Cloud vSwitch Project Proposal

Vision and Goals Of The Project:

Cloud vSwitch aims to build a connection between remote clients and a network in the cloud so that the clients are able to access different kinds of services hosted in the cloud as they were in a local network. High-level goals of Cloud vSwitch includes:

- Simple setup for clients through a browser
- Automated deployment of the cloud environment for the organization
- To provide secure connections
- To provide an authorization scheme for clients
- To develop a RESTful API for clients to interact with vSwitch network
- Provide a modular structure for new services

Users/Persons of the Project

Cloud vSwitch will be used by customers with access to a cloud network, such as small organizations with no IT department or departments which work as independent bodies as part of large organizations.

It targets both owners and end-users of the Cloud network.

Cloud vSwitch is not targeted for large enterprises with IT departments or expert users.

1. Scope and Features Of The Project:

Cloud vSwitch

- A user management interface.
    - User add/edit/delete functionalities
    - Client certificate management will be provided in order to
        - Create certificate requests
        - Issuing client certificates
        - Certificate verification (client authentication)
    - Group management
        - Group add/edit/delete functionalities
        - Everyone has equal authority in the network to invite new users.
- A virtual local network in the cloud with the ability to connect multiple remote clients.

    - A Local network access to all the clients based on the IP configurations of Gateway and Subnet addresses.
    - Ensuring a secure communication for all the users in the network through encryption.
- Service modules (An abstract module based on Amazon or OpenStack services)
    - Each kind of service such as file storage, Amazon EC2 services, etc, will be represented as a standardized module.
    - A standardized interface will be provided to manipulate the modules.
    - Service modules discovery
        - An inbuilt feature in the cloud to discover new service modules.
    - Developers can create vSwitch modules to manipulate the services required.
- Automation in deployment of the cloud key components (see section 2)
- Client UI
    - Providing a basic interface for registration and connection.
        - The client account will be linked with other accounts through a social media interface for registration.
        - Every login or connection attempt will be checked by the server
    - API interaction with modules
        - Introduce uniform interface for different services
    - Management of user specific requirements such as IP addresses and authorization of certificates(insertion and deletion)
    - Cross platform integration

2. Solution Concept

2.1 Global Architectural Structure Of the Project and a Walkthrough:

Below is a description of the key components of Cloud vSwitch which allow the deployment of this network and client access. These components are conformed by: cloud network side and client components

Cloud side:

- Service modules
- Web Portal for client registration
- Certificate Authentication(Issue certificates and verify certificates)
- DHCP server (Dynamically assigns IP address)
- Virtual Gateway: client's entry point to the network
- An interface for communication between vSwitch and clients.

Client side:

- Browser for initial registration through the Web portal.
- An Application to connect to the Cloud and interact with the modules
- A Client side Application on mobile devices for initial login to request certificates, receive certificates and to access services.

