

PIACERE General Presentation

Leire Orue-Echevarria (TECNALIA)



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 101000162.



General Information

PIACERE -Programming trustworthy Infrastructure As Code in a sEcuRE framework

- → December 2020-2023 (36 months)
- →Overall budget: 4.424.250 euros
- → Project coordinator: TECNALIA
- → Technical coordinator: XLAB



















Overall objective

- The main objective of the PIACERE project is to provide means (tools, methods and techniques) to enable most organizations to fully embrace the Infrastructure-as-Code approach, through the DevSecOps philosophy, providing an end-to-end solution from Dev to Ops
- Focus on quality, security, trustworthiness and evolvability of infrastructural code
- Ensure business continuity by providing self-healing mechanisms anticipating to failures and violations, and self-learning



Target users

Target users

- → Developers of IaC
- → Operators of IaC

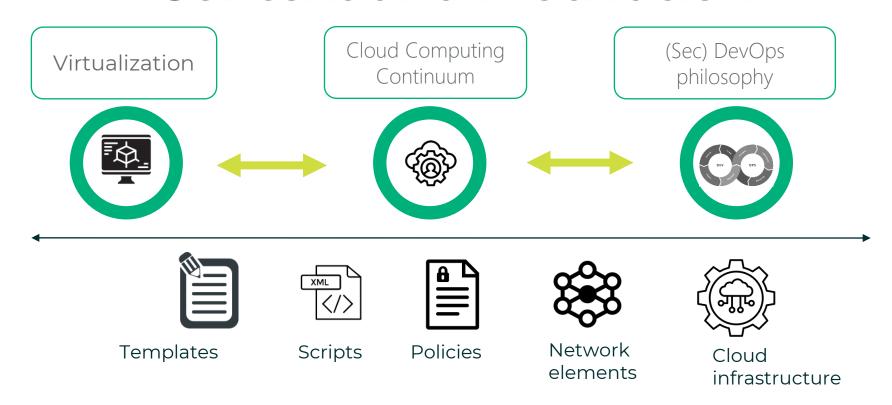


Infrastructural Code





Context and motivation



Infrastructure as Code (IaC)

Enables the automation of several deployment, configuration and management tasks that otherwise would have to be performed manually by an operator



Context and motivation: general challenges

Large variety of competing tools with different programming languages for writing infrastructural code

Focused on a single or a small set of automation steps and of types of resources (e.g., VMs)

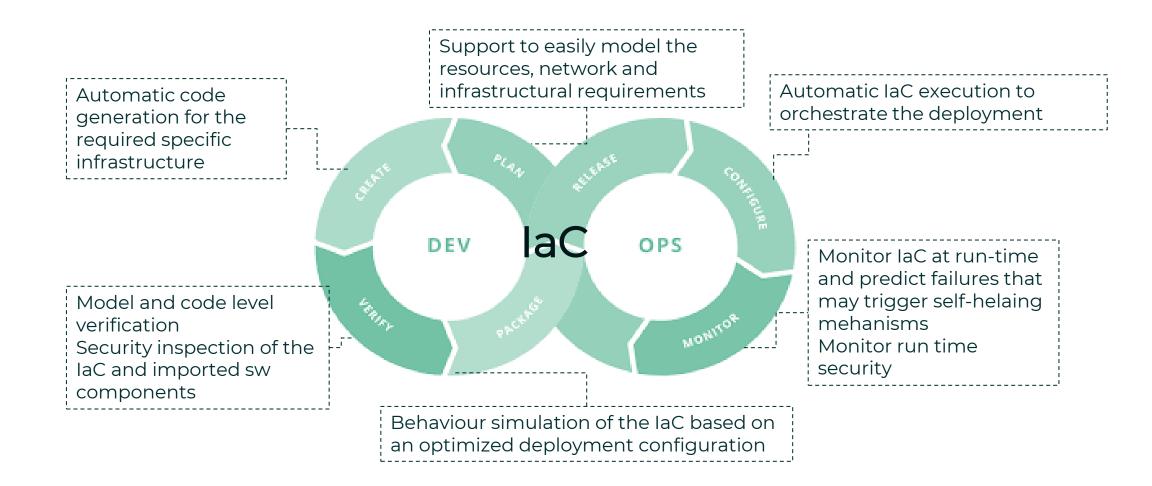
Reactive solutions to anomalies, instead of seeking to predict them. IaC not self-healable

Not really an end-to-end solution covering Devs and the Ops

Trustworthiness and security aspects of the IaC are often left for the end of the cycle



Approach and workflow





IaC DevSec



DevSecOps

Modelling Language
(DOML)



Infrastructural Code Generator laC code security inspector

Component Security inspector

Verification Tool

Creation of IaC models based on the NFRs

- Definition of the topology and properties of the infrastructure
- Abstraction from the specifities of the IaC language and protocol
- Extensible

Automatic IaC generation based on the models

- Most prominent target IaC environments and languages (e.g., Terraform, Ansible, TOSCA)
- Code generation for provisioning and deployment orchestrators, configuration management environments, monitoring platforms and other necessary elements

Models and code verification

- Verification of the correctness of the models and code
- SAST and security inspector components

Modelling

Code generation

IaC Development

IaC design, development and verification

Integration of the IaC Sec Dev process





Canary Sandbox Environment laC Optimized Platform

laC Execution Manager Self-learning and selfhealing mechanisms Performance and Runtime security monitoring

