RED HAT SUMMIT

USING ANSIBLE TO MANAGE YOUR HYBRID CLOUD

Ryan S. Brown
Senior Software Engineer, Red Hat

Tom Melendez
Senior Software Engineer, Google
5/3/2017

AGENDA

- About Ansible
- GCP integrations and use cases
- Ansible best practices
- Multi-Provider planning and automation
- Demo: cross-cloud service w/ global redundant DNS

ANSIBLE AND CLOUD

WHAT IS ANSIBLE?

- YAML-driven automation tool
- Tower web interface for collaboration, auditing, and API

AUTOMATION NEVER SLEEPS

On any platform

- Automating dull work reduces risk
- Talented IT pros don’t want to repeat the same task over and over
- Handle more projects more safely with automated deployments
- Ansible is a force multiplier for your team

USING ANSIBLE TO MANAGE YOUR CLOUD

- Take full advantage of provider flexibility
- Google Cloud Platform bills VMs by the minute
- New instances can be ready in < 60 seconds
- APIs are there for automation and Ansible makes them accessible

LEGACY

It’s what works today!

- Existing datacenters
- Colo/IaaS deployments

The Dalles, Oregon. Google Data Center.
Photo: Google/Connie Zhou.

HYBRID CLOUD

The combination of your computers and someone else’s computers.

- We’re using “someone else’s computers” for this talk.
  - Generally, IaaS; providers with APIs.
- The principles are the same.

Photos: Google/Connie Zhou.

SENSIBLE HYBRID CLOUD

- Insurance policy for provider-specific downtime, pricing, or regionality
- Be conscious of your data
  - Transfer costs over WAN add up quickly for data-heavy applications
- Splitting a workload is harder than running some workloads in each provider
- Automation can be shared between clouds
- Use playbooks/roles to smooth the differences between providers

Council Bluffs, Iowa. Google Data Center.
Photo: Google/Connie Zhou.

HYBRID != HOMOGENOUS

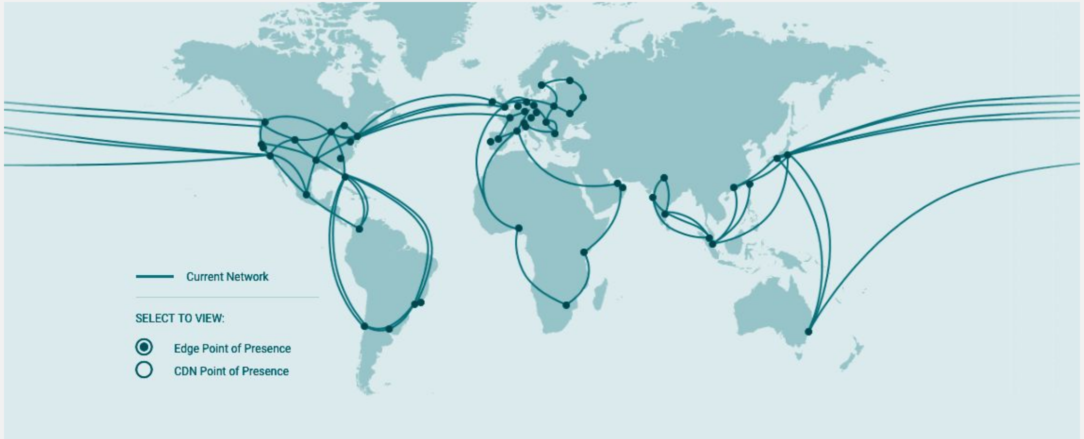
- Find best-of-breed services to fit your needs
- Different apps have different requirements

Douglas County. Georgia. Google Data Center.
Photo: Google/Connie Zhou.

DATA HEAVY APPS

Considerations for “big-ish” data

- Transfer costs
- WAN/leased line speeds
- Site-to-site VPNs
- Estimated daily transfers (GB/day)



Google Edge PoP.
Source: https://cloud.google.com/about/locations/#network-tab
Date Taken: 5/3/2017

HYBRID PRINCIPLES

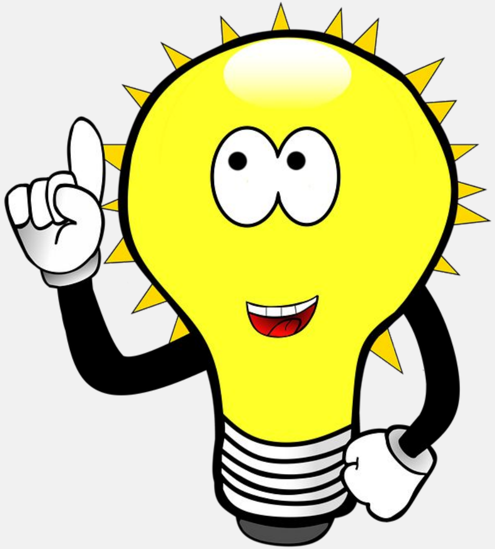


Image: Christian Dorn

- Be aware of provider-specific choices
- User common platforms like OpenShift and Kubernetes over provider APIs
- Provider APIs built into the application are a tradeoff - velocity vs. portability
- Incorporating Hybrid into your dev process early can have a huge ROI

ANSIBLE AND GOOGLE CLOUD PLATFORM

WHY GOOGLE CLOUD?

Author: Plutoforpres
Source: https://commons.wikimedia.org/wiki/File:Cat_lawyer.jpg

- I’m biased, so I can’t really say.
- Visit our booth to find out more.
- Or, just “Google” it. :-)

QUICK FACTS

- Weekly meetings with Ansible Engineering the last 12 months
- AnsibleFests: We’ve attended almost all of them and have sponsored a few, too.
- Ansible usage on GCP is Significant and Growing
- Google engineers work on Ansible and other Open Source projects (full-time)
    - Feature development
    - Bug fixes
    - User issues
- Actively reviewing and accepting PRs for GCP functionality in OSS

GCP ANSIBLE MODULES

GCE

Scalable virtual machines running in Google's innovative data centers.

Networking

More than 100 global network points of presence close to your users.

Spanner

Scalable, globally distributed relational database service that speaks SQL.

