VIAVI

VIAVI Solutions

VIAVI
TeraVM
for Amazon Elastic Compute Cloud (EC2)

Introduction

Amazon Elastic Compute Cloud (EC2)* is a central part of Amazon.com's cloud computing platform, Amazon Web Services (AWS). EC2 allows users to rent virtual computers on which to run their own computer applications. EC2 allows scalable deployment of applications by providing a web service through which a user can boot an Amazon Machine Image (AMI) to create a virtual machine, which Amazon calls an "EC2 instance". A user can create, launch, and terminate server instances as needed, paying by the hour for active servers, hence the term "elastic".

With the evolution of EC2 some network equipment manufacturers (NEMs), enterprises and service providers have begun to use EC2 as a test or deployment platform for their virtual or cloud offerings. To meet the test needs of these companies Shenick offers a TeraVM AMI which can be deployed in EC2 (VPC service only) on demand. Simply put a TeraVM AMI (Amazon Machine Image) is TeraVM software delivered in an Amazon specified format that contains all information necessary to boot an EC2 instance.
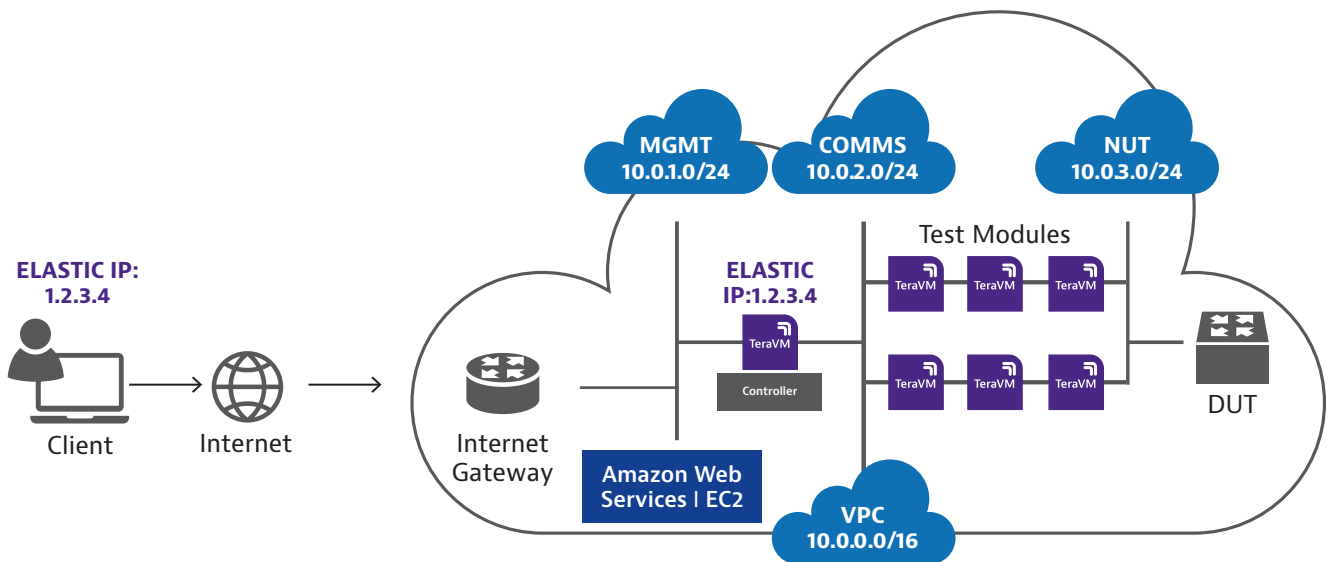

Figure 1: AWS Deployment

Amazon VPC (Virtual Private Cloud)

TeraVM AMI requires use of the Amazon VPC (Virtual Private Cloud) service. Amazon "EC2-Classic" is not supported. Amazon VPC lets users provision a logically isolated section of the Amazon Web Services (AWS) Cloud where they can launch AWS resources (e.g. EC2 instances) in a virtual network that they define. With VPC, the customer has complete control over the virtual networking environment, including selection of their own IP address range, creation of subnets, and configuration of route tables and network gateways.

