

Making Apache CloudStack market ready with a native rating solution

Daniel Augusto Veronezi Salvador
<gutoveronezi@apache.org>



Agenda

- ❖ Introduction
- ❖ Usage (accounting system)
- ❖ Quota (rating system)
- ❖ Use cases
- ❖ Current limitations
- ❖ Future works
- ❖ Questions

Introduction

- ❖ **Rating:** process of applying a monetary value to a computing resource usage
- ❖ **Billing:** process of invoicing and collecting/processing payments
 - We do not do that on CloudStack; that is responsibility of other systems

Introduction

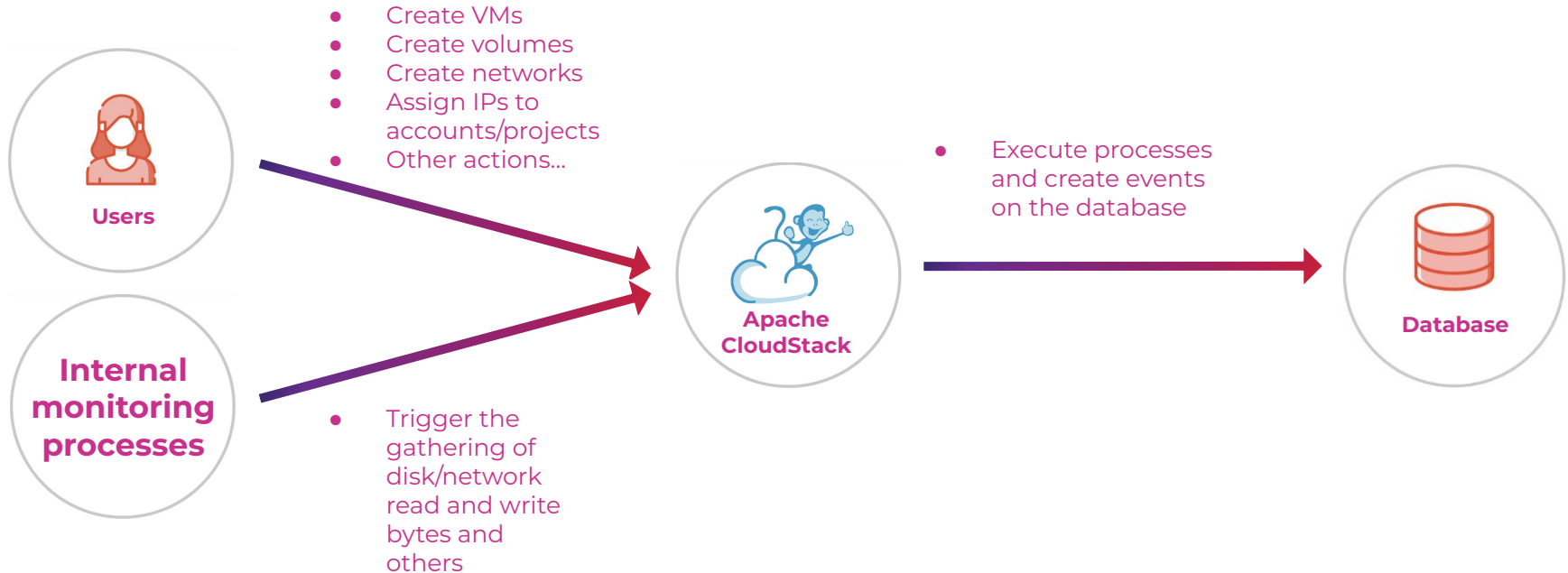
- ❖ Why CloudStack as a cloud orchestrator?
 - Solid option to deploy a cloud environment
 - Turn-key solution
 - Easy-to-use graphical interface
- ❖ What it was missing to be market ready?
 - **A native solution for rating the cloud consumption**
(third party softwares and tools were necessary)
 - Runtime white-label sub-system (presented in the previous session)
- ❖ **We will address the rating module we improved**

Usage

- ❖ Service for accounting the consumption of cloud resources on CloudStack
- ❖ Data generated based on platform events
- ❖ Per account/project
- ❖ 2 steps: event generation and event processing

