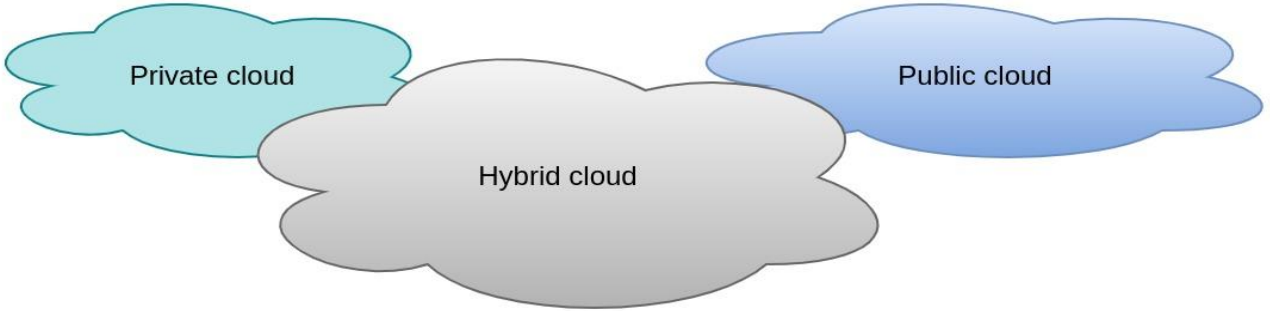
Containerisation of Web Application & Deploying in Hybrid Cloud

Members : Mohammad Sayedur Rahman, Saiful Islam, Shrabanti Saha Rimi, Ashis Banik, Ta-Seen Junaid

Introduction

Hybrid Cloud

- Combination of private and public cloud
- Mixed computing, storage, and services environment
- Made up of on-premises infrastructure, private cloud services, and public cloud services



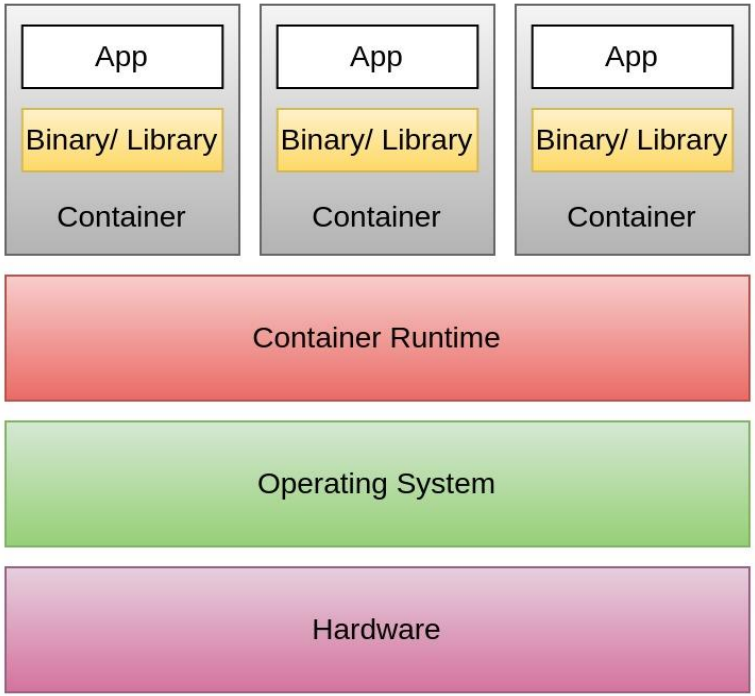
Hybrid Cloud

Why Hybrid Cloud

- The perfect world
- Taking the best from both world
- More control over costs
- Separating critical workloads from less sensitive workloads
- Modernization and migration at your own pace
- Maintain regulatory compliance
- Flexibility for the future

Containerization

- Encapsulation of software code with its run time environment
- Application code as a single, portable, lightweight executable package
- Package together with related OS, libraries, configuration files, dependencies etc



Containerization

Benefits of Containerization

- Portability
- Agility
- Speed
- Scalability
- Dev and Ops separation of concerns
- Developer-Friendly
- Continuity
- Environmental consistency across development, testing, and production