Storage

Unified object storage from live data serving to data analytics/ML to data archiving.

DNS

Reliable, authoritative name lookups using our global network of anycast name servers.

PubSub

A global service for real-time and reliable messaging and streaming data

GCP ANSIBLE PLAYBOOK YAML

```
# Compute
gce:
    instance_names: my-test-instance
    zone: us-central1-a
    machine_type: n1-standard-1
    state: present
    metadata: '{"db":"postgres", "group":"qa"}’
    tags: '[http-server, my-other-tag]'
    disks:
      - name: disk-2
        mode: READ_WRITE
      - name: disk-3
        mode: READ_ONLY
    disk_auto_delete: false
    network: foobar-network
    subnetwork: foobar-subnetwork-1
    preemptible: true
    ip_forward: true
```

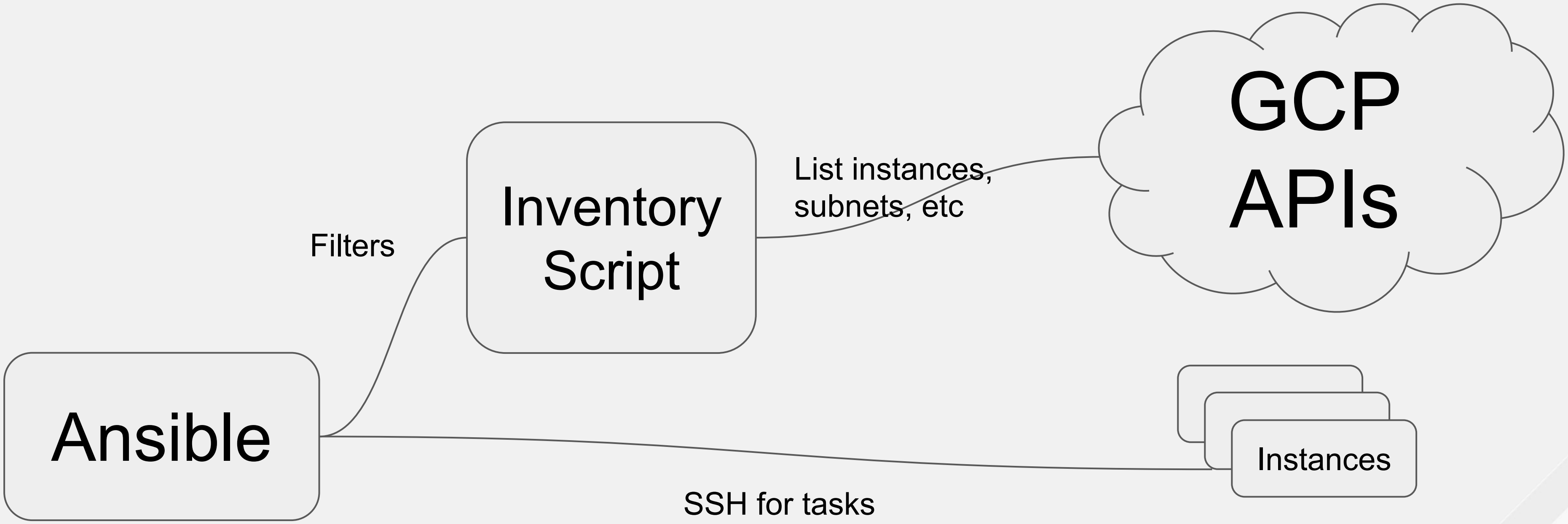
```
# Networks
gce_net:
    name: privatenet
    mode: custom
    subnet_name: subnet_example
    subnet_region: us-central1
    ipv4_range: 10.0.0.0/16
```

```
# Disks
gce_pd:
    disk_type: pd-standard
    snapshot: myinstance1-snap
    name: ansible-disk-from-snap
    state: present
    zone: us-central1-b
```

GCP DYNAMIC INVENTORY

- Grouping by zone, networks, tags and more
- Caching support
- Configurable via `gce.ini` config file
- Keeps up with host churn from automated scaling

