

POSTGRESQL

в гетерогенной инфраструктуре



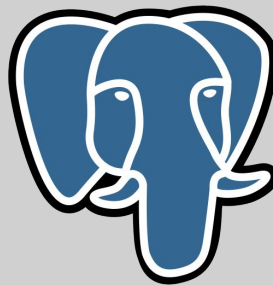
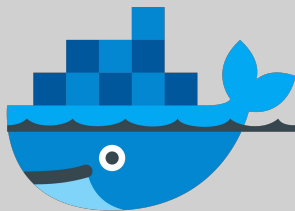
2ГИС

Волков Николай

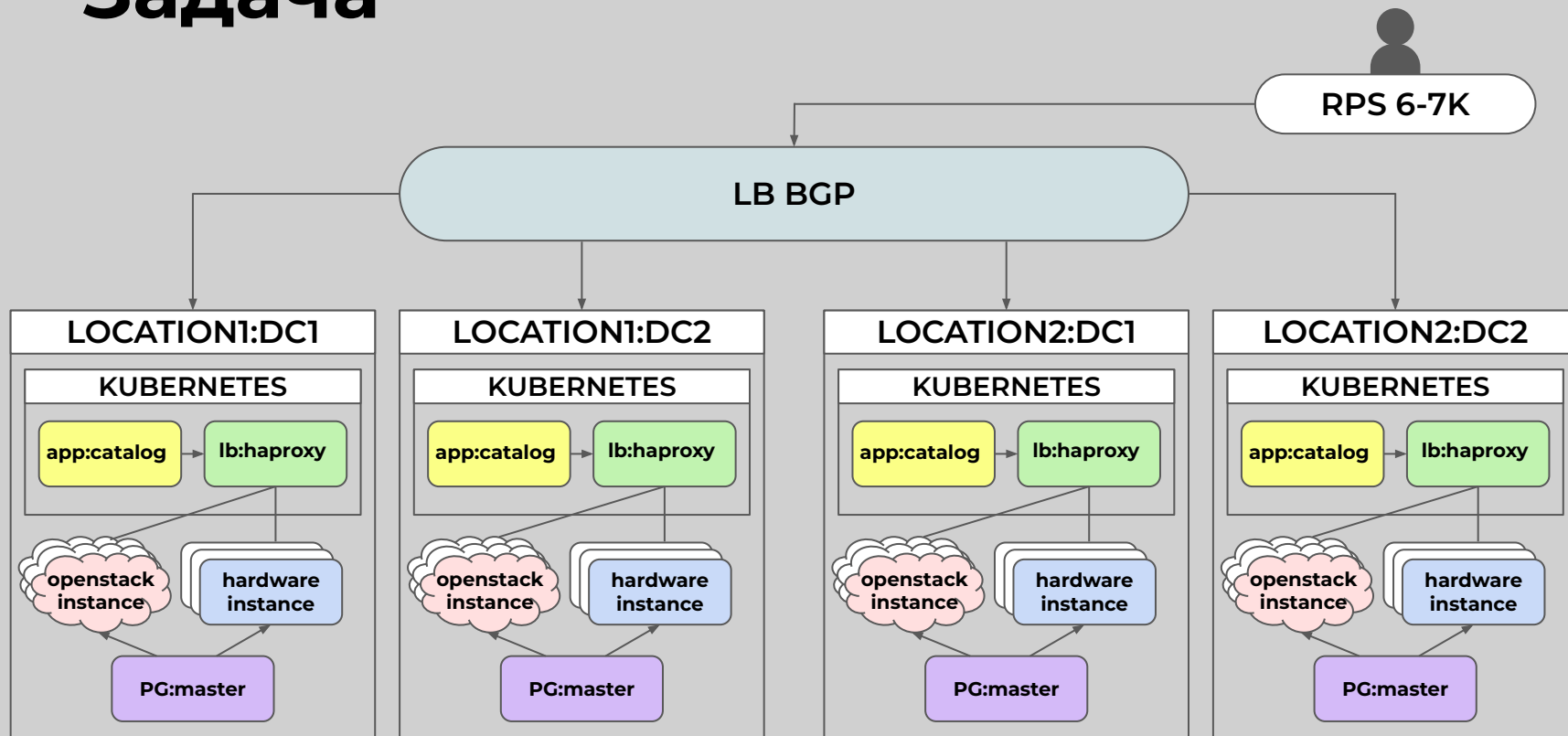


@rmrfthm

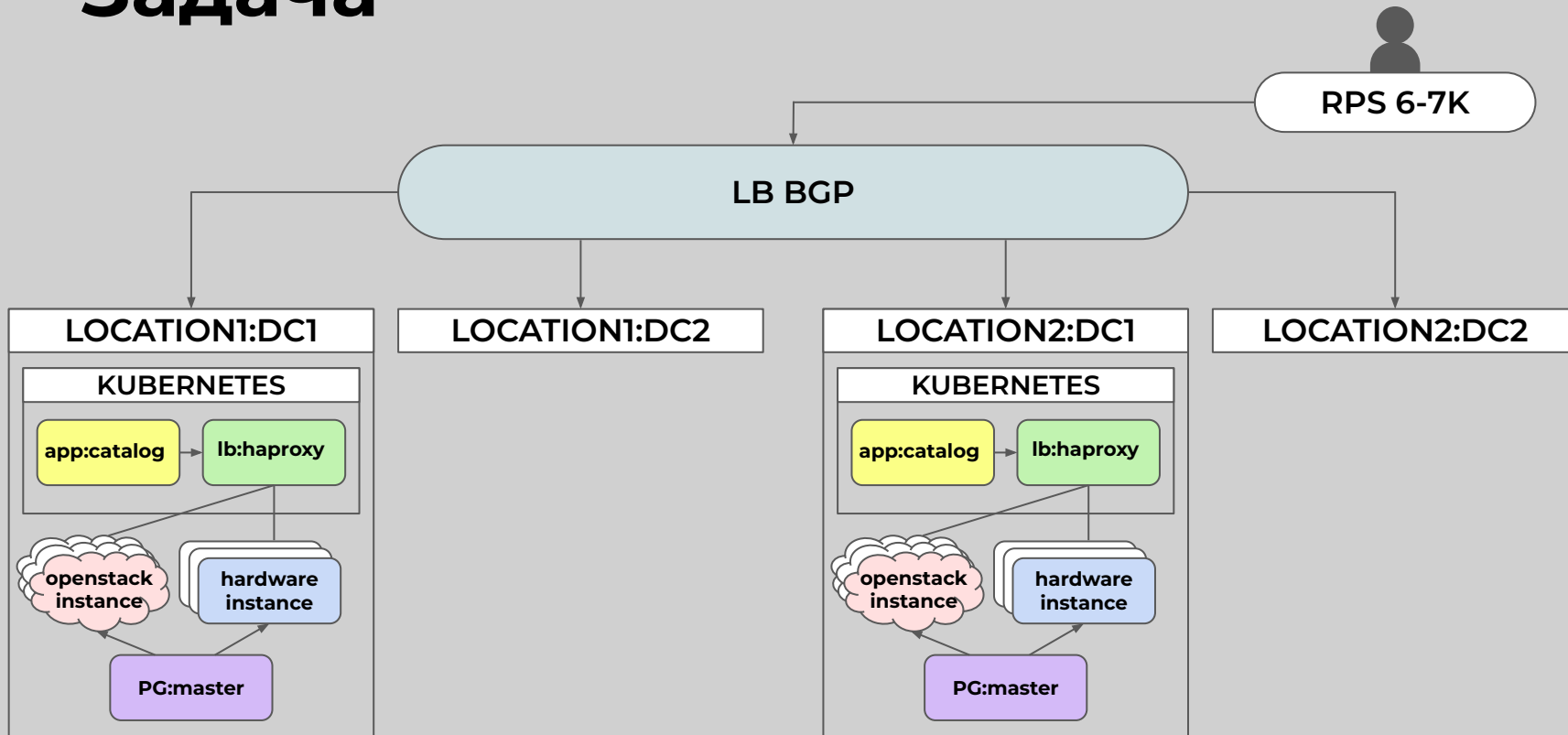
