Modello Dichiarativo per l'Automazione del Cloud GARR

GIUSEPPE ATTARDI

Dipartimento CSD, Consortium GARR

Roma 30/5/2018

Workshop GARR 2018

Intent-Based Deployment

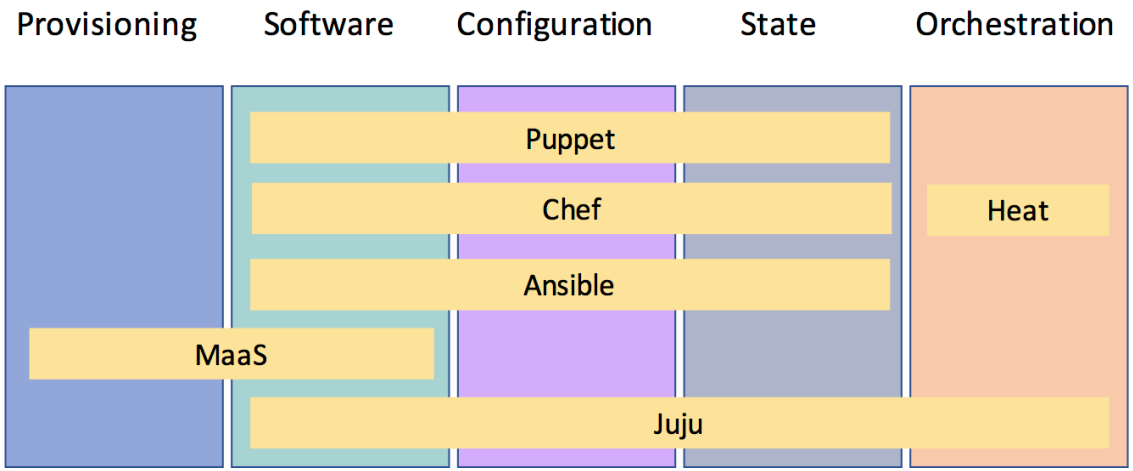
- Describe **what**, not how
- Workflow **Engine generates execution plan** from the desired model
- Asynchronous process that converges by computing the **differences** between the **current** and the **desired state**

Benefits of Declarative Modeling

- Portability
  - Models can be deployed across platforms
- Consistency
  - Both physical and virtual infrastructure can be modeled
  - Relationships between components
  - Changes are propagated
- Automation
  - Mapping model to infrastructure delegated to orchestrator
- Evolution
  - Scaling up/down
  - Upgrades
  - maintenance