Containerization

Benefits of Containerization

- Ease of Management
- Plug and play
- Security
- Continuous development, integration, and deployment
- Fault Isolation
- Loosely coupled, distributed, elastic, liberated micro-services
- Efficiency
- Resource utilization
- Lightweight
- Resource isolation

Container Orchestration

- Automation of managing and coordinating the life cycles of containers in dynamic environments
- Automation of much of the operational effort required to run containerized workloads and services
- Provisioning, coordinating, deployment, scheduling, networking, configuring, scaling, availability, and lifecycle management of containers

Container Orchestration

Benefits of Containerization Orchestration

- Load balancing
- Traffic routing
- Service discovery of containers
- Storage orchestration
- Automated rollouts and rollbacks
- Automatic bin packing
- Disaster recovery or backup and restore
- Keeping interactions between containers secure
- Resource allocation

Container Orchestration

Benefits of Containerization Orchestration

- Self-healing
- High availability or no downtime
- Provisioning of containers
- Deployments of containers
- Secret management
- Configuration management
- Scalability or high performance
- Monitoring container health
- Configuring and scheduling of containers

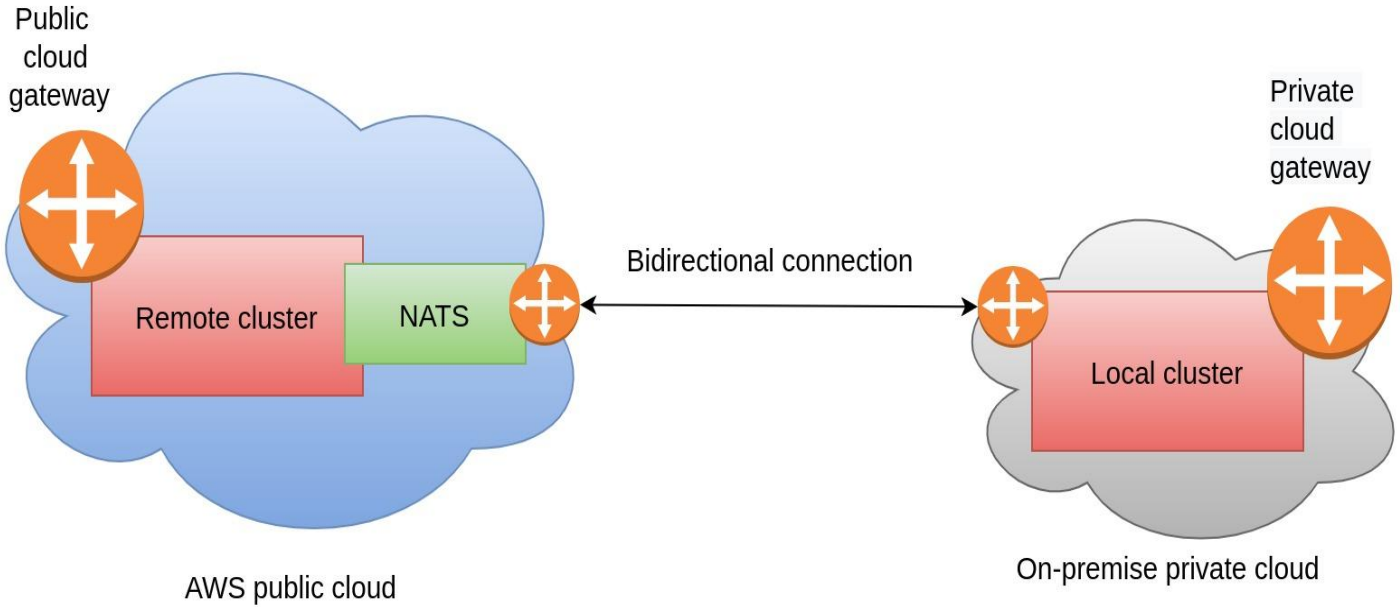
Technology Used

- Docker
  - Container runtime
- Kubernetes
  - Container orchestration tools
- AWS
  - Public cloud computing platforms
- AWS EKS
  - Managed Kubernetes service for run Kubernetes on AWS
- Helm
  - The package manager for Kubernetes
- NATS
  - Connective technology for adaptive edge & distributed systems