Команда 2GIS



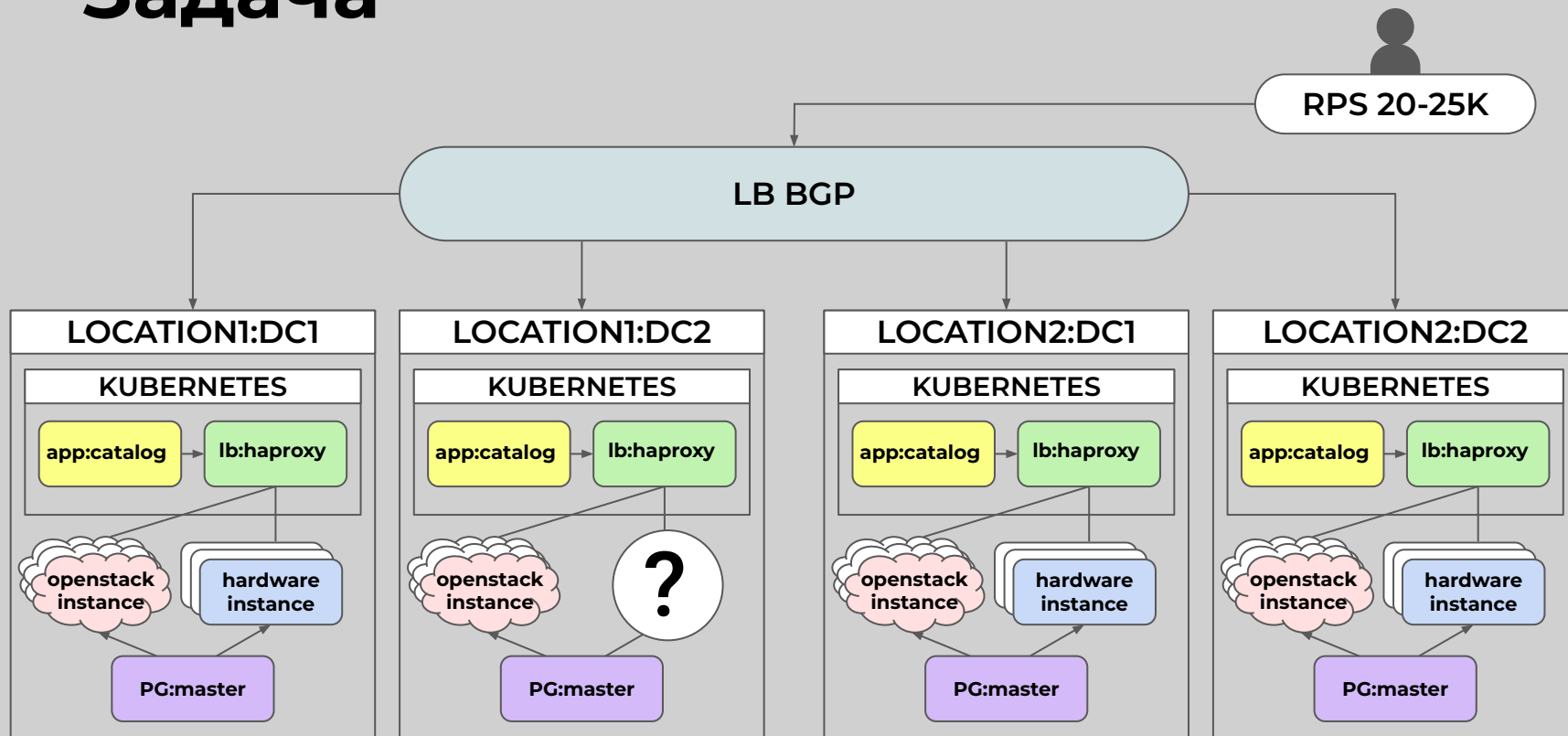
Задача



Задача



Задача



Задача

приложение периодически интенсивно
записывает в мастер

балансируем чтение на репликах

необходимо вписаться в SLA 99.9

размер кластера 300Gb, 4.5к таблиц