GCP DYNAMIC INVENTORY


GCP APIs

Inventory Script

List instances, subnets, etc

Filters

Ansible

Instances

SSH for tasks

ANSIBLE BEST PRACTICES

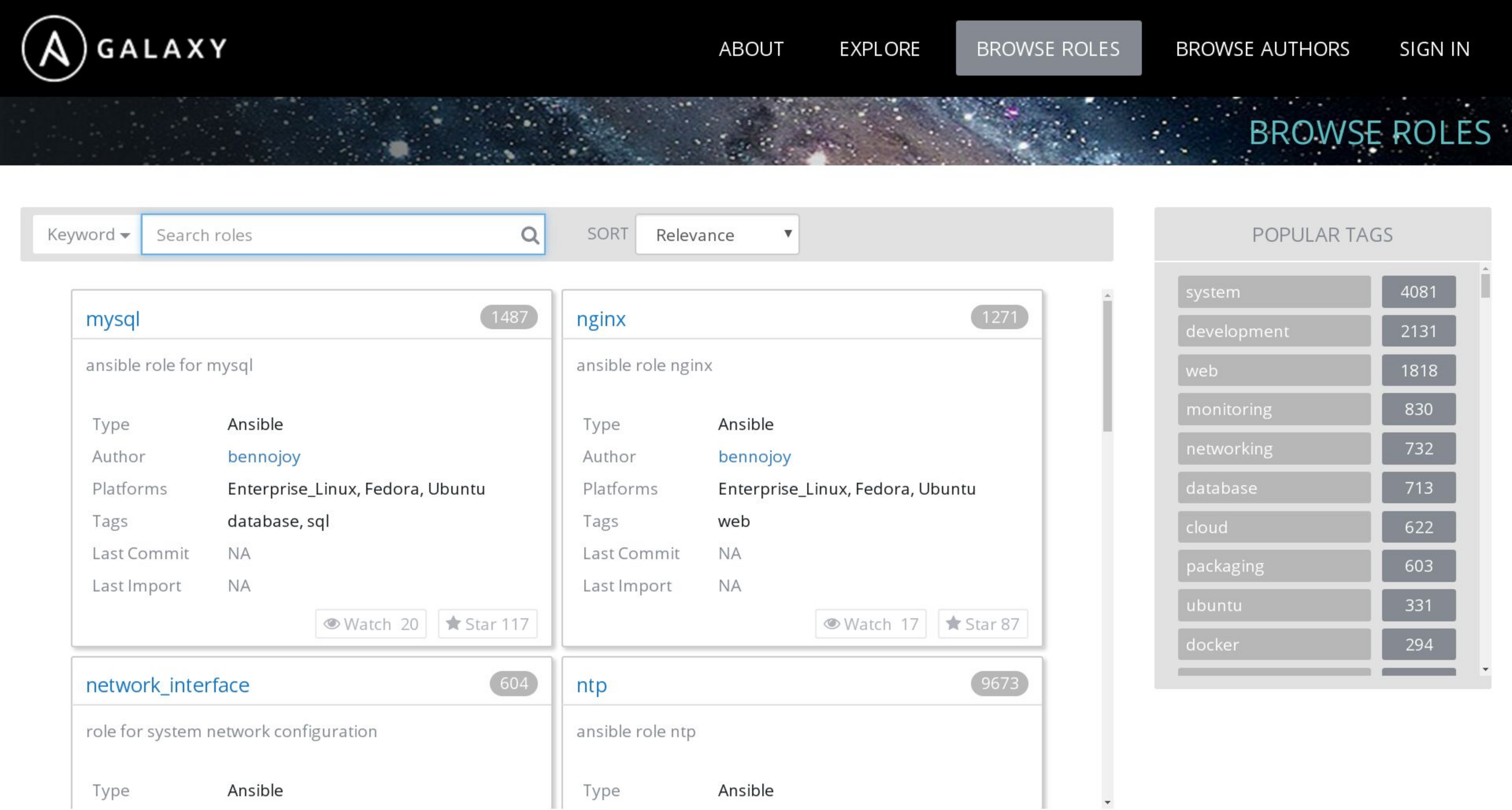
PRACTICES

- Roles & directory structures
- Variables and tagging
- Idempotency (the right way)
- Using cloud APIs
- Dynamic inventories

KEY BENEFITS

Ansible Roles

- Simple enough to be shared across teams
- Document procedures in a readable **and** executable format
- Support any combination of cloud/colo/on-prem systems
- Extensible via
  - Galaxy Community
  - Custom modules
  - Custom roles
  - Your own Galaxy



mysql 1487

ansible role for mysql

Type	Ansible
Author	bennojoy
Platforms	Enterprise_Linux, Fedora, Ubuntu
Tags	database, sql
Last Commit	NA
Last Import	NA

Watch 20 Star 117

nginx 1271

ansible role nginx

Type	Ansible
Author	bennojoy
Platforms	Enterprise_Linux, Fedora, Ubuntu
Tags	web
Last Commit	NA
Last Import	NA

Watch 17 Star 87

network_interface 604

role for system network configuration

Type	Ansible

ntp 9673

ansible role ntp

Type	Ansible

POPULAR TAGS

system	4081
development	2131
web	1818
monitoring	830
networking	732
database	713
cloud	622
packaging	603
ubuntu	331
docker	294

ROLE STRUCTURE

```
myco.netsec/
  tasks/
    main.yml
    firewall.yml
    ipv6.yml
  defaults/
    main.yml
  meta/
    main.yml
    container.yml
```

DIRECTORY STRUCTURE

```
mysite-automation/
  vars/
    ...yml
  playbooks/
    ci_deploy_webapp.yml
    roll_dep_updates.yml
  roles/
    myco.netsec/
      tasks/
        ...
```

SHARE PROD AND STAGE PLAYS

Conditional love

```
- name: Set up a production-only service
  some_module:
    arg1: abc
  when: environment == “production”
```

SPLIT PROVIDER ACTIONS

```
# Create separate tasts for provision_gcp.yml and provision_aws.yml
- include: provision_{{ provider }}.yml
```

IDEMPOTENCY THE RIGHT WAY

- Modules aren’t always consistent
  - shell
  - command
- Check status of these resources **before** changing state
- Use **`changed_when`** to avoid extra “changed” counts when running plays
- Tower keeps track of changed/failed/ok tasks for every job

PLANNING WHAT TO AUTOMATE

LOW HANGING FRUIT

Incrementally automating your job

- No need for huge migration project
- Find daily tasks, start there

AUTOMATING HYBRID ENVIRONMENTS

- Double the credentials to manage
- Start with one provider if you’re just learning
- More diverse environments mean more conditionals, roles, and special cases
- Find tasks common to both

HOW MANY (RENTED) DATACENTERS

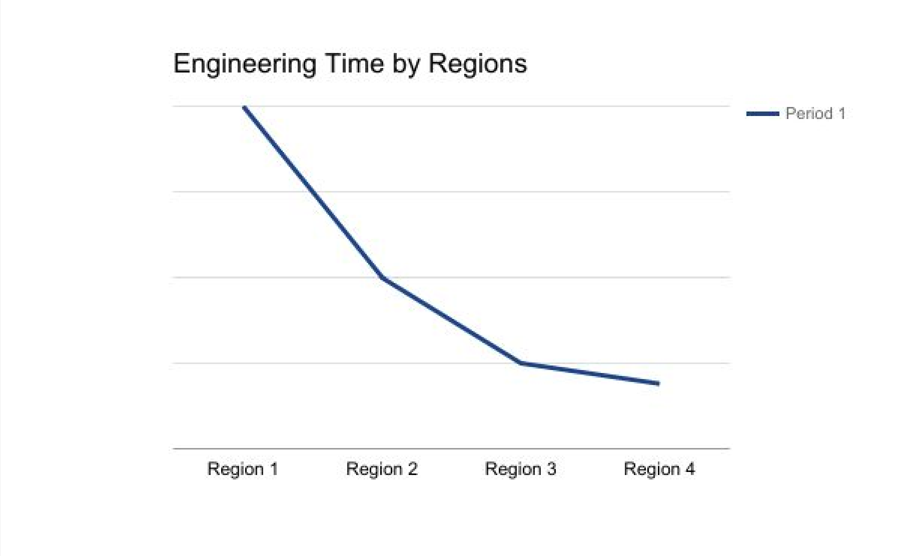
Latency-sensitive users and the speed of light

- Trans-American latency is ~100ms in fiber
- Content Distribution Networks commonly have 300-1,500 PoP’s
- Redundancy in case of disasters

PARTITIONED FAILURE DOMAINS

If we use one computer, only one thing can possibly fail...

