AriaToscaProposal

/!\ FINAL (!)

Abstract

ARIA's mission statement is to drive the adoption of TOSCA by offering an easily consumable Software Development Kit(SDK) and a Command Line Interface(CLI) to implement TOSCA based solutions which will help in market consolidation around TOSCA based Orchestration. One of the key attributes of the TOSCA specification by OASIS is that it is a vendor neutral and technology agnostic specification, allowing to describe applications and then to orchestrate them using various technologies of choice, such as Amazon AWS, Google GCP, OpenStack, VMware, Puppet, Ansible Chef, etc'. The reality is such that TOSCA is a complex specification,making software vendors trying to implement the specification take implementation decisions, ending up with different and incompatible implementations, creating even more market segmentation instead of consolidating around a single standard for orchestration. ARIA is an open source python library that helps orchestrators and management tools developed TOSCA enabled orchestration solutions. ARIA aims to become the main reference implementation of the OASIS TOSCA(Topology and Orchestration Specification for Cloud Applications) specification for orchestrating cloud applications, and descriptor for VNFs in Telecom NFV. ARIA can be executed via CLI to orchestrate application templates, additionally it allows embedding its python library code for creating TOSCA compatible services, and includes rich set of plugins for Iaas orchestration, such as Amazon AWS, Google GCP, OpenStack and VMWare; It Includes plugins for Container orchestration, with Docker and Kubernetes plugins, configuration management tools(Puppet,Chef, Ansible) and a rich API that allows to create plugins for any new technology. ARIA serves as a code base that allows to test the TOSCA specification and experiment with new approaches to modeling and orchestration of applications, providing feedback and real world use cases OASIS to further refine and advance the standard specification.

Proposal

ARIA offers a command line tool and a set of embeddable python libraries implementing an engine for orchestration using TOSCA declarative language blueprints. TOSCA allows describing application's topology with its interconnections and interactions among multiple components using declarative language. This involves combining tasks into workflows, provisioning and management of various components and their associated resources in an automated manner, and in multi-cloud environments it involves interconnecting processes running across heterogeneous systems in multiple locations.

ARIA aims to implement several main use cases, and can be used as a command line tool to orchestrate TOSCA based application templates serving as a lightweight tool to onboard and orchestrate applications in a repeatable and robust manner. ARIA can be used by software vendors and VNF providers as a development environment for creating TOSCA templates for onboarding and managing the lifecycle of software products and Virtual Network Functions (VNFs). Additionally ARIA can be used for building orchestration platforms and enabling TOSCA based orchestration using the ARIA TOSCA orchestration engine as the kernel of the orchestrator.

- ARIA TOSCA Parser* ARIA includes a TOSCA DSL parser, the parser's role is to interpret the TOSCA template, create an in-memory graph of the application and validate templates' correctness. TOSCA provides a type system of normative node types to describe the possible building blocks for constructing a service template, as well as relationship types to describe possible kinds of relations. Both node and relationship types may define lifecycle operations to implement the behavior an orchestration engine can invoke when instantiating a service template. The template files are written in declarative YAML language using TOSCA normative types. Technology specific types can be introduced via ARIA Plugins without any modifications of the parser code.

TOSCA Templates include:

- YAML Topology Template
- Plugins
- Workflows
- Resources such as scripts, etc'

- ARIA Plugins* ARIA Plugins allow extending the TOSCA normative types dynamically by adding new technology specific node types and relationship types with their implementation, without changing the code of the ARIA TOSCA Parser. The plugin based types are isolated, allowing different versions of the same plugin in a single blueprint - for example support OpenStack Kilo and OpenStack Juno in the same template. It also allows combining types of different technologies - for example OpenStack nodes with VMware, Amazon, or other types such as Router, Firewall, Kubernetes and others. The work of interacting with IaaS APIs and running scripts, Configuration Management tools, Monitoring tools and any other tools used when managing applications is done by the ARIA Plugins. Plugins can be included as part of the application template package and loaded dynamically. ARIA includes plugins that can be used as is or as reference for implementing for new plugins, ARIA Includes plugins for the following:

- Plugins for IAAS: OpenStack, AWS, VMWAre, GCP Azure
- Plugins for CM: Puppet, Chef, Ansible, SaltStack
- Plugins for Containers: Kubernetes, Docker, Mesos, Swarm
- Plugins for SDN: ODL, ONOS
- Script Plugin: Bash, Python others
- Fabric Plugin via SSH
- Custom Plugins

ARIA Embedded Workflows Workflows are automated process algorithms that allow to interact dynamically with the graph described in the application topology template. Workflows describe the flow of the automation by determining which tasks will be executed and when. A task may be an operation, optionally implemented by a plugin, but it may also be other actions, including arbitrary code or scripts. Workflows can be embedded within the TOSCA Template to be able to access the graph dynamically. They are implemented as Python code using dedicated APIs and a framework to access the graph context, the context provide access to the object graph described in the TOSCA template. ARIA comes with a number of built-in workflows - these are the workflows for install, uninstall, scale and heal. Additionally it is possible to write custom workflows. Built-in workflows are not special in any way - they use the same API and framework as any custom workflow is able to use.