необходимо в пределах 10 мин
получить X2 / X3 репик postgresql



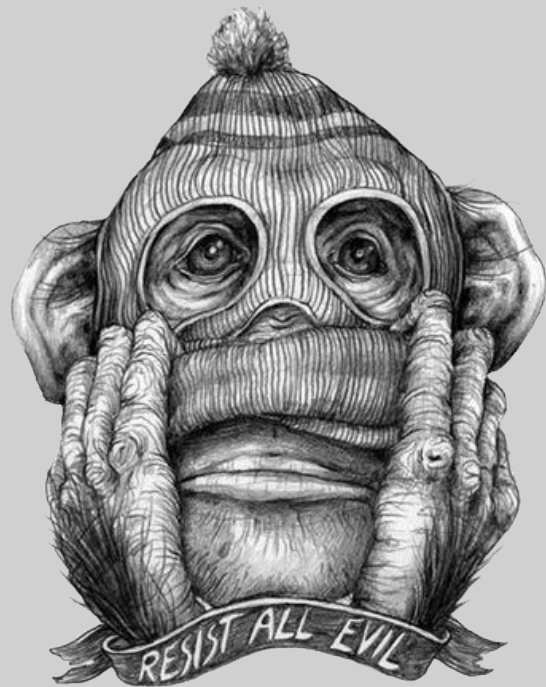
Муки выбора

держат горячий резерв на HW

отдать postgresql оператору

использовать DBaaS

облачный провайдер ?



Муки выбора

держат горячий резерв на HW

