



Subject :- Simulating centralized identity and access management in a distributed network with high server availability.

Guided By:

Ms. Tejaswini Apaté

Presented By:

Abhishek Raj

Sahil Tadas

Sayyad Nayda

Madhavi Jangam

INTRODUCTION :-

Simulating centralized (means bringing things to one central place or under unified control) identity and access management in a distributed network with high server availability refers to creating a system where user identities and their access permissions are managed centrally, even though the network spans multiple locations or servers. This system ensures that users can securely access resources regardless of their physical location and that the network remains operational even if some servers experience downtime.

Problem statement:- In today's distributed computing environments, maintaining centralized control over identity and access management while ensuring high server availability poses a significant challenge. As organizations expand their network infrastructure across multiple locations and employ a diverse array of servers, ensuring consistent access controls and minimizing downtime becomes increasingly complex. The task at hand is to develop a solution that can effectively simulate centralized identity and access management in such a distributed network environment, all while guaranteeing high server availability.

Key Objectives:

1. **Centralized Identity Management:** Develop a system that centrally manages user identities and access permissions across the distributed network, ensuring consistency and security.
2. **Access Control:** Implement robust access control mechanisms to regulate user access to resources, regardless of their physical location within the network.
3. **High Server Availability:** Design the system with redundancy and failover capabilities to maintain network operation even in the event of server failures or maintenance activities.
4. **Scalability:** Ensure that the solution can scale seamlessly to accommodate the growing demands of the network, supporting an increasing number of users and resources.

5. **Performance:** Optimize the system's performance to provide fast and reliable access to resources, minimizing latency and ensuring a seamless user experience.

Scope: The solution will focus on developing a software-based approach to simulate centralized identity and access management within a

distributed network environment. It will include the design and implementation of identity management protocols, access control mechanisms, and server availability strategies. However, hardware-level solutions or physical network configurations will be beyond the scope of this project.

Objective:- Integrating LDAP (Lightweight Directory Access Protocol), Jenkins, Docker, Amazon Elastic Compute Cloud (EC2), and Amazon Elastic Container Service (ECS) collectively supports a range of goals related to modern software development, deployment, and management in the cloud. Here are several key objectives they help achieve:

1. **Centralized Identity and Access Management:** LDAP serves as a centralized directory service for managing user identities and access permissions. Integrating LDAP with Jenkins, Docker, ECS, and EC2 ensures consistent authentication and access control across all stages of the development and deployment lifecycle. This centralization simplifies user management and enhances security.
2. **Automated CI/CD Pipelines:** Jenkins orchestrates automated CI/CD pipelines for building, testing, and deploying applications. By integrating with LDAP, Jenkins can authenticate users and enforce access controls on pipeline execution. Docker facilitates consistent packaging and deployment of applications, while ECS manages the orchestration of containerized workloads, enabling seamless deployment of applications across environments.
3. **Containerization and Microservices Architecture:** Docker enables containerization of applications, promoting consistency and portability. ECS provides a managed environment for deploying and scaling containerized applications, ensuring high availability and resource efficiency. Leveraging Docker and ECS together allows for the adoption of microservices architecture, enhancing scalability and agility.

4. **Scalable Infrastructure Management:** EC2 offers resizable compute capacity in the cloud, while ECS simplifies the management of containerized workloads. Together, they provide scalable infrastructure resources for hosting applications. With auto-scaling capabilities, EC2 and ECS dynamically adjust resource allocation based on workload demands, ensuring optimal performance and cost efficiency.
5. **High Availability and Fault Tolerance:** ECS and EC2 support high availability and fault tolerance by distributing workloads across multiple availability zones and automatically replacing unhealthy instances. Docker's container orchestration features, combined with ECS's service auto-recovery mechanisms, enhance resilience and minimize downtime during failures.
6. **Efficient Resource Utilization:** Integrating Docker with ECS optimizes resource utilization by allowing multiple containers to run on a single EC2 instance. This maximizes resource efficiency and reduces operational costs associated with managing infrastructure resources.
7. **Standardized Development and Deployment Processes:** Using LDAP, Jenkins, Docker, ECS, and EC2 together promotes standardized development and deployment processes. Developers can leverage automated CI/CD pipelines to build, test, and deploy applications consistently across different environments, leading to improved efficiency and reliability.