Background

GigaSpaces have been offering a cloud orchestration product - Cloudify - which is an open source and open platform for orchestration based on the OASIS TOSCA specification. TOSCA, introduced by OASIS in 2013, is a Domain Specific Language(DSL) to describe cloud applications and Virtual Network Functions(VNFs), TOSCA allows to orchestrate cloud applications and VNFs for NFV based on the description of an application topology, its workflows and policies.
A key attribute of the TOSCA specification is that it is a vendor neutral and technology agnostic specification, allowing to describe applications and then to orchestrate their installation and workflows using technologies of choice, such as Amazon AWS, Google GCP, OpenStack, VMware, Puppet, Ansible Chef, etc'. Several software vendors introduced specific implementations of the TOSCA specification, including Cloudify by GigaSpaces, each implementation offered TOSCA based orchestration solution making implementation decisions, due to the complexity of the spec, that prevents the portability of described applications, or making the implementation dependent on a specific technology, making TOSCA application templates specific to the orchestrator and not portable across orchestration solutions.

The reality is such that TOSCA is a complex specification,making software vendors trying to implement the spec make implementation decisions, ending up with several different and incompatible implementations, creating even more market segmentation instead of consolidating around a single standard for orchestration, making it difficult for the industry to come around a single standard for orchestration, the original promise of OASIS TOSCA to the market.

Based on our work the past several years and experience implementing a TOSCA orchestrator, Cloudify, we have realized that there should be a simple to consume open source library and framework of tools and services for software companies,such as GigaSpaces and others, to implement TOSCA based orchestration solutions, where each orchestrator would be TOSCA compliant, therefore making the TOSCA application templates portable across different orchestrators, making it easier for consumers of orchestrator solutions create rich library of application templates which are cross orchestrator compatible.

ARIA, stands for Agile Reference Implementation for Automation, aims to become the reference implementation library for TOSCA, allowing software vendors such as GigaSpaces to use ARIA as the kernel for TOSCA orchestration and to implement compatible orchestrators, making sure that application templates written for any of orchestrator are supported by any other ARIA TOSCA enabled orchestrators.

Rationale Initial Goals

The TOSCA based orchestration ecosystem is currently fragmented due to the complexity involved in developing a TOSCA based orchestration solutions, preventing wide adoption of TOSCA. In order for TOSCA to become a widely accepted orchestration standard, an independent and neutral open source reference implementation with SDK for developing TOSCA based solution has to emerge. Project ARIA(Agile Reference Implementation for Automation) under the Apache Software Foundation will become a vendor independent open source library for building TOSCA solutions aiding in consolidating the TOSCA community around a single robust and neutral TOSCA library. In addition ARIA strives to become a development and testing framework for new modeling methods by OASIS as OASIS TC explorers and proposes changes to the TOSCA specification.

Current Status

ARIA 0.1 release with it's initial code is based on Cloudify 3.4 mature core orchestration libraries, with rich set of plugins capable of orchestrating most major private and public clouds. As the project's goal is to offer a robust software library and a TOSCA SDK, ARIA is being refactored to become a general purpose library for TOSCA Orchestration. The refactoring process includes alignment with most recent OASIS TOSCA DSL specification, reflecting the workflow engine which drives the execution of the described TOSCA templates, Aiming for initial 1.0 release in November 2016.

Meritocracy

ARIA being a reference implementation of a vendor independent and neutral standard specification, we strongly believes in meritocracy, where individual committers and companies can help drive the implementation of the standard and take leading roles in steering the project. ARIA's started it's life as the Kernel of Cloudify, an open source and open platform orchestration product, we intend to bring our experience operating in open source communities to create an open governance structure for project leadership to encourage individual and company involvement and contributions.