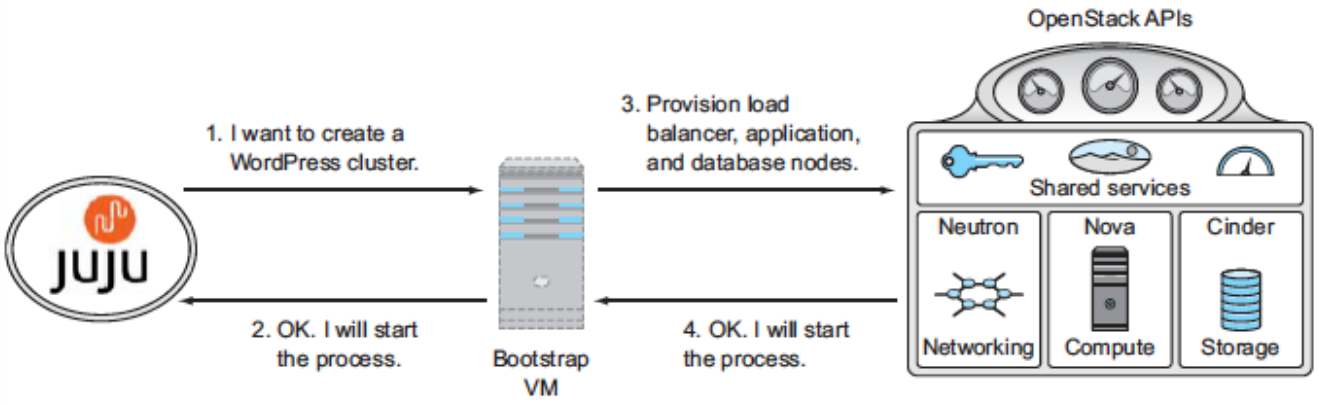
Automazione

Automation Tools



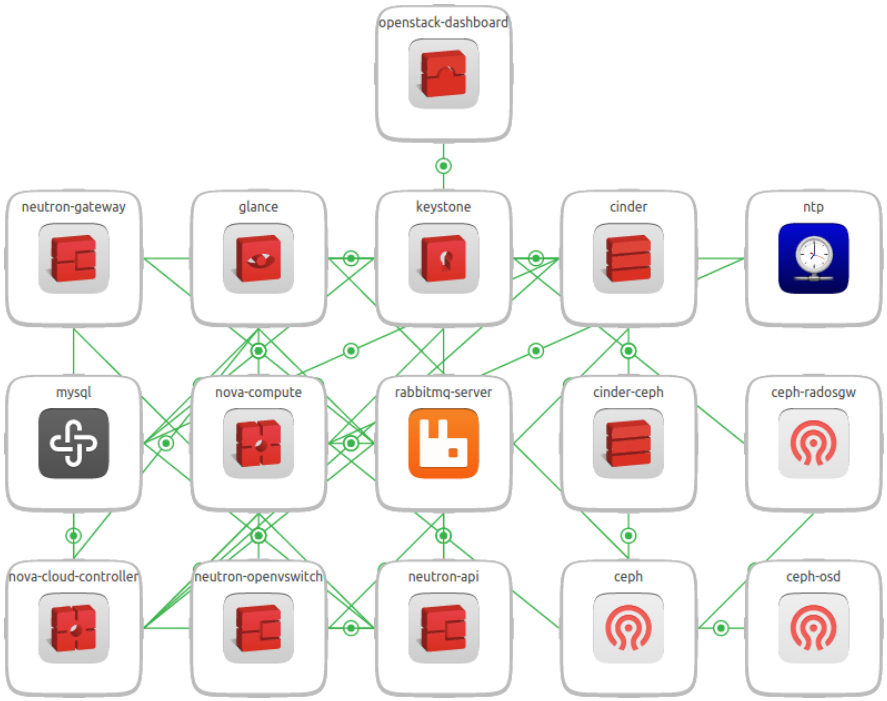
Juju

- Declarative modelling tool for composing software applications
- Charms express the steps through the lifecycle of a software component
- Performs installation on any cloud: OpenStack, AWS, Azure, Google
- Automates deployment of both OpenStack and cloud applications
- Similar t
  - AWS C
  - Brook

