gslab | GAVS

CASE STUDY

Effective Showcasing of Automated Fabric Capabilities through On-Demand VMware Lab

Customer Overview

The customer is a global leader in computer networking products.

The Context

The customer provides an IaaC cloud based platform featuring virtual sandboxed lab environments for each of their products to enable potential customers to get familiar with product features and functionalities.
One of their solutions automatically provisions networks configured in vSphere in physical fabric, delivering Network-as-a-Service. The customer required a pre-provisioned, fully configured lab module with nested virtualization where VMware's vCenter server would be integrated with the controller of this automated fabric.

Type of Service Provided

Product Engineering

Technologies Used

Networking, Nested Virtualization, KVM, VCSA, AWS

The Solution

With rich expertise in product engineering, networking, nested virtualization, and automation, the GS Lab | GAVS team designed the architecture and networking configurations for the lab module and took full responsibility for its development from ground up, testing, bug fixing, documentation, and maintenance.
The team created a nested virtualized environment with two ESXis and the automated fabric controller and developed a tool using Mininet for creating virtual network topology. KVM networking was configured between all components inside the EC2 instance in such a way that the two ESXis get connected to the automated fabric. The controller, Mininet and vCenter appliances were exposed for user access. The VCSA was built with the two ESXis and one host on each ESXi. These hosts were pre-configured with networking capabilities so users could verify connectivity.

The successful delivery of this lab module enabled the customer to effectively showcase the capabilities of their automated fabric.

To find out how GS Lab | GAVS can help your organization, please visit
www.gavstech.com

Challenges

- Increasing demands from potential customers for hands-on experience with products
- Rising need to showcase several major features of wide variety of products
- Need for expertise in product engineering, networking, nested virtualization, and automation to build effective lab module for this product

Solution Highlights

- Designed the architecture and networking configurations of the lab module
- Developed a fully provisioned and configured lab from scratch
- Developed module on AWS EC2 instance to cater to on-demand provision of fully equipped sandboxed environments
- Built EC2 instance using i3.metal type instance for nested virtualization support
- Used the Mininet component inside i3.metal instance for better security and cost savings
- Created an end-to-end automated setup to deliver ready-to-use environment whenever the lab module was launched
- Created automation scripts to spawn components with required networking configurations inside KVM VM and destroy them when done
- Tightly integrated vCenter server with the automated fabric controller
- Enabled high security through internal networking exposing only necessary ports
- Provided cost effective lab module through nested virtualization

Solution Impact

- Enabled effective showcasing of VMware integration capability of automated fabric
- Provided real-time contextual visibility into virtual infrastructure, vSphere activities from the fabric's UI
- Empowered potential customers to gain deep insights into the fabric's capabilities
- Helped the team win the customer's confidence resulting in development requests for
  - New labs for other products (Network Detection and Response, AN + RN, etc.)
  - Labs to showcase integration between multiple products from the customer

Effective Showcasing of Automated Fabric Capabilities through On-Demand VMware Lab

Solution Details

The implementation details of the different components of the lab module are given below:

1. Type 1 Hypervisor

Since a fully equipped sandboxed environment had to be provided to users on demand, the team decided to develop this module on AWS EC2 instance, which would spawn as and when users launched the lab module. The EC2 instance was built using i3.metal type instance because it supports nested virtualization. This served as the type 1 hypervisor. KVM - which is a type 1 hypervisor implemented as a Linux kernel module was used, since KVM has the ability to pass on the virtualization capability to its guest OS.

