ARGON: A Tool for Modeling Cloud Resources

Julio Sandobalin[1, 2], Emilio Insfran[2], Silvia Abrahao[2]

[1] Escuela Politécnica Nacional, Ladrón de Guevara, E11-253, P.O. Box 17-01-2759, Ecuador
julio.sandobalin@epn.edu.ec
[2] Universitat Politècnica de València, Camino de Vera, s/n, 46022, España
{jsandobalin,einsfran,sabrahao}@dsic.upv.es

Abstract. Configuration Management Tools (CMTs) have achieved automating the orchestration of cloud infrastructure provisioning. However, CTMs are managed through scripting languages in a manually intensive manner. As a result, CMTs manage is a time-consuming and error-prone activity. To face this issue in previous works, we have presented ARGON tool which models the cloud infrastructure and generate scripts to manage CMTs. This paper aims to show an extension of ARGON tool with functionalities both to modeling computing, storage, networking and elasticity in the Cloud and to model the middleware needed for the proper operation of cloud-based software applications.

Keywords: Cloud Computing, DevOps, Infrastructure as Code, Cloud Infrastructure Provisioning, Model-Driven Development.

1 Introduction

DevOps (Development & Operations) is a paradigm which brings principles, practices and tools that optimize the time of software delivery. The cornerstone of DevOps is Infrastructure as Code [1] which is the infrastructure automation based on practices from software development. There exist several Configuration Management Tools (CMTs) from DevOps community which orchestrate the cloud infrastructure automation, such as Ansible[1], Chef[2] or Puppet[3]. However, CTMs are managed through scripting languages in a manually intensive manner. As a result, CMTs management is a time-consuming and error-prone activity. To face this issue we have presented ARGON tool [2] which models the cloud infrastructure and generate scripts to manage CMTs. This paper aims to present an extension of ARGON with the functionalities both to modeling computing, storage, networking and elasticity in the Cloud and to model the middleware needed for the proper operation of cloud-based software applications. Moreover, ARGON is the keystone of our research about a model-driven approach for continuous delivery of cloud resources based on end-to-end automated toolchain for cloud provisioning [3].

[1] https://www.ansible.com
[2] https://www.chef.io
[3] https://puppet.com

2 ARGON

ARGON (An infRastructure modelinG tool for clOud provisioNing) is a modeling tool for specifying the final state of the infrastructure provisioning of cloud resources and generating scripts to manage the Configuration Management Tools (CMTs) used in the DevOps community. ARGON supports a Domain Specific Language (DSL), which includes an abstract syntax and a concrete syntax. The former is an Infrastructure Metamodel [2] which abstracts the capacities of the Cloud Computing such as computing, storage, networking, and elasticity. The latter defines graphical notation to render the metamodel elements in the modeling editors of Eclipse Modeling Framework [4]. Moreover, ARGON includes a transformation engine based on model-to-text transformations to generate the scripts needed to manage the CMTs. The transformation engine is defined by transformation rules that abstract the features of the proprietary scripting languages that the CMTs use.

2.1 Functionalities

The functionalities of the ARGON's elements are described below.

Table 1. Functionalities to model computing capacity

Figure	Element	Description
	Virtual Machine	Create, terminate, start and stop virtual server instances
	Security Group	Act as a firewall to protect one or many virtual machines
	Inbound	Specify inbound rules to enforce in Security Groups
	Outbound	Specify outbound rules to enforce in Security Groups
	Static IP	Allocate or release a static IP address to a Virtual Machine
	Load Balancer	Register or unregister a virtual machine to a Load Balancer
	Listener	Check the connection requests to a Load Balancer

Table 2. Functionalities to model storage capacity

Figure	Element	Description
	Database	Create, terminate, start and stop databases instances
	File Server	Create and delete File Servers

Table 3. Functionalities to model elasticity capacity

Figure	Element	Description
	Launch Configuration	Specify a template with hardware features of a virtual machine
	Auto Scaling Group	Specify the number of virtual machines to create or delete
	Scaling Policy	Specify the conditions to create or delete a virtual machine
	Alarm	Monitor a metric in a time frame to trigger a Scaling Policy

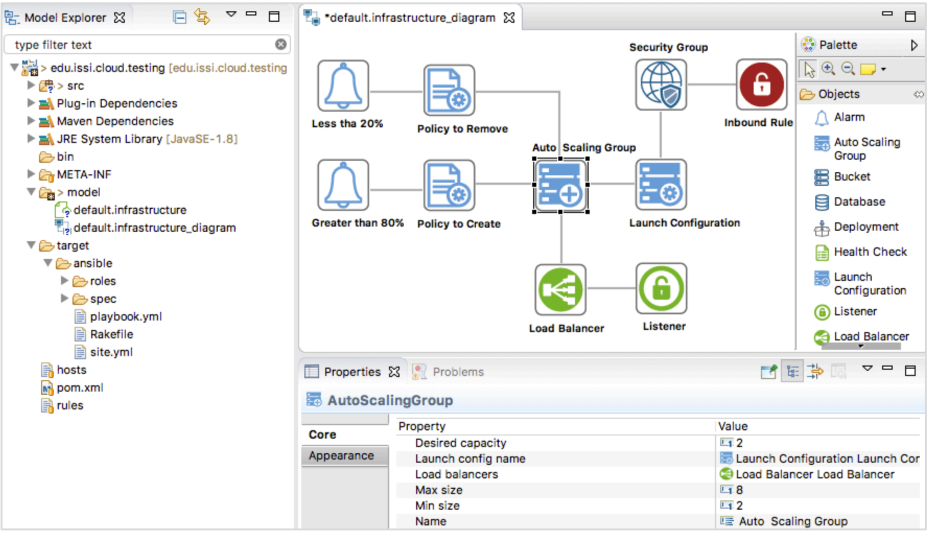
Table 4. Functionalities to model middleware

Figure	Element	Description
	Role	Specify software packages to be installed as middleware
	Deployment	Specify path, permission, and user to install software applications
	Template	Specify configuration files to be replaced in a virtual server

Functionalities to model networking capacity is represented by associations among metaclasses specified in the Infrastructure Metamodel [2].

