Unit - 4

1. Federation in Cloud

Cloud federation is the practice of interconnecting the cloud computing environments of two or more service providers for the purpose of load balancing traffic and accommodating spikes in demand.

Cloud federation requires one provider to wholesale or rent computing resources to another cloud provider. Those resources become a temporary or permanent extension of the buyer's cloud computing environment, depending on the specific federation agreement between providers.

Cloud federation offers two substantial benefits to cloud providers. First, it allows providers to earn revenue from computing resources that would otherwise be idle or underutilized. Second, cloud federation enables cloud providers to expand their geographic footprints and accommodate sudden spikes in demand without having to build new points-of-presence (POPs).

Service providers strive to make all aspects of cloud federation—from cloud provisioning to billing support systems (BSS) and customer support— transparent to customers. When federating cloud services with a partner, cloud providers will also establish extensions of their customer-facing service-level agreements (SLAs) into their partner provider's data centers.

Microsoft’s GENEVA framework focuses on issues involved in cloud federation. It is a claims-based access platform used to simplify the user access based on claims to applications and other systems. Claims describe identity attributes and allows multiple providers to interact seamlessly with others. It enables developers to incorporate various authentication models that will work with any corporate identity system, including Active Directory, LDAPv3-based directories, application-specific databases, and new user-centric identity models such as LiveID, OpenID, and InfoCard systems. It also supports Microsoft's Card-Space and Novell's Digital Me.

XMPP Protocols

Jabber XCP is a highly scalable, extensible, available, and device-agnostic presence solution built on XMPP and supports multiple protocols such as Session Initiation Protocol for Instant Messaging and Presence Leveraging Extensions (SIMPLE) and Instant Messaging and Presence Service (IMPS). Jabber XCP is a highly programmable platform, which makes it ideal for adding presence and messaging to existing applications or services and for building next-generation, presence-based solutions. Cloud services are being talked up as a fundamental shift in web architecture that promises to move us from interconnected silos (In business management and information technology (IT) a *silo* describes any management system that is unable to operate with any other system.) to a collaborative network of services whose sum is greater than its parts. The problem is that the protocols powering current cloud services, SOAP (Simple Object Access Protocol) and a few other assorted HTTP-based protocols, are all one-way information exchanges. Therefore cloud services aren't real-time, won't scale, and often can't clear the firewall. Many believe that those barriers can be overcome by XMPP (also called Jabber) as the protocol that will fuel the Software-as-a-Service (SaaS) models of tomorrow. Google, Apple, AOL, IBM, Livejournal, and Jive have all incorporated this protocol into their cloud-based solutions in the last few years.

XMPP’s profile has been steadily gaining since its inception as the protocol behind the open source instant messenger (IM) server jabberd in 1998. XMPP’s advantages include:

- It is decentralized, meaning anyone may set up an XMPP server.
- It is based on open standards.
- It is mature—multiple implementations of clients and servers exist.
- Robust security is supported via Simple Authentication and Security Layer (SASL) and Transport Layer Security (TLS).
- It is flexible and designed to be extended.

XMPP is a good fit for cloud computing because it allows for easy two way communication; it eliminates the need for polling; it has rich publish subscribe (pub-sub) functionality built in; it is XML-based and easily extensible, perfect for both new IM features and custom cloud services; it is efficient and has been proven to scale to millions of concurrent users on a single service (such as Google’s GTalk); and it also has a built-in worldwide federation model

XMPP is not the only pub-sub enabler getting a lot of interest from web application developers. An Amazon EC2-backed server can run Jetty and Cometd from Dojo. Unlike XMPP, Comet is based on HTTP, and in conjunction with the Bayeux Protocol, uses JSON to exchange data. Given the current market penetration and extensive use of XMPP and XCP for federation in the cloud and that it is the dominant open protocol in that space.

The ability to exchange data used for presence, messages, voice, video, files, notifications, etc., with people, devices, and applications gain more power when they can be shared across organizations and with other service providers. Federation differs from peering, which requires a prior agreement between parties before a server-to-server (S2S) link can be established.

Four Levels of Federation

Technically speaking, federation is the ability for two XMPP servers in different domains to exchange XML stanzas. According to the XEP-0238: XMPP Protocol Flows for Inter-Domain Federation, there are at least four basic types of federation

1. Permissive Federation -- a server accepts a connection from any other peer on the network, even without verifiying the identity of the peer based on DNS lookups. The lack of peer verification or authentication means that domains can be spoofed. Permissive federation was effectively outlawed on the Jabber network in October 2000 with the release of the jabberd 1.2 server, which included support for the newly-developed Server Dialback (XEP-0220) [2] protocol.
2. Verified Federation -- a server accepts a connection from a peer only after the identity of the peer has been weakly verified via Server Dialback, based on information obtained via the Domain Name System (DNS) and verification keys exchanged in-band over XMPP. However, the connection is not encrypted. The use of identity verification effectively prevents domain spoofing, but federation requires proper DNS setup and is still subject to DNS poisoning attacks. Verified federation has been the default service policy followed by servers on the open XMPP network since the release of the open-source jabberd 1.2 server.
3. Encrypted Federation -- a server accepts a connection from a peer only if the peer supports Transport Layer Security (TLS) as defined for XMPP in (RFC) 3920.The peer must present a digital certificate. However, the certificate may be self-signed, in which case mutual authentication is typically not possible. Therefore, after STARTTLS negotiation the parties proceed to weakly verify identity using Server Dialback. This combination results in an encrypted connection with weak identity verification.

