Nispero: a cloud-computing based Scala tool specially suited for bioinformatics data processing

Evdokim Kovach, Alexey Alekhin, Marina Manrique, Pablo Pareja-Tobes, Eduardo Pareja, Raquel Tobes and Eduardo Pareja-Tobes

Oh no sequences! Research Group. Era7 bioinformatics

Introduction

Nispero is a Scala library for declaring stateless computations and scaling them using cloud computing, in particular a combination of services from AWS (Amazon Web Services).

Features

- For all configuration of Nispero we are using Scala, it makes possible to check it statically before any Amazon instance will be launched
- Nispero specially designed for bioinformatics data processing, for example it has built-in tools for working with FASTQ files.

Applications

Nispero is used as a core component of metapasta — microbial community profiling cloud tool.

Instructions

Instructions represent computations that will be launched by Nispero. They can be:

- Scala/Java/other JVM code that implements the special interface
- Any BASH script following simple conventions. It makes possible to use any programming language with Nispero.

The input data for instructions is called a task.

Why nispero? It's tasty!

(photo from http://malagasecome.es/)

Intercrossing

This project is funded in part by the ITN FP7 project INTERCROSSING (Grant 289974).

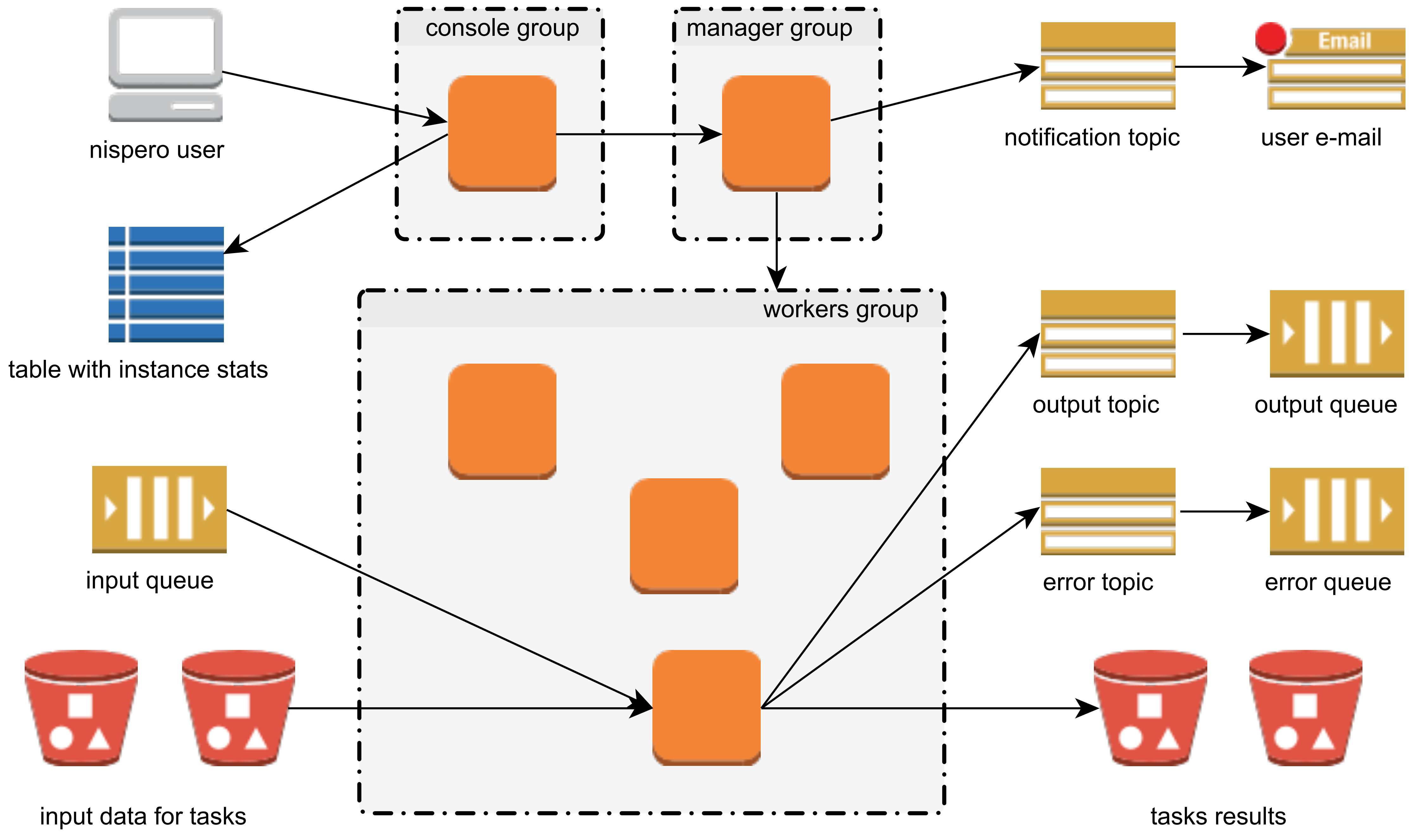
Why AWS?

- Hire as much resources as you want
- pay-as-you-go.

Why Scala?

- Type safety
- Java compatibility.