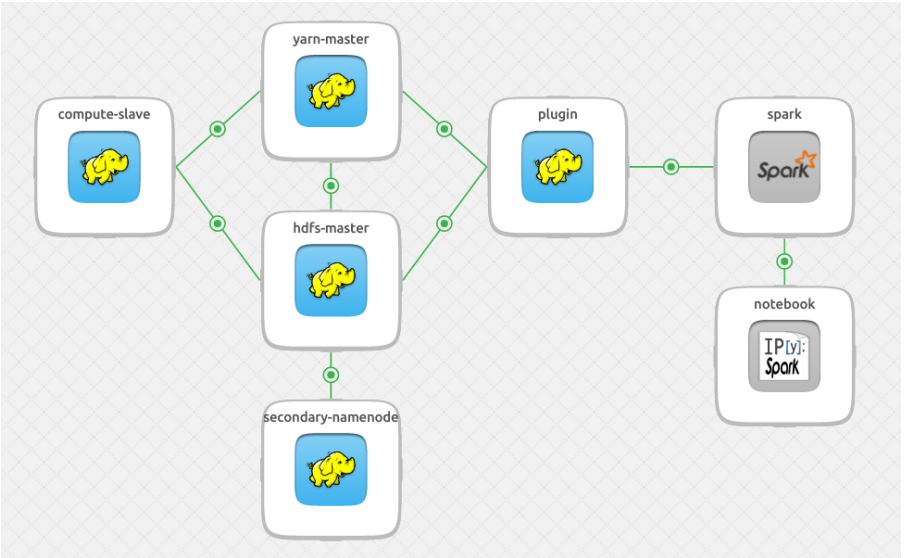


A Single Automation Tool for all Tasks

Platform Deployment: OpenStack



Application Deployment: Big Data Analytics



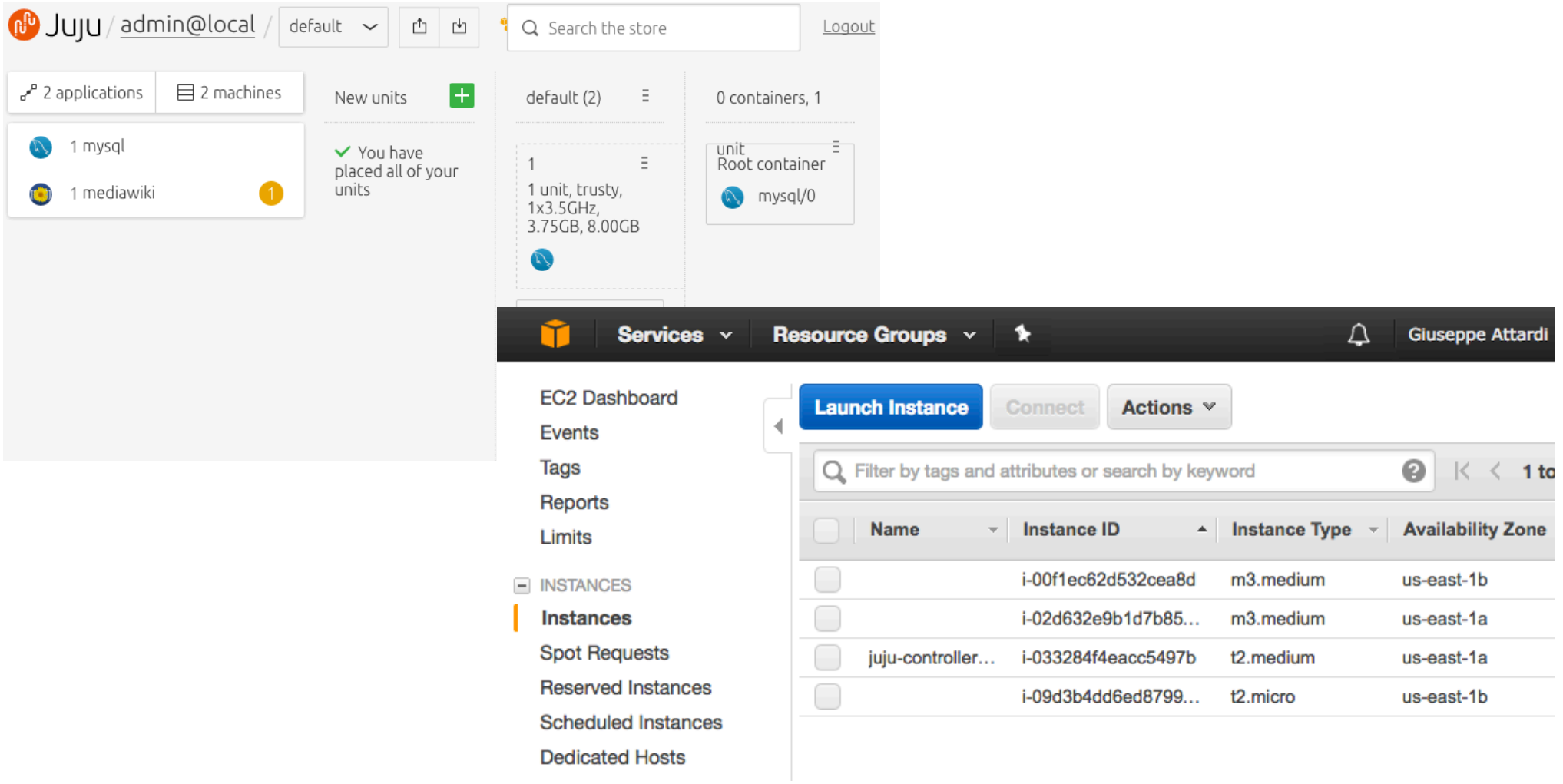
Deployment Duration

- Deploying an OpenStack region from bare metal
- Half a day
- Automated upgrade to OpenStack releases
- Half an hour
- Deploying a Container Platform
- 2 days