2. Components

All components are guest VMs inside the KVM VM.

a. Automated Fabric Controller - A specific version ISO was used to bring up this guest VM.

b. Mininet VM – Although this was a separate VM instance for other labs, the team made this a guest VM to gain security and cost benefits. The backend code for VMware lab had to be modified to achieve this.

c. Two EXSi VMs - Two guest VMs with VMware ESXi. Virtualization Acceleration (VT-x) had to be enabled.

d. VCSA Setup - The process of bringing up VCSA on top of KVM is different from the process for a normal guest VM. There needs to be atleast one ESXi first. A major requirement before bringing up the VCSA is to have full DNS resolution and reverse lookup for both the ESXis and the to-be-created VCSA host. The subnet of the ESXis was used for VCSA bring-up. Then, a small guest OS was brought up, a VCSA image was mounted to this VM, and configuration commands were run after VCSA configuration was done.

3. KVM VM Networking

This VM required two types of networking:
a. Internal networking to handle communication between components
b. External networking to enable the user to get hands-on experience on the lab module

4. Automation

Since the lab module is a sandbox environment, it needs to be created when a user launches it and destroyed either when the user is done using it or when its time duration expires. To serve this purpose, the team created an end-to-end automated setup to deliver a ready-to-use environment whenever the user launched the lab module for hands-on experience, and to automatically destroy it as needed. Automation scripts created for this spawned all the different components with the required networking configurations inside KVM VM.

Value Add

The GS Lab | GAVS team suggested using the Mininet component inside the i3.metal instance. This benefited the customer in two ways:

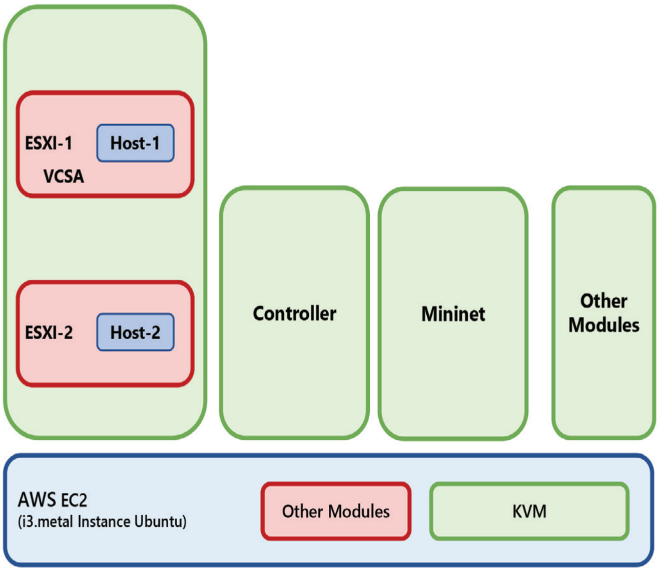
Security – There was no need to expose the external/public ips of the controller and VCSA, thereby increasing the security of the lab module.

Cost – Typically, each lab has 2 instances, where one instance provides information about the lab module and the other instance is the product VM. AWS provides nested virtualization only on the i3.metal type instance but the cost of this type instance is more than that for usual lab VMs. This implied that the customer would have incurred more cost on the lab module had there been two instances. To prevent any additional costs, the team used nested virtualization to incorporate the functionalities of both VMs inside one VM, which helped keep the cost of this lab on par with the cost of other labs.

CASE STUDY

Effective Showcasing of Automated Fabric Capabilities through On-Demand VMware Lab