4. Trusted Federation -- a server accepts a connection from a peer only if the peer supports Transport Layer Security (TLS) and the peer presents a digital certificate issued by a trusted root certification authority (CA). The list of trusted root CAs is determined by local service policy, as is the level of trust accorded to various types of certificates (i.e., Class 1, Class 2, or Class 3). The use of trusted domain certificates effectively prevents DNS poisoning attacks but makes federation more difficult since typically such certificates are not easy to obtain.

How Encrypted Federation Differs from Trusted Federation

Verified federation serves as a foundation for encrypted federation, which builds on it concepts by requiring use of TLS for channel encryption. The Secure Sockets Layer (SSL) technology, originally developed for secure communications over HTTP, has evolved into TLS. XMPP uses a TLS profile that enables two entities to upgrade a connection from unencrypted to encrypted. This is different from SSL in that it does not require that a separate port be used to establish secure communications. Since XMPP S2S communication uses two connections (bi-directionally connected), encrypted federation requires each entity to present a digital certificate to the reciprocating party.

Not all certificates are created equal, and trust is in the eye of the beholder.

In the trusted federation scenario, Dialback can be avoided if, after using TLS for channel encryption, the server verifying identity proceeds to use the SASL protocol for authentication based on the credentials presented in the certificates. In this case, the servers dispense with server Dialback, because SASL (in particular the EXTERNAL mechanism) provides strong authentication.

Federation services and Applications

Clouds typically consist of all the users, devices, services, and applications connected to the network. In order to fully leverage the capabilities of this cloud structure, a participant needs the ability to find other entities of interest. Such entities might be end users, multiuser chat rooms, real-time content feeds, user directories, data relays, messaging gateways, etc. Finding these entities is a process called discovery.

XMPP uses service discovery (as defined in XEP-0030) to find the aforementioned entities. The discovery protocol enables any network participant to query another entity regarding its identity, capabilities, and associated entities. When a participant connects to the network, it queries the authoritative server for its particular domain about the entities associated with that authoritative server.

In response to a service discovery query, the authoritative server informs the inquirer about services hosted there and may also detail services that are available but hosted elsewhere. XMPP includes a method for maintaining personal lists of other entities, known as roster technology, which enables end users to keep track of various types of entities

Protecting and controlling Federated communication

Some organizations are wary of federation because they fear that real-time communication networks will introduce the same types of problems that are endemic to email networks, such as spam and viruses. While these concerns are not unfounded, they tend to be exaggerated for several reasons:

- Designers of technologies like XMPP learned to prevent address spoofing, unlimited binary attachments, inline scripts, and other attack tactics in XMPP.
- The use of point-to-point federation will avoid problem that occur with multihop federation. This includes injection attacks, dataloss, and unencrypted intermediate links.
- Using certificates issued by trusted root CAs ensures encrypted connections and strong authentication, both of which are currently feasible with an email network.

- Employing intelligent servers that have the ability to blacklist (explicitly block) and whitelist (explicitly permit) foreign services, either at the host level or the IP address level, is a significant mitigating factor.

2. Presence in the Cloud

Presence data enables organizations to deploy innovative real-time services and achieve significant revenue opportunities and productivity improvements.

At the most fundamental level, understanding presence is simple:

- It provides true-or-false answers to queries about the network availability of a person, device, or application.
- Presence is a core component of an entity's ***real-time*** identity.
- Its purpose is to signal availability for interaction over a network.
- It is being used to determine availability for phones, conference rooms, applications, web-based services, routers, firewalls, servers, appliances, buildings, devices, and other applications.

The management of presence is being extended to capture even more information about availability, ***or even the attributes associated with such availability***, such as a person's current activity, mood, location (e.g., GPS coordinates), or preferred communication method (phone, email, IM, etc.).

Presence is an enabling technology for peer-to-peer interaction. It first emerged as an aspect of communication systems, especially IM systems such as ICQ, which allowed users to see the availability of their friends. The huge role that IM has had in establishing presence is evident with the protocols available today, such as Instant Messaging and Presence Service (IMPS), Session Initiation Protocol (SIP) for Instant Messaging and Presence Leveraging Extensions (SIMPLE), the Extensible Messaging and Presence Protocol (XMPP).

Implementation of presence follows the software design pattern known as publish-and-subscribe (pub-sub). This means that a user or application publishes information about its network availability to a centralized location and that information is broadcast to all entities that are authorized to receive it. The authorization usually takes the form of a subscription. In IM implementations, contacts or buddies are the authorized entities. The popularity of these services among millions of people validated the value of the concept of presence.

