TRESBU
technologies

# NVIDIA

## Cloud-Based GPU Management Enables Deep Learning Appliance Control

*white paper*

## Abstract

NVIDIA is perhaps the world's most innovative chip maker, single handedly responsible for inventing the GPU and igniting a global universe of industries built on parallel computing.

Its capacity as a world leader means the level of commitment to leadership in innovation requires massive investment in research and development. Continuous product ingenuity is the essence of the company DNA that drives most or all decision making at every level.

As such, in late 2016, engineering leaders sought ways to improve the management of bought hardware and software in a multi-tenant test environment where deep learning appliances could be more effectively managed.

The previous environment had on the capability to manage deep learning appliances with a stand alone command line interface. Although each appliance had its own CLI interface, there was no way for customers to manage multiple deployments from a single, central console.

## Proof of Concept

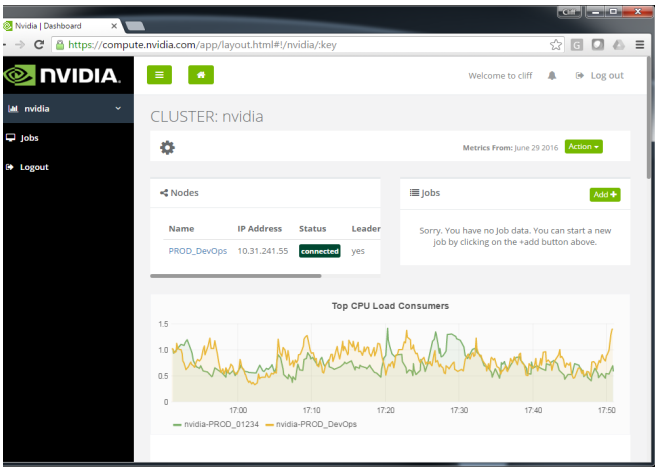
In order to efficiently evaluate and test the viability of the concept, a proof-of-concept was conceived and built using Agile methods and a scrum framework with a small team from Tresbu Technologies, Inc.

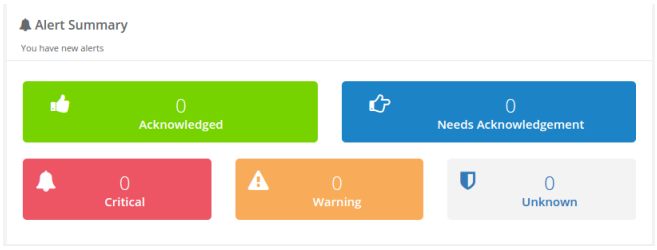
Using results from the PoC, Tresbu then built a cloud-based portal for managing the deep learning appliances.

The cloud solution also brought into focus a host of new challenges that had not been considered initially – like multi-tenancy, reliability, scalability, security, etc.

The proof of concept was completed for NVIDIA in 4 weeks, demonstrating all of the functional needs for final project build

## Solution






Tresbu built a secure, scalable, reliable, multi-tenant, AWS-based portal which is used by customers to manage deep learning hardware. The hardware gets deployed in the customers' own data centers, while the management portal running on the public cloud provided a browser based GUI.

The GUI had a unified multi-tier dashboard – to view the resource (CPU, GPU, RAM, HDD) utilization – both collectively and individually of all the deployed hardware. It

allowed customers' users to add/remove docker containers via the GUI. It also had an admin dashboard to add/remove tenants; view basic analytics of the deployed hardware; and perform other housekeeping functions.

https://www.nvidia.com/en-us/gpu-cloud/


# Specifications

## Components

- DGX appliances – robust systems with top-end NVIDIA GPUs. Each appliance contains 8x GPUs (12GB+ VRAM each), 8x CPUs (12 cores each), 512GB RAM, & 1TB SSDs and run a customized version of Ubuntu which is optimized for running DL compute loads.

- Cloud server – this has a web server, an application server, a docker repo, and databases. This allows multiple tenants to login to their own environments, view all their appliances, schedule DL jobs on their appliances, monitor the vital stats of each appliance, or a cluster of appliances, & set customized alerts on any of the vital stats. Stats include CPU/GPU/RAM utilization, disk I/O & usage.

## Architecture

The appliances are located on-premises for each tenant. They communicate to the cloud only via a 2-way authenticated secure connection.

All the cloud components are hosted in Amazon AWS, & use various features of the AWS cloud, eg: Virtual Private Cloud, Elastic Load Balancer, ElasticCache, S3, etc. - Illustrated visually in the images on the next page.

## Design

**Appliance Components**

- Collectd – for collecting all the necessary system stats & posting them as a time series to the cloud server.

- Marathon & Chronos – job scheduling & orchestration services for starting/stopping/queueing jobs. Jobs are instances of a docker container, with the necessary data-sets to run the DL loads. Marathon is for running long-running and/or repeating jobs, while Chronos is used to run one-time jobs.