System Architecture

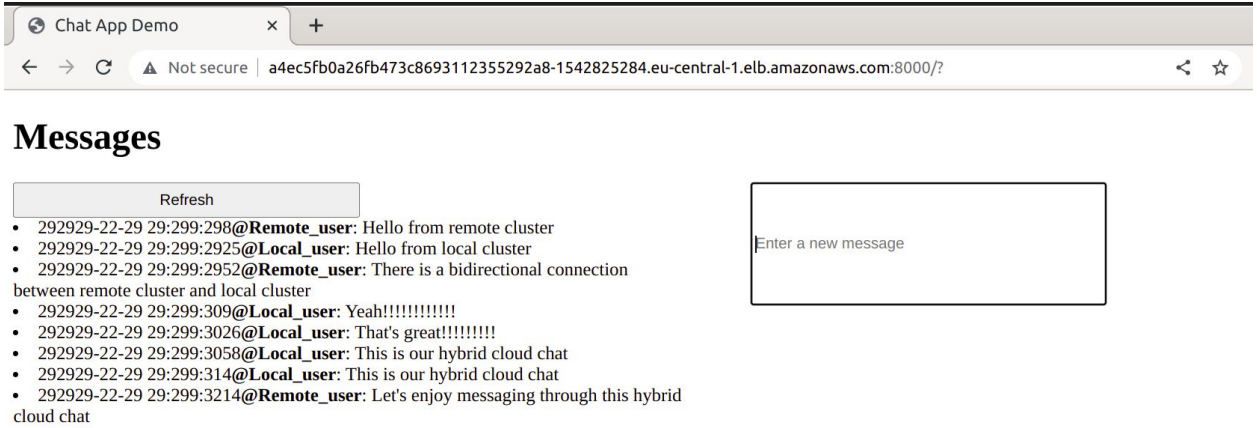
- Remote cluster
  - AWS EKS
- Local cluster
  - On-premises cluster with Kubernetes
- Tunnel connection between remote and local cluster via NATS
  - Deployed on remote cluster with the help of Helm
- Bi-directional conversation in between
  two systems running in local and Remote cluster

System Architecture



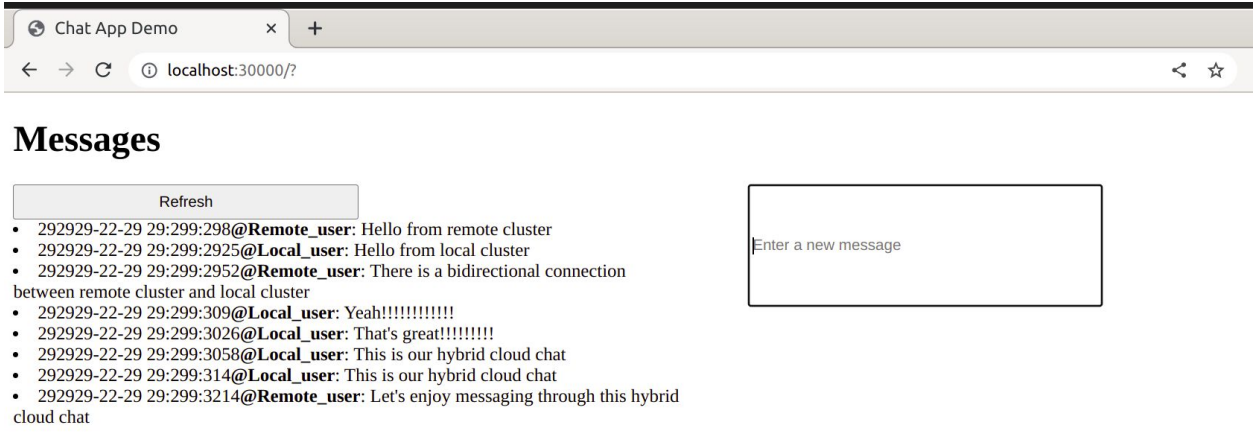
Demo

User interface from remote cluster



Demo

User interface from local cluster



Demo

Live demo

Conclusion

Questions??

Conclusion

Thank You!