CloudsStorm: An Application-driven DevOps Framework for Managing Networked Infrastructures on Federated Clouds

Huan Zhou, Cees de Laat, Zhiming Zhao
University of Amsterdam

Overview

Why do we need it?

- Application Requirements
- Project
- Cloud Infrastructure
- Topologies
- Custom

Where are the data sources?

- What is the data or workload size?
- When more or less resources is needed?

Geographic related Computing capacity Runtime scaling...

Key Approaches and Implementation

Networking

We adopt a tunnelling-based technique to realize the Virtual Network Function (VNF) between different Clouds.

Controlling Model

Our runtime controlling model supports multiple control scenarios.

YAML-based syntax

YAML (YAML Ain’t Markup Language) is human-readable and easy to learn. Topology Description Syntax (a) ≅ Infrastructure Code Syntax (b)

Experimental Results

- Recovery Performance
  - Test Clouds: EC2, EoGENI and EGI
  - VM Information: CPU: 2 CPU, MEM: 5G
  - OS: Ubuntu 1404
  - Corresponding types defined in Clouds: EC2: t2.large, EoGENI: XOMedium, EGI tetra.medium

- Performance Evaluation
  - Scaling Performance
  - Test Cloud: EoGENI
  - VM type: XOMedium
  - Scaling group: defined as 8 VMs in a sub-topology
  - Scaling at different scales
  - Each scaling group is from different datacenters

- Performance Comparison
  - Test Cloud: EC2 (California datacenter)
  - VM type: t2.large
  - Scenario:
    - (a) 5 VMs without deployment
    - (b) 5 VMs with deployment
    - (c) 5 VMs with Traffic 1 VM installs MySQL for database
  - Comparison tools: jclouds, jcloudscloudinit.d, CloudsStorm, Application-driven DevOps

Background

The IaaS (Infrastructure-as-a-Service) model in Cloud computing allows applications to customize its VM individually and configure its own network, but with limited programmability and controllability on the entire infrastructure. This gap hinders the application to customize its infrastructure at development phase and dynamic control at operation phase.

Research Problem and Challenges

Traditional DevOps (development and operations) approaches are suitable for today’s IaaS cloud environments, because of the slow, manual and error-prone collaboration between developers and operations personnel. The main challenges are as follows:

- Networked infrastructure: The network configuration for computing resources cannot be predefined, if adopting the unpredictable public IP addresses provided by Clouds. Most distributed applications rely on infrastructure with a customized network topology however.
- High-level controllability: There should be unified and application-driven controllability for the entire infrastructure, including failure recovery and auto-scaling, in order to ensure a high quality of service (QoS).
- Extensibility for federated Clouds: Clouds provide access to large quantities of computing resources. However, each Cloud has its own API for leveraging those resources. We therefore need an extensible framework to support different Clouds.
- Operation efficiency: Public Clouds adopt a pay-as-you-go business model. We need an efficient way to manage and operate these resources, with less manual work involved, in order to reduce monetary cost.