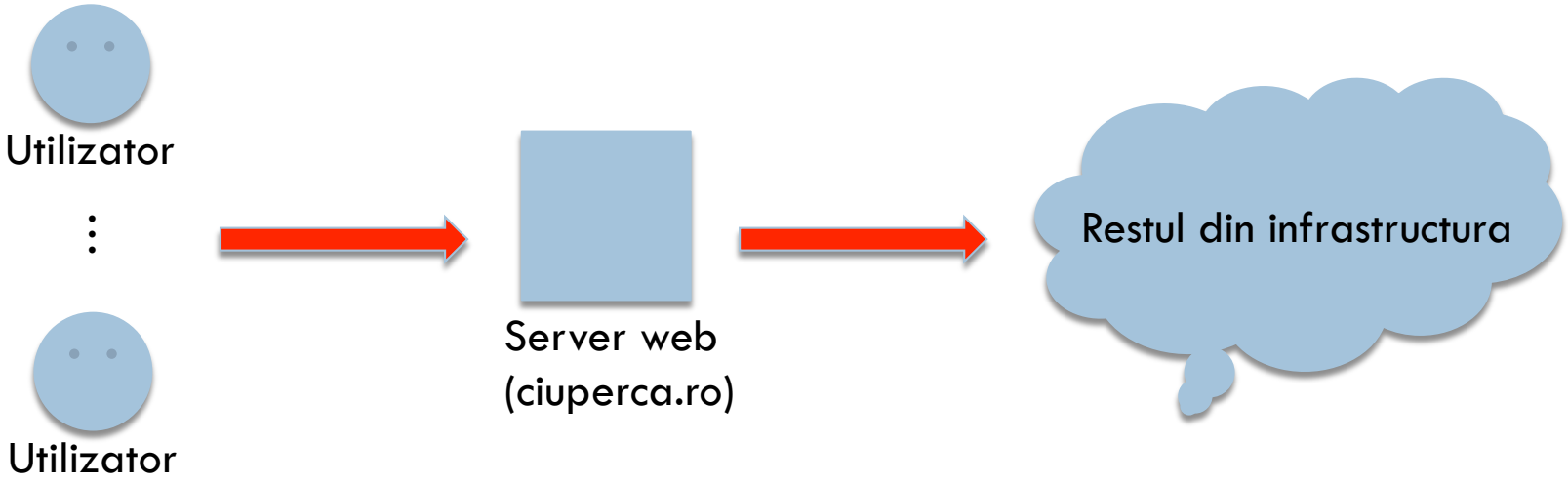
# BIG DATA SI SCALABILITATE

"Load balancer" si exemple de sisteme

# Load balancer

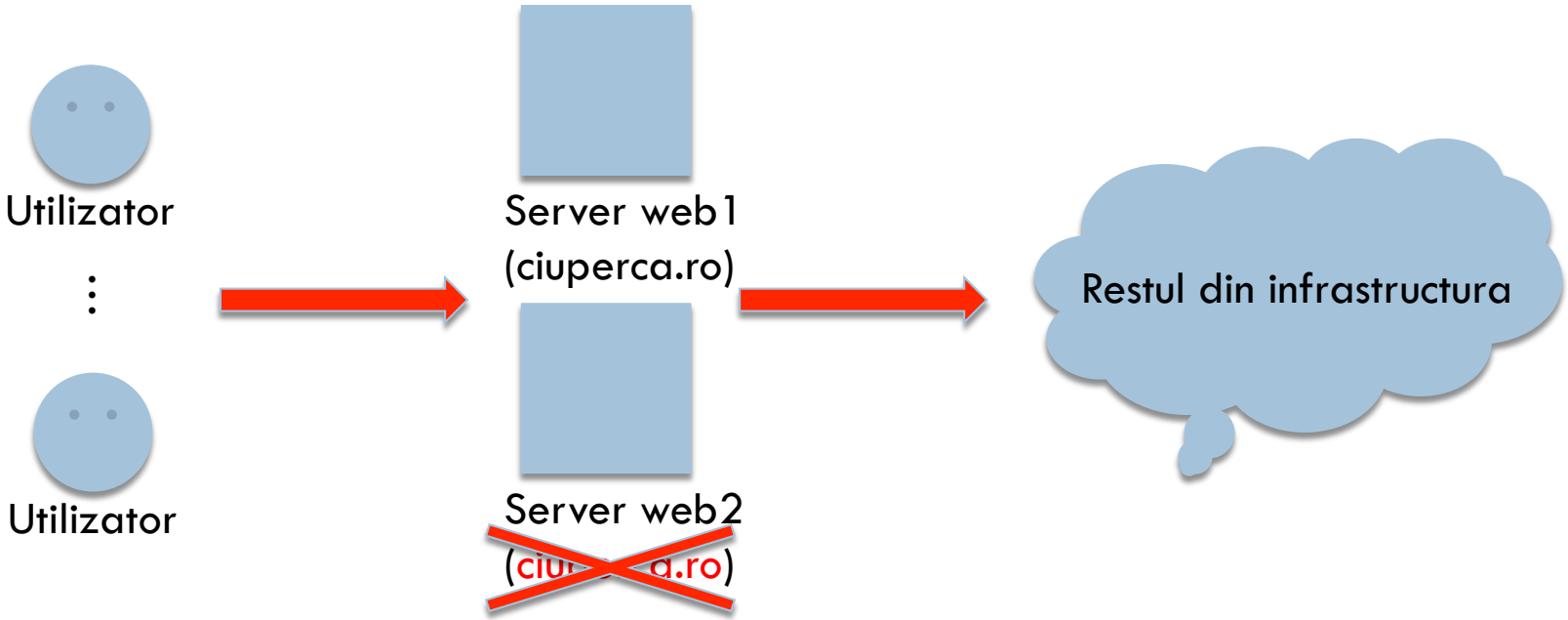
Considerati urmatorul scenariu:




Serverul web poate servi X utilizatori simultan. Cum putem servi X+1 utilizatori simultan?

# Load balancer

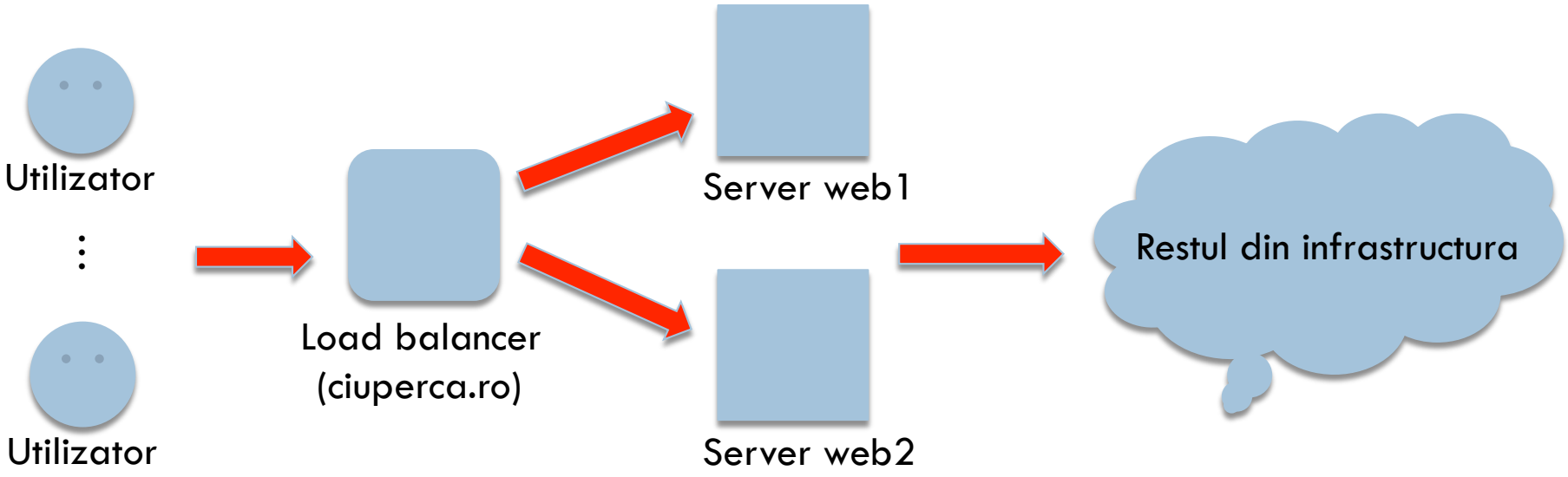
## Adaugam inca un server




Cel de-al 2 lea server este functional dar nu poate folosi acelasi nume de domeniu.

# Load balancer

## Solutie: alta masina care doar distribuie cererile




Numele de domeniu este atribuit load balancer-ului. Astfel ambele server-e web sunt accesibile prin intermediul lui.

# Load balancer

- Masina pe care ruleaza load balancer-ul are in general mai multe interfete de retea ( astfel poate avea mai multe conexiuni concurente ).
- Din cauza ca nu el face procesarea, masina pe care ruleaza poate avea resurse limitate.
- Daca serverele web sunt in internetul public, load balancer-ul poate sa redirecteze cererea direct intr-un server fara sa tina el conexiunea deschisa.