For enterprise solutions, the limits of consumer-based IM services quickly became clear when enterprises tried to integrate presence into business-critical systems and services. Because business organizations require a great deal more control and flexibility over the technologies they deploy, they needed a presence solution that could provide separation between the presence service and the communication mechanisms (e.g., IM or VoIP) that presence enables. Any solution had to be scalable, extensible, and support a distributed architecture with its own presence domain. It should not overload the network and should support strong security management, system authentication, and granular subscription authorization. Also, any device or application should be able to publish and subscribe to presence information. Enterprise solutions should have the ability to federate numerous cross-protocol presence sources and integrate presence information from multiple sources. Any solution should be able to access presence data via multiple methods. The ability to integrate presence information with existing organizational infrastructure such as active directory is very important. Being able to publish content and allow other people and/or applications to subscribe to that information ensures that updates and changes are done in real time based on the presence/availability of those people/applications.

Presence Protocols

Proprietary, consumer-oriented messaging services do not enable enterprises or institutions to leverage the power of messaging and presence protocol suite based on SIP and managed by the Internet Engineering Task Force (IETF). XMPP is the IETF's formalization of the core XML messaging and presence protocols originally developed by the open source Jabber community in 1999. These protocols have been in wide use on the Internet for over five years.

The modern, reliable method to determine another entity's capabilities is called ***service discovery***, wherein applications and devices exchange information about their capabilities directly, without human involvement. Even though no framework for service discovery has been produced by a standards development organization such as the IETF, a capabilities extension for SIP/SIMPLE and a robust, stable service discovery extension for XMPP does exist.

The SIMPLE Working Group is developing the technology to embed capabilities information within broadcasted presence information. A capability already exists in a widely-deployed XMPP extension. Together, service discovery and capabilities broadcasts enable users and applications to gain knowledge about the capabilities of other entities on the network, providing a real-time mechanism for additional use of presence-enabled systems.

Leveraging Presence

The real challenge today is to figure out how to leverage the power of presence within an organization or service offering. This requires having the ability to publish presence information from a wide range of data sources, the ability to receive or embed presence information in just about any platform or application, and having a robust presence engine to tie ubiquitous publishers and subscribers together.

It is safe to assume that any network-capable entity can establish presence. The requirements for functioning as a presence publisher are fairly minimal. As a result, SIP software stacks are available for a wide range of programming languages and it is relatively easy to add native presence publishing capabilities to most applications and devices. Enabling devices and applications to publish presence information is only half of the solution, however; delivering the right presence information to the right.

Presence Enabled

What does it mean to be "presence-enabled"? The basic concept is to show availability of an entity in an appropriate venue. Some modern applications aggregate presence information about all of a person's various connections. For communication devices such as phones and applications such as IM, presence information is often built into the device itself. For less communication-centric applications, such as a document or web page, presence may be gathered by means of a web services API or channeled through a presence daemon. Providing presence data through as many avenues as possible is in large measure the responsibility of a presence engine, as described below.

The presence engine acts as a broker for presence publishers and subscribers. A presence broker provides aggregation of information from many sources, abstraction of that information into open and flexible formats, and distribution of that information to a wide variety of interested parties. As presence becomes more prevalent in Internet communications, presence engines need to provide strong authentication, channel encryption, explicit authorization and access control policies, high reliability, and the consistent application of aggregation rules. Being able to operate using multiple protocols such as IMPS, SIMPLE, and XMPP is a basic requirement in order to distribute presence information as widely as possible. Aggregating information from a wide variety of sources requires presence rules that enable subscribers to get the right information at the right time.

The Future of Presence

It will remain to be seen if XMPP is the future of cloud services, but for now it is the dominant protocol for presence in the space. Fixing the polling and scaling problems with XMPP has been challenging but has been accomplished by providers such as Tivo, and the built-in presence functionality offers further fascinating possibilities. Presence includes basic availability information, but it is extensible and can also include abilities.

The Interrelation of Identity, Presence, and Location in the Cloud

Digital identity refers to the traits, attributes, and preferences on which one may receive personalized services. Identity traits might include government-issued IDs, corporate user accounts, and biometric information. Two user attributes that may be associated with identity are presence and location. Over the last few years, there has been an aggressive move toward the convergence of identity, location, and presence. This is important because a standard framework tying identity to presence and location creates the ability to develop standards-based services for identity management that incorporate presence and location. Identity, presence, and location are three characteristics that lie at the core of some of the most critical emerging technologies in the market today: real-time communications (including VoIP, IM, and mobile communications), cloud computing, collaboration, and identity-based security.

Presence is most often associated with real-time communications systems such as IM and describes the state of a user's interaction with a system, such as which computer they are accessing, whether they are idle or working, and perhaps also which task they are currently performing (reading a document, composing email etc.). Location refers to the user's physical location and typically includes latitude, longitude, and (sometimes) altitude. Authentication and authorization mechanisms generally focus on determining the "who" of identity, location defines the "where," and presence defines the "what"—all critical components of the identity-based emerging technologies listed above, including cloud computing.♦