3 Modeling infrastructure and middleware

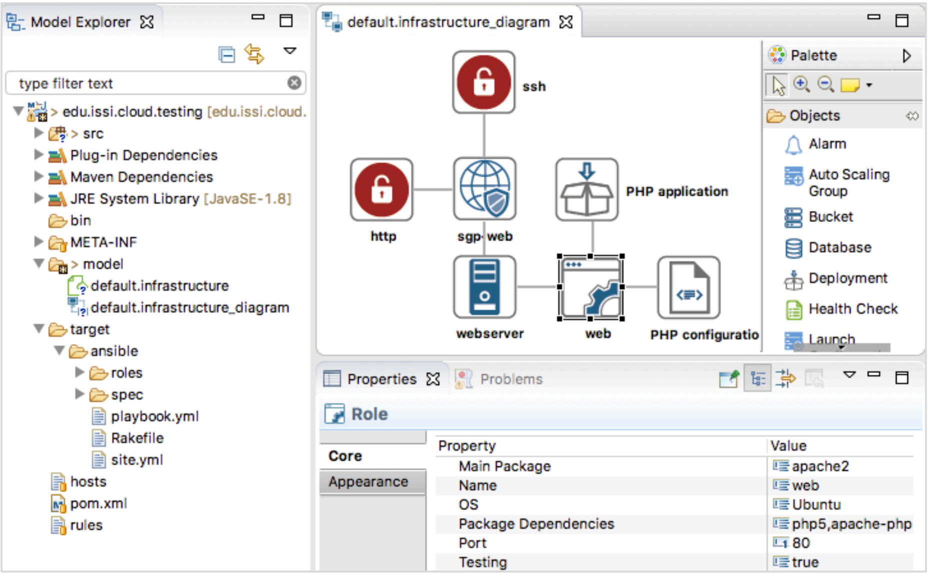
ARGON allows modeling a scalable architecture to provide the elasticity capacity to cloud-based software applications. **Fig. 1** shows an infrastructure model which has a ***Launch Configuration*** with hardware features of a virtual machine. ***Auto Scaling Group*** specify the minimum and maximum number of virtual machines to be created or eliminated according to hardware features from ***Launch Configuration***. Creation of a virtual machine is done based on ***Policy to Create*** which is executed by reaching ***Greater than 80%*** of CPU usage of a virtual machine. Elimination of a virtual machine is done based on ***Policy to Remove*** which is executed by reaching ***Less than 20%*** of CPU usage of a virtual machine. ***Security Group*** performs as a firewall to all virtual machines created in the scalable architecture. ***Security Group*** enables the virtual machines to respond requests through ports as ***Inbound Rules***. ***Load Balancer*** allows distributing incoming application traffic between multiple virtual machines and with an input rule or ***Listener*** that checks the connection requests.



Property	Value
Desired capacity	2
Launch config name	Launch Configuration Launch Cor
Load balancers	Load Balancer Load Balancer
Max size	8
Min size	2
Name	Auto Scaling Group

Fig. 1. Scalable architecture for cloud infrastructure provisioning.

On the other hand, ARGON allows modeling the middleware and cloud-based software applications. ***Fig. 2*** shows a virtual machine (***webserver***) which has a security group (***sgp-web***) to enable the virtual machine to respond requests through port 22 (***ssh***) and port 80 (***http***). To model the middleware, we define a role (***web***) where we specify software packages to be installed in the virtual machine, kind of operative system, port to respond requests to the virtual server, and the option to testing the infrastructure deployed. Moreover, ARGON allows modeling the deployment (***PHP application***) of software applications and specify a template (***PHP Configuration***) or configuration file that will be replaced in the virtual server.



Property	Value
Main Package	apache2
Name	web
OS	Ubuntu
Package Dependencies	php5,apache-php
Port	80
Testing	true

Fig. 2. Modeling middleware and software applications.

Acknowledgments. This research is supported by the Value@Cloud project (TIN2013- 46300-R).

References

[1] K. Morris, *Infrastructure As Code: Managing Servers in the Cloud*, 1st ed. O’Reilly Media, Inc., 2016.

[2] J. Sandobalin, E. Insfran, and S. Abrahao, “An Infrastructure Modelling Tool for Cloud Provisioning,” in *Proceedings - 14th IEEE International Conference on Services Computing, SCC (in press)*, 2017.

[3] J. Sandobalin, E. Insfran, and S. Abrahao, “End-to-End Automation in Cloud Infrastructure Provisioning,” in *Proceedings - 26th International Conference on Information Systems Development, ISD (in press)*, 2017.

[4] D. Steinberg, F. Budinsky, E. Merks, and M. Paternostro, *EMF: eclipse modeling framework*. 2008.