# Load balancer: algoritmi

## Round robin

- Cererile sunt distribuite uniform dupa formula: $cerere_x$->$masina_{x \bmod n}$ unde ‘x’ este numarul de ordine al cererii si ‘n’ este numarul de servere.
- Presupune ca serverele nu mentin date despre cererile anterioare. Orice server poate procesa orice cerere in orice moment.

# Load balancer: algoritmi

## In functie de utilizarea resurselor

- Cererea este procesata de serverul care are factorul de incarcare cel mai mic ( cpu, io, ram etc ).
- Se foloseste cand timpul/resursele necesare procesarilor variaza mult in functie de tipul de cerere.
- Presupune ca serverele nu mentin date despre cererile anterioare. Orice server poate procesa orice cerere in orice moment.

# Load balancer: algoritmi

## Sticky sessions

- Daca o cerere are asociata un id de sesiune, cererea merge la serverul web care a creat sesiunea respectiva. Daca nu are asociata un id de sesiune se foloseste alt algoritm.
- Cererile nu sunt distribuite uniform.
- De exemplu un magazin virtual trebuie sa foloseasca sticky sessions, altfel cosul de cumparaturi nu va functiona.

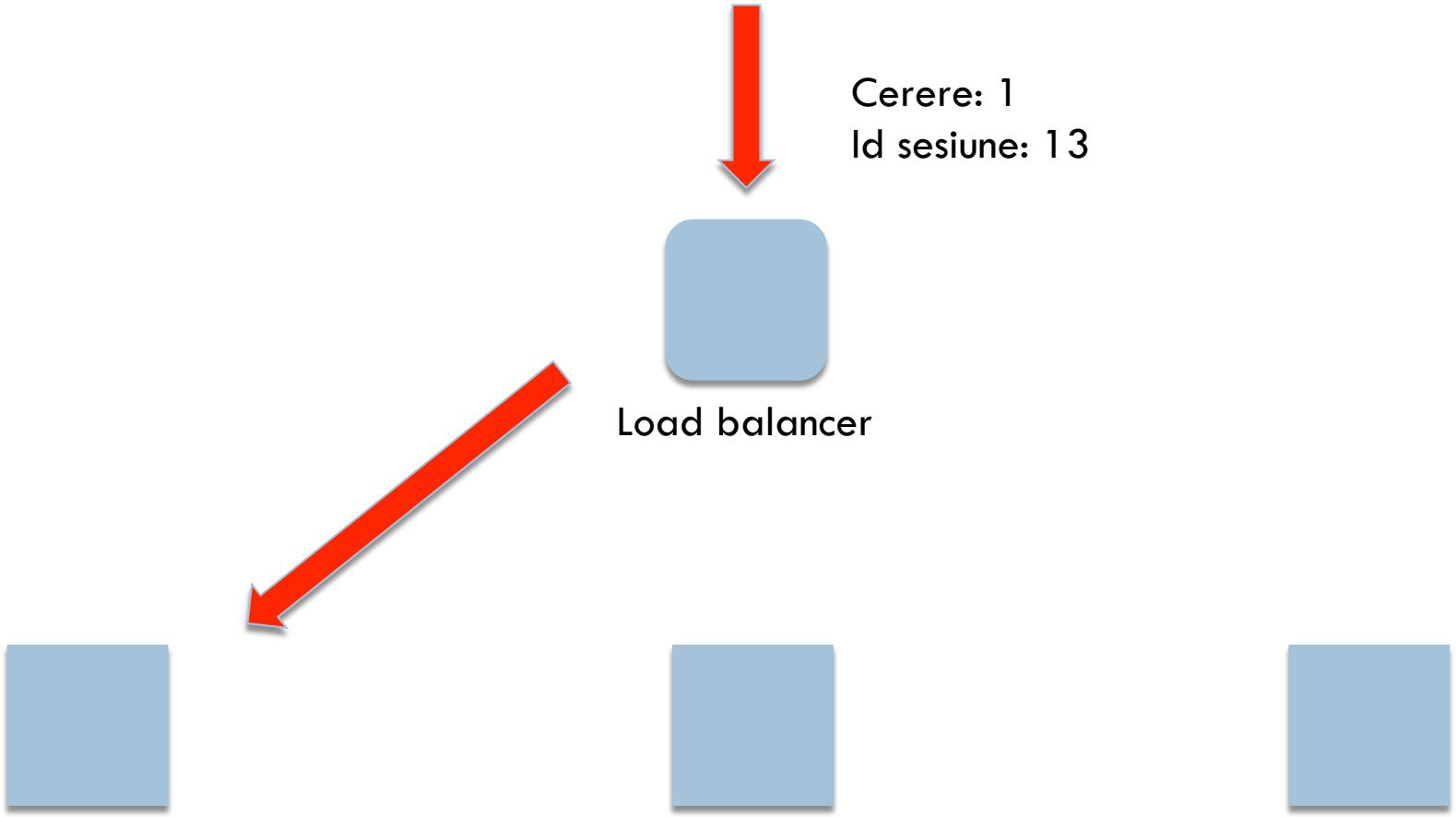
# Load balancer: algoritmi

Eliminarea nevoii de sticky sessions:

Sesiunile pot fi stocate intr-o baza de date ( nu pe serverul web ) iar serverul web extrage informatiile de acolo cand cererea are asociata un id de sesiune. Astfel nu este nevoie de sticky sessions dar trebuie luat in considerare ca pentru fiecare cerere venita de la utilizator, serverul web interogheaza baza de date.