Architecture Design Diagram



ESXI-1

Host-1

VCSA

ESXI-2

Host-2

Controller

Mininet

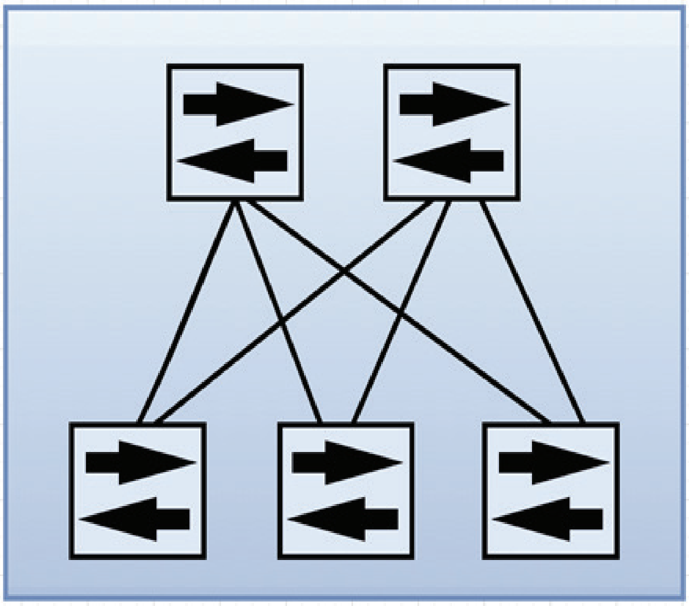
Other Modules

AWS EC2
(i3.metal Instance Ubuntu)

Other Modules

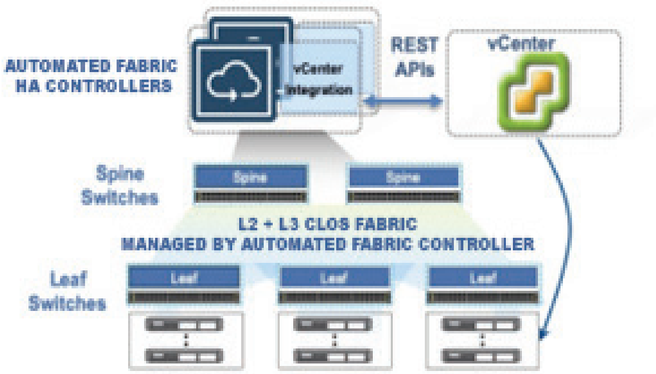
KVM

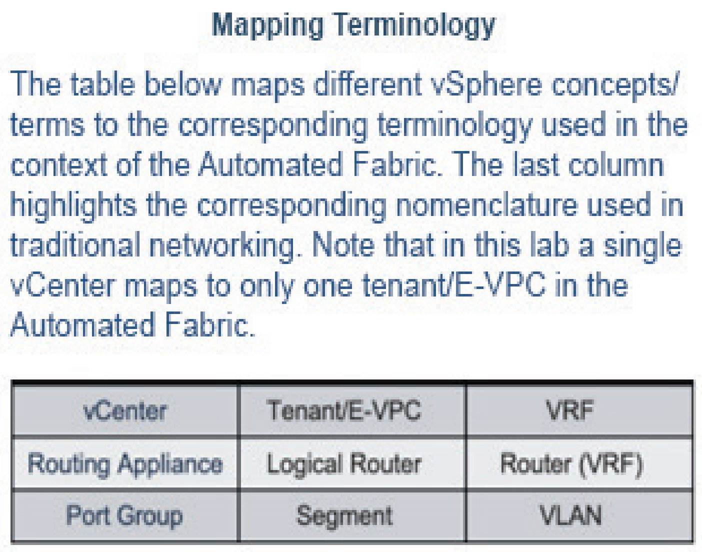
Spine-Leaf Architecture



VMware Integration with the Automated Fabric

Architecture


AUTOMATED FABRIC HA CONTROLLERS
vCenter Integration
REST APIs
vCenter
Spine Switches
Spine
Spine
L2 + L3 CLOS FABRIC MANAGED BY AUTOMATED FABRIC CONTROLLER
Leaf Switches
Leaf
Leaf
Leaf