- Docker daemon – for running the individual jobs. Each job is posted to docker only via Marathon or Chronos.

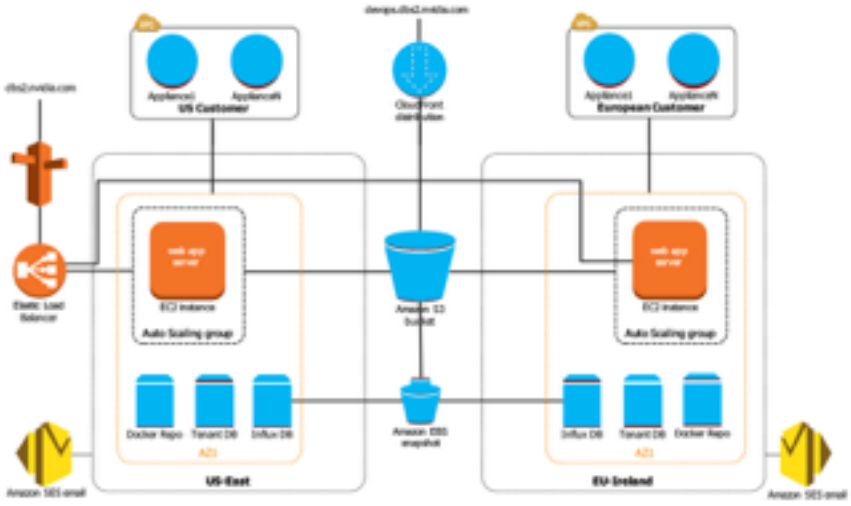
**Back End Components**

- Nginx – acts as a web server & a reverse proxy for all the other services running in the system.
- Tomcat – application server, which provides the control for each tenant to do their work via a web based GUI. All the server-side functionality is exposed thru REST APIs. It interacts with Marathon & Chronos on the appliance as well.
- MySQL – DB for storing tenants, users, their appliances, their privileges, & other data.
- InfluxDB – a time-series DB for storing all the data getting logged from Collectd.
- Bosun – an alert system for time-series data.
- Quay – a docker repository, where tenants could upload their own docker images & containers. Each container is customized for executing specific types of DL jobs.

**Front End Components**

- Angular.js GUI – for client side computation, GUI rendering, & validation. It allows creation of a sophisticated browser based UI – for adding/updating/deleting docker containers, scheduling & monitoring/deleting jobs, view vital stats for any appliance or a cluster of appliances.
- Graphana – for displaying all the time-series data in interactive graphs.

