Business white paper

HP Cloud Application PaaS Architecture

How it works and why.

HP Cloud Application PaaS: How it works and why.

Traditionally, applications were hosted directly on the physical server that ran them, with only the OS layer between the application software and the "bare metal". This method yielded some efficiencies because the software stack was relatively thin, but the model was more difficult to scale because physical server deployment is a very manual process.

With Moore's Law providing faster processors and more memory, there was room for another layer of abstraction. With IaaS providers and private clouds, virtual machine (VM) instances can be easily created on demand. This speeds up the process of server allocation, but applications still require supporting software to be installed and configured on each server.

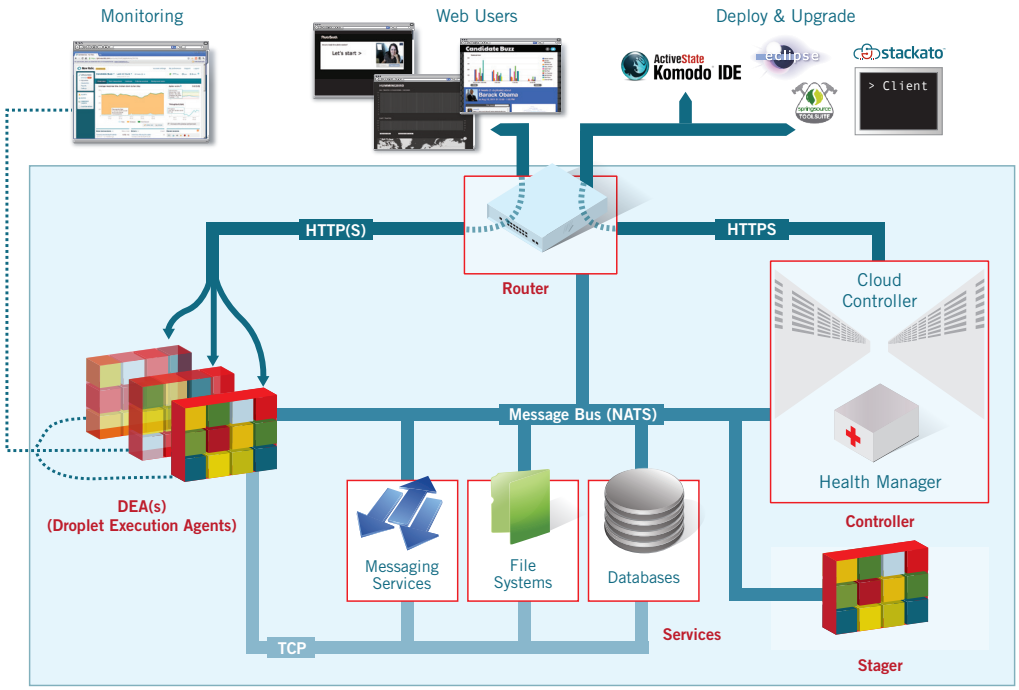
The PaaS model provides an additional level of automation while sharing resources between multiple tenants and applications. This is why application hosting from PaaS providers is generally less expensive than VM hosting from an IaaS. Typically PaaS providers have to limit the range of runtimes and services offered and strictly control access to the system.

HP Cloud Application PaaS (aPaaS) combines the flexibility offered by direct VM access on IaaS with the highly automated configuration provided by PaaS. Computing resources are shared efficiently and securely by giving each application its own Linux container (using LXC) which can be extensively customized to suit the application it is hosting.

Elements of the HP Cloud aPaaS System

A single HP Cloud aPaaS VM instance can run all services required for operation. This configuration, called a micro cloud, can be used on a desktop hypervisor such as VirtualBox or VMware Fusion to emulate a HP Cloud aPaaS.

This same VM is also the building block used to construct a full-scale PaaS cluster, using a simple command to assign a role to each VM and connect them together.



How HP Cloud aPaaS works on a VM cluster.

HP Cloud aPaaS powered by stackato from ActiveState

These HP Cloud aPaaS roles are:

Cloud Controller: responsible for directing all of the elements of the system. The attached Health Monitor keeps track of DEA availability. All HP Cloud aPaaS VMs communicate with the Cloud Controller over NATS, "a lightweight publish-subscribe and distributed queuing messaging system." This system handles the orchestration of the components and sets up other communication channels between components as appropriate (e.g. TCP connections between apps and databases). During cluster configuration, all VMs are connected to the Cloud Controller via NATS.

Stager: responsible for assembling all of the software required to run each application. Application prerequisites are bundled together in an application "droplet" ready for deployment to a Droplet Execution Agent (DEA).

Router: maps application URLs to the application instances running on the DEAs. Application users connecting from the web are redirected transparently to an internal URL and port. Connections from HP Cloud aPaaS clients are routed directly to the Cloud Controller.

DEAs: the worker nodes of the system. Each hosts multiple applications within separate Linux containers. Application droplets are pulled from the Cloud Controller and launched inside a pre-allocated container. If a DEA becomes unresponsive for any reason, the Health Manager will notice and have the Cloud Controller redeploy the assigned applications to "healthy" DEAs.

Services: database, messaging, file system, and other services can be automatically provisioned by the Cloud Controller. Services are bound to applications by way of a special environment variable that exposes the connection information in the application container. Any services requested by a user can be bound to any application deployed by that user, so multiple services can be bound to an application and vice versa. Services can be run on separate VMs, or can share VMs as required. Applications can use external services as well. For example, if an existing high-availability database cluster is exposed to the DEAs, the applications can connect directly, as it would in a traditional application hosting scenario.

Pushing Applications

The HP Cloud aPaaS client 'targets' the API URL of the HP Cloud aPaaS system and is authenticated with the 'HP Cloud aPaaS login' command. Users have a quota of application instances, services and reservable memory available to them.

The 'HP Cloud aPaaS push' command initiates the process of deploying application code to HP Cloud aPaaS. The client either prompts for configuration information (application name, required services, how much memory to reserve, etc.) or reads information from a local YAML configuration file.

HP Cloud aPaaS checks to see if all the resources required by the application are available to the user (within quotas, services and runtimes available). If they are, the client bundles the application directory and uploads it to the Cloud Controller.

An application URL is created and any requested services are provisioned. A temporary container is created in the Stager, application dependencies are assembled, and a "droplet" package is created for deployment. The Cloud Controller queries the DEAs and requests app containers. An available DEA starts a container and downloads the droplet, then executes any 'pre-running' hooks specified in stackato.yml before starting the application.

4AA0-xxxxENW, Created November 2012

This is an HP Indigo digital print.