TeraVM Deployment and Operation

Figure 1 shows a deployment of TeraVM AMI in an Amazon VPC. Components are logically separated from each other using different subnets to provide ease of configuration and maximum security.
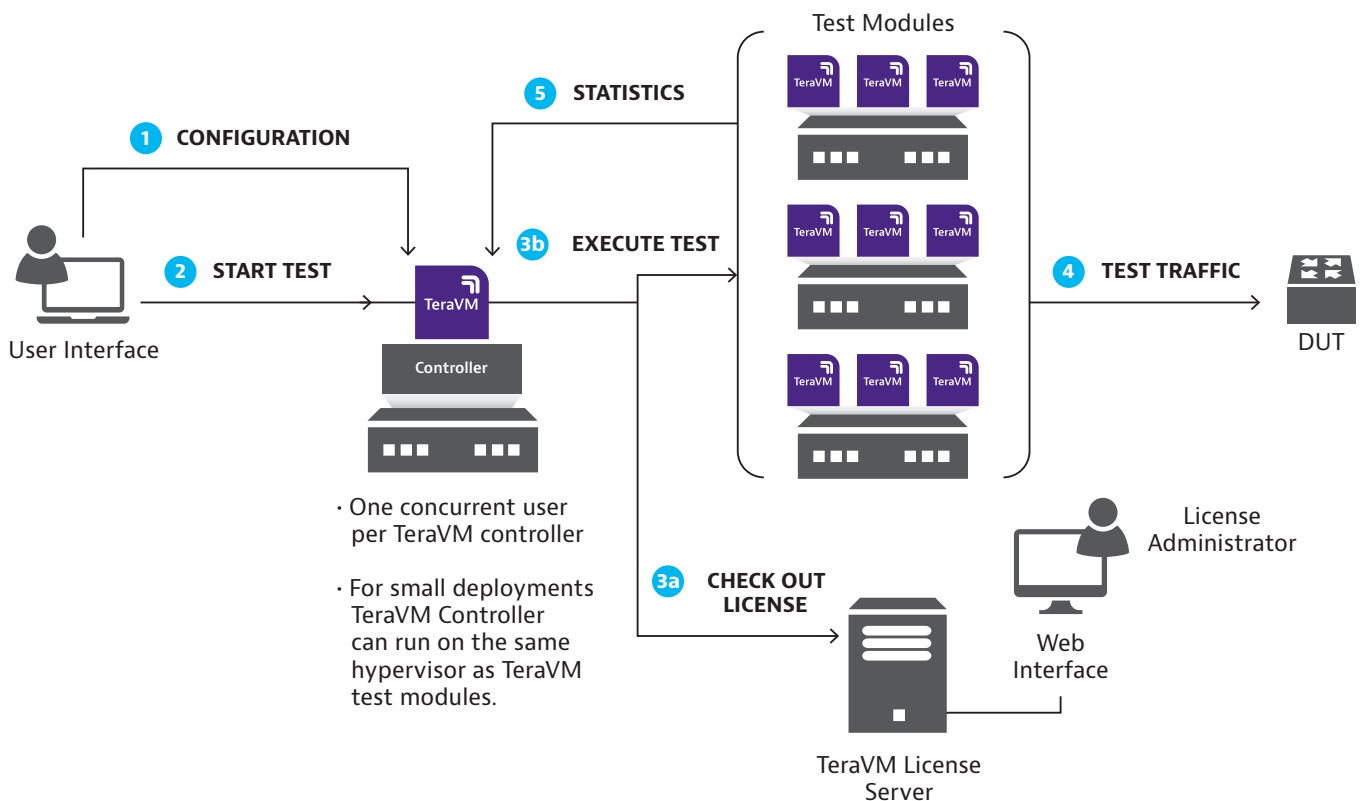

Figure 2: Typical TeraVM on Premise Deployment

1. Client Interface – A graphical or command line interface used to configure and execute tests as well as output related statistics or metrics.