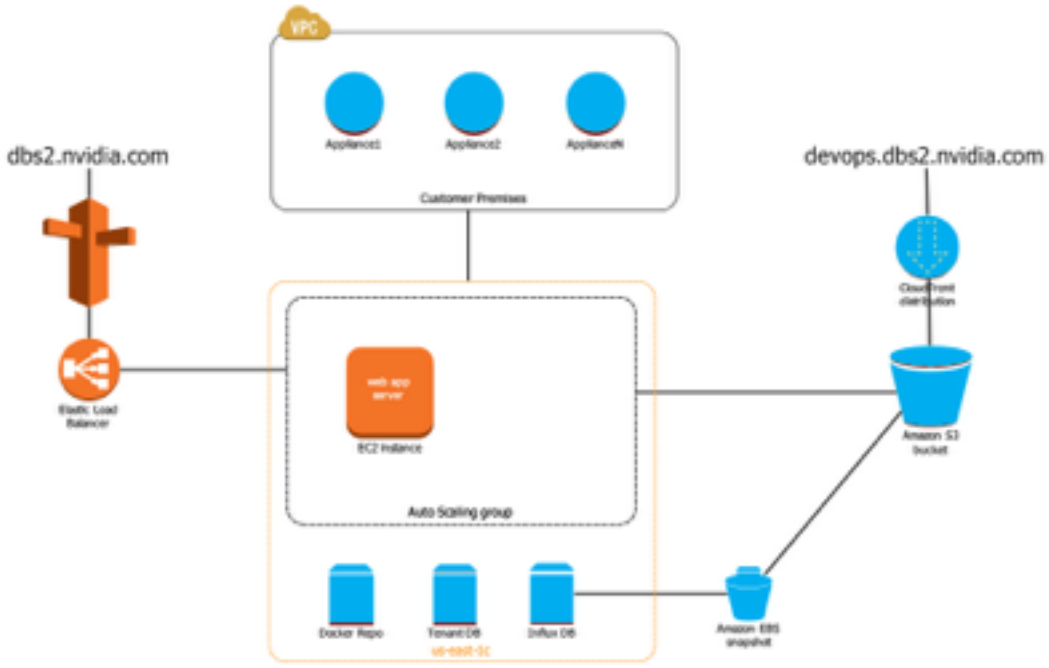
Global Delivery Architecture



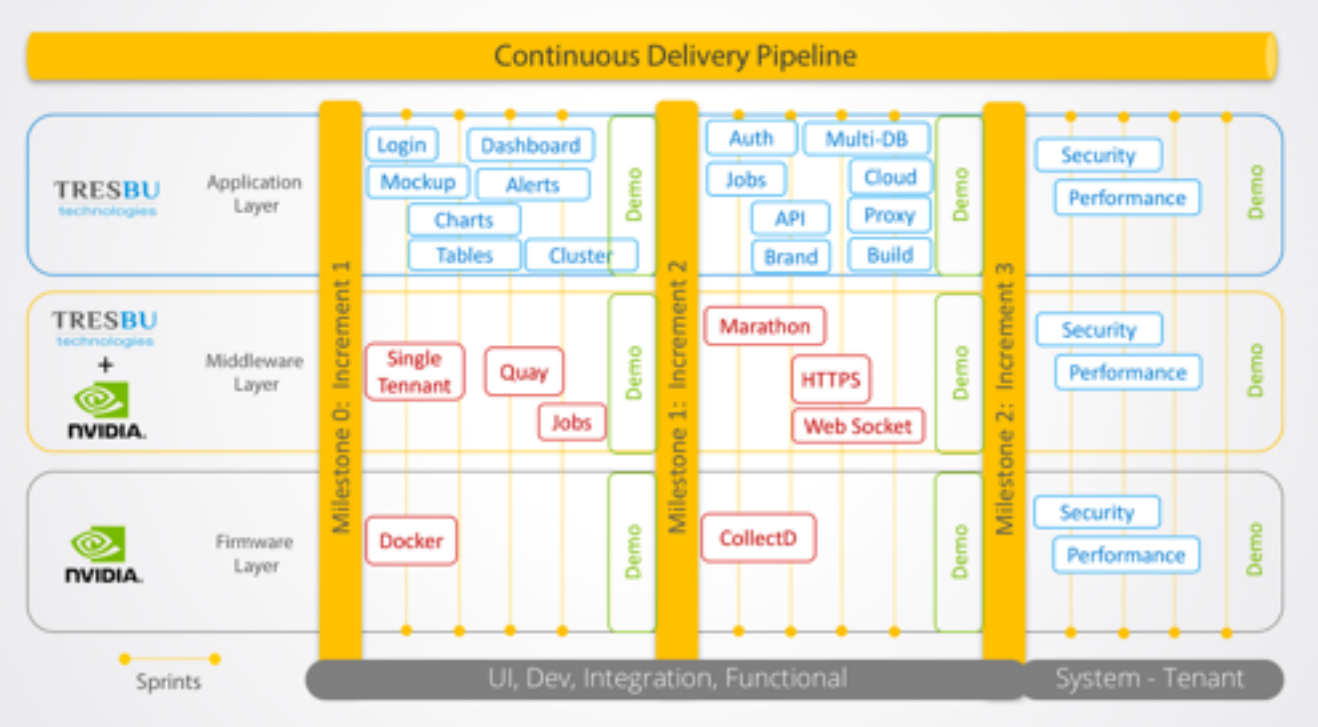
## Workflow

- The user creates their docker containers to perform specific DL computations. Then uploads the containers to the docker repo in the cloud. This is done outside the purview of the system.
- After the upload is successful, the container becomes visible in the GUI. Now the user can schedule a job by selecting a container, the data path (training set data), & other attributes necessary for the job.
- The GUI posts the details to the server via REST APIs, & the server posts the job to the 1st free appliance it finds. If all appliances are busy, it posts it to the least-recently used appliance, where Marathon or Chronos will queue it, depending on whether it's a repeating or a one-time job.
- The user can then view their job in the queue, or if it's running interact with it via the container's UI, which can also be opened from the browser UI.
- The user can view the vital stats of the appliance while their job is running to understand the efficiency of their DL algorithms.
- Finally, they can view historical reports of the jobs, appliances, etc.

MVP Architecture



## Scaled Agile Integration




## Organizational Impact

Many look to NVIDIA for the future of parallel computing. As the leading GPU producer in the world, NVIDIA breaks barriers, processing large volumes of data faster than anyone had thought possible. In order to meet such high standards, setting the pace for research and development is a sacrosanct commitment that is tied directly to NVIDIA's success.

By building a performance testing portal that allowed users to manage multiple GPU deployments across multiple deep learning appliances from a single interface, NVIDIA was able to have greater test environment visibility, compress development cycles, and reduce costs. More importantly, NVIDIA is now able to offer their customers more efficiency, improved usability, and an overall better experience.

In order to better support their customers, NVIDIA needed a trusted resource and partner that they could rely on to meet the quality standards that make NVIDIA the world leader in their industry. They found that partner in Tresbu Technologies.

www.tresbu.com