

# Cloud Gateway

Qianli Ma, Ayush Singh, Rahul Bahal and Mania Abdi  
College of Computer and Information Science, Northeastern University

## Abstract

Using Cloud gateway, we are proposing a cost effective hybrid cloud model to seamlessly expand resources on-demand.

- Provisioning resources for peak workloads is expensive.
- Public Cloud resources are added to private cloud resources to provide more resources for peak workload.
- All traffic flows from Public through the Private Cloud, secured using an IPsec tunnel protocol.

## Introduction

- The average-sized workload stays in the private cloud, and peak workload is provided via public cloud burst.
- Cloud Gateway provides the admin of private cloud seamless and secure integration between an organization's on-premises IT environment and cloud service providers.
- Machines will utilize the CG to send traffic between the private and public cloud subnets.
- Workers on public side communicates to internet via the cloud gateway on private side to ensure they follow the organization firewall rules.

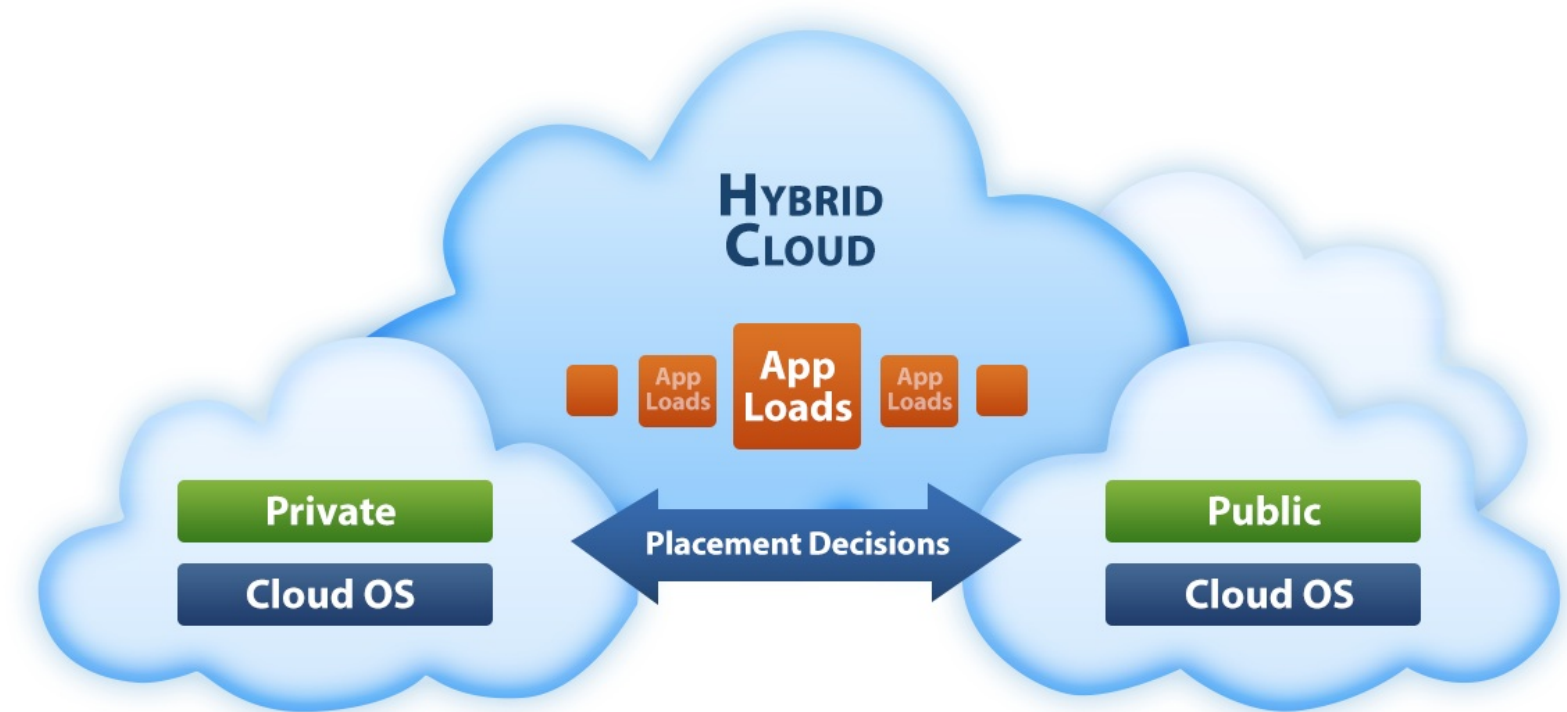


Figure 1: Hybrid cloud using Cloud Gateway

## Architecture

We designed an efficient architecture for expanding resources to withstand high workload.