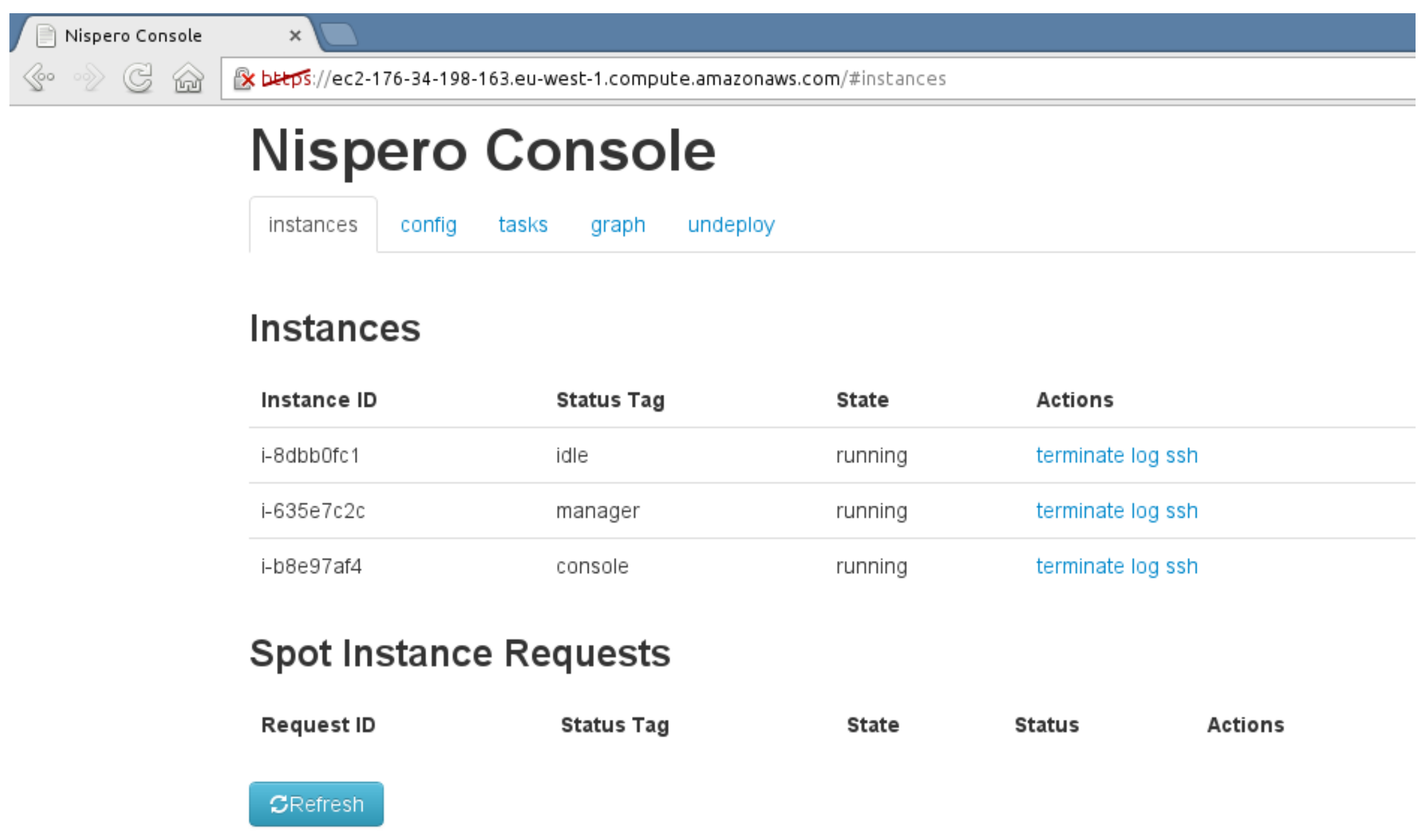
Architecture



Components

- A console instance that tracks at any moment the status of the whole system giving the user the opportunity to check at any point the current status of the computations, workers, etc
- A manager instance that is in charge of deploying and undeploying the group of workers
- A set of workers that performs the computations/tasks in a parallel, independent way
- SQS queues for input, output and error messages
- S3 objects for input and output files.

Console


Nispero Console
~~https~~://ec2-176-34-198-163.eu-west-1.compute.amazonaws.com/#instances
instances
config
tasks
graph
undeploy
Refresh

Nispero Console

Instances

Instance ID	Status Tag	State	Actions
i-8dbb0fc1	idle	running	terminate log ssh
i-635e7c2c	manager	running	terminate log ssh
i-b8e97af4	console	running	terminate log ssh

Spot Instance Requests

Request ID	Status Tag	State	Status	Actions

Monoids

$$Reads \otimes Reads \xrightarrow{merge} Reads \xrightarrow{BLAST} AssignTable \otimes Reads$$

We are using idempotent commutative monoids to describe distributed systems in Nispero. They are:

- powerful enough to build complex system
- easy to work with them abstractly (graphical workflow editor that produces distributed systems).

Availability

Nispero is an open-source project released under the AGPLv3 license.
The source code is available at
https://github.com/ohnosequences/nispero