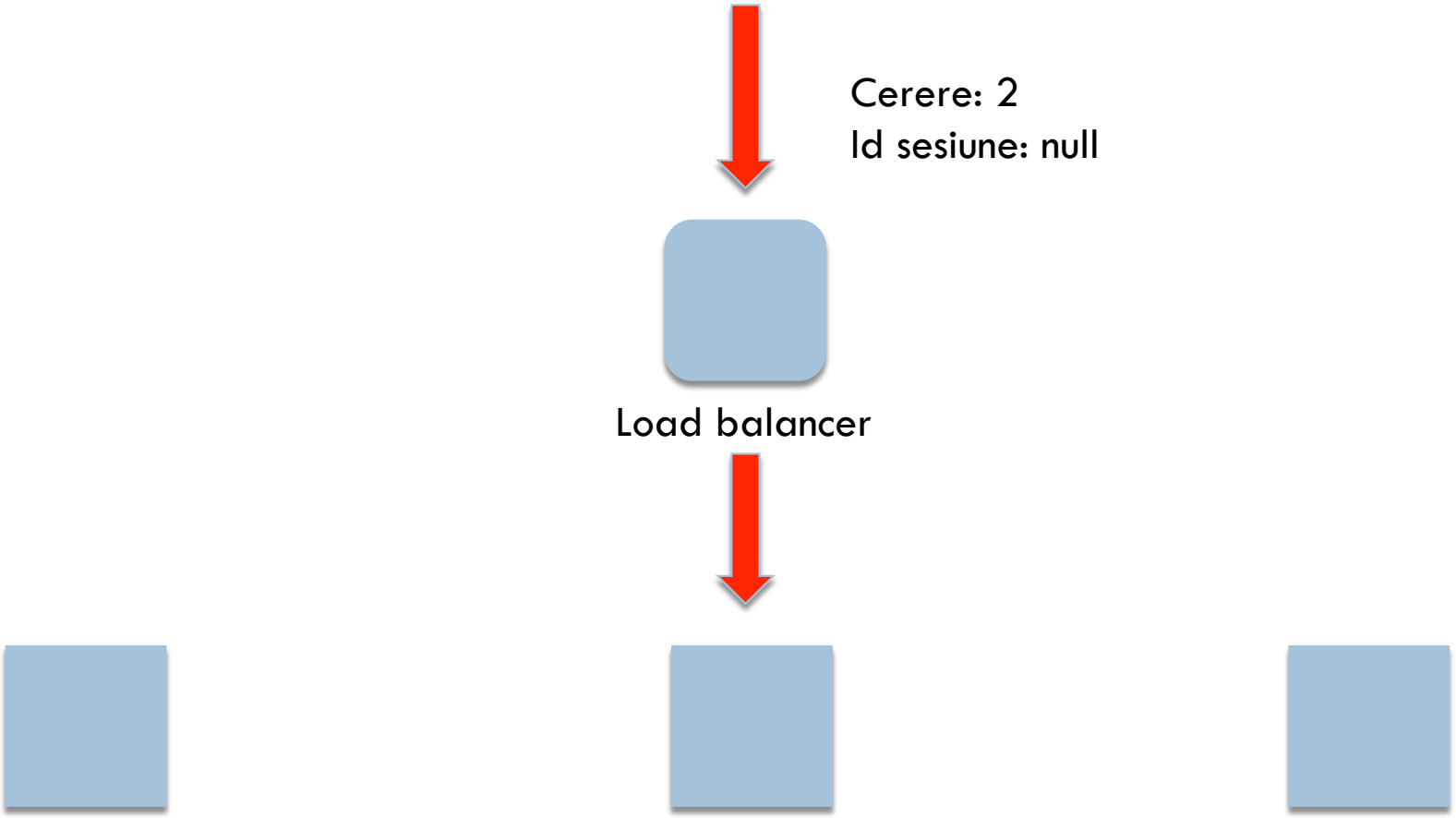
# Load balancer

Exemplu sticky sessions



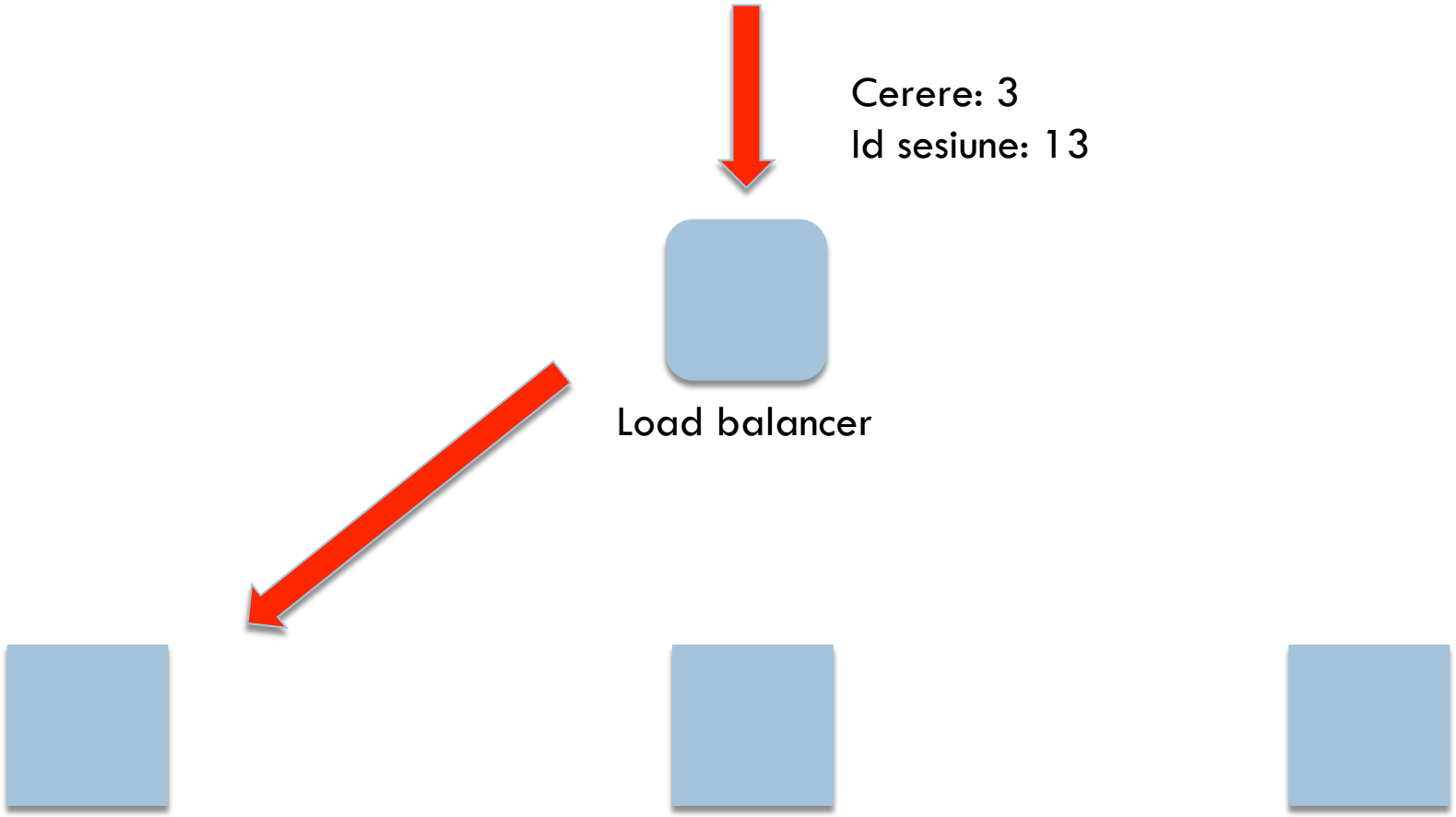
# Load balancer

Exemplu sticky sessions



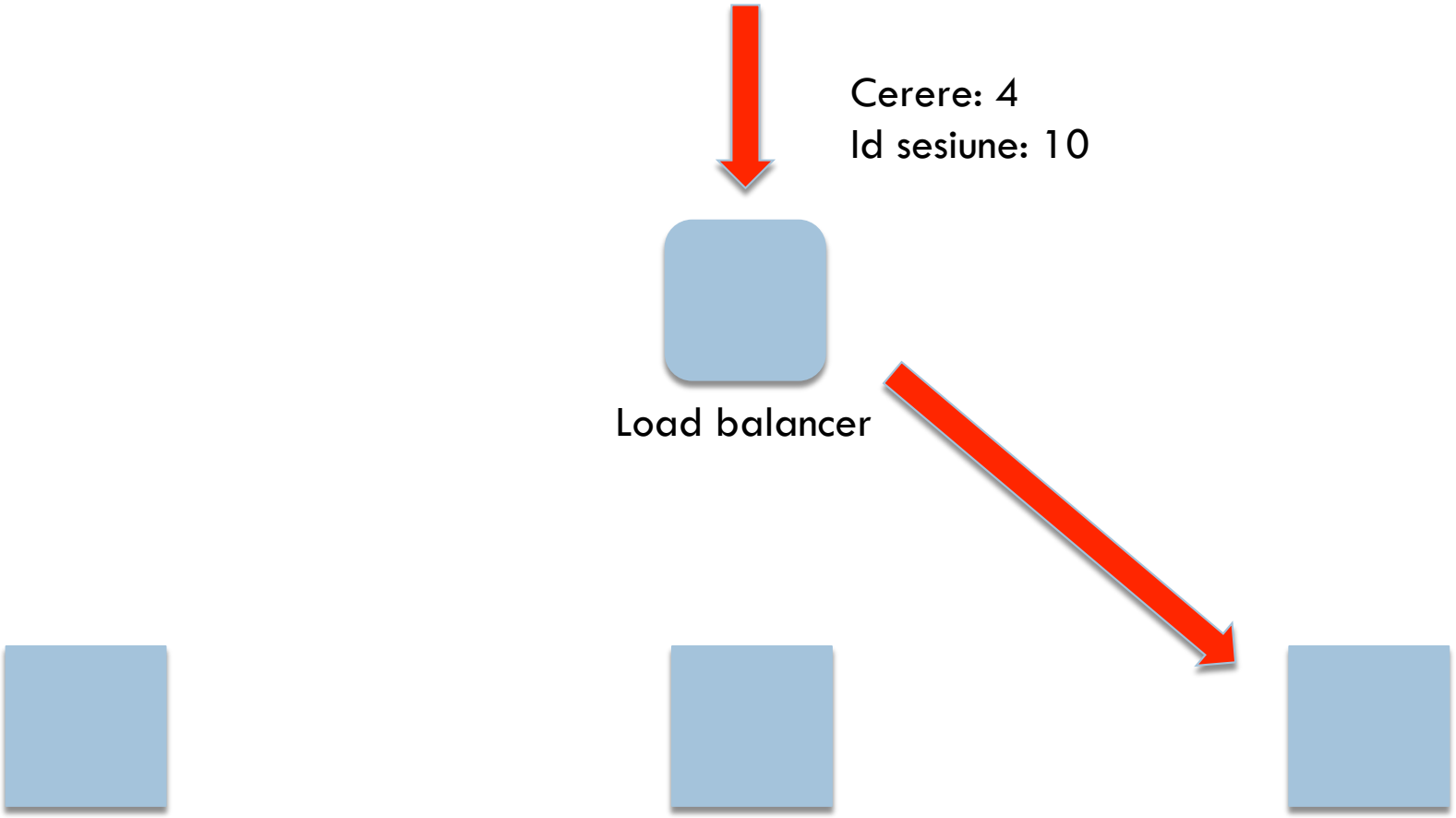
# Load balancer

Exemplu sticky sessions



# Load balancer

Exemplu sticky sessions



# Load balancer: consistenta

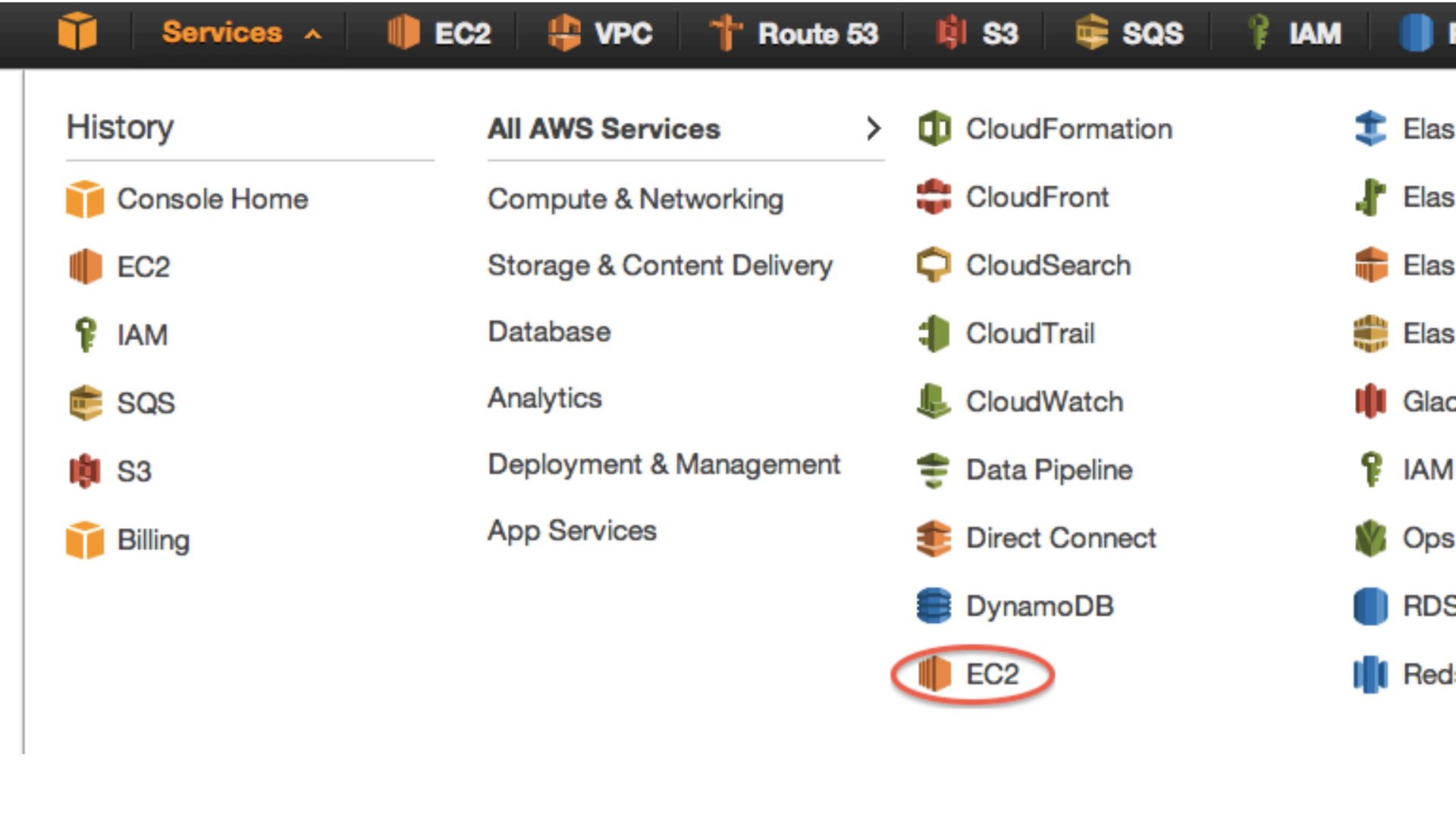