2. TeraVM Controller (TVM-C) – A centralized management entity that acts as the interface between the client interface and the test modules. We refer to it as a Controller because it "controls" the actions of one or more test modules.

3. TeraVM (TVM) test modules – The entities which generate the actual test traffic such as HTTP, VoIP or email. The test module is the reason why a customer buys TeraVM and hence when we generically refer to "TeraVM" we are referring to the actual test modules and not the controller or anything else.

4. TeraVM License Server (TVM-LS) – A common, centralized repository that contains all authorized test licenses. A TVM-C consults with the TVM-LS before executing a test to make sure a valid license is available.

The aforementioned TeraVM components have been adapted to run in the Amazon EC2 environment. The major change is with the TeraVM Controller (TVM-C) and TeraVM (TVM) test modules. These have both been adapted to the Amazon Machine Image (AMI) format and hence when we refer to "TeraVM AMI" we are referring to the TeraVM product that is specifically designed to run in Amazon EC2. TeraVM AMI is not publicly available. It is a private AMI that is shared by VIAVI with customers on an individual basis. As a result, TeraVM AMI is not available in the Amazon AWS Marketplace or any other public forum at this time.

Customers who wish to evaluate or purchase the TeraVM AMI should contact their sales representative.

To begin deploying TeraVM AMI, first an AWS VPC has to be created. Any other private address space can be used with subnets created accordingly.

With the evolution of EC2 some network equipment manufacturers (NEMs), enterprises and service providers have begun to use EC2 as a test or deployment platform for their virtual or cloud offerings. To meet the test needs of these companies Shenick offers a TeraVM AMI which can be deployed in EC2 (VPC service only) on demand. Simply put a TeraVM AMI (Amazon Machine Image) is TeraVM software delivered in an Amazon specified format that contains all information necessary to boot an EC2 instance.

Let's look at each TeraVM AMI component individually in the context of Amazon EC2. The client interface is a Java GUI which is installed locally on any computer. This Java client is part of the overall TeraVM AMI package. The Java GUI client directly accesses the TVM-C via an Amazon elastic IP address which must be configured as part of an Elastic Network Interface (ENI) for the TVM-C. The TVM-C is on a management subnet (shown as subnet 10.0.1.0/24) with an ENI associated with the TVM-Cs private IP address. The TVM test modules are deployed and communicate with the TVM-C via the "communication" or "comms" network (shown as subnet 10.0.2.0/24). Once a TVM-C is instantiated it can control all TVM test modules on the same comms network and facilitate the user to generate the required test traffic profiles. If more test traffic is required the user simply deploys more TVM test modules on the same comms network. A single TVM-C can manage or control up to 500 TVM test modules.

Finally the other side of the TVM test modules communicate on a "NUT" or "Network Under Test" network (shown as subnet 10.0.3.0/24) to a device under test (DUT). The DUT can be located in AWS or across the Internet in some other location such as a private data center.

Note: It is recommended that all TeraVM EC2 instances are deployed from a contiguous and consistent block of IP addresses.

For the TVM-LS there are two deployment options which can be selected based upon convenience.

- Deploy a system specific TVM-LS in Amazon EC2, where a number of TVM test modules are licensed in advance and dedicated to a single specific TVM-C.

- Deploy TVM-LS in a private data center. Clearly, the private data center must be IP reachable from Amazon EC2 for this deployment model to work. A secure connection between TVM-C in Amazon EC2 and TVM-LS in the private data center is recommended if this deployment model is chosen.

Note: All TeraVM AMIs are deployed via automation scripts and only one TVM-C per VPC subnet is possible.

EC2 Instance Types

Amazon EC2 instances are the fundamental building blocks within the AWS cloud. One can think of instances as virtual servers that can run applications such as TVM-C or TVM. Instances are created by choosing an Amazon Machine Image (AMI) and the appropriate instance type.