- The data stream in private cloud is forwarded to cloud gateway.
- The cloud gateway forwards the packet to private cloud by reforming IP header with gateway address.
- Used iptables DNAT and Port forwarding to maintain consistent iptable in both public and private clouds.

## Cloud Configuration

- We install CG assuming a virtual private cloud is already setup in an organization.
- A CloudFormation template which take some parameter to set up VPC, subnet, cloud gateway VM etc.
- Amazon AWS and MOC OpenStack were used as test framework implying that the design is applicable to any cloud.
  - Amazon AWS virtual gateway is setup with source/destination checking disabled.
  - MOC OpenStack virtual gateway is configured with source/destination checking disabled.
  - Private virtual clouds are formed in different subnet from cloud gateway.
  - All data from virtual cloud are routed to cloud gateway.

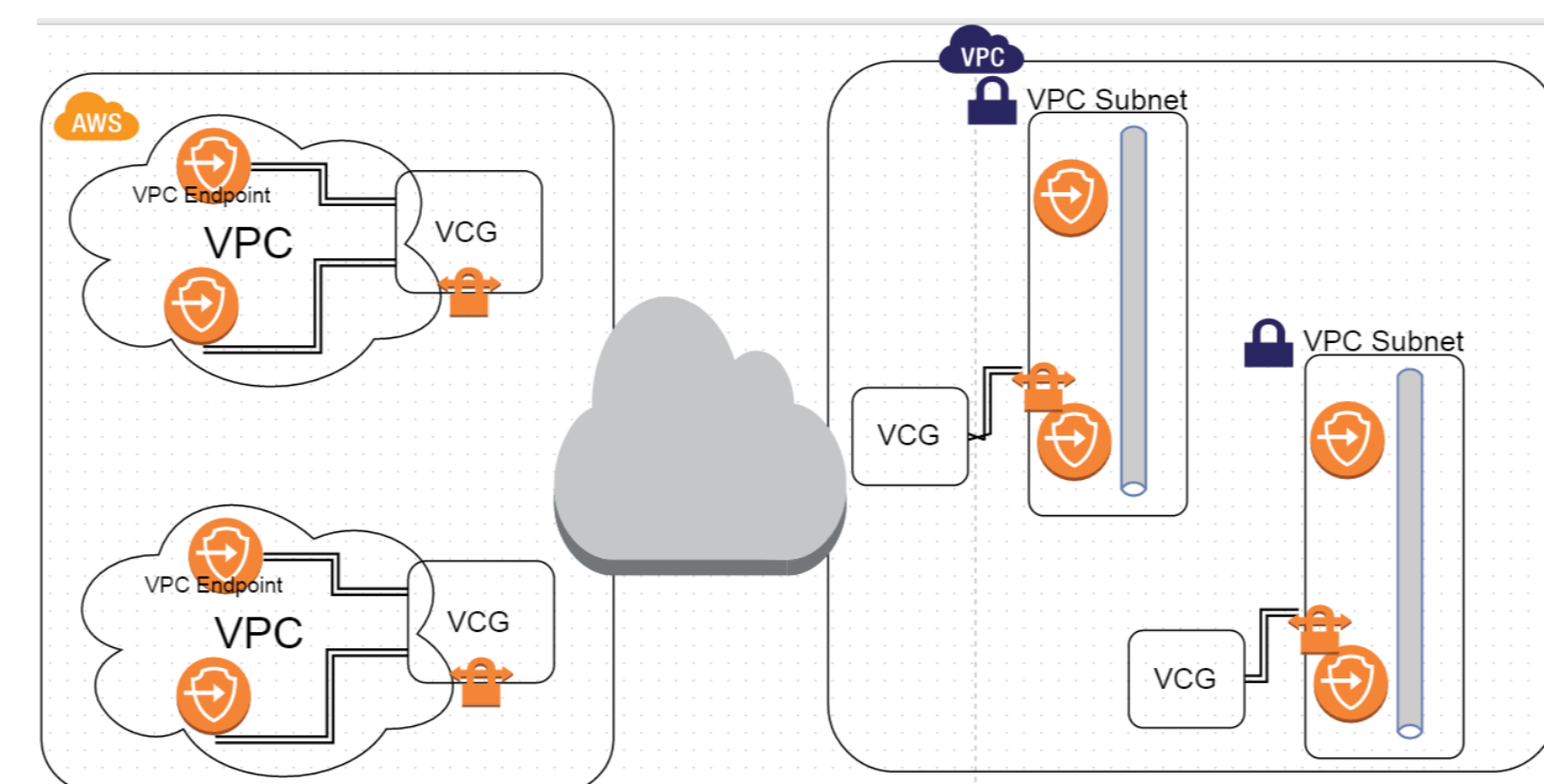


Figure 2: Cloud Gateway Architecture

## User Interface

User interface is used to configure cloud gateway.

- It exposes web service api, master-slave mode between VCG to facilitate control of network.
- It is open source under Apache 2.0 License.
- The webAPP allows the user to do DNAT, enable/disable internet and modify port forwarding tables.
- It makes sure that IP tables on both virtual cloud gateways are consistent.
- It is written in Flask (Python) backend, sqlite3 database.

## Cloud Gateway Deliver

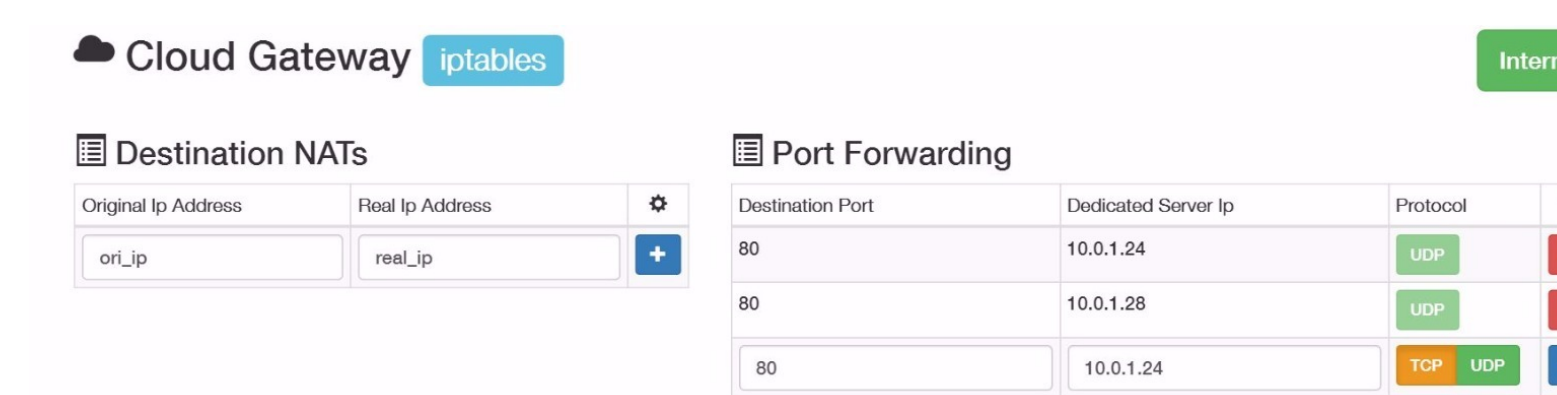


Figure 3: User Interface

- Rests on private and scales on demand via cloud burst to Public Cloud.
- Open source User-friendly interface to deal with configuration.
- Open source console tools to configure public cloud and private cloud.
- Cloud formation Templates to setup public cloud infrastructure.