- Load balancer-ul verifica permanent starea serverelor web (healthcheck).
- Daca un server web se defecteaza, load balancer-ul il scoate din lista lui de clienti.

# Load balancer in AWS

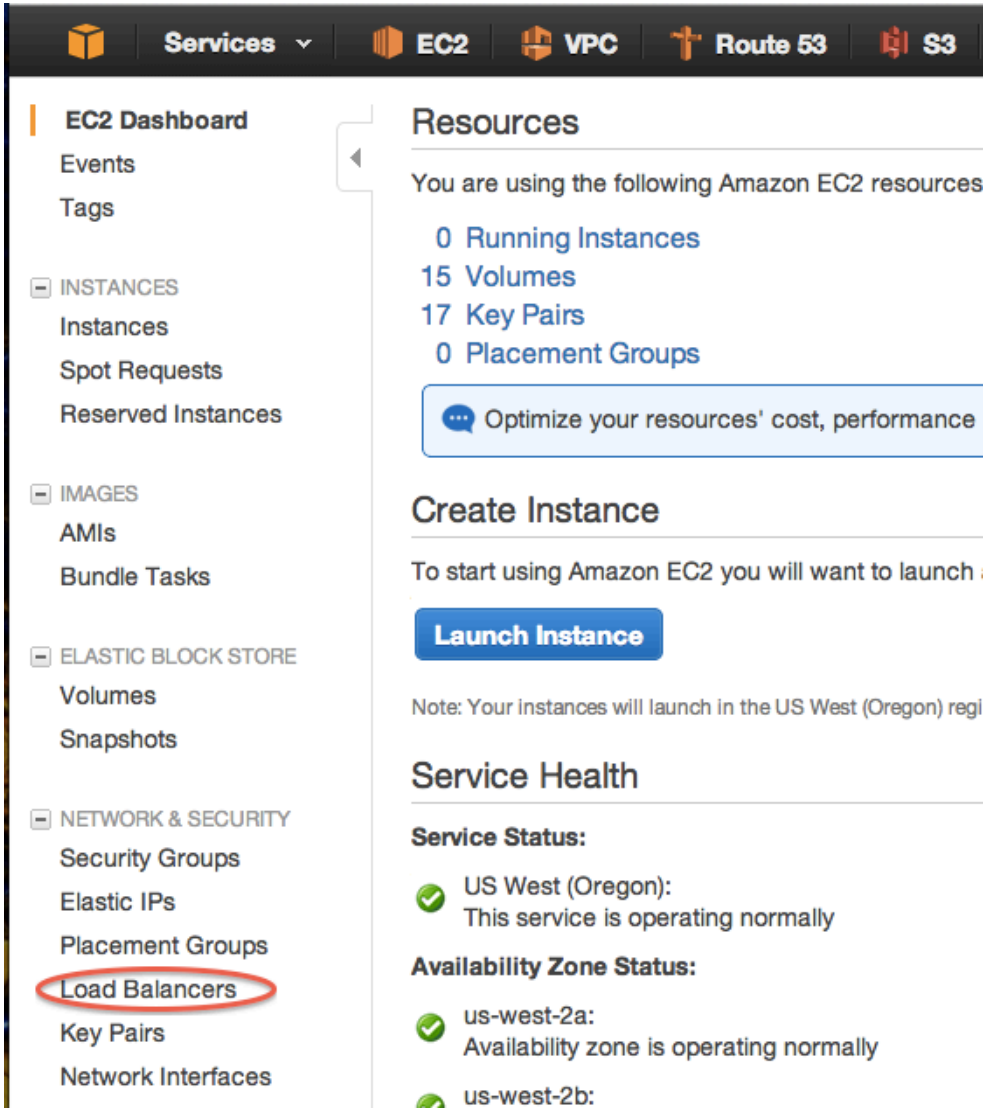
ELB: Elastic load balancer

- Gestionat de AWS.
- Scaleaza automat in functie de cantitate de trafic.
- Suporta ipv6.
- Suporta sticky sessions.

