RADON: DevOps for Serverless Computing

Speaker & Project coordinator:
Giuliano Casale
Imperial College London
RADON Consortium

• ICT-16-2018: Software Technologies

• 30 months project (Jan 2019 - Jun 2021) – 8 organizations
The Rise of Serverless FaaS

Serverless Function-as-a-Service (FaaS)

- Fine-grained functions hosted in the cloud and fully managed by the provider
- Cost-savings in event-driven workloads (e.g., IoT)
- Strong synergy with microservices
- Resource decoupling
  - stateless functions
  - state persisted via storage
  - state change can produce events

Source: Google Images
Challenges in DevOps for Serverless

• How to deploy and update in continuous, DevOps fashion, hybrid serverless-based applications?
• How to avoid vendor lock-in?
• How to choose an optimal deployment configuration respecting QoS requirements?
• How to debug infrastructure-as-code?
• How to best do testing and monitor outcomes?
Benefits for the end user

A DevOps framework to help developing FaaS-based products without code & data lock-in

Embrace a technology shift
DevOps methods and tools to automatically build & deploy microservices & FaaS

Low cost and low risk
Multi-cloud serverless FaaS
Move/synch data between clouds
Compare and test alternatives and their costs

Select best vendor yet avoiding code & data lock-in risks

Embrace portability across business use cases
Hw/platform independent
TOSCA model facilitate reuse and customization
Multi-tenant orchestration
Model-based approach
RADON: modelling for serverless FaaS

- Easy-to-use composition of functions, microservices, storage, VMs, ...
- Reuse modular element to assemble complex applications

1. Graphical Modeling
2. Model-based Delivery
3. Quality Engineering
• Technical objectives:
  • A single **multi-user environment** to access all RADON artifacts;
  • A **front-end of the RADON methodology** enabling users to invoke RADON tools;
RADON IDE / Eclipse Che
RADON IDE Integration

RADON Kubernetes components

RADON Plugins

RADON Workspace
RADON Graphical Modeling
RADON Graphical Modeling
RADON Models

Models **automatically deployable** using the RADON orchestrator. Compatible with OASIS TOSCA standard.
RADON Delivery toolchain
Quality Guardrails in RADON

1. Create model
2. Run verification

Verification Tool

[All constraints verified]

1. Extract product metrics
   (e.g. # lines of code)
2. Extract delta metrics
   (between two successive releases)
3. Extract process metrics
   (e.g., # modifications to the file in a release)
4. Run detection

Defect Prediction Tool

Continuous Testing Tool

DEPLOY

Application Source Code

```
int div(a, b):
    return a/b
```

Possible division by zero

Infrastructure Code

```
- name: "foo"
  include: es-templates.yml
  when: es_templates
- when: es_templates
```

this makes the application behave wrongly
More about RADON

https://github.com/radon-h2020

https://radon-h2020.eu/

cf. YouTube videos