Automatic Migration and Deployment of Cloud Services for Healthcare Application Development in FIWARE

Stelios Sotiriadis[1], Lenos Vakanas[1], Euripides Petrakis[1], Paolo Zampognaro[2], Nik Bessis[3]

[1]Technical Univeristy of Crete, Kounoupidiana Campus 73100, Crete, Chania
[1]{s.sotiriadis, petrakis}@intelligence.tuc.gr, lenosvakanas@gmail.com

[2]Engineering Ingegneria Informatica S.P.A, Napoli, Italy
[3]paolo.zampognaro@eng.it

[3]Edge Hill Univerity, Ormskirk, UK
Nik.Bessis@edgehill.ac.uk

Abstract— **Over the latest years due to the emergency of new requirements in the area of healthcare domain, the software to data cloud model attracted significant attention by bridging the gap between cloud service deployment and sensitive data storage and manipulation. In particular, in FI-STAR FP7 we develop innovative cloud services for the healthcare domain using the FIWARE platform. This offers new opportunities for building innovative applications that utilize FIWARE Generic Enablers as the building blocks of Future Internet (FI) applications. In this work we focus on (a) the automatic migration requirements in FIWARE systems and (b) on the automatic shipment and deployment of FI-STAR services among service providers infrastructures that is the FIWARE cloud nodes and service consumer infrastructures (FI-STAR use cases). We describe the design and implementation enabling the automatic shipment and deployment to support the software to data paradigm and to allow marketplace development of the FI-STAR catalogue. The deployment automation solution offers less error-prone and highly repeatable processes, efficient deployment and management of updates and new releases.**

***Keywords*— Cloud Computing, Automatic service deployment, Cloud software to data, FIWARE, FI-STAR**

I. Introduction

FI-STAR FP7 project [1] utilizes cloud computing and Internet of Things (IoT) concept to build innovative Future Internet services [1] for healthcare provision in the information and communication technology domain. FI-STAR experimentation sites and use case scenarios involve seven healthcare providing sites, servicing around 6 Million people throughout Europe. This encompasses various use cases including monitoring, rehabilitation and other of patients utilizing cloud services[2]. In general, cloud computing has been utilized with success in various areas such as industry, agriculture, energy and environment yet not in the healthcare area. Data that are stored in cloud systems and related with healthcare information are usually confidential and restricted for open access (e.g. ISO-25010). Since there are various standards, regulations and recommendations described in [2], there are severe restrictions to data transfer and storage.

To overcome this hurdle, FI-STAR develops cloud services based on the software to data [8] solution and according to an automatize shipment and deployment model. Specifically, based on [6] we aim to create a framework that allows GEs to be delivered to different physical locations. That is to say that the services are encapsulated as cloud Virtual Machines (VMs) in OpenStack environments [3] and [5], that could be transferred from a public cloud service provider to a private cloud system seamless and without any manual intervention by the user. The FI-STAR platform is developed following the FI-WARE conceptual model and offers a set of software modules, called as Specific Enablers (SEs) that allow flexible developments of healthcare applications. The proposed solution allows SEs to be transferred near to the data source and to be executed and managed automatically. The solution is developed based on a RESTFul architecture, wherein an agent allows two-fold communication and is responsible for updating and versioning features. Based on this discussion next we present the FI-STAR automatic deployment model.

In this work we focus on (a) the automatic migration requirements in FIWARE systems and we present the VM migration tool among OpenStack cloud platforms and (b) on the automatic shipment and deployment of FI-STAR services among service providers infrastructures that is the FIWARE cloud nodes and service consumer infrastructures (FI-STAR use cases). The work is organized as follows. Section II present the requirements for software to data approaches, Section III the software to data service functionality and the Section IV FI-STAR automatic deployment process.

II. Requirements for Software to Data Approaches

A vital concept of the FI-STAR project is that aims to eliminate restrictions, which are responsible for the minimal usage of cloud technology by various application fields especially health care. Software to data refers to the idea of transferring the software to the data location, thus near to the source of data generation. In order to accomplish this goal,

[1] http://www.fi-star.eu
[2] https://www.fi-star.eu/use-cases.html

hybrid cloud technology is utilized [7]. This cloud model is the combination of a public and a private cloud and it provides the ability, for an entity, to own and manage a private cloud and use functionality offered by public cloud providers, with the advantages of both models. It comes as a solution to problems found in public cloud models, such us the problem with patient data in e-Health environments. As a result, the provider offers his services via public cloud and the consumer uses these services in his private cloud achieving maximum security over his data (patient data or otherwise).