Details on all the available instance types can be found on the AWS website as follows:

http://aws.amazon.com/ec2/instance-types/#instance-details

There are many different EC2 instance types to choose from within AWS. At a high level, instance types are grouped in to 6 families as follows:

1. General Purpose – Provide a balance of compute, memory, and network resources, and are a good choice for many applications.

2. Compute Optimized – Optimized for applications that benefit from high compute power. Compute-optimized instances have a higher ratio of vCPUs to memory than other families and the lowest cost per vCPU among all Amazon EC2 instance types.

3. Memory Optimized – Optimized for memory-intensive applications. Memory-optimized instances use Intel Xeon processors and have the lowest cost per GiB of RAM among Amazon EC2 instance types.

4. Storage Optimized – Provides user with Intel Xeon processors and direct-attached storage options optimized for applications with specific disk I/O and storage capacity requirements. Currently, there are two types of storage-optimized instances. HI1 instances provide very fast SSD-backed instance storage capable of supporting over 120,000 random read IOPS, and are optimized for very high random I/O performance and low cost per IOPS. HS1 instances are optimized for very high storage density, low storage cost, and high sequential I/O performance.

5. GPU Instance – Allows users to take advantage of the parallel performance of NVidia Tesla GPUs using the CUDA and OpenCL programming models for GPGPU computing. GPU instances also provide high CPU capabilities (Intel Xeon X5570) and support cluster networking.

6. Micro Instance – Micro instances are a very low-cost instance option, providing a small amount of CPU resources. Micro instances may opportunistically increase CPU capacity in short bursts when additional cycles are available. They are well suited for lower through-put applications and websites that require additional compute cycles periodically, but are not appropriate for applications that require sustained CPU performance.

For TeraVM AMI we recommend the general purpose instances because they provide a good balance between compute and networking which are the key drivers for TeraVM AMI. Below are specific instance types that are recommended.

TeraVM Controller Recommended Instance Types

For TeraVM Controller (TVM-C) we recommend the "m1.medium" general purpose instance type. This instance type provides a good balance between compute power and networking performance.

TeraVM Test Module Recommended Instance Types

For TeraVM test modules we recommend the "m1.medium" general purpose instance type. This instance type provides a good balance between compute power and networking performance.

Use Cases

TeraVM AMI can be used to test the security, performance and functionality of the hybrid enterprise in both predeployment and post-deployment stages. TeraVM AMI can also be used to provide a cost-effective test platform for testing virtualized network functions such as switches, routers, firewalls and load balancers in Amazon EC2.

Hybrid Enterprise

An automatic level control facility is provided for both of the external modulation inputs and provides correctly calibrated modulation for input levels varying from 0.7 V to 1.4 V RMS. HI and LO indications show when the input level is outside the range of the ALC system.

A data center is common place at most organizations and according to industry research up to half of private enterprises will implement a private cloud within the next three years. Along with this push towards "cloudification" is the inevitable trend towards the hybrid enterprise where there is both a private and public cloud component to the IT infrastructure. The economics of utilizing public clouds is very compelling and the major driving factor behind ongoing adoption. However, with the move towards a hybrid enterprise comes additional complexity in terms of security, predictable performance and even simple reachability. To cope with this additional complexity a simple but comprehensive test and validation regime must be put in to place. TeraVM and specifically TeraVM AMI can help in both predeployment and post-deployment scenarios.

TeraVM Test Module Recommended Instance Types

For TeraVM test modules we recommend the "m1.medium" general purpose instance type. This instance type provides a good balance between compute power and networking performance.