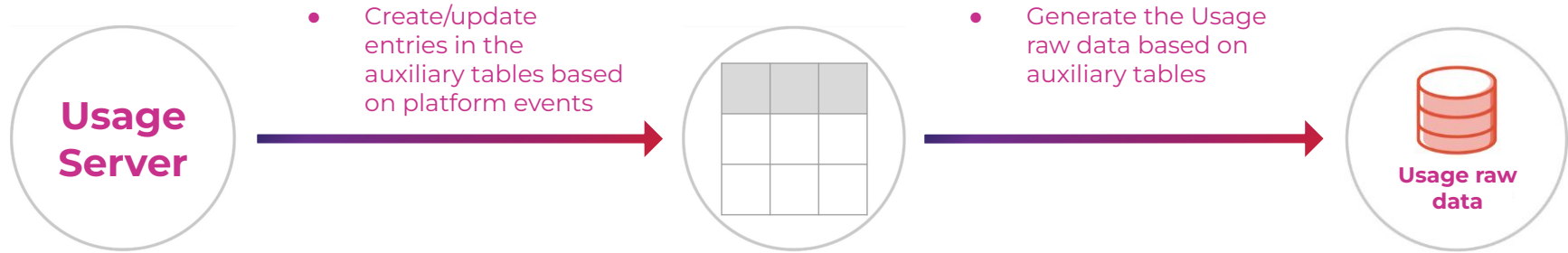
Usage

Event generation workflow



Usage

Event processing workflow



Quota

v1.0

- ❖ Plugin to generate a monetary value based on computing resource consumption/usage
- ❖ Hard-coded per resource tariffs
- ❖ Same values applied to every resource
- ❖ Very limited
- ❖ It did not work as expected
- ❖ It was not addressing the cloud providers needs

Quota

v2.0

- ❖ Multiple tariffs per resource type
- ❖ Tariffs can use rating rules to define/generate monetary value
- ❖ Context-based tariffs
- ❖ Per entry and monthly processing
- ❖ Ordered tariffs processing
- ❖ Prepaid/postpaid (PAYG) approaches

Quota

Context-based tariffs

- ❖ Rules are written in JavaScript
- ❖ Context injected in the rules as attributes
- ❖ Common context for all resource types and specific contexts for some resources types

Use cases

#1 - Applying different prices to VMs on different hosts

#2 - Pricing differently based on the contracts

#3 - Applying discount over a certain amount of consumption in the end of the month

Use cases

#1 - Applying different prices to VMs on different hosts

#2 - Pricing differently based on the contracts

#3 - Applying discount over a certain amount of consumption in the end of the month

Use cases

#1 - Applying different prices to VMs on different hosts

We have 3 clusters:

- cluster X has 3 hosts with Intel Xeon Platinum processors and they have the tag **platinum**;
- cluster Y has 5 hosts with Intel Xeon Gold and they have the tag **gold**;
- cluster Z has 12 hosts with other processor and without tag/with other tags.

For VMs running on cluster X, the cost will be 50 monetary units per month; for VMs running on cluster Y, the cost will be 40 monetary units per month; and for VMs running in cluster Z, the cost will be 30 monetary units per month.

Use cases

#1 - Applying different prices to VMs on different hosts

```
if (value.host.tags[0].indexOf('platinum') !== -1) {  
  50  
} else if (value.host.tags[0].indexOf('gold') !== -1) {  
  40  
} else {  
  30  
}
```

Use cases

#1 - Applying different prices to VMs on different hosts

#2 - Pricing differently based on the contracts

#3 - Applying discount over a certain amount of consumption in the end of the month

Use cases

#2 - Pricing differently based on the contracts

Due to the long partnership, client T will receive 10% discount on running VMs.

- As tariffs can be applied in order, this one could be configured to run after the previous one, with its output as an attribute.

Use cases

#2 - Pricing differently based on the contracts

```
If (account.id == 'f0336ed8-e869-413f-9002-33d7dde11e14') {  
    previousValue = lastTariffs[0].value  
    0 - (previousValue * 0.1)  
} else {  
    0  
}
```

Use cases

#1 - Applying different prices to VMs on different hosts

#2 - Pricing differently based on the contracts

#3 - Applying discount over a certain amount of consumption in the end of the month

Use cases

#3 - Applying discount over a certain amount of consumption in the end of the month

Clients get 5% discount at the end of the month if they use they VMs more than 100 hours during the month.

Use cases

#3 - Applying discount over a certain amount of consumption in the end of the month

```
totalQuota = 0
totalUsage = 0

for (i = 0; i < processedData.length; i++) {
  entry = processedData[i]
  totalQuota += entry.aggregatedTariffsValue
  totalUsage += entry.usageValue
}

thresholdInTimeRunning = 100

if (totalUsage > thresholdInTimeRunning) {
  0 - (totalQuota * 0.05)
} else {
  0
}
```

Current limitations

- ❖ Tariff rules processed with ECMAScript 5.1
- ❖ Official documentation is limited ([#5891](#))
- ❖ Not all changes are on upstream (yet)
- ❖ Hardcoded Usage types bounded to the platform events

Future works

- ❖ Support of newer ECMAScript versions
- ❖ Improve official documentation
- ❖ Push the lasts changes upstream
- ❖ Support dynamic Usage types and data beyond the platform events

Let's look at a
demonstration!

Questions?

gutoveronezi@apache.org