IaC simulation

 Isolated execution and testing of Infrastructure as Code behavior

A

- Identification of potential vulnerabilities and bottlenecks
- Catalogue of services and infrastructural elements
- Optimization combination of services and infrastructural elements

Automatic IaC execution

- Creation of the deployment plan
- · Interdependencies management
- Execution manager that performs the actual configuration, provisioning, and deployment orchestration

laC intelligent monitoring

- Monitoring of the infrastructure (e.g., performance, availability)
- Run time security verification
- QoS assurance through self learning and self-healing mechanisms

IaC (Pre-)deployment

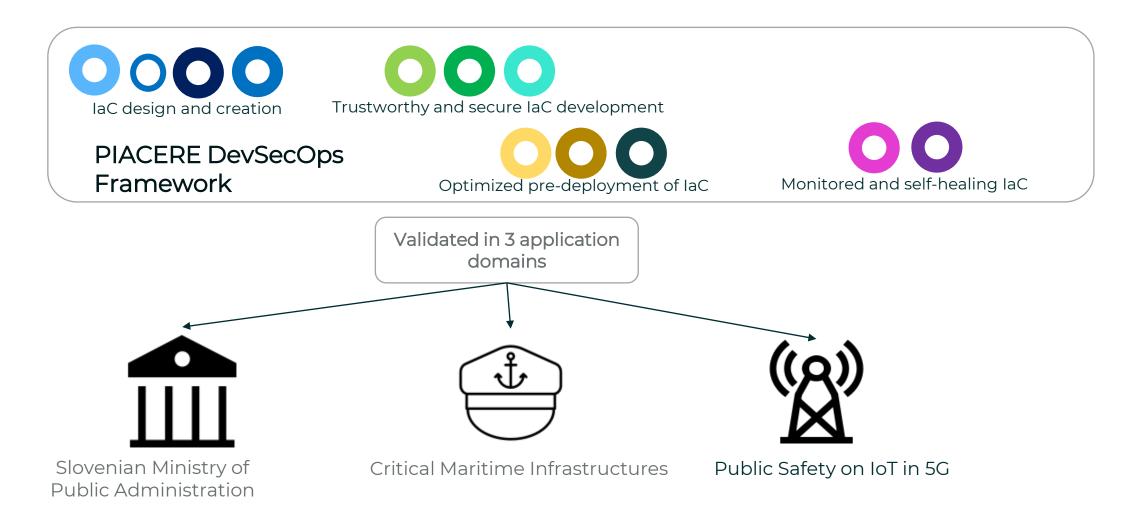
laC Operation

Automatic adaptation

To ensure that their infrastructural code is always running and available, even when an anomaly is predicted
or identified

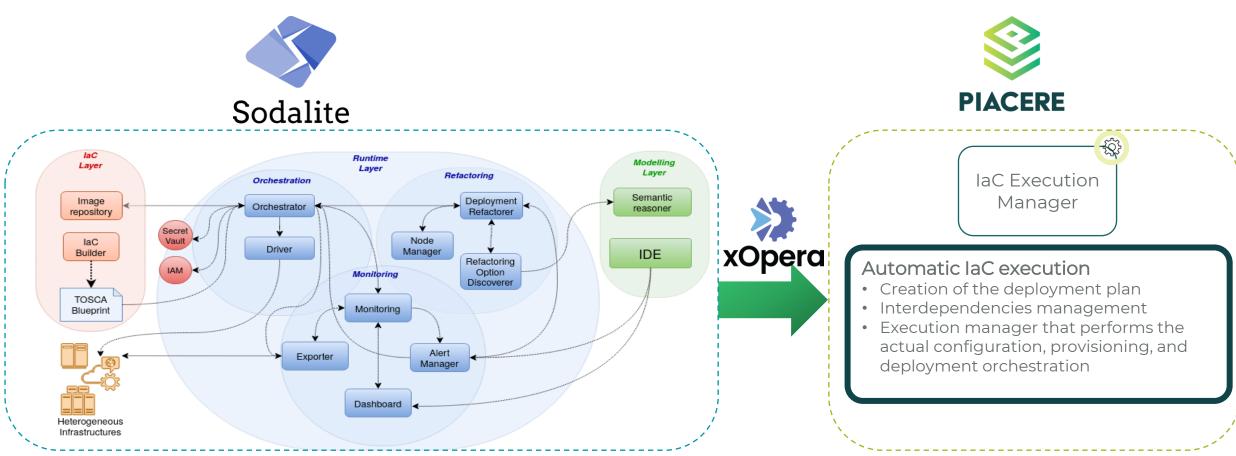


PIACERE Key Results





Relationship between SODALITE and PIACERE





Relationship between SODALITE and PIACERE







- Focus on
 - Resource Discovery
 - Deployment automation
- Deployment blueprints described in TOSCA
- (Application is considered and details known)

- Focus on IaC Deployment automation
 - Configuration
 - Provisioning
 - Deployment orchestration

Directly In Terraform and Ansible in addition to TOSCA

 Verification of the IaC, especially security (security inspector and component inspector)



Thank you!

https://www.piacere-project.eu/// Leire.Orue-Echevarria@tecnaliacom