Community

Cloudify currently has a rich active community of users and developers. Most of the code for core orchestration framework is created by GigaSpaces. Additionally a rich set of plugins and blueprints, which extend the capabilities of the core orchestrator, created by GigaSpaces, and community contributors. Cloudify fosters live community discussion over google groups, a mailing list, IRC, and SLACK. It is important to foster and expand this community and attract more diversity to the core team. For ARIA community to thrive and success It is important to plug into dependent communities that consumes ARIA(such as Cloudify, Open-O and others) encourage and facilitate discussions and joint contributions.

Core Developers

Lior Mizrahi, Tal Liron, Maxim Orlov, Ran Ziv. New core developers (initial committers) for the ARIA project are welcome to join the community by code contributions, broad involvement in the community through code reviews, mailing lists and IRC.

Alignment

Project ARIA's main goal is to become a healthy, neutral, open-governance open-source project, we strongly believe that the Apache Software Foundation provides the most substantial framework to achieve such community.

Known Risks

Orphaned products

Starting from ARIA's first release is the core orchestration library in Cloudify, commercially offered by GigaSpaces. There's a strong commitment by GigaSpaces to keep a leadership role in the ARIA community, to maintain and advance the project.

Inexperience with Open Source

The team behind ARIA has vast experience in many open source communities, and has been working on Cloudify, an open source project of it's own since 2013. All development is in public on GitHub, backed up by mailing lists on Google groups and IRC. The team of committers are committed to the principles of open source, and count amongst their number existing Apache committers.

Homogenous Developers

The initial list of committers includes developers from several different geographically distributed locations, spanning across the U.S and Europe,they are experienced with working in a distributed environment.

Reliance on Salaried Developers

The initial committers are affiliated with GigaSpaces. The majority of the commits to the project to date have been GigaSpaces employees in the line of their work. Our community is growing, and we hope to grow the community significantly and bring more diversity into the list of committers.

Relationships with Other Apache Products

A Excessive Fascination with the Apache Brand

While we expect the Apache brand may help attract more contributors, our interests in starting this project under the Apache Software Foundation is build a neutral community consisted of users, developers and vendors alike. We feel that the Apache Software Foundation is the natural home for such a community. We will be sensitive to inadvertent abuse of the Apache brand and will work with the Incubator PMC and the PRC to ensure the brand policies are respected.

Documentation

www.ARIATOSCA.org Complete project documentation is being planned for upcoming releases.

Initial Source

The initial source of ARIA 0.1 is based on Cloudify 3.4 mature code base. Reaching to ARIA 1.0 release after refactoring to make ARIA easily consumable library(and corresponding Cloudify 3.5 release which consumes ARIA as the Orchestration Kernel) all core orchestration capabilities are developed in ARIA.

Source and Intellectual Property Submission Plan

The code is licensed under Apache License V2, and all contributions are subject to an Apache-style ICLA. In addition to the code, there is a logo currently used for the project which would be contributed to the ASF. The PPMC will work with GigaSpaces and project's stakeholders in order to identify additional properties and donate them to the Apache Foundation. There are also a number of other assets related to the project, such as its domain name, Twitter account, and IRC channel. During incubation the PPMC will identify all these assets, and arrange the transfer, replacement, or deprecation of these assets as appropriate.

External Dependencies

The dependencies all have Apache compatible licenses. These include BSD, CDDL, CPL, MPL and MIT licensed dependencies.

Cryptography

Required Resources

Mailing lists

We seek "ARIA-Dev" and "ARIA-User" lists to replace the lists currently in use. In alignment with Apache's standard practices, we would also have a "ARIA-Private" list for the PMC members.

Source Control

We would require a Git repository named "ARIA-TOSCA" to hold the ARIA source code, and "ARIA-SITE" to hold the web site source code. We would like these to be mirrored to GitHub repositories, where we intend to follow the same model currently used by Apache projects.

Issue Tracking

Jira, with a project name of "ARIA-TOSCA".

Initial Committers

Lior Mizrahi

Ran Ziv

Maxim Orlov

Tal Liron

Dan Kilman

Idan Moyal

Nir Biran

Avia Efrat

Adam Levin

Lukasz Maksymczuk

Arthur Berezin

Affiliations

The majority of the commits to the project so far have been made by people who were employees of GigaSpaces at the time of the contribution, and the vast majority of those commits were in the line of their employment. All named initial committers are employees of GigaSpaces. As stated in the Known Risks section, we appreciate that this cannot continue long term, and a key goal of the podling will be to encourage people not affiliated with GigaSpaces to join the project as committers and PMC members.

Sponsors

Champion

Suneel Marthi

Nominated Mentors

Jakob Homan

John D. Ament

Suneel Marthi

Sponsoring Entity

We request sponsorship from the Incubator PMC.