- Dollar cost of adding a new region
  - No new real estate
  - No new leased lines
- Uptime requirements, more is **usually** better
- Automation makes adding new regions a sublinear time investment



DEMO TIME

RED HAT SUMMIT

THANK YOU

plus.google.com/+RedHat

facebook.com/redhatinc

linkedin.com/company/red-hat

twitter.com/RedHatNews

youtube.com/user/RedHatVideos

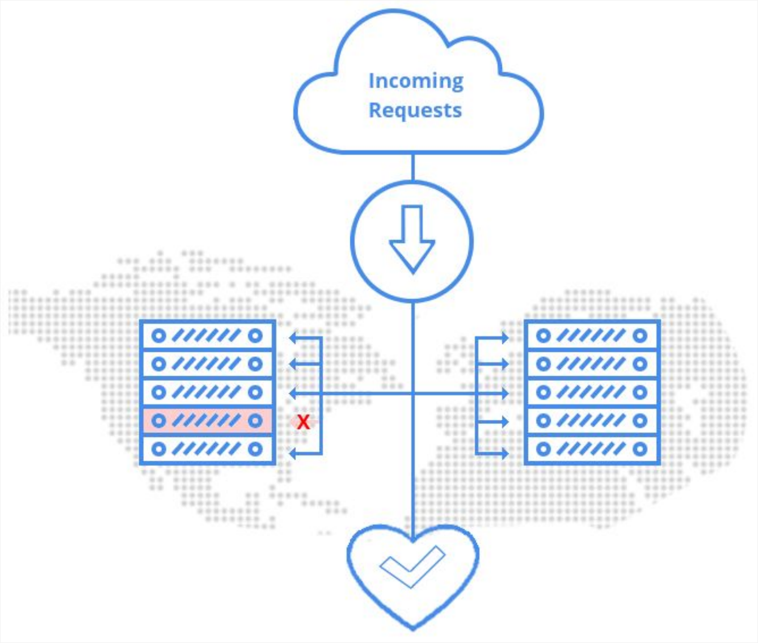
RED HAT
SUMMIT

LEARN. NETWORK.
EXPERIENCE
OPEN SOURCE.