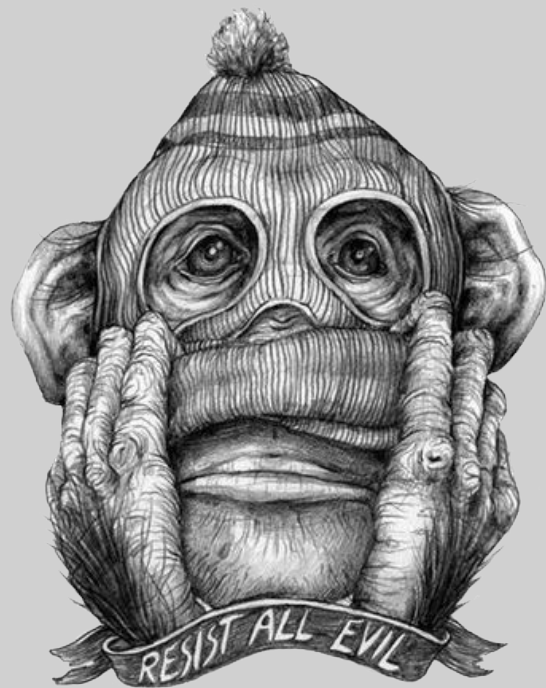
*не рационально содержать
статический набор реплик*

*нет гарантии, что доступного набора
хватит для того, чтобы обработать всплеск*

отдать postgresql оператору

использовать DBaaS

облачный провайдер ?