Deployment as a Service

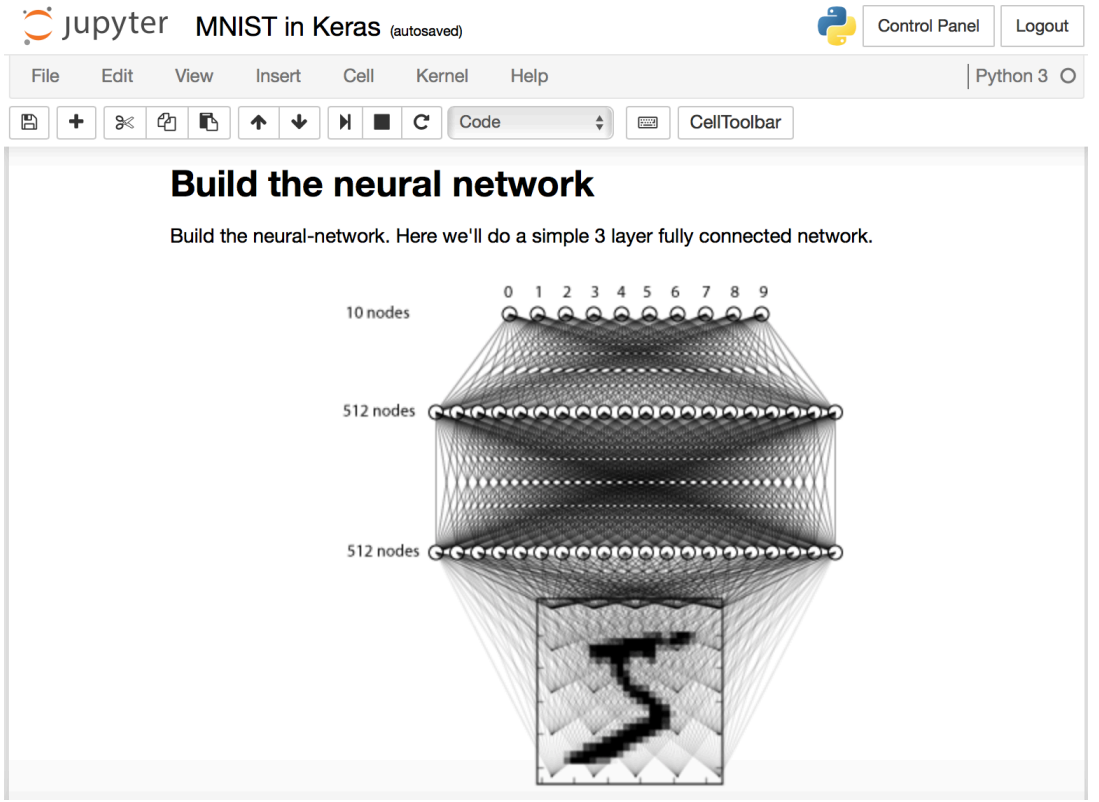
Self-service app deployment

App Deployed on AWS



Jupyter Notebook Server

- Experiment live with Machine Learning and GPUs



Juju Details

Typical Fabric, Ansible, Chef code

- Connect to each server
- Install packages for a web application
- Configure web application, styles nd database credentials
- Connect database server XXX: NEEDS REPLICA
- Create table and populate data

Charm Interfaces

- Interfaces define how different charms can be related to share data
- One charm is the provider, like a socket
- Any char can consume the interface, like a p;ug
- Juju operates as the information exchange broker beween the two charms

Repeatability

- A model can be described through a YAML file
- The model can be deployed with a singe command
- The file can also contain the deployment details
  - The number of instances of an application
  - Where the applications should be locate
  - Whether the application are deployed into containers or machines
- Same model can be used for pre-production testing on a small scale and then scaled up

Managing Evolution

- From development to production
- Security updates
- Monitoring
- Log aggregation
- Certificates

Day Two Operations: Scaling and Adapting

- Scaling applications, while keeping related applications notified
- Perform configuration updates
- Relation configuration updates

Upgrades

- Upgrading complex software requires coordination between components
  - Juju provides this coordination point
- OpenStack ugrades:
  - Mitaka -> Newton -> Ocata
- Kubernetes upgrades:
  - 1.8 -> 1.9
- Upgrading the charms themselves as their functionality is improved

Lifecycle Events

- install
  - Invoked just once when the charm is deployed
- config-changed
  - Invoked whenever a configuration parameter is changed (either grom the GUI, or from the CLI)
- relation-joined, relation-changed
  - When a relation is added to a charm relation-joined fires first, so that the two units can communicate with each other, and then relation-changed fires
- leader-elected
  - Occurs when many nodes require a “leader” node to coordinate among them
- pool-storage-attached, pool-storage-detached
  - Actions to take when a storage pool is attached/detached