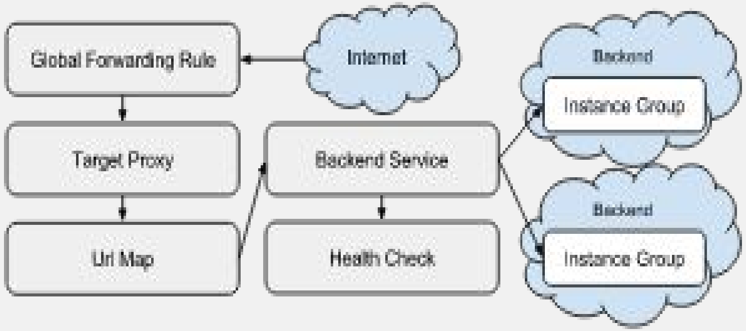
BONUS SLIDES

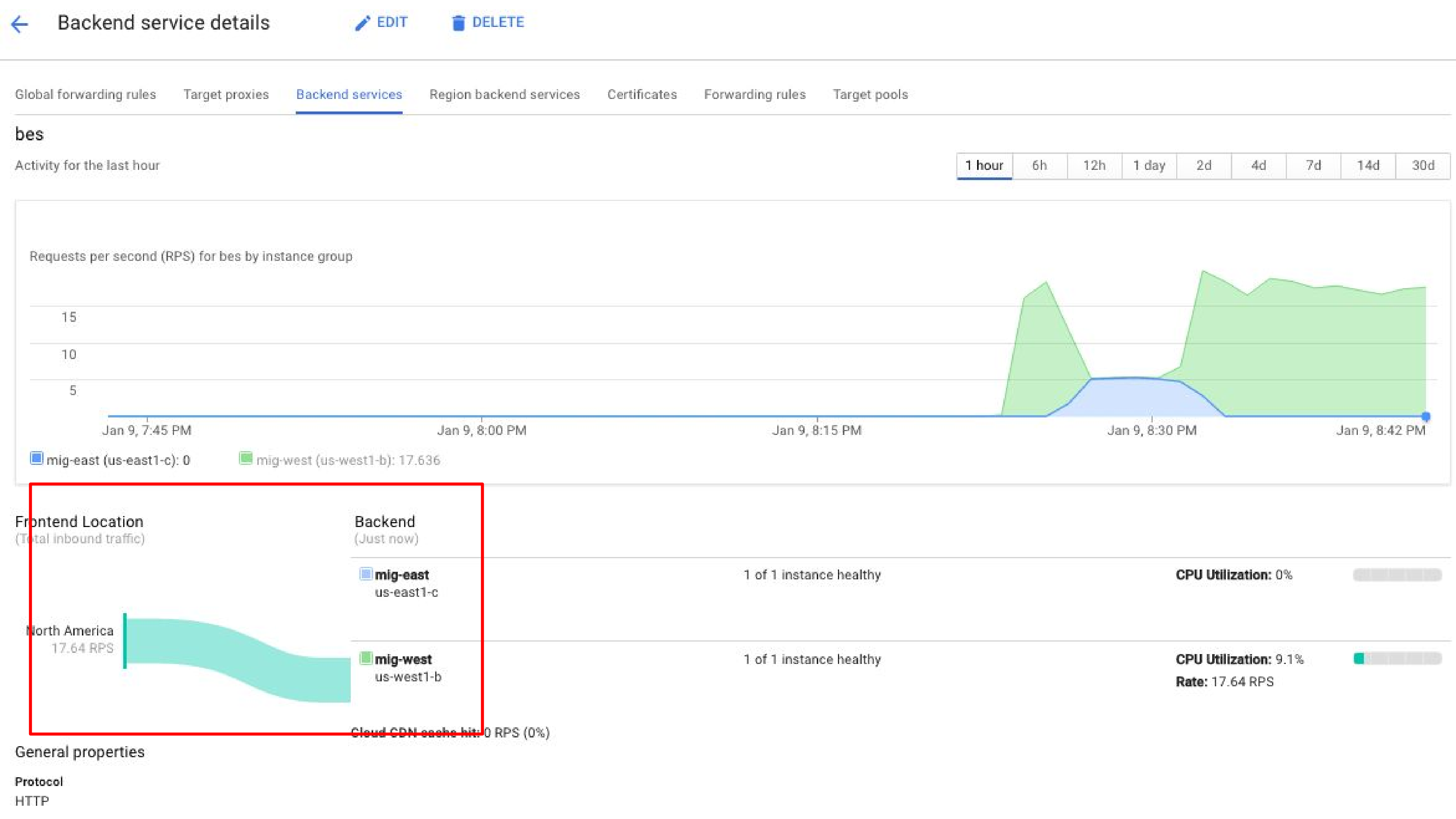
Demo

Request to the Closest Region:



In GCP terms...





bes

Activity for the last hour

Requests per second (RPS) for bes by instance group

🟦 mig-east (us-east1-c): 0
🟩 mig-west (us-west1-b): 17.636

Frontend Location
(Total inbound traffic)

North America
17.64 RPS

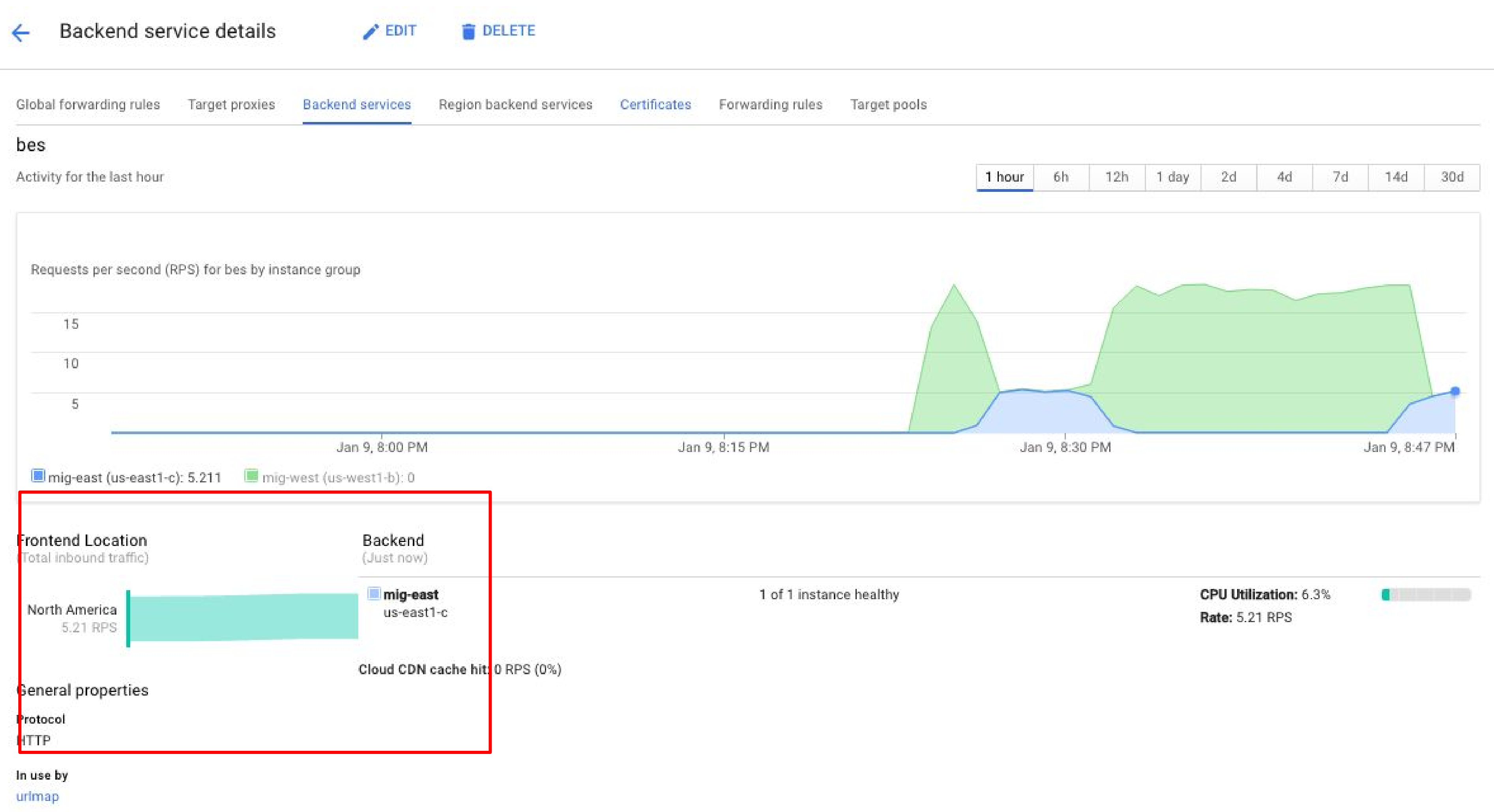
Backend
(Just now)

🟦 **mig-east** us-east1-c	1 of 1 instance healthy	**CPU Utilization:** 0%
🟩 **mig-west** us-west1-b	1 of 1 instance healthy	**CPU Utilization:** 9.1% **Rate:** 17.64 RPS

Cloud CDN cache hit: 0 RPS (0%)

General properties

Protocol
HTTP



bes

Activity for the last hour

🟦 mig-east (us-east1-c): 5.211 🟩 mig-west (us-west1-b): 0

Frontend Location
(Total inbound traffic)

North America
5.21 RPS

Backend
(Just now)