In summary, integrating LDAP, Jenkins, Docker, ECS, and EC2 supports goals related to centralized identity management, automated CI/CD pipelines, containerization, scalable infrastructure management, high availability, fault tolerance, efficient resource utilization, and standardized development and deployment processes in the cloud.

Requirements:-

Platform:- Amazon Web Services (AWS)

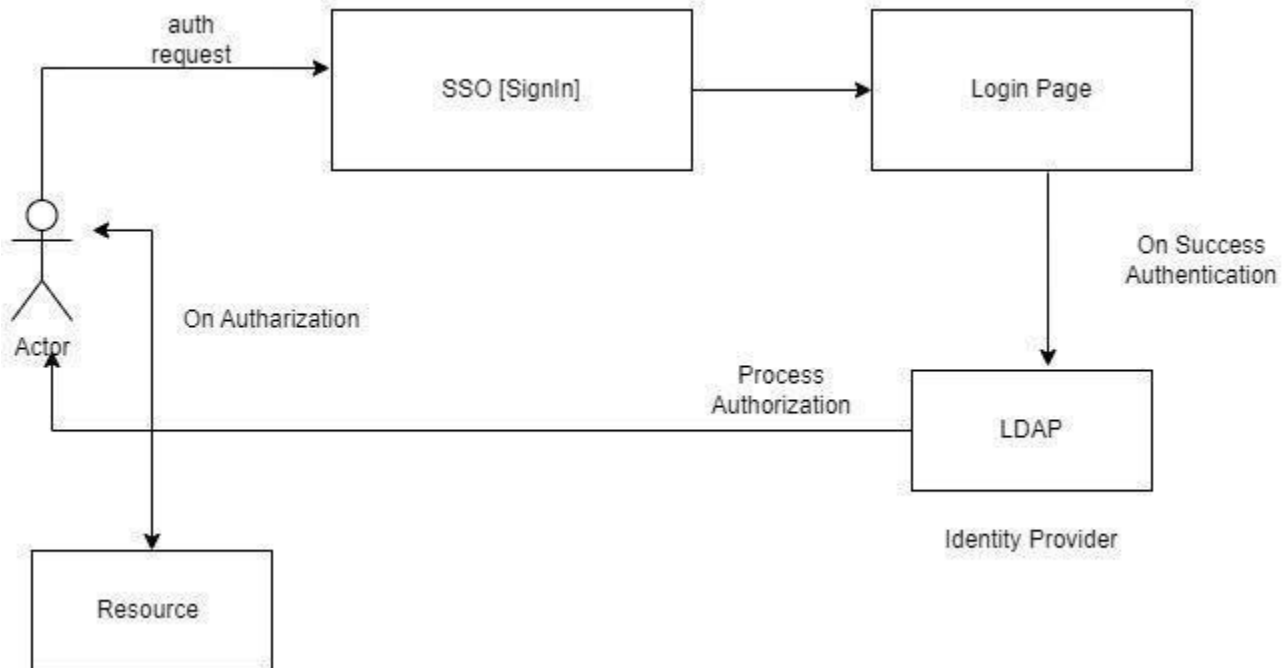
Services: -

- **AWS:-**

- EC2 Instances:- To host an application that will act as a service provider.
- ECS:- Plan to containerize and orchestrate the application.
- VPC:- To get a private cloud where we can securely manage all of our infrastructural resources.
- Auto scaling:- The ability of a system to readily add extra processing resources to handle the increased loads.
- Load Balancing:- It involves distributing incoming network traffic across multiple servers.

- **LDAP Server:-** Act as an identity provider.

Workflow Diagram



1. First client/user accesses the SSO login page where the user will provide his/her credentials.
2. After Successful Authentication on the SSO login page it sends a request to the identity provider in this case (LDAP).
3. When a request is received by LDAP it checks the USER entry in its active directory.
4. If this user entry is present there then the authorization process is completed for that user and the user gets access to those services which is being allowed to use.