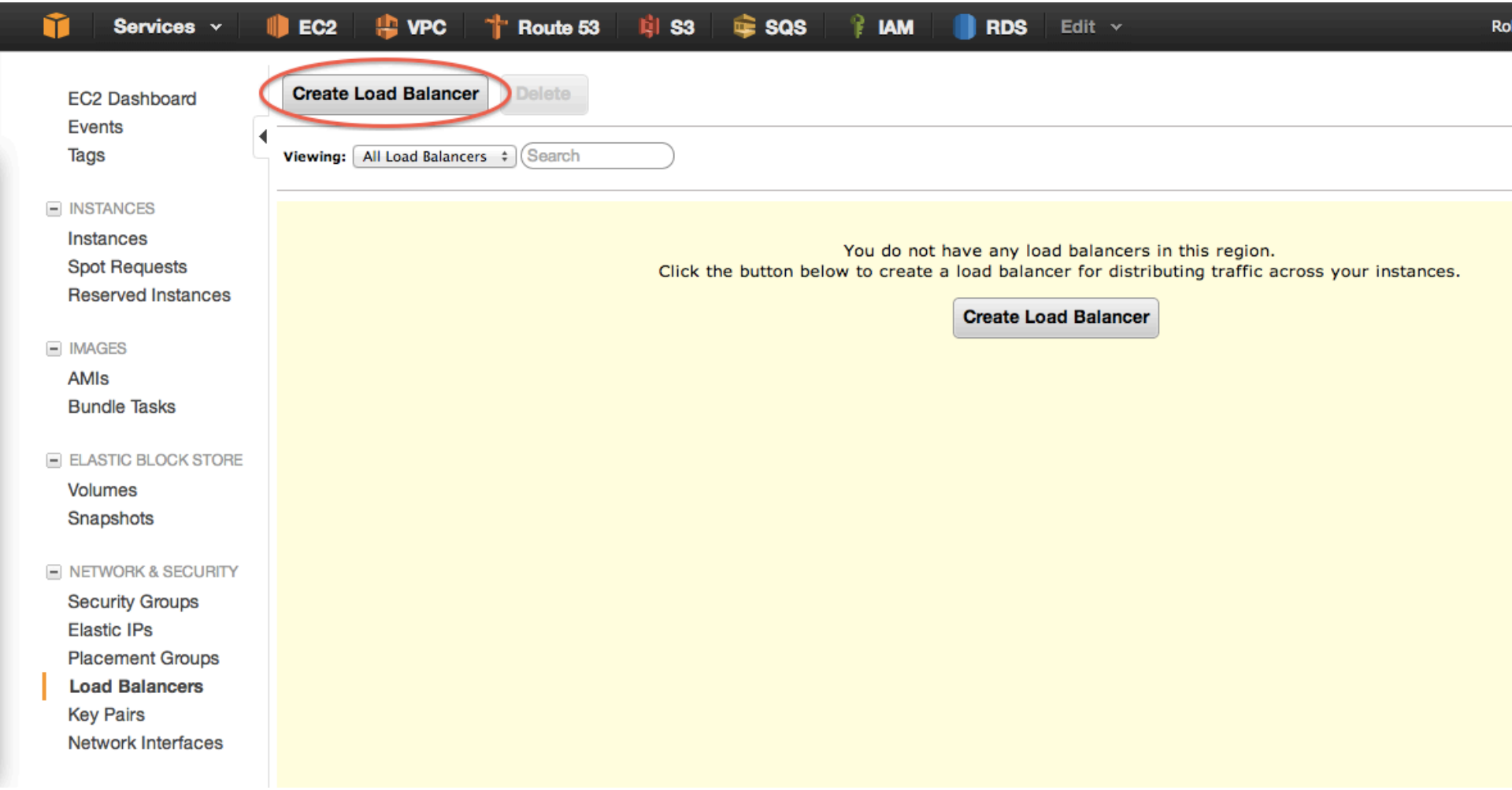
# ELB


Services
EC2
VPC
Route 53
S3
SQS
IAM
History
Console Home
EC2
IAM
SQS
S3
Billing
All AWS Services
Compute & Networking
Storage & Content Delivery
Database
Analytics
Deployment & Management
App Services
CloudFormation
CloudFront
CloudSearch
CloudTrail
CloudWatch
Data Pipeline
Direct Connect
DynamoDB
EC2

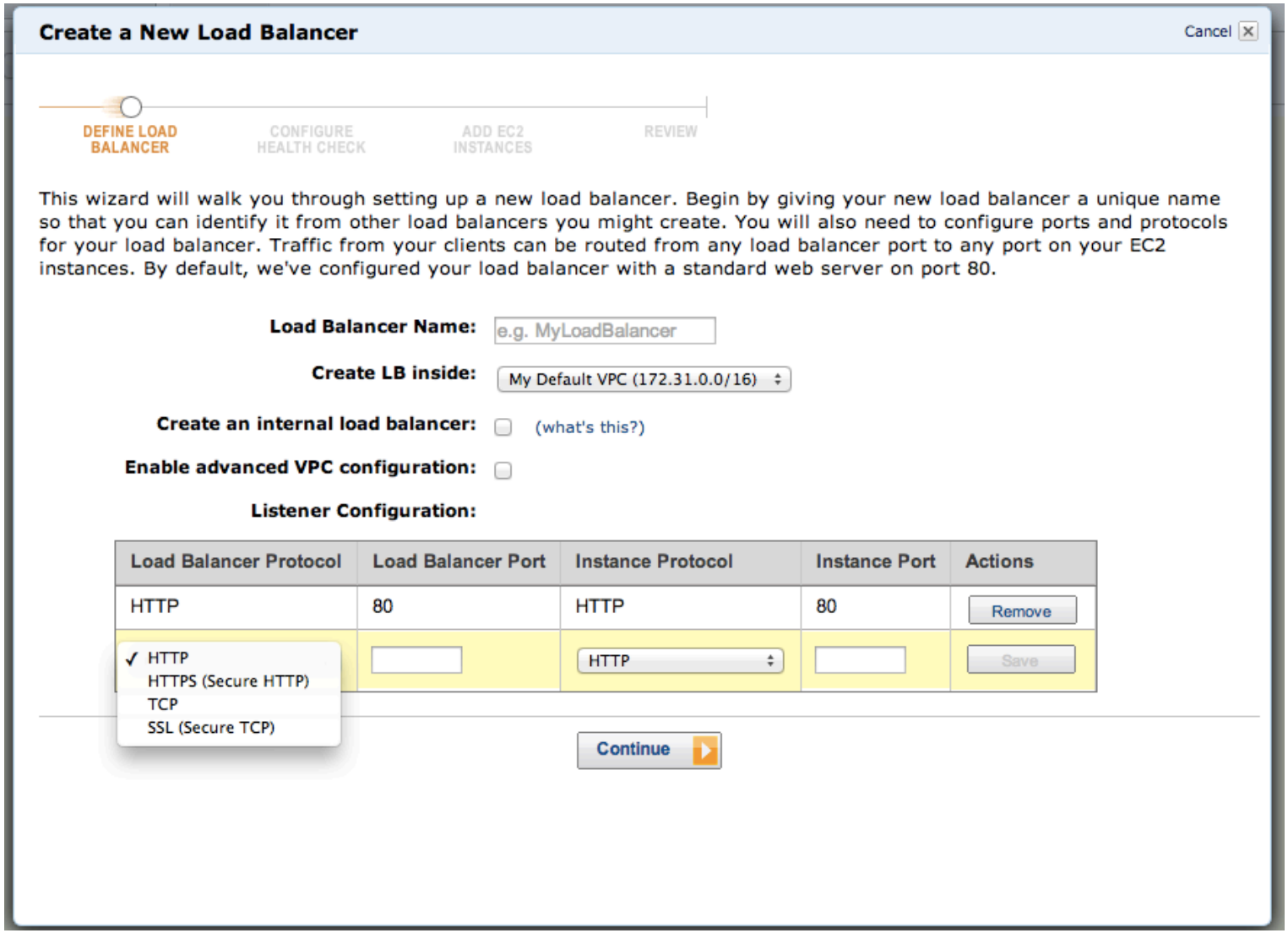
# ELB


Services
EC2
VPC
Route 53
S3
EC2 Dashboard
Events
Tags
INSTANCES
Instances
Spot Requests
Reserved Instances
IMAGES
AMIs
Bundle Tasks
ELASTIC BLOCK STORE
Volumes
Snapshots
NETWORK & SECURITY
Security Groups
Elastic IPs
Placement Groups
Load Balancers
Key Pairs
Network Interfaces
Resources
You are using the following Amazon EC2 resources
0 Running Instances
15 Volumes
17 Key Pairs
0 Placement Groups
Optimize your resources' cost, performance
Create Instance
To start using Amazon EC2 you will want to launch
Launch Instance
Note: Your instances will launch in the US West (Oregon) regi
Service Health
Service Status:
US West (Oregon):
This service is operating normally
Availability Zone Status:
us-west-2a:
Availability zone is operating normally
us-west-2b:

