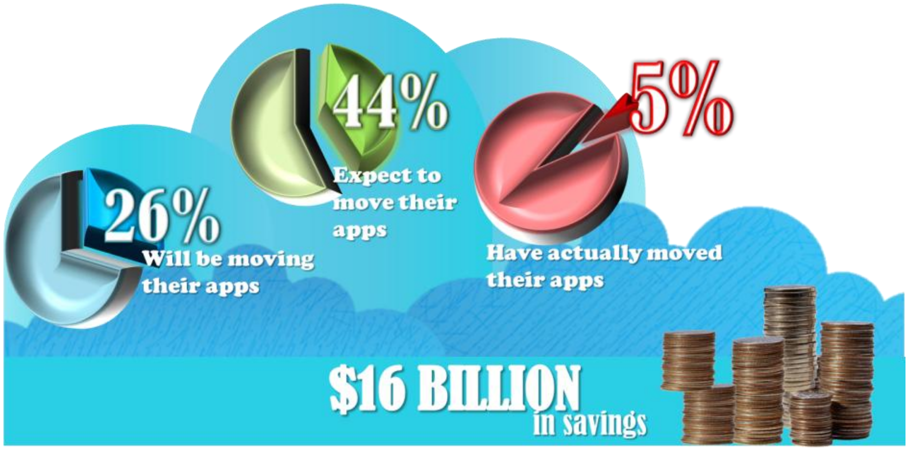
Cloudify for HP Cloud Services

About HP Cloud Services



HP Cloud Services delivers leading edge public cloud infrastructure, platform services, and cloud solutions for developers, ISVs, partners, service providers, and enterprises along with a developer-friendly integration of OpenStack™ technology. HP is uniquely positioned to provide enterprise-ready public cloud infrastructure that's business grade, open source-based, and developer focused, with its long-standing footprint and focus on enterprise technology.

Cloudify & HP Cloud Services

Moving Enterprise Workloads to the Cloud on a Massive Scale

Cloudify is designed to bring enterprise workloads to the cloud by enabling enterprise applications to quickly benefit from cloud automation and elasticity. Cloudify enables enterprise organizations to maximize application onboarding and automation by externally orchestrating the application deployment and runtime. This partnered with HP's robust Cloud Services portfolio, enables enterprises to this on a massive scale by facilitating a number of core principles:

- **No Code Change**
  Any change, even the slightest one, to your existing apps will slow down migration efforts substantially. Furthermore, many apps in use in enterprises today are third-party apps, where there is often times no control over the application code, so changing the app is many times not an option. The primary benefits of moving apps to the cloud include: automation, ease of change, and configuration. There are obviously plenty more benefits if the apps are changed to take better advantage of the cloud infrastructure, but the right order would be to do this after migrating the app to the cloud in the first place. In many cases, just automating the way apps are run and making them available on-demand is sufficient enough. Because Cloudify works on the process level, and uses a recipe-based approach it doesn't require any code change to get your application Cloudified.

- **Plug-in to the Current Way of Running Apps**
  The common way for running apps today is either through custom shell scripts in the simple case, or through configuration management tools such as Chef or Puppet in the more advanced case. To migrate apps on a massive scale, there needs to be a way to plug-in to the most common environments, and make it possible to import them nearly as is. This ensures that the knowledge and know-how that has already been invested into these systems is still leveraged, as well as significantly shortens the migration process. Cloudify recipes were designed to plug-in to diverse application environment configurations and setups. This currently includes basic shell/batch files in the simple scenario, as well as configuration management with tools such as Chef or Puppet in more advanced scenarios.

- **Using a Shared Infrastructure Across the Stack Regardless of the Language**

Using different frameworks for each element of the application stack based on the language and environment is not going to scale for obvious reasons, such as complexity, a large number of moving parts, inconsistency, among other reasons. Cloudify uses the same infrastructure to automate the deployment, failover and scaling, or in other words, a large variety of workloads – starting with simple web apps to big data apps. All this is possible due to the process + recipe-based approach outlined earlier. In addition, Cloudify's abstraction of the underlying infrastructure details makes it possible to use the exact same application recipe on different target platforms whether they happen to be on public, private and bare-metal environments.