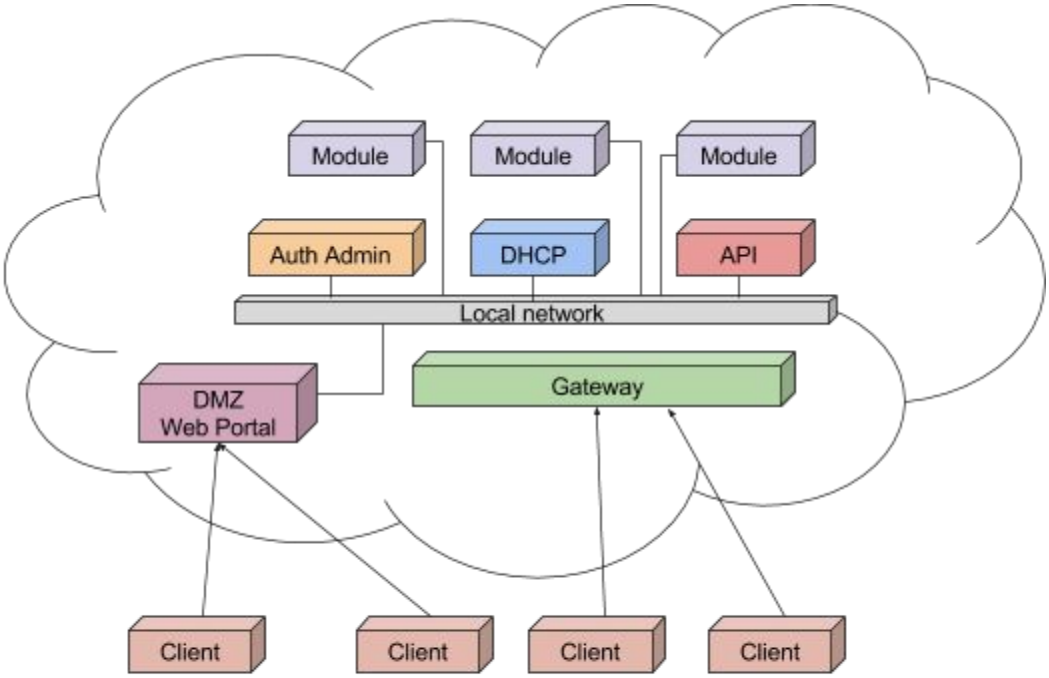

Figure 1. Cloud vSwitch Architecture

Figure 1. presents our global architecture design for Cloud vSwitch. A client first registers through the Web Portal which provides him/her with the necessary UI to interact with the Cloud and the certificates required to authenticate against the platform. The UI will allow the Client to connect to the cloud gateway which will assign a local address to the client's device. Also this UI will provide mechanisms to interact with the module services.

Cloud vSwitch is designed in a modular way to ensure growth and incorporation of new services based on an interface that dictates the basic operations these modules must offer.

2.2 Design Implications and Discussion

Key design decisions and motivation behind them.

- Automated deployment: Since our project aims at inexperienced users, different key components in the system must be automatically generated in accordance with the services a user chooses : This will be designed and implemented by initializing the configuration protocols for specific features like IP gateway addresses, subnet addresses and individual client IP addresses through shell scripting on a linux platform.
  The main motivation behind considering shell scripting is due to its inbuilt advantage of ease and flexibility compared to other configuration management softwares like puppet.
- Keep the network local: In order to preserve user authenticity in the internal network, we have decided to make the user creation process rely on invitations from other users in the network. This solution avoids the need of a supervisor who has the authority to include someone in the local network. Under certain circumstances where an employee leaves the organization, we expect the user to send a request to revoke his access to the cloud.
- Service as a module: Every service will be abstracted as a module inside the cloud. Basic behavioral characteristics of a module will be designed and implemented. With this formalized and standardized interface, an API from the client side will gain easier access to the service inside the cloud. For example, if we consider file systems as a service, the file service module can be created and run functionally when it is imported. This gives access to the Clients to add and remove their files. After the file service module is disabled, clients will no longer have access to manipulate files in the cloud.
- Client UI: The user interface should be made simple while providing the basic manipulation function of the maintenance service such as setting up, checking the current status, stopping and refreshing the service. The UI will provide uniform dashboard for different services deployed.
- Client Authentication: Cloud vSwitch uses certificate authentication schema. Therefore one of the key components is Certificate Authority responsible of issuing client certificates that will allow them to connect and access the network. The certificate will be checked automatically every time when connected to the local network.
- The vGateway is the entry point to the network and it will implemented by using openVPN. This will allow us to immediately inherit key aspects like connection security and extend the network to remote clients.

Acceptance criteria

Minimum acceptance criteria is

- Clients must be able to connect to the cloud network as a local network.
- Client application with a proper UI.
- Cloud side will have certificate authentication service, one service module(for demonstration purpose), vGateway, and also interface to manipulate services in the cloud.
- Ability to automatically deploy the cloud

Release Planning

Detailed user stories and plans are on the Trello board:

Release #1 (due by Week):
User stories: Web interface, vGateway

Web interface

(Cloud vSwitch init):
As a user when I enter a valid invitation code on the initial web page, I will be redirected to a registration webpage.

(New User Registration):
After submitting basic information (e.g., email, password, username), the user is registered with the Cloud vSwitch-Authority and receives his/her certificate.

vGateway

As a registered user I would connect to the network through a virtual gateway after having my credentials verified by the server.

Release #2 (due by Week):
Interface design
…

Release #3 (due by Week):
Client Application
Automation of deployment
…

Release #4 (due by Week):
Client Application
Module design
…

Release #5 (due by Week):
Integration and testing, possible extension