🟦 **mig-east**
us-east1-c

1 of 1 instance healthy

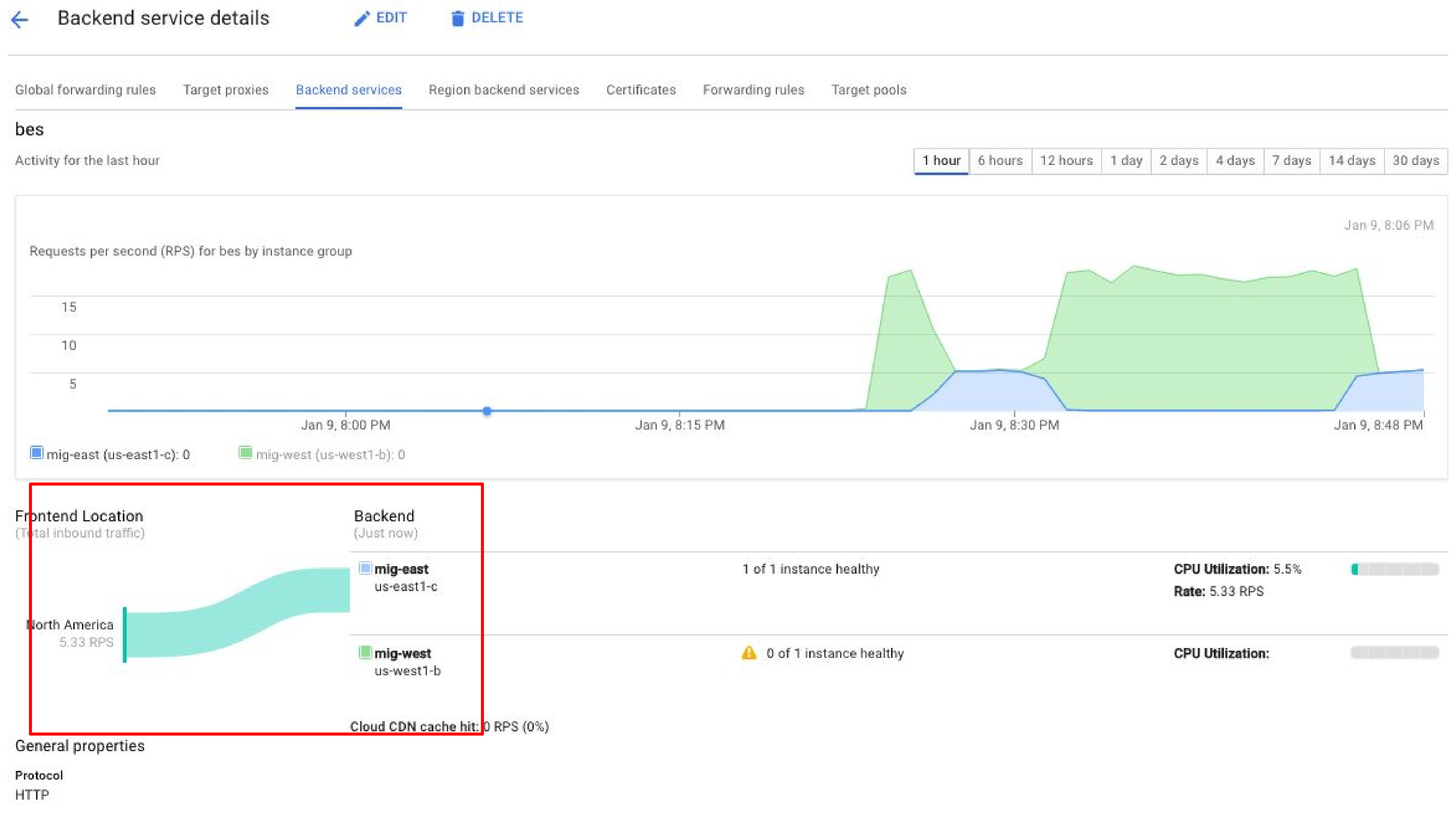
CPU Utilization: 6.3%
Rate: 5.21 RPS

Cloud CDN cache hit: 0 RPS (0%)

General properties

Protocol
HTTP

In use by
urlmap



Backend service details

bes

Activity for the last hour

Jan 9, 8:06 PM

Requests per second (RPS) for bes by instance group

🟦 mig-east (us-east1-c): 0

🟩 mig-west (us-west1-b): 0

Frontend Location
(Total inbound traffic)