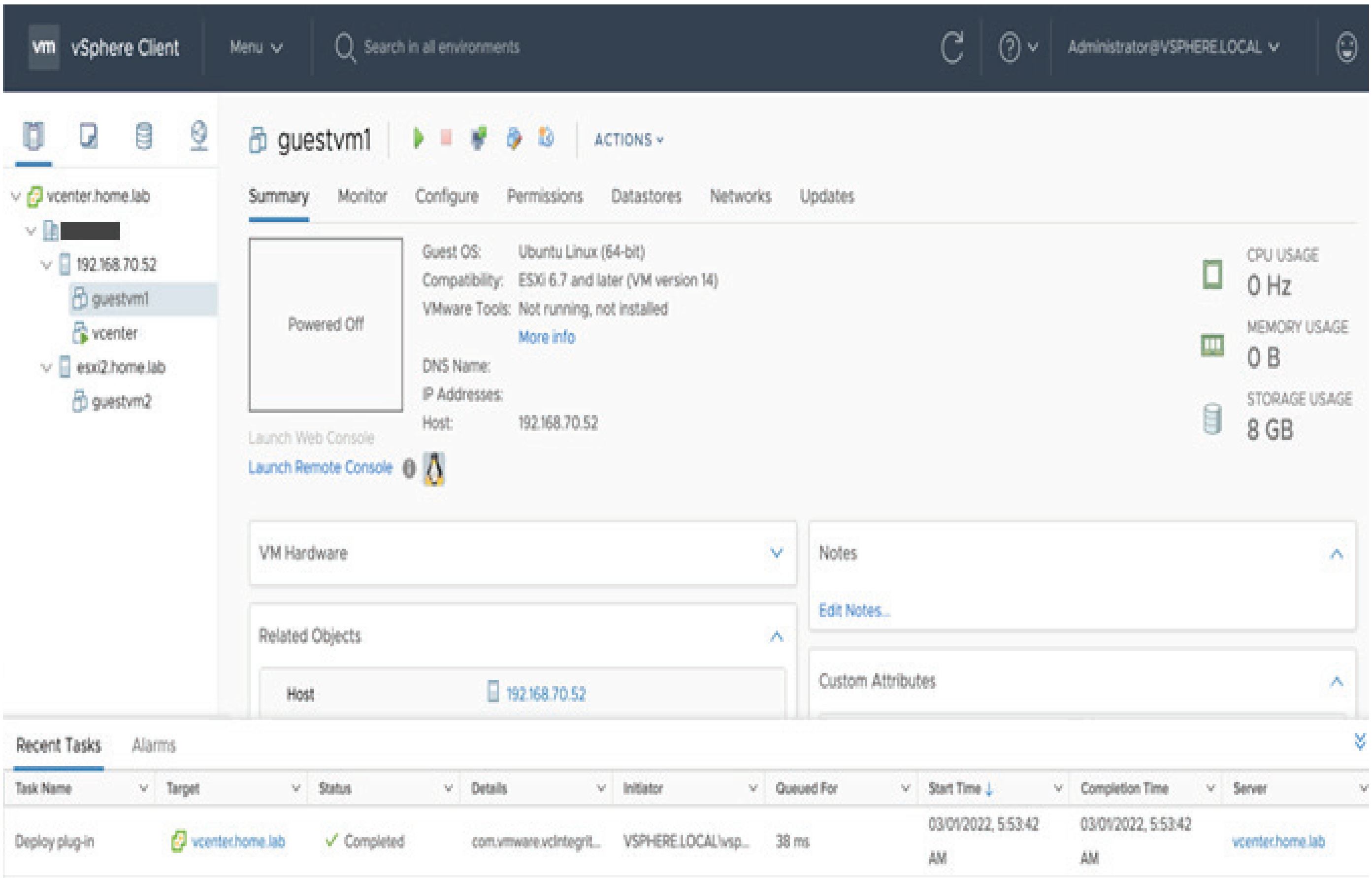
Mapping Terminology

The table below maps different vSphere concepts/ terms to the corresponding terminology used in the context of the Automated Fabric. The last column highlights the corresponding nomenclature used in traditional networking. Note that in this lab a single vCenter maps to only one tenant/E-VPC in the Automated Fabric.

vCenter	Tenant/E-VPC	VRF
Routing Appliance	Logical Router	Router (VRF)
Port Group	Segment	VLAN