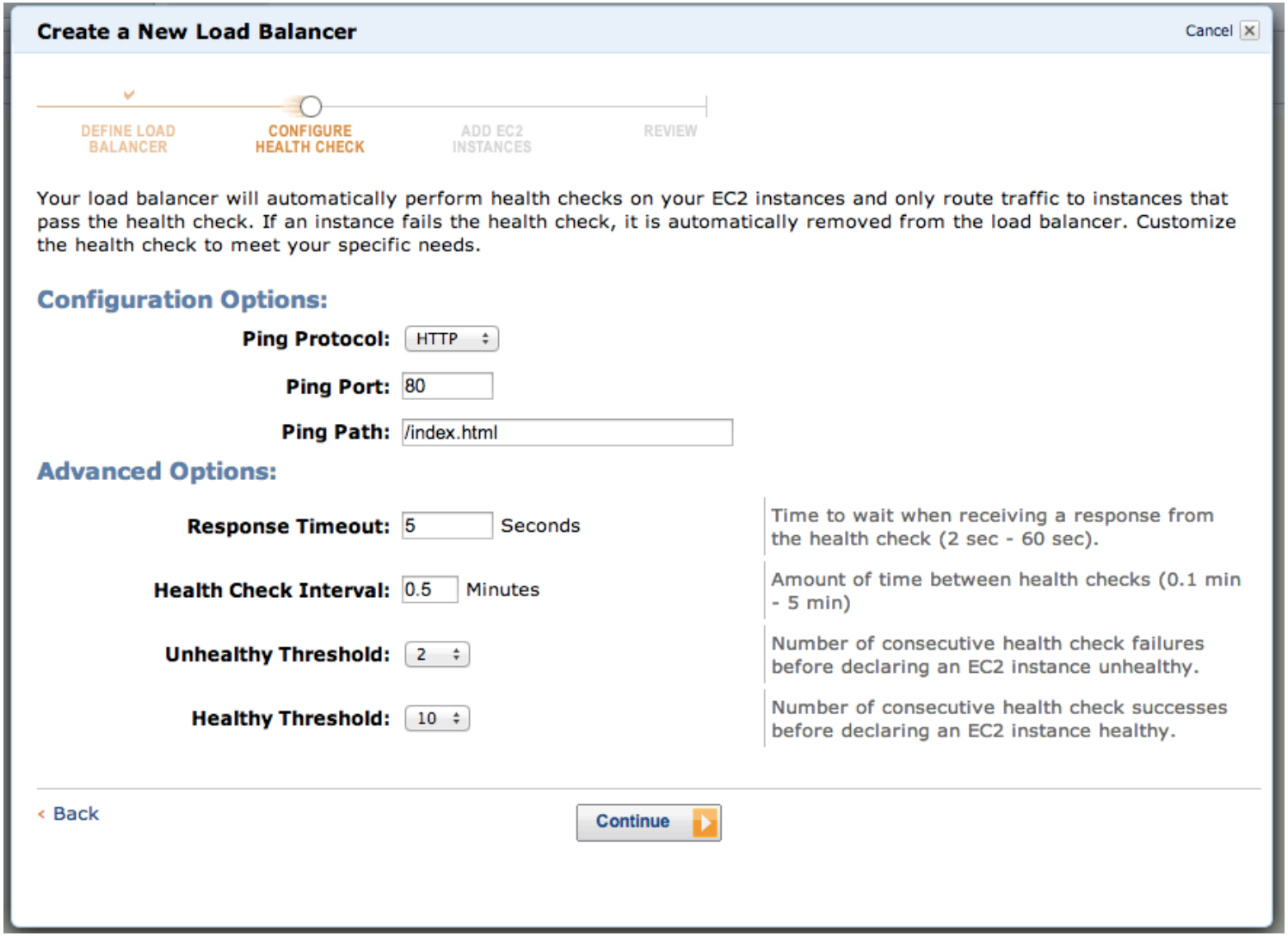
# ELB


Services
EC2
VPC
Route 53
S3
SQS
IAM
RDS
Edit
Create Load Balancer
Delete
EC2 Dashboard
Events
Tags
INSTANCES
Instances
Spot Requests
Reserved Instances
IMAGES
AMIs
Bundle Tasks
ELASTIC BLOCK STORE
Volumes
Snapshots
NETWORK & SECURITY
Security Groups
Elastic IPs
Placement Groups
Load Balancers
Key Pairs
Network Interfaces
Viewing:
All Load Balancers
Search
You do not have any load balancers in this region.
Click the button below to create a load balancer for distributing traffic across your instances.
Create Load Balancer

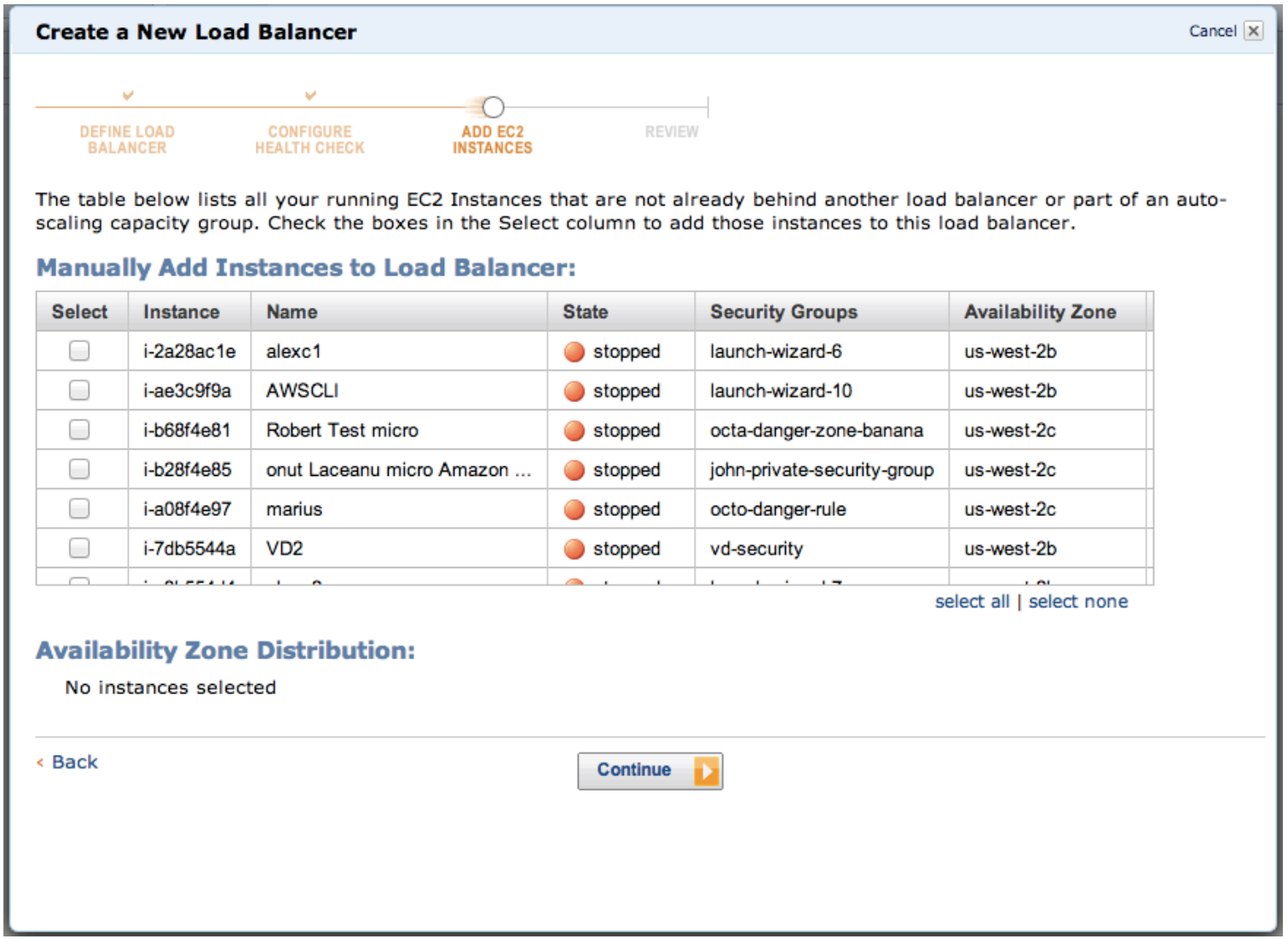
# ELB


Create a New Load Balancer
Cancel
DEFINE LOAD BALANCER
CONFIGURE HEALTH CHECK
ADD EC2 INSTANCES
REVIEW
This wizard will walk you through setting up a new load balancer. Begin by giving your new load balancer a unique name so that you can identify it from other load balancers you might create. You will also need to configure ports and protocols for your load balancer. Traffic from your clients can be routed from any load balancer port to any port on your EC2 instances. By default, we've configured your load balancer with a standard web server on port 80.
Load Balancer Name:
e.g. MyLoadBalancer
Create LB inside:
My Default VPC (172.31.0.0/16)
Create an internal load balancer:
(what's this?)
Enable advanced VPC configuration:
Listener Configuration:
Load Balancer Protocol
Load Balancer Port
Instance Protocol
Instance Port
Actions
HTTP
80
HTTP
80
Remove
HTTP
HTTPS (Secure HTTP)
TCP
SSL (Secure TCP)
HTTP
Save
Continue