Hooks

- Represent the handlers to be run when an event occurs
- Hooks must be *idempotent*
  - To avoid inconsistencies or divergence if run more than once

Bundles

- Bundles describe a service consisting of several charms
- They express constrains, configuration parameters and relationships between charms that provide/implement an interface
- Can be configured before/after deployment
- They provide scalability options

Juju Engine

- The Juju engine follows a reactive pattern, triggered by events that cause corresponding hook handlers to run
- Multiple handlers may match for a given hook and will be run in a non-determined order
- Running the handlers or issuing Juju commands may cause additional events
- The state engine is evaluated every time an event occurs
- The engine runs until convergence to a stable state

Actions

- Actions are executable scripts defined in the charms
- High level functionality related to the application:
  - Pause and resume replication for postgresql
  - Creating, renaming or deleting pools in ceph
- Can be executed on one or more instances of the application

Developing Bundles

- Expand a shared Catalogue of services
- Examples:
  - Moodle as a Service
  - Jupyter Notebooks as a Service

Status

- **Resources**
  - ~9000 vCPU
  - 10 PB Storage
- **Usage**
  - Over 700 users
  - Over 1000 VM
- **Guarantees**
  - Service Continuity
  - Data Protection

Usage

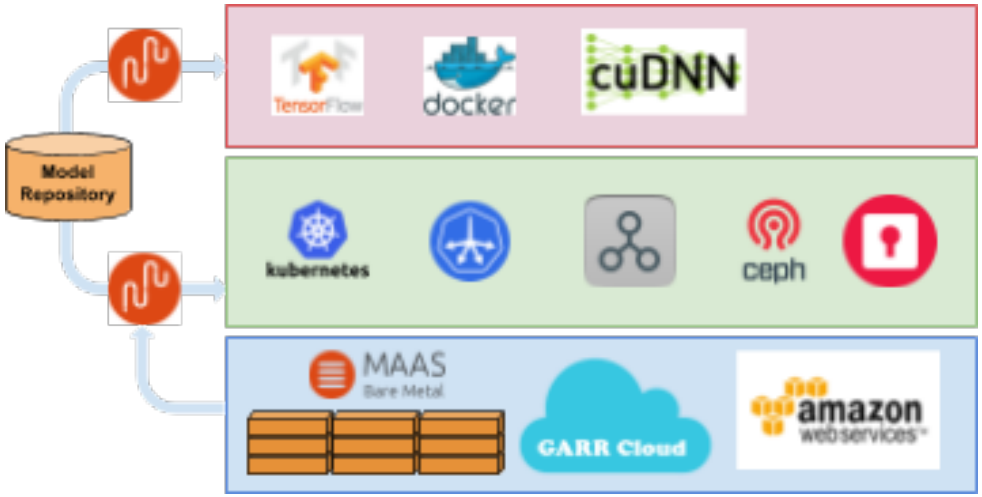
Download CSV Summary (?start=2016-11-01&end=2017-03-22&format=csv

Project Name	VCPUs	Disk	RAM	VCPU Hours	Disk GB Hours	Memory MB Hours
isti	368	2.8TB	656GB	164333.47	1322184.30	279763673.66
unipa	336	7.5TB	1.2TB	693206.14	16207815.52	2692199747.8
ws2017ipv6	128	2.5TB	256GB	14955.82	299116.37	30629516.41
lns-prj1	54	543GB	71.5GB	118279.27	1629777.45	189620873.70
garrdemo318	32	20GB	32GB	16914.14	10571.34	17320083.68
INFN-FI	32	40GB	16GB	601.92	771.89	309779.80
wsosadmin	27	540GB	54GB	4510.02	90200.37	9236517.59
GEANT	25	481GB	48.5GB	29681.28	591502.48	60615623.66
infn-vlabs	24	480GB	48GB	8052.53	161050.61	16491582.24
demo	17	340GB	34GB	9877.99	197559.85	20230128.54
SSSUP	16	80GB	16GB	12651.09	63258.04	12954892.40
garrdemo125	8	100GB	128GB	13429.14	167864.31	220023110.77

Container Platform

Container Platform for AI

- Automated deployment on bare metal, AWS or other clouds by Juju
- Workloads deployed by Juju
- Distributed storage system using Ceph
- NFS cluster for sharing big data
- Docker containers managed Kubernetes