To find out how GS Lab | GAVS can help your organization, please visit
www.gavstech.com

VCSA



vm vSphere Client | Menu | Search in all environments | Administrator@VSPHERE.LOCAL

- vcenter.home.lab
  - [redacted]
    - 192.168.70.52
      - guestvm1
      - vcenter
    - esxi2.home.lab
      - guestvm2

guestvm1 | ACTIONS

Summary | Monitor | Configure | Permissions | Datastores | Networks | Updates

Powered Off

Guest OS: Ubuntu Linux (64-bit)
Compatibility: ESXi 6.7 and later (VM version 14)
VMware Tools: Not running, not installed
More info
DNS Name:
IP Addresses:
Host: 192.168.70.52

Launch Web Console
Launch Remote Console

CPU USAGE 0 Hz
MEMORY USAGE 0 B
STORAGE USAGE 8 GB

VM Hardware

Notes
Edit Notes...

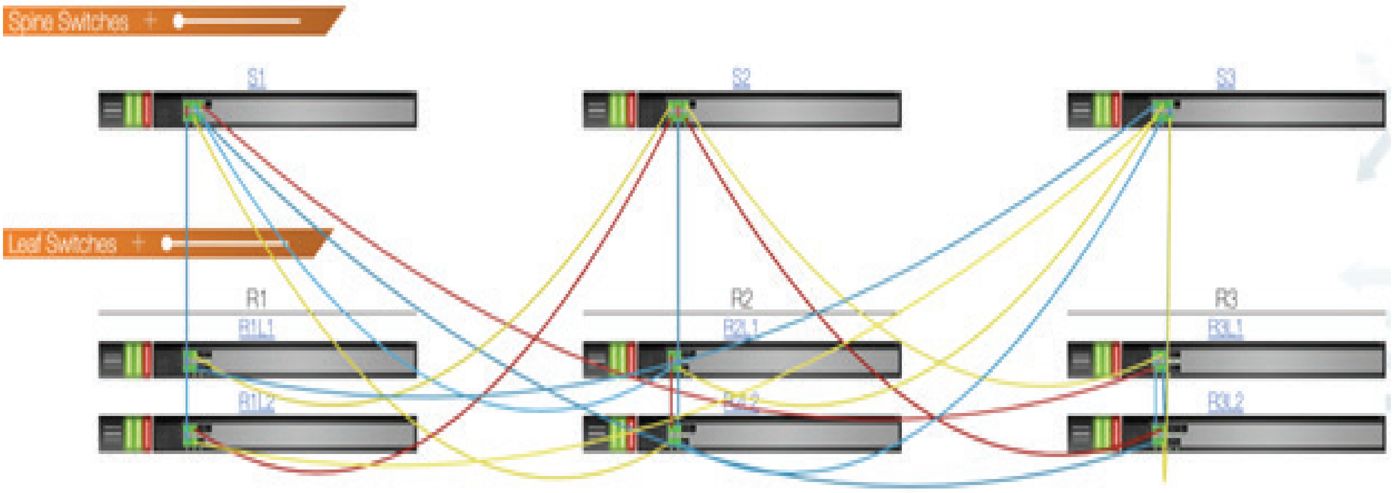
Related Objects
Host 192.168.70.52

Custom Attributes

Recent Tasks | Alarms

Task Name	Target	Status	Details	Initiator	Queued For	Start Time ↓	Completion Time	Server
Deploy plug-in	vcenter.home.lab	✓ Completed	com.vmware.vcIntegrit...	VSPHERE.LOCAL\vsp...	38 ms	03/01/2022, 5:53:42 AM	03/01/2022, 5:53:42 AM	vcenter.home.lab

Spine-Leaf in Controller UI


Spine Switches
S1
S2
S3
Leaf Switches
R1
B1L1
B1L2
R2
B2L1
B2L2
R3
B3L1
B3L2