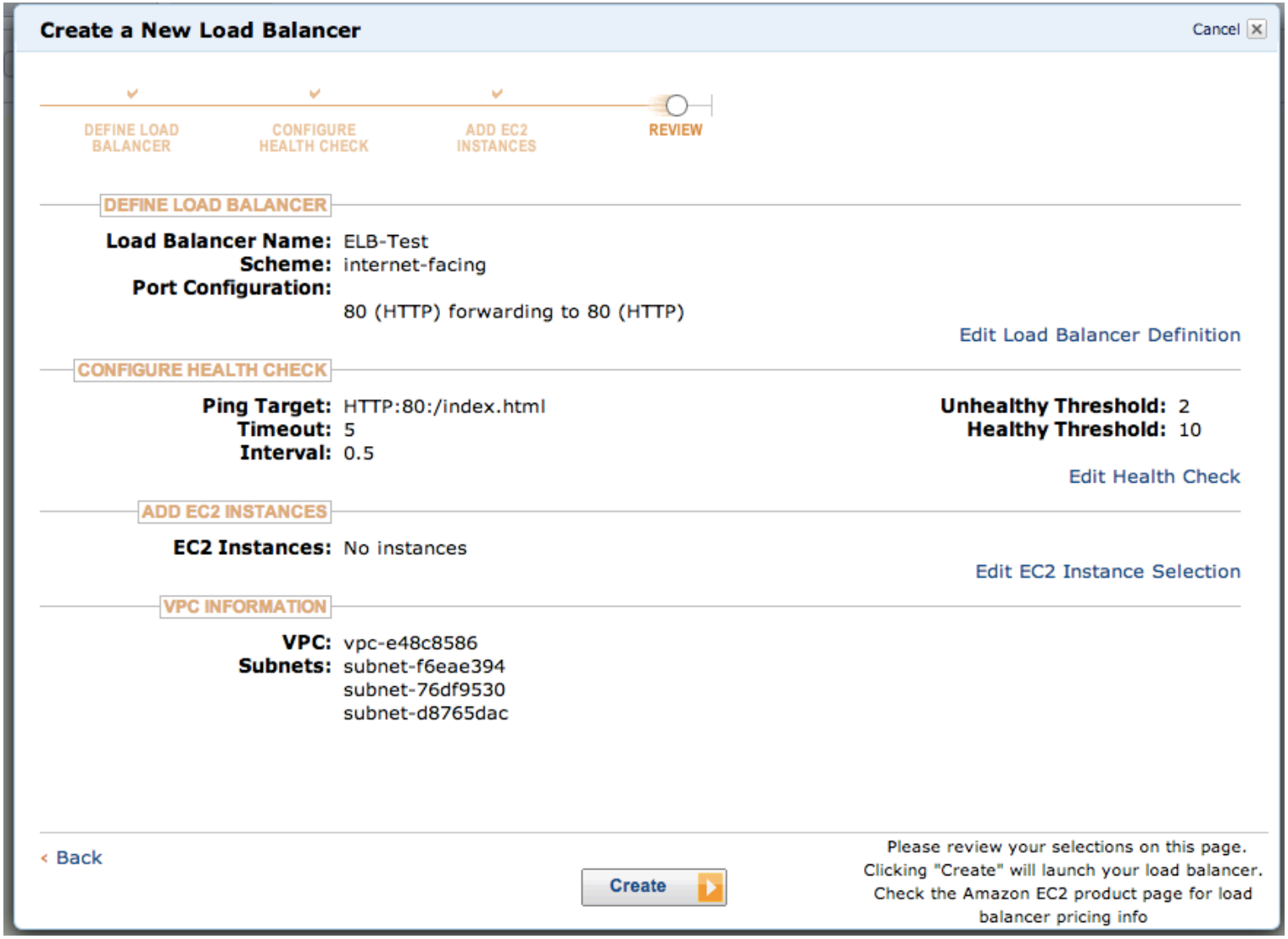
# ELB


Create a New Load Balancer
Cancel
DEFINE LOAD BALANCER
CONFIGURE HEALTH CHECK
ADD EC2 INSTANCES
REVIEW
Your load balancer will automatically perform health checks on your EC2 instances and only route traffic to instances that pass the health check. If an instance fails the health check, it is automatically removed from the load balancer. Customize the health check to meet your specific needs.
Configuration Options:
Ping Protocol: HTTP
Ping Port: 80
Ping Path: /index.html
Advanced Options:
Response Timeout: 5 Seconds
Time to wait when receiving a response from the health check (2 sec - 60 sec).
Health Check Interval: 0.5 Minutes
Amount of time between health checks (0.1 min - 5 min)
Unhealthy Threshold: 2
Number of consecutive health check failures before declaring an EC2 instance unhealthy.
Healthy Threshold: 10
Number of consecutive health check successes before declaring an EC2 instance healthy.
Back
Continue

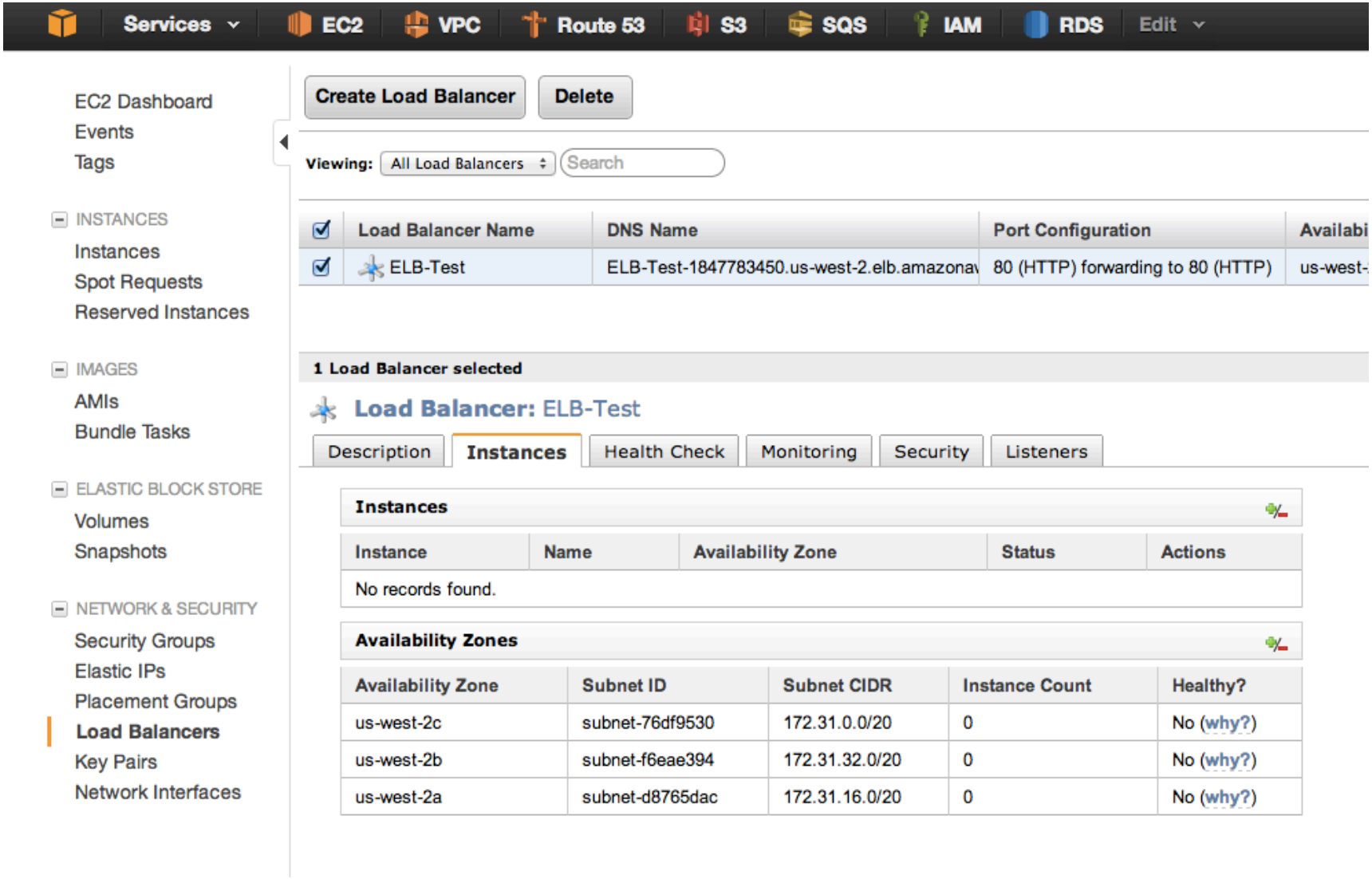
# ELB


Create a New Load Balancer
Cancel
DEFINE LOAD BALANCER
CONFIGURE HEALTH CHECK
ADD EC2 INSTANCES
REVIEW
The table below lists all your running EC2 Instances that are not already behind another load balancer or part of an auto-scaling capacity group. Check the boxes in the Select column to add those instances to this load balancer.
Manually Add Instances to Load Balancer:
Select
Instance
Name
State
Security Groups
Availability Zone
i-2a28ac1e
alexc1
stopped
launch-wizard-6
us-west-2b
i-ae3c9f9a
AWSCLI
stopped
launch-wizard-10
us-west-2b
i-b68f4e81
Robert Test micro
stopped
octa-danger-zone-banana
us-west-2c
i-b28f4e85
onut Laceanu micro Amazon ...
stopped
john-private-security-group
us-west-2c
i-a08f4e97
marius
stopped
octo-danger-rule
us-west-2c
i-7db5544a
VD2
stopped
vd-security
us-west-2b
select all | select none
Availability Zone Distribution:
No instances selected
Back
Continue

# ELB


Create a New Load Balancer
Cancel
DEFINE LOAD BALANCER
CONFIGURE HEALTH CHECK
ADD EC2 INSTANCES
REVIEW
DEFINE LOAD BALANCER
Load Balancer Name: ELB-Test
Scheme: internet-facing
Port Configuration:
80 (HTTP) forwarding to 80 (HTTP)
Edit Load Balancer Definition
CONFIGURE HEALTH CHECK
Ping Target: HTTP:80:/index.html
Timeout: 5
Interval: 0.5
Unhealthy Threshold: 2
Healthy Threshold: 10
Edit Health Check
ADD EC2 INSTANCES
EC2 Instances: No instances
Edit EC2 Instance Selection
VPC INFORMATION
VPC: vpc-e48c8586
Subnets: subnet-f6eae394
subnet-76df9530
subnet-d8765dac
Back
Create
Please review your selections on this page.
Clicking "Create" will launch your load balancer.
Check the Amazon EC2 product page for load balancer pricing info