Figure 3 depicts a typical hybrid enterprise deployment with the enterprise data center or private cloud on the left and Amazon EC2 as the public cloud on the right. The connection between the private and public clouds is a secure tunnel with an Internet or Cloud Gateway device on both sides. TeraVM is deployed on both sides to simulate realistic usage scenarios (e.g. emulate voice, video and data traffic) in order to verify the security, performance, and scalability of the entire private and public infrastructure. At the simplest level TeraVM can be used to verify that
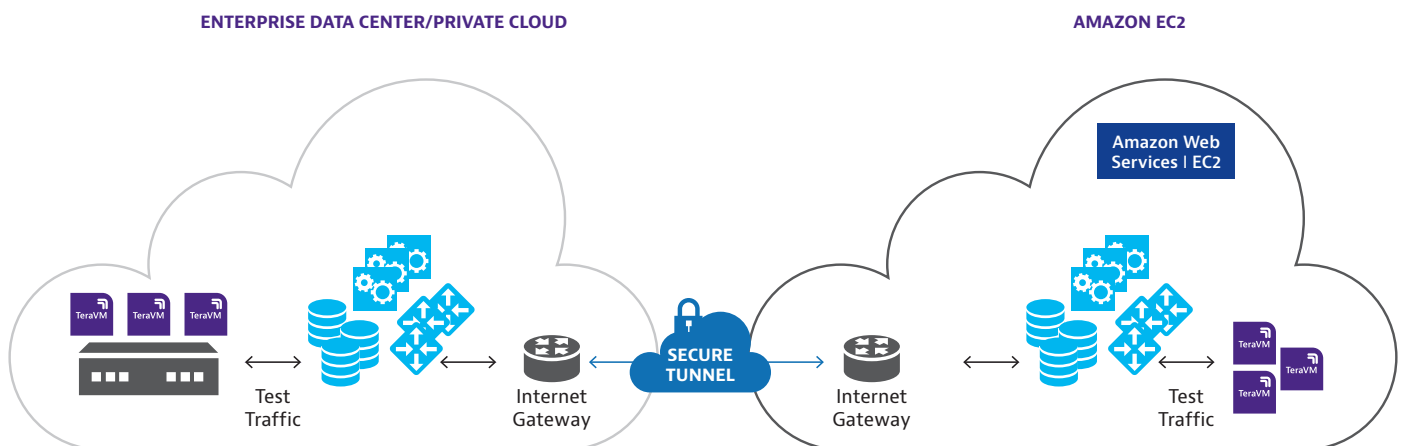

Figure 3: Hybrid Enterprise

*Amazon Elastic Compute Cloud (EC2) is owned by Amazon Internet Services Private Limited.

there is end to end application level connectivity within the entire infrastructure and if there are problems TeraVM can help pinpoint where the problem is occurring. TeraVM can also be used to exercise the various components in the infrastructure such as the Internet Gateway to ensure that all scalability requirements are met. Finally TeraVM can be used to verify that security policies are acting properly and that there is no associated security leakage.

Post Deployment

Once TeraVM has been used to validate the initial deployment, all instances should be left in place for post deployment validation purposes. The post deployment validation might be periodic tests to verify security, performance and functionality as network changes are made or active monitoring where TeraVM instances are configured to send a continuous stream of traffic between them. In the active monitoring scenario traffic could be used to measure or record the loss, latency and jitter of the hybrid network over time. Also as VMs are dynamically deployed in the cloud or infrastructure is changed (e.g. new code versions) the TeraVMs can be used to perform quick and easy regression tests to ensure the network continues to be healthy.

Network Functions Virtualization (NFV)

With ever-demanding schedules placed on network equipment manufacturers, Amazon EC2 offers a rapid deployment, hassle-free environment for both functional and regression testing. Many traditional IP networking products (e.g. switches, routers, firewalls, load balancers, mobile LTE components) have virtual versions which are released for use in a cloud environment and TeraVM AMI offers the ability to easily test functionality directly within Amazon EC2.

Using orchestration, each daily product build can be uploaded as an EC2 instance and surrounded by TeraVM so that functionality and performance can be measured on a daily basis within EC2. The additional flexibility helps reduce test environment set up costs and works efficiently with a distributed development team where daily changes to code are verified remotely in the Amazon cloud.