Муки выбора

держат горячий резерв на HW

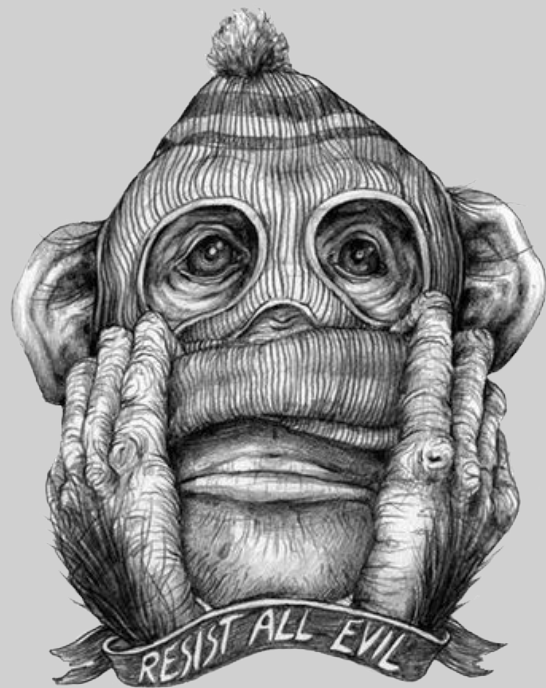
отдать postgresql оператору

*необходимо резервирование ресурсов
CPU / memory*

*восстановление большого количества
реплик pg_basebackup + wal = долго*

использовать DBaaS

облачный провайдер ?



Муки выбора

держат горячий резерв на HW

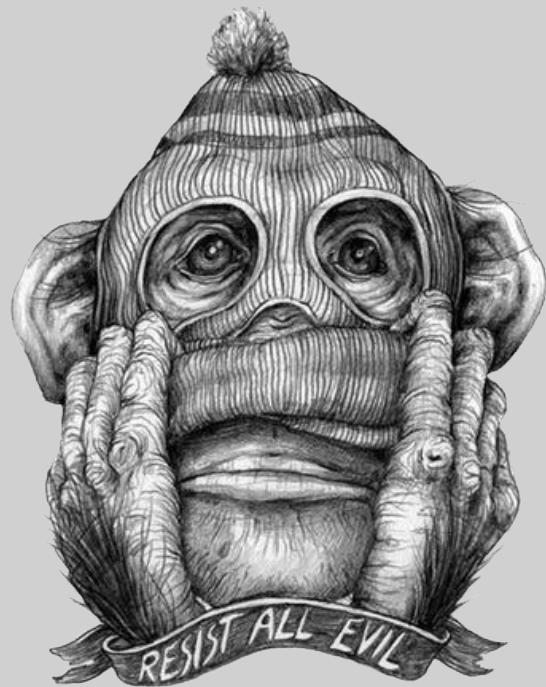
отдать postgresql оператору

использовать DBaaS

мы не планируем использовать мастера

*нет возможности использовать
custom extension*

облачный провайдер ?



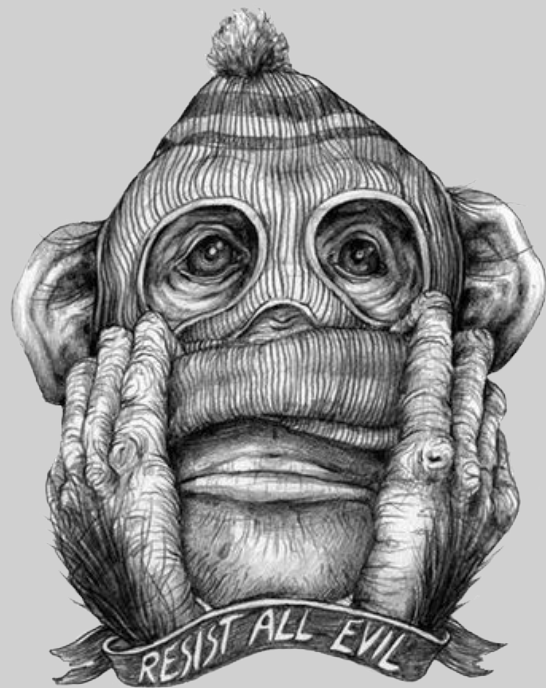
Муки выбора

держат горячий резерв на HW

отдать postgresql оператору

использовать DBaaS

облачный провайдер ?