North America
5.33 RPS

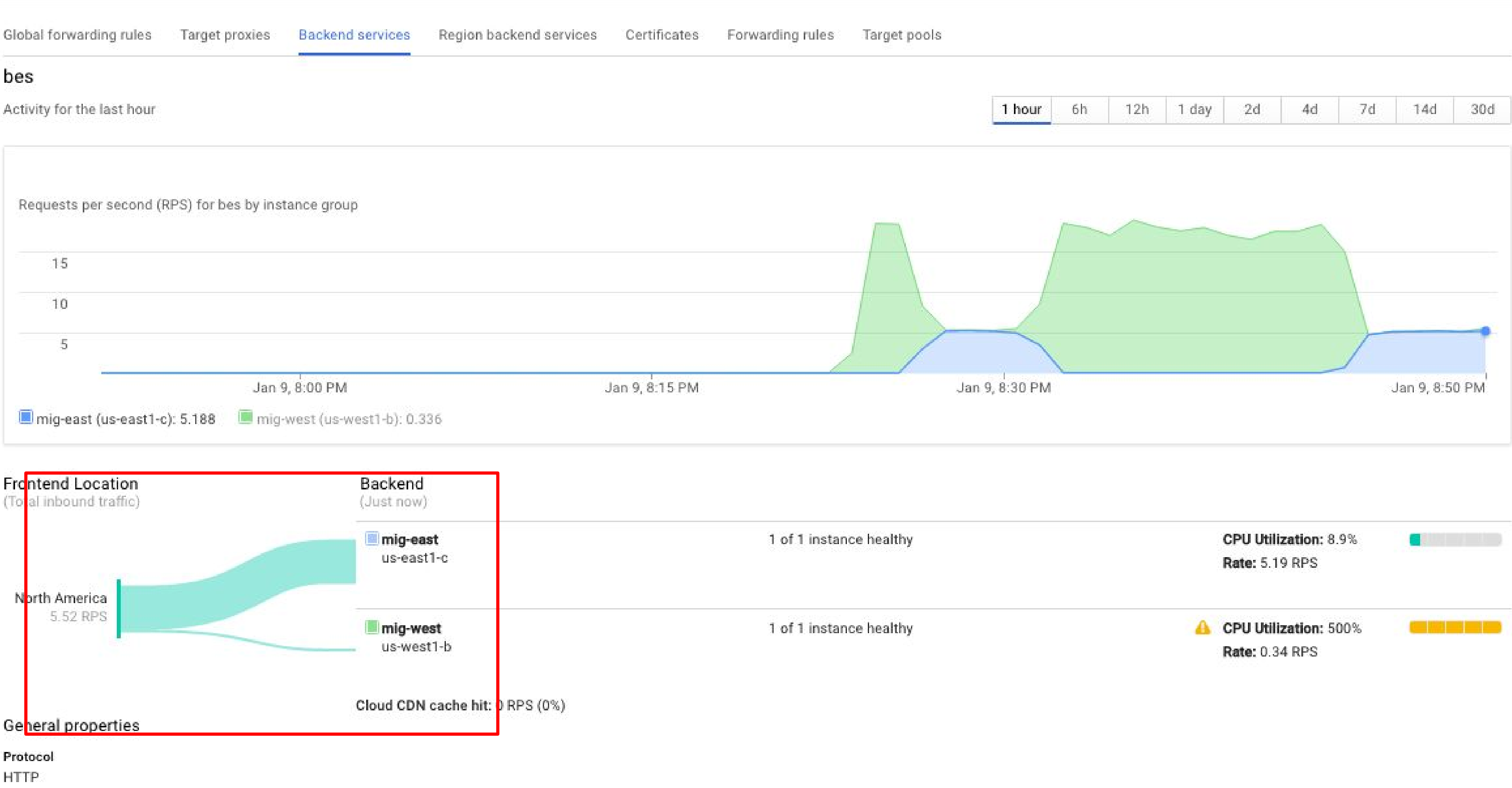
Backend (Just now)		
🟦 **mig-east** us-east1-c	1 of 1 instance healthy	**CPU Utilization:** 5.5% **Rate:** 5.33 RPS
🟩 **mig-west** us-west1-b	⚠ 0 of 1 instance healthy	**CPU Utilization:**

Cloud CDN cache hit: 0 RPS (0%)

General properties

Protocol

HTTP



bes

Activity for the last hour

Requests per second (RPS) for bes by instance group

🟦 mig-east (us-east1-c): 5.188 🟩 mig-west (us-west1-b): 0.336

Frontend Location
(Total inbound traffic)

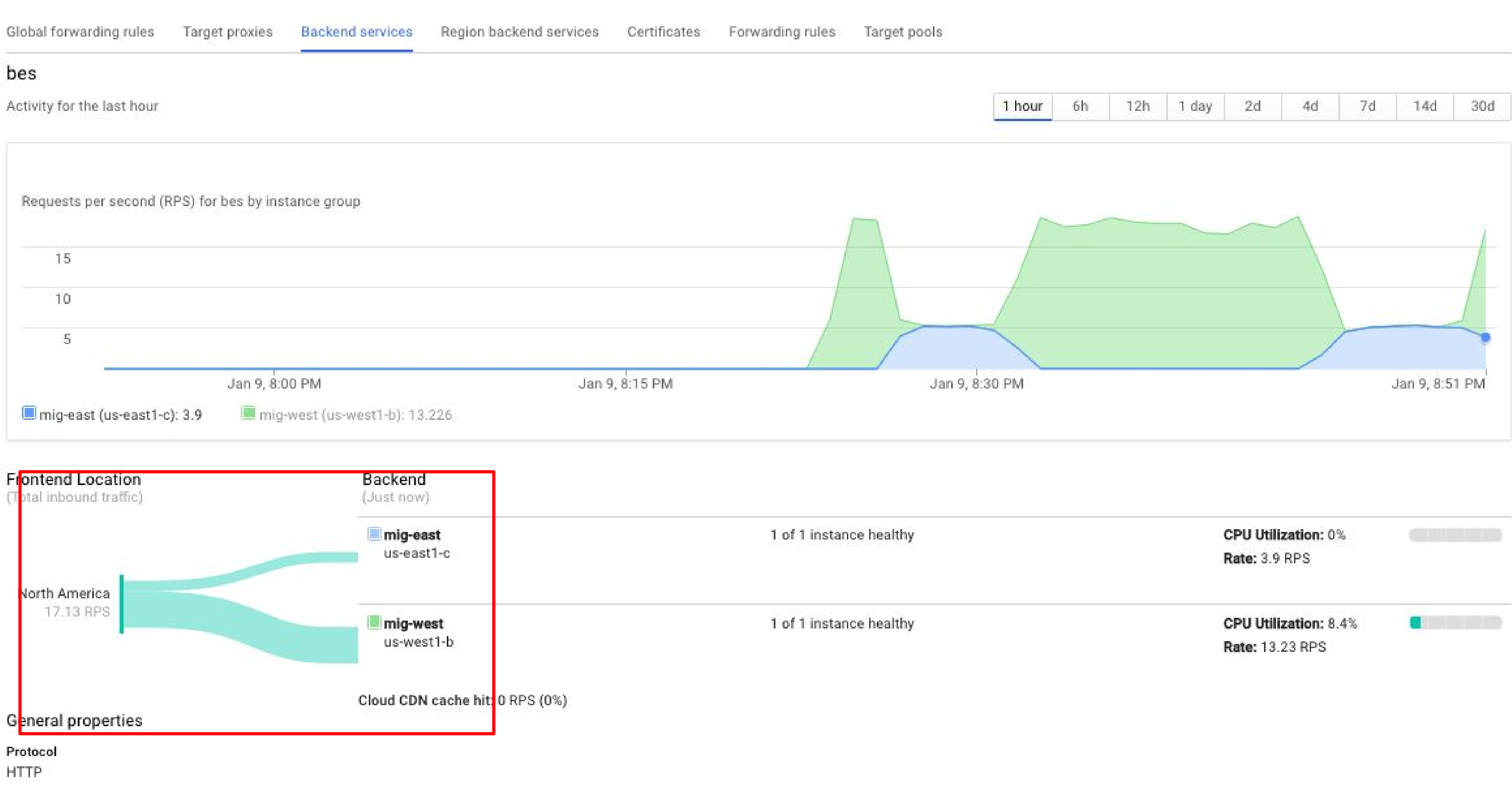
Backend
(Just now)