An important requirement for software to data approaches involves services that are utilizing cloud APIs (such as Openstack REST API) to allow its users to migrate a virtual machine between two clouds. Technically users can perform the migration using the migration tool presented in [4] that includes the web-based user interface or the migration based on sending an xml document containing all the needed information. If a user selects to perform migration through the Web interface, we design deferent configuration steps (that are available in the interface) in order the migration service to gather all needed information. The user provides information about credentials, the virtual machine (VM) require migrating, configuration of the newly deployed machine and other. Apart from the web interface, the user can also send directly to the service all this information in an xml document. The xml document has a predefined structure validated by an xml schema and the user has to fill all the fields, which will be used by the service.

The migration service executes the three modes of operation referred to above as follows:

(a) The first is responsible for transferring and deploying the software

(b) The second is responsible for monitoring the usage of the software (for example VM up-time)

(c) The third one is an alternative to the user interface but only for the first module, meaning it offers to the user the alternative for using the migration service as an API call containing all the needed information in an XML document.

We define as successful migration the process of moving a running instance (and its software) from a cloud A, to be deployed it in cloud B while maintaining its hardware, software and network configurations. Thus when the process is finished the user could find in cloud B a running instance including pre-installed software of the same VM size (known also as VM flavour), same ports and rules in the security group and a new floating IP that will not be different because it varies based on geographical location, organization.

III. Software To Data Service Technical Solution

To achieve the software to data VM migration, the service guides the user through the process and at the same time performs some automated actions resulting in a less complex and time consuming process. Currently, the service offers this functionality for Intellicloud[3] and FI-Lab[4] with both clouds using Openstack. In this section we describe the functionality provided by the service through both the user interface and the API call.

- Authentication: The user provides his tenant ID, username and password in order to be authenticated by the cloud. Authenticating generates a token, which is being used in every action the user (or the service itself) performs on the cloud. The token is unique for every user and it has a lifecycle predefined by the cloud. For example, Intellicloud's token is valid for 24 hours.
- Get Instances: The service retrieves a detailed list of the instances registered to the user.
- Get Images: The service retrieves the images registered to the user including snapshots.
- Get Instance's Details: Retrieves the information that describes an Instance.
- Create Snapshot: Creates a snapshot of the running instance that the user selected for migration. In addition, the service stores all the properties of this instance, which will be later, be used for launching the new instance in cloud B with the same configurations such as security group rules or flavor. For this action, the user provides a name for the snapshot and the name of the instance required to snapshot.
- Download Snapshot: The service downloads and stores temporarily the previously created snapshot.
- Upload Snapshot: Because a snapshot is also an image, the service creates a new a new image containing the data of the previously downloaded snapshot. The user provides the name of the new image, the format and if it is going to be public or private (access). The service will perform the following actions to achieve this result without exposing the user to each action.
- Create a new blank image with the given name
- Update the blank image according to the format (e.g. with the format qcow2) and access (e.g. public) that the user specified.
- Upload the data the snapshot's data to the new image. The snapshot's data contain information about the instance's operating system and software that will be used to launch the same instance in the target cloud.
- Keypair Actions: The keypair must be allocated to the user and provided to the new instance before its creation. As a result, the service will save the name of the keypair that the user chooses.
- Create Keypair: The user provides the name of the new keypair and the service will create a new one with that name and allocate it to the user.
- Import Keypair (from Cloud A): The user selects the keypair from cloud A and the service imports it in cloud B. For this action to be completed, the service firsts retrieves the list of user's keypairs from cloud A and after the selection of the user, imports it in cloud B.
- Select Keypair (from Cloud B): The service retrieves the list of keypairs allocated to the user and the user selects one.

[3] http://cloud.intellicloud.tuc.gr/

[4] https://cloud.lab.fiware.org/

- Launch Instance: The user provides the name of the instance and its security group name. Afterwards, the service will set the following.
- Keypair: Keypair of the instance will be the previously selected by the user.
- Image: Image will be the new image which the service created from the data of the snapshot.
- Flavor: The service will fetch it from Cloud A and set the same flavor in Cloud B.
- Security Group: The service will either create a new security group or will allocate to the instance the default one depending on the users input. If the user sets the name as "default" or if it leaves it empty, the service will select the default security group. Now, if the user sets a different name, the service will generate a new security group with that name. Independent of the user's input, the service will:
  - Get the security group that the instance had in Cloud A.
  - Get all of the rules inside that security group.
  - Insert all of the rules in the security group that the user selected for the instance in Cloud B.
- The rationale behind all these actions is that we want the instance to be launched in Cloud B and immediately be operational (including all of its software) without any further configuration by the user.
- IP Number Actions: The user can choose from a list of free IPs or chooses to create a new one to allocate to the new instance. Free IP is one that belongs to the specific user but it is not allocate to any other of his instances. If the user chooses to create a new IP, the service will
- Create a new IP:
  - Allocate it to the user's account
  - Allocate the new IP to the instance.
- Instance Overview: The service fetches all the information about the instance that the user created. This is also a way for the user to check that the process was successful (the other way is directly through the cloud's dashboard). The fetched information is directly from the cloud's system and not by any information given by the user during the process ensuring the success of the process.
- Get Instance's Usage Data: The user provides his cloud credentials and after his successful authentication by the cloud's system, the service fetches the usage data of his instances. The usage data includes the total of CPU hours used, total GB of storage, RAM and so on.
- Reset System: This action allows the user to reset the state of the service. This means that the user's session will be deleted along with any information provided till that point, including any images that required to be downloaded to the service's server. However, any actions that had already been executed on any of the two clouds are not affected meaning if the user wants an action done on a cloud reverted, needs to do it through the cloud's dashboard (e.g. deleting a snapshot). This action is only available when the user uses the migration tool of the service because in the case of the monitoring tool the user does not provide anything more than his cloud credentials.