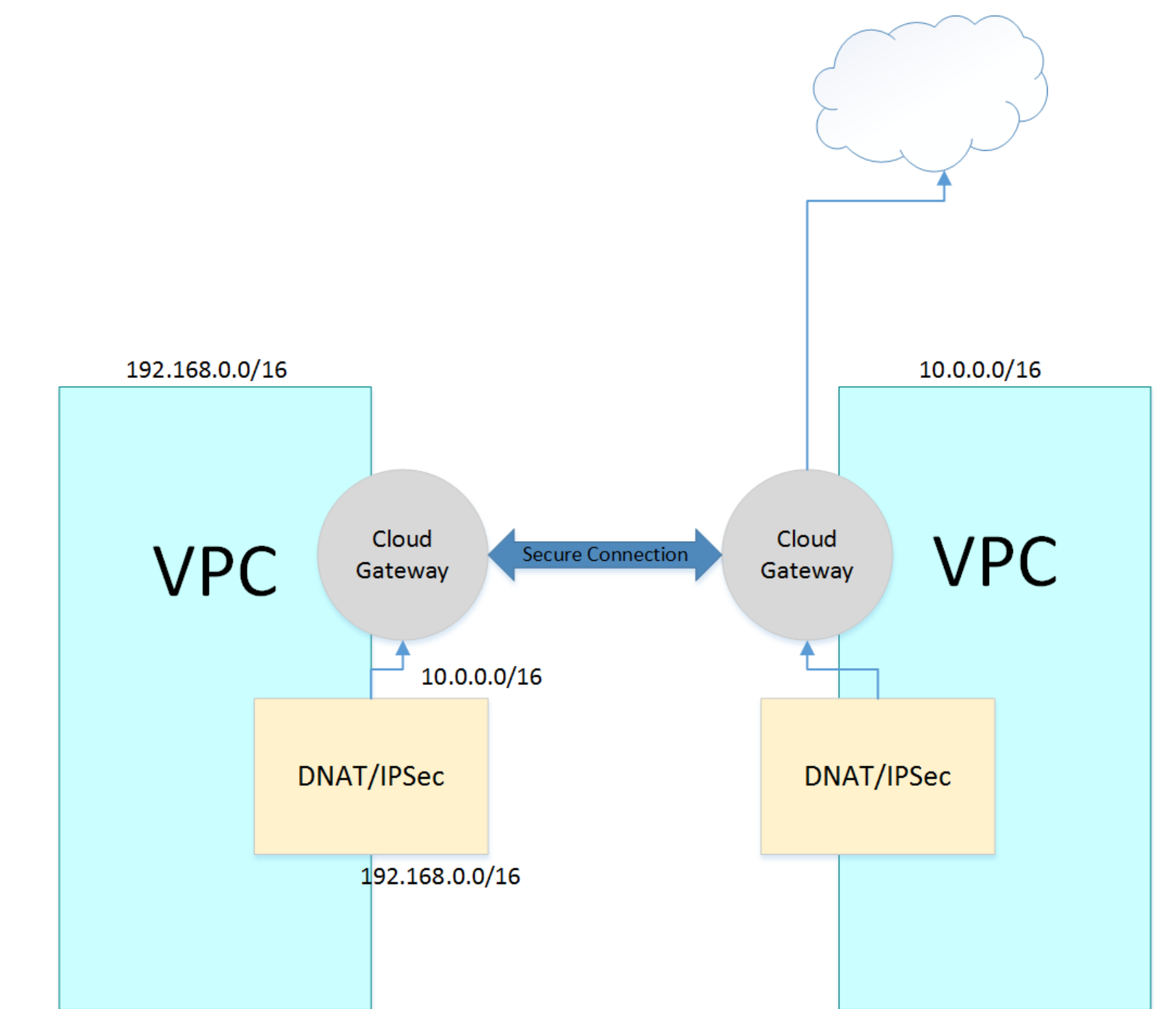


Figure 4: Gateway Setup

## Conclusion

- The hybrid cloud model is a cost-effective method.
- The average-size workload stays in private cloud.
- Machines use CG to send traffic b/w different cloud subnets.

## ACKNOWLEDGEMENTS

Thanks to Kyle Forster and Ted Elhourani in BigSwitch.

## Contact Information

- Massachusetts Open Cloud
- Git Repository