🟦 **mig-east** us-east1-c	1 of 1 instance healthy	**CPU Utilization:** 8.9% **Rate:** 5.19 RPS
🟩 **mig-west** us-west1-b	1 of 1 instance healthy	⚠ **CPU Utilization:** 500% **Rate:** 0.34 RPS

Cloud CDN cache hit: 0 RPS (0%)

General properties

Protocol
HTTP



bes

Activity for the last hour

Requests per second (RPS) for bes by instance group

🟦 mig-east (us-east1-c): 3.9 🟩 mig-west (us-west1-b): 13.226

Frontend Location
(Total inbound traffic)

Backend
(Just now)

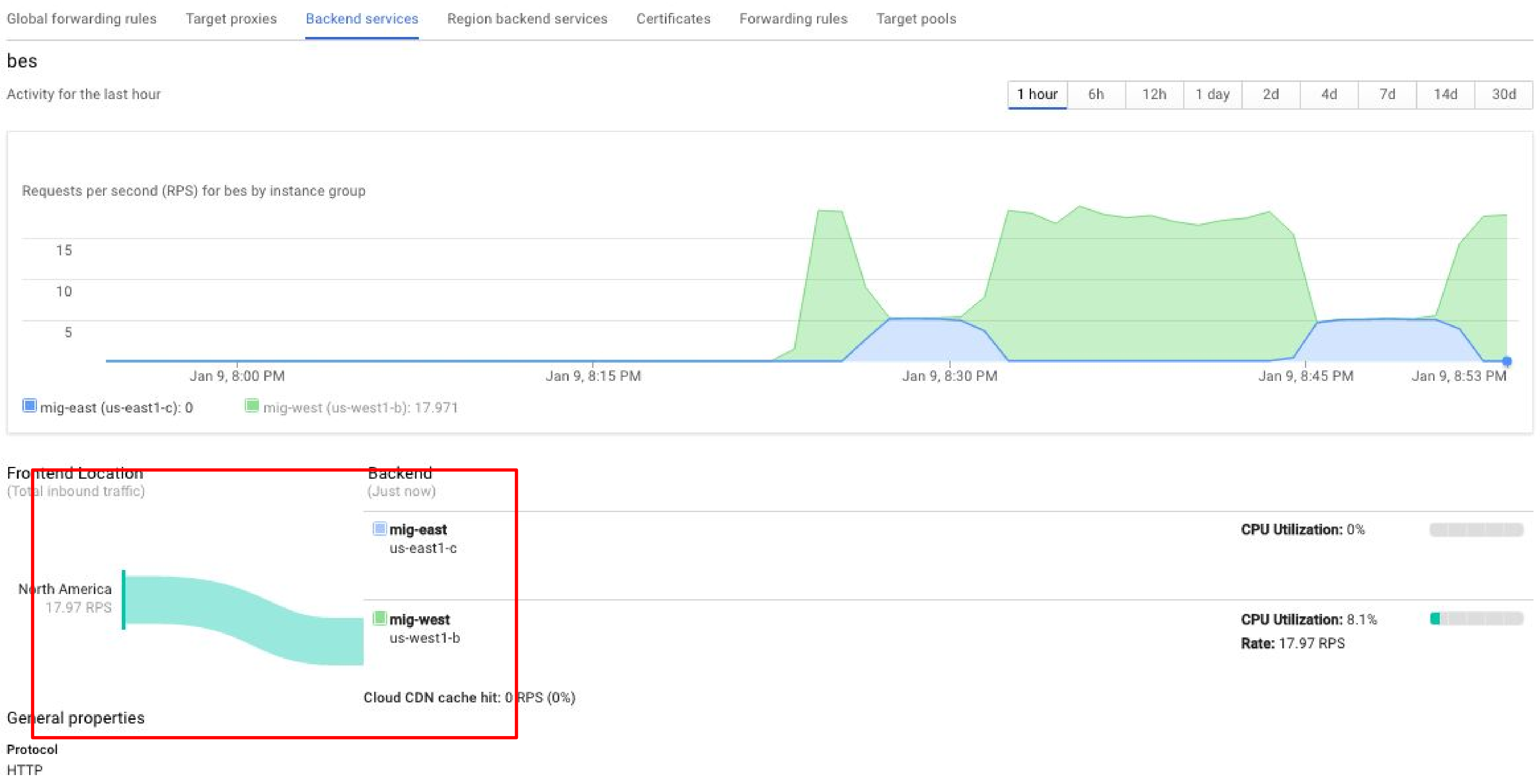
🟦 **mig-east**
us-east1-c
1 of 1 instance healthy
CPU Utilization: 0%
Rate: 3.9 RPS

🟩 **mig-west**
us-west1-b
1 of 1 instance healthy
CPU Utilization: 8.4%
Rate: 13.23 RPS

Cloud CDN cache hit: 0 RPS (0%)

General properties

Protocol
HTTP



bes

Activity for the last hour

Requests per second (RPS) for bes by instance group

🟦 mig-east (us-east1-c): 0 🟩 mig-west (us-west1-b): 17.971

Frontend Location
(Total inbound traffic)

North America
17.97 RPS

Backend
(Just now)

🟦 **mig-east**
us-east1-c

CPU Utilization: 0%

🟩 **mig-west**
us-west1-b

CPU Utilization: 8.1%
Rate: 17.97 RPS

Cloud CDN cache hit: 0 RPS (0%)

General properties

Protocol
HTTP