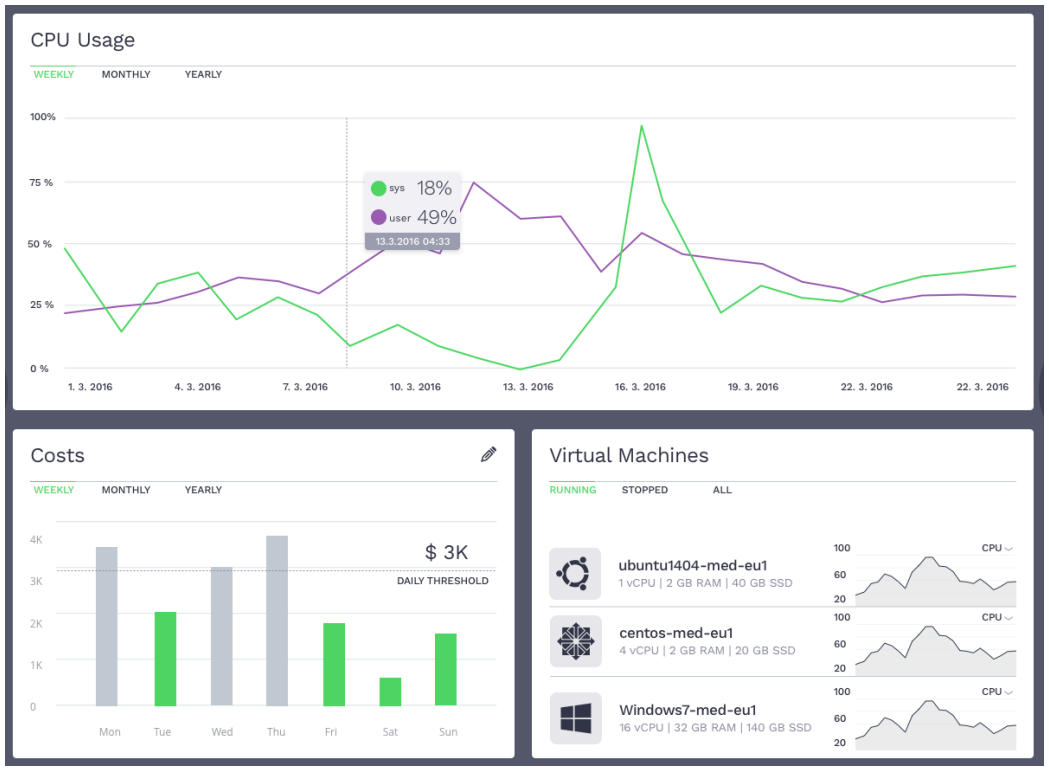


Kubernetes Deployment by Juju

- Kubernetes Nine node Kubernetes cluster with one master and a configurable number (> 3) of worker nodes.
- EasyRSA performs the role of a certificate authority serving self-signed certificates to the requesting units of the cluster.
- Etcd provides a distributed key value store: three node cluster for reliability.
- Ceph provides distributed resilient storage
- CephFS provides shared storage
- Keystone enables authenticating registered OpenStack users to the Container Platform
- Flannel provides a CNI (Container Network Interface) among the nodes

Charging and Billing

Domain Administration Dashboard



Conclusioni

Pro

- Esplicitare l'architettura desiderata
- Livello di astrazione più alto
- Assicura consistenza tra le parti
- Opera su strutture anziché su file di configurazione
- Delega al tool dei passi elementari
- Delega al tool scelte non cruciali
- Riduce il rischio di sviste

–> 90% delle interruzioni del servizio dovuti a interventi manuali

Cons

- Conoscere strumento
- Sistemisti abituati ad operare direttamente sui file di confiurazione
- Dipendenza dallo strumento e dalle sue evoluzioni
- Seguire l'evoluzione dello strumento/i
- Standards (Tosca?)

IL PROSSIMO OPENSTACK DAY È A
ROMA IL 21.09.2018
SCOPRI DI PIÙ AL #MEETUP APERITECH DI OPENSTACK

14 GIUGNO H.19

14 GIUGNO H.19
#MEETUP APERITECH
@LUISS ENLABS
ROMA, VIA MARSALA 29H

ITALY
OpenStack Days

Iscrizioni su Eventbrite:
https://bit.ly/2xodHCI