IV. FI-STAR Automatic Deployment

The FI-STAR platform makes the shift to an innovative technology that incorporates a cloud platform with open access SEs and "a software to data cloud model" for secure SE deployment. This section presents the automation system that allows both the deployment of the SEs in the form of VM files that are shipped and deployed in the final cloud node. The solution involves features such as: (a) deployment could be executed by the customer or an external (trusted) platform provider, (b) the application provider specifies the client preferences, (c) the option to choose whether the automation will include downloading and\or deployment, (d) the option to check the subscription status for an enabler and to trigger an automatic download and optionally an automatic redeployment (in case of an update) and (e) to consider specific customer configuration before proceeding with the redeployment, for example specified time slot for deployment.

The automatic solution includes various components that are demonstrated in Figure 1. The Agent Service Component that is installed within the application provider domain periodically checks the status of the SEs subscriptions submitted on the FI-STAR platform by the application provider. The communication and polling activity is secured by exploiting an access token obtained by interacting with an identity manager server[5] during the initialization step. If a subscription status changed (e.g. a new SE version is available because of an update) the Agent notifies the deployment manager component to serve the updating requests.

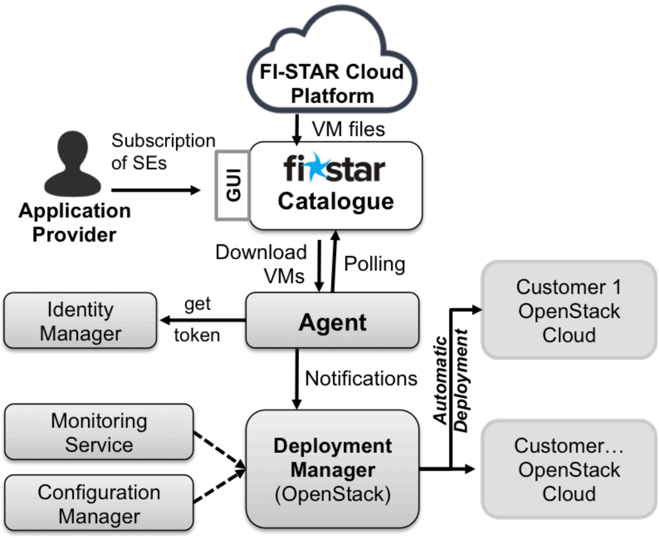

Fig. 1. Automatic VM Shipment and Deployment Architecture of FI-STAR

The Configuration Manager Component includes operations to store and retrieve customer preferences and it holds preferences related to the geographic constraints and the

[5] http://catalogue.fiware.org/enablers/identity-management-keyrock

Deployment Configuration associable to each single SE. The geographic constraints allow the use of cloud nodes to host SEs locally or into a trusted cloud platform provider infrastructure by capturing the customers preferences related to the physical infrastructure in which the deployment will take place. Within the Deployment Configuration component the customer indicates "how" and "when" such deployment should occur. In fact the user can choose to be notified when a deployment action should be performed and\or can enable a full automatic deployment. In the last two cases the Customer can also specify the time slot when the deployment should be performed automatically. The Deployment Manager component notified by the Agent, to execute the deployment or un-deployment actions of the VMs of SEs as consequence of new subscriptions status. For instance if the subscription policy of an SE is set to "always updated" and a new version is uploaded the module will retrieve from the Configuration Manager the needed deployment options in order to proceed. In principle an Application Provider could delegate to a third trusted (cloud) Platform Provider the deployment task. The solution involves the following services:

- The catalogue service provides operations to get user token, create a new token and provide subscriptions.
- The agent service provides a graphical interface (GUI in Figure 1) to manage the agent activity and to monitor the application provider's subscriptions.
- The configuration manager service provides a graphical interface to configure customer preferences to be mapped into customer constraints and deployment configuration objects.
- The deployment manager service that provides a graphical interface allowing the cloud platform provider to configure preferences and to define the cloud nodes of his own cloud environment. The notification parameter indicates whether to trigger an update of a VM or a subscription is due to expiration. In first case the service checks the cloud status and updates/deploys the new VM using also the customer preferences according to the Configuration Manager and in the second case un-deploy all the VM instances.