- **Add Cloud Properties from the Outside**

There are many cloud properties that can be applied to existing apps without forcing any changes. For example, automating the scaling from small to bigger machines, automating the configuration, as well as the setup. This significantly simplifies the way the app is run by making it available on-demand. All these cloud properties can be plugged into almost any existing app without forcing a real change, as noted in this **MySQL use case**. Cloudify automates the deployment and configuration of the application even in the case of complex multi-tier applications, making it possible to launch the application on-demand on any given environment. In addition, Cloudify can plug-in to the application-specific metrics which makes it possible to detect application-specific SLA breaches, and thus automate the fail-over process as well. On top of that, Cloudify comes with a scaling rules plug-in that makes it extremely simple to add more instances to the application for scale-out purposes, or simply to enable it to move from smaller to bigger instances, in the case of a scale-up model.

Scan the barcode to view the MySQL use case.

- **Use the Baby Steps Approach**

Scan to watch the demo.

One of the main inhibitors to migrating apps was the all or nothing approach. There are many risks involved in migrating apps that go well beyond the technology challenge. The easiest way to mitigate such risks is by taking them one at a time. For example, we can first make our apps cloud ready, in which case we will abstract the automation part from the target infrastructure. By doing so, the decision where to run the app becomes tactical and less strategic. We can then choose to run our production in our private data center, and run only testing and demos on the cloud. In this way, we can also make the migration gradual and smoother. With Cloudify all you need to get started is a start and stop script. All the other elements such as custom metrics, auto-scaling and more can be added at a later stage, making it possible to grow gradually with the value, and add more cloud properties as we grow. Additionally, the Cloud portability enables the automation of our apps on our existing environment first, and then to move to the cloud at a later stage. A good example for this would be to keep our production on our existing environment, and testing on the public environment. A more advanced incarnation of this could be to combine the two into a complete hybrid-cloud deployment.

- **Incorporate Bare-Metal into our Cloud**

One of the main challenges involved with moving true enterprise workloads to the cloud is not necessarily related to the cloud, but to the virtualization overhead that is often used as the underlying infrastructure of many clouds. Today it is possible, to gain access to bare metal clouds which make it possible to run almost any workload including I/O intensive ones without any performance overhead. Cloudify was also designed to work on non-virtualized environments. In such a case, it takes a list of IP addresses as a fixed pool and manages the workloads elastically within the boundary of that pool as noted in this post: **bare-metal clouds.**

Scan this to read the bare-metal clouds blog post.

Technical Jargon

Moving the Enterprise Stack to the Cloud

Application	Product name	Comments
Application Servers	**WebLogic, WebSphere, JBoss,**	**Via Cloudify recipes**
Messaging Middleware	**RabitMQ, ServiceMix, Mule**	**Via Cloudify & Chef/ Puppet**
Web Servers	**Tomcat, Ruby, Lamp,..**	
Databases	**MySQL, Oracle, SQL Server**	**Via Cloudify & Chef/Puppet**
Big Data (NoSQL, Hadoop, Storm etc.)	**Cassandra, MogoDB, CoachBase, Hadoop**	**Via Cloudify recipes**
ERP and CRM	**SAP, SugarCRM,**	**Via Cloudify, Puppet**
Search Engine	**Elastic Search**	
Content Management	**Drupal, ..**	**Via Cloudify**

References

Cloudify recipes on Github:
https://github.com/CloudifySource/cloudify-recipes/tree/master/services

Chef Cookbooks
http://community.opscode.com/search?query=

Puppet Cookbooks
https://forge.puppetlabs.com/modules

Cloudify & Chef Integration (Technical Overview)

http://www.cloudifysource.org/guide/2.2/integrations/chef_documentation.html