# ELB

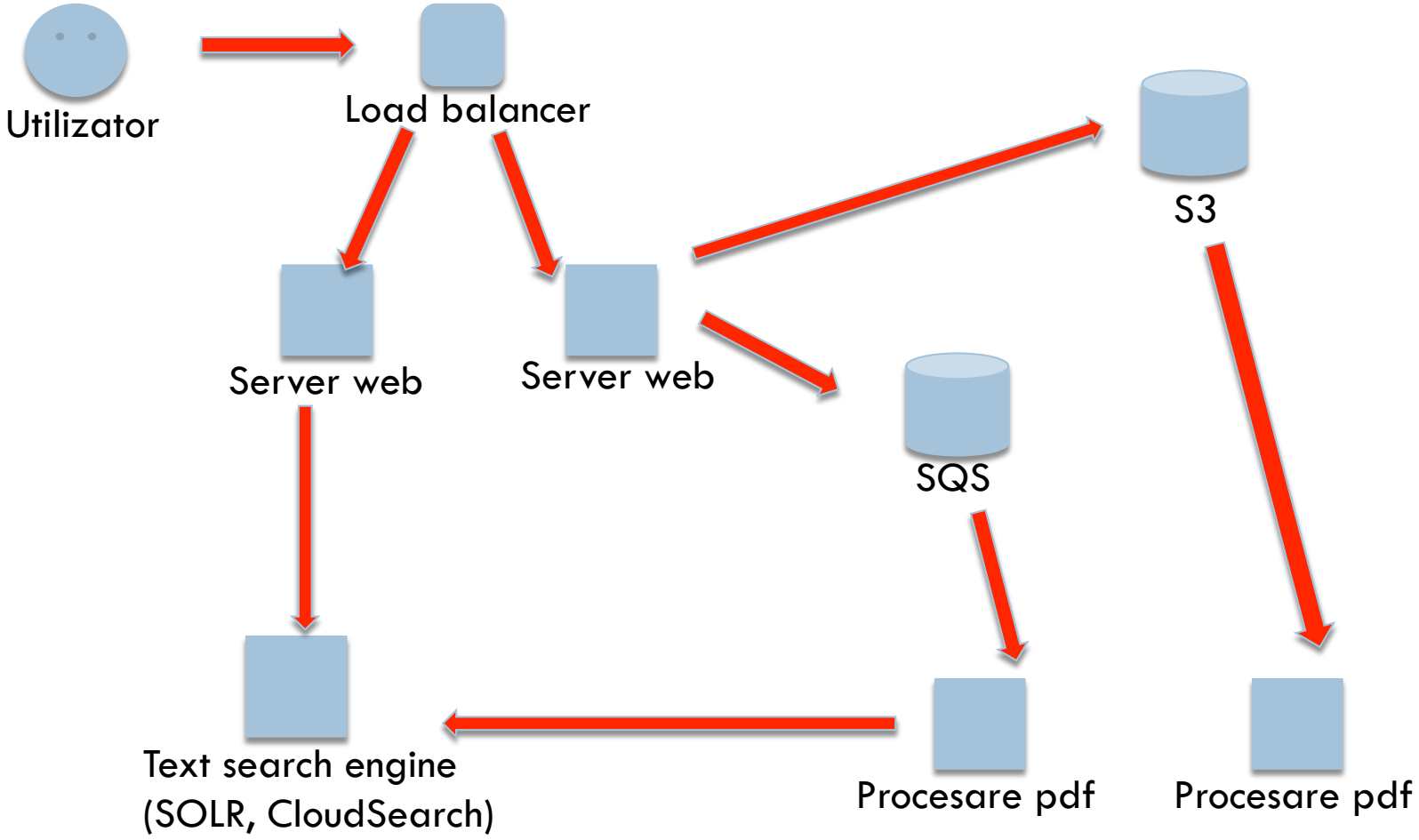

Services
EC2
VPC
Route 53
S3
SQS
IAM
RDS
Edit
EC2 Dashboard
Events
Tags
INSTANCES
Instances
Spot Requests
Reserved Instances
IMAGES
AMIs
Bundle Tasks
ELASTIC BLOCK STORE
Volumes
Snapshots
NETWORK & SECURITY
Security Groups
Elastic IPs
Placement Groups
Load Balancers
Key Pairs
Network Interfaces
Create Load Balancer
Delete
Viewing: All Load Balancers
Search
Load Balancer Name
DNS Name
Port Configuration
ELB-Test
ELB-Test-1847783450.us-west-2.elb.amazonav
80 (HTTP) forwarding to 80 (HTTP)
1 Load Balancer selected
Load Balancer: ELB-Test
Description
Instances
Health Check
Monitoring
Security
Listeners
Instances
Instance
Name
Availability Zone
Status
Actions
No records found.
Availability Zones
Availability Zone
Subnet ID
Subnet CIDR
Instance Count
Healthy?
us-west-2c
subnet-76df9530
172.31.0.0/20
0
No (why?)
us-west-2b
subnet-f6eae394
172.31.32.0/20
0
No (why?)
us-west-2a
subnet-d8765dac
172.31.16.0/20
0
No (why?)

# Exemple de sisteme scalabile

Motor de cautare in documente pdf.

Vrem sa construim un sistem scalabil care este capabil sa indexeze o cantitate mare de documente pdf si sa caute text in ele.

# Motor de cautare in documente pdf

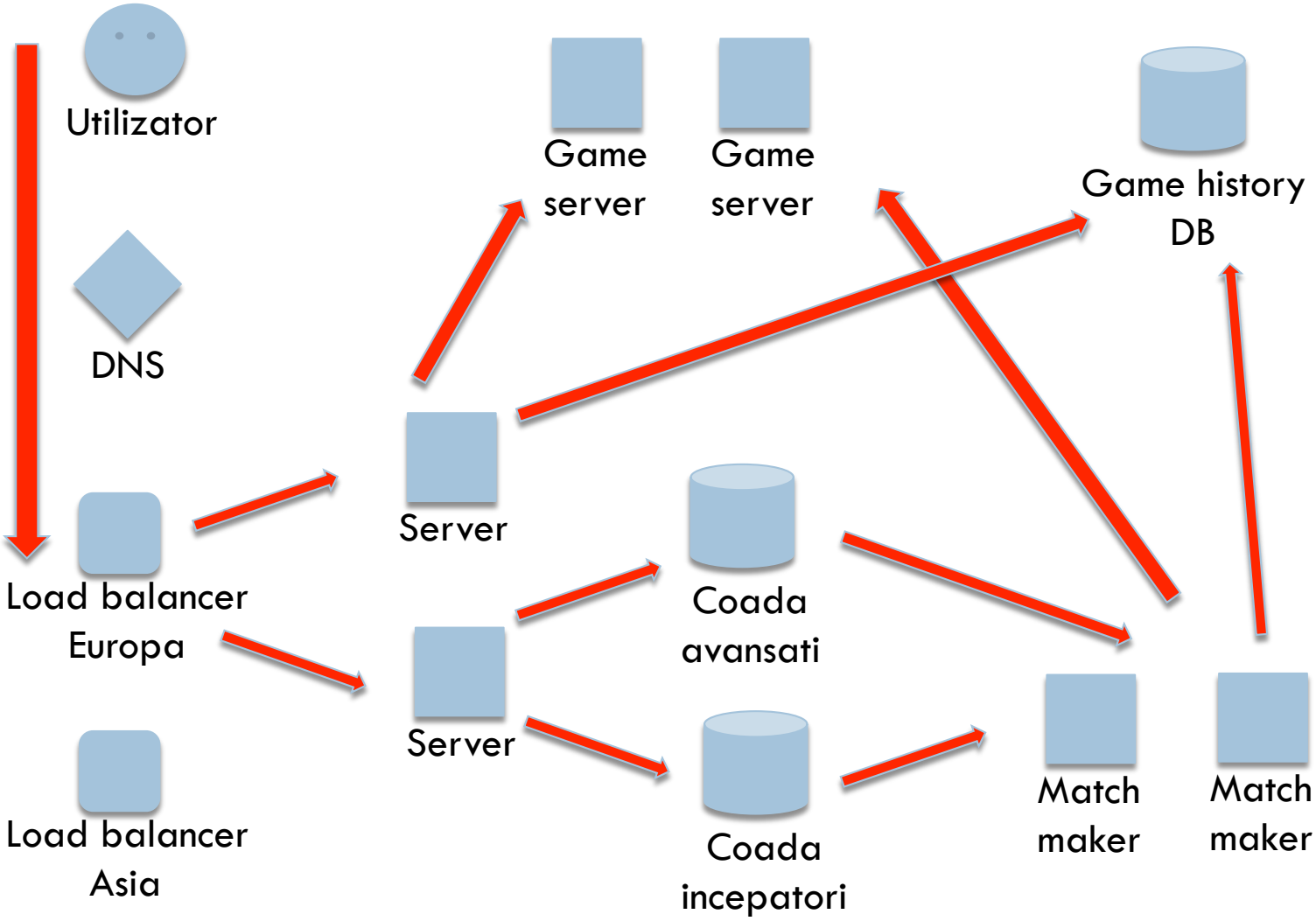


# Exemple de sisteme scalabile

Joc multiplayer online

Vrem sa construim un sistem scalabil care grupeaza jucatorii in functie de nivelul fiecaruia si simuleaza jocul real time.

# Joc multiplayer online



# Exemple de sisteme scalabile

- Un cluster de SOLR Cloud cu un forum de zooKeeper
- Hadoop