC (via ENI) |
| **Trust Boundary** | Internet Boundary | Between public and VPC |
| **Trust Boundary** | Kubernetes Cluster Boundary | Secure zone for pods |
| **Process** | CI/CD Pipeline | DevOps pipeline |
| **Process** | Amazon ECR | Stores Docker images |

**⚠️ Part 4: Add STRIDE Threats (Examples)**

**🔒 1. Network Load Balancer**

| **STRIDE** | **Title** | **Description** | **Score** |
| --- | --- | --- | --- |
| S | DNS Spoofing | Attacker sends users to fake app | High |
| I | SSL Stripping | Load balancer fails to enforce HTTPS | Critical |
| D | SYN Flood Attack | DoS on entry point | High |

**🧩 2. service-ui → service-app (Internal Communication)**

| **STRIDE** | **Title** | **Description** | **Score** |
| --- | --- | --- | --- |
| T | API Parameter Tampering | Input manipulation changes logic | Medium |
| I | Internal API Info Leak | Verbose error messages expose backend logic | Medium |
| R | Lack of Logging | No trace of inter-service calls | Medium |

**💾 3. User DB (on VMware via ENI)**

| **STRIDE** | **Title** | **Description** | **Score** |
| --- | --- | --- | --- |
| I | Cross-cloud Data Leak | Sensitive data exposed over hybrid connection | Critical |
| T | Unencrypted Queries | DB communication lacks encryption | High |
| S | Identity Mapping Failure | Compromised EKS identity accesses VM workloads | High |

**🔁 4. CI/CD Pipeline (CodePipeline, CodeBuild)**

| **STRIDE** | **Title** | **Description** | **Score** |
| --- | --- | --- | --- |
| T | Compromised Docker Image | Malicious container is built and deployed | Critical |
| E | Privilege Escalation via IAM | Poor role boundaries allow access to production | Critical |
| R | No Build Logs | Lack of audit trail on builds | Medium |

**🧠 Part 5: Contextual Threats**

| **Context** | **Threat Description** | **STRIDE** | **Score** |
| --- | --- | --- | --- |
| Multi-cloud | Misconfigured route between EKS and VMware | I, T | High |
| Open Ingress | Publicly exposed pods via wildcard routes | S, I | Critical |
| No Image Scanning | CI/CD does not scan Docker images | T | High |

**📤 Part 6: Final Steps**

1. Validate model in Threat Dragon.
2. Add contributors and reviewers in metadata.
3. Export report as PDF or Markdown.

**📘 Optional Extension**

**Ask students to:**

* Add mitigation strategies
* Propose secure design alternatives
* Compare STRIDE vs DREAD threat scoring