Based on the aforementioned discussion, the FI-STAR Automation service will provide significant benefits. The SEs could be easily deployed within a homogeneous OpenStack while complex dependencies including versioning management and geographically constraints are easily orchestrated in a centralized manner. The dynamic provisioning of resources includes an efficient specification of each utilization and dependencies control. In addition, the OpenStack federation (adopted in the FI-STAR project) allows transfer of SEs to private clouds realizing the software to data model. Finally, monitoring features easily issue detection and deployment checking. The seven FI-STAR use case trials site in various locations in Europe including Germany, Italy, Spain, UK, Romania, Norway and Poland adopts the solution. Nevertheless, the solution is limited to the homogeneous infrastructures of OpenStack and reflects the transfer of SEs in the form of VM images and not as running instances. An extended model of running VMs deployment has been described in [3] and implemented in [4].

V. Conclusions

FI-STAR highlights new opportunities and openings for wider adoption of FI technologies in the health healthcare domain. This work presented the migration of VMs in FIWARE systems and the deployment automation solution to achieve the software to data solution in a less error-prone and highly repeatable way. The FI-STAR catalogue[6] highlights new openings in the area of health care provision including services for personalized applications utilizing IoT devices and acting as the integration plan to secure Future Internet healthcare services.

Acknowledgment

The research leading to these results received funding from the European Unions Seventh Framework Programme (FP7/2007-2013) under grant agreement 604691 (FI-STAR).

References

[1] Galis, A. and Gavras, A. (2013) *The Future Internet: Future Internet Assembly 2013 Validated Results and New Horizons*, Springer Publishing Company, Incorporated, 2013.

[2] Sotiriadis, S. Petrakis, E. Covaci, S. Zampognaro, P., Georga, E., and Thuemmler, C. (2013) *An architecture for designing future internet applications in sensitive domains: Expressing the software to data paradigm by utilizing hybrid cloud technology*, In Bioinformatics and Bioengineering (BIBE), 2013 IEEE 13th International Conference on, pages 1-6, Nov 2013.

[3] Sotiriadis, S., Bessis, N., and Petrakis, E. (2014) *An inter-cloud architecture for future internet infrastructures*, In Pop, F. and Potop-Butucaru, M., editors, Adaptive Resource Management and Scheduling for Cloud Computing, Lecture Notes in Computer Science, pages 206-216. Springer International Publishing.

[4] Vakanas, L., Sotiriadis, S. and Petrakis, E. (2015) *Implementing the Cloud Software to Data approach for OpenStack environments*, Adaptive Resource Management and Scheduling for Cloud Computing, Held in conjunction with PODC-2015, Donostia-San Sebastián, Spain, on July 20th, 2015

[5] Sotiriadis, S., Bessis, N., *An Inter-Cloud Bridge System for Heterogeneous Cloud Platforms*, An inter-cloud bridge system for heterogeneous cloud platforms, Future Generation Computer Systems, Volume 54, January 2016, Pages 180-194, ISSN 0167-739X, http://dx.doi.org/10.1016/j.future.2015.02.005

[6] L. Schubert, K. Jeffery, B. Neidecker-Lutz (2010) "The Future of Cloud Computing –Opportunities for European cloud computing beyond 2010", European Commission [Online]. Available: http://cordis.europa.eu/fp7/ict/ssai/docs/cloud-report-final.pdf, Accessed: 8 June 2013

[7] Open Network Foundation (2012), "OpenFlow-Enabled Hybrid Cloud Services Connect Enterprise and Service Provider Data Centers", ONF Solution Brief 2012, Available: https://www.opennetworking.org/solution-brief-openflow-enabled-hybrid-cloud-services-connect-enterprise-and-service-provider-data-centers, Accessed: 15 November 2015

[8] C. Thuemmler, J. Mueller, S. Covaci, T. Magedanz, S. D. Panfilis, T. Jell and A. Gavras, "Applying the Software-to-Data Paradigm in Next Generation E-Health Hybrid Clouds", *In Proc. Proceedings of the 10th International Conference on Information Technology (ITNG2013)*, IEEE Computer Society, ISBN 978-0-7695-4967-5

[9] ISO-25010, Available: http://sa.inceptum.eu/sites/sa.inceptum.eu /files/Content/ISO_25010.pdf, Accessed: 15 November 2015

[6] http://catalogue.fi-star.eu