Муки выбора

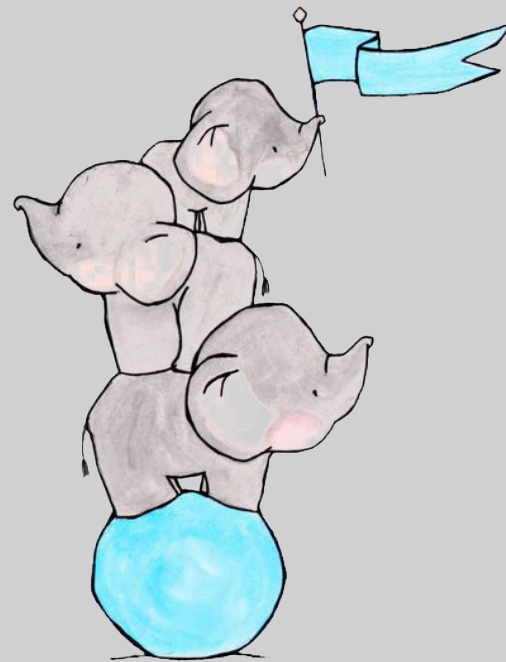
держат горячий резерв на HW

отдать postgresql оператору

использовать DBaaS

облачный провайдер ?

SberCloud ?!



CLOUD

Муки выбора

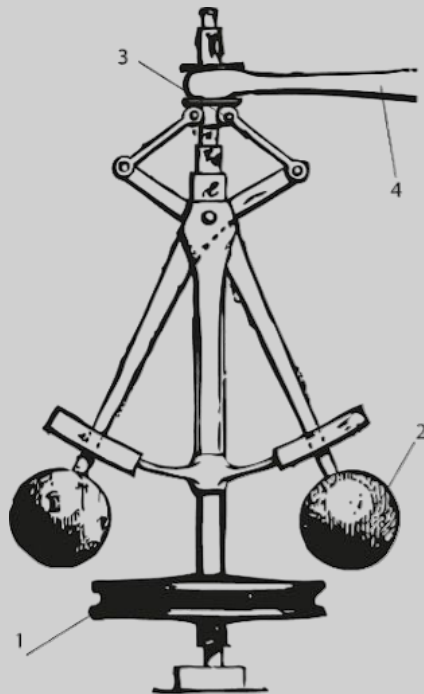
держат горячий резерв на HW

отдать postgresql оператору

использовать DBaaS

облачный провайдер ?

SberCloud ?!



Муки выбора

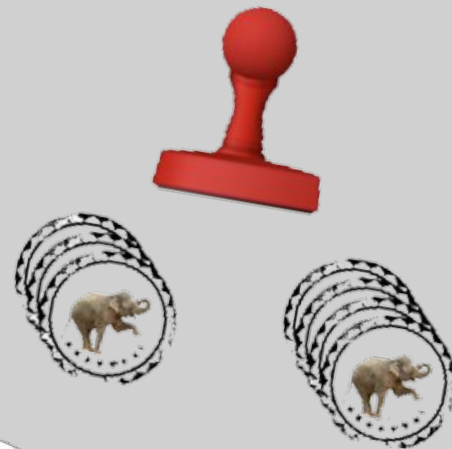
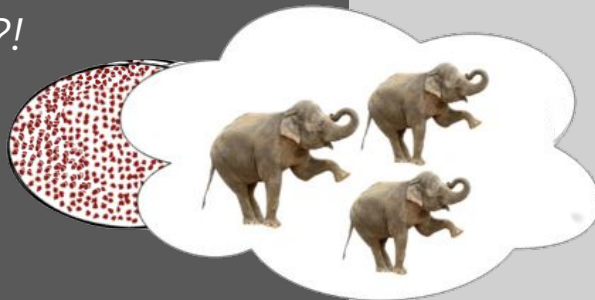
держат горячий резерв на HW

отдать postgresql оператору

использовать DBaaS

облачный провайдер ?

SberCloud ?!



Муки выбора

держат горячий резерв на HW

отдать postgresql оператору

использовать DBaaS

облачный провайдер ?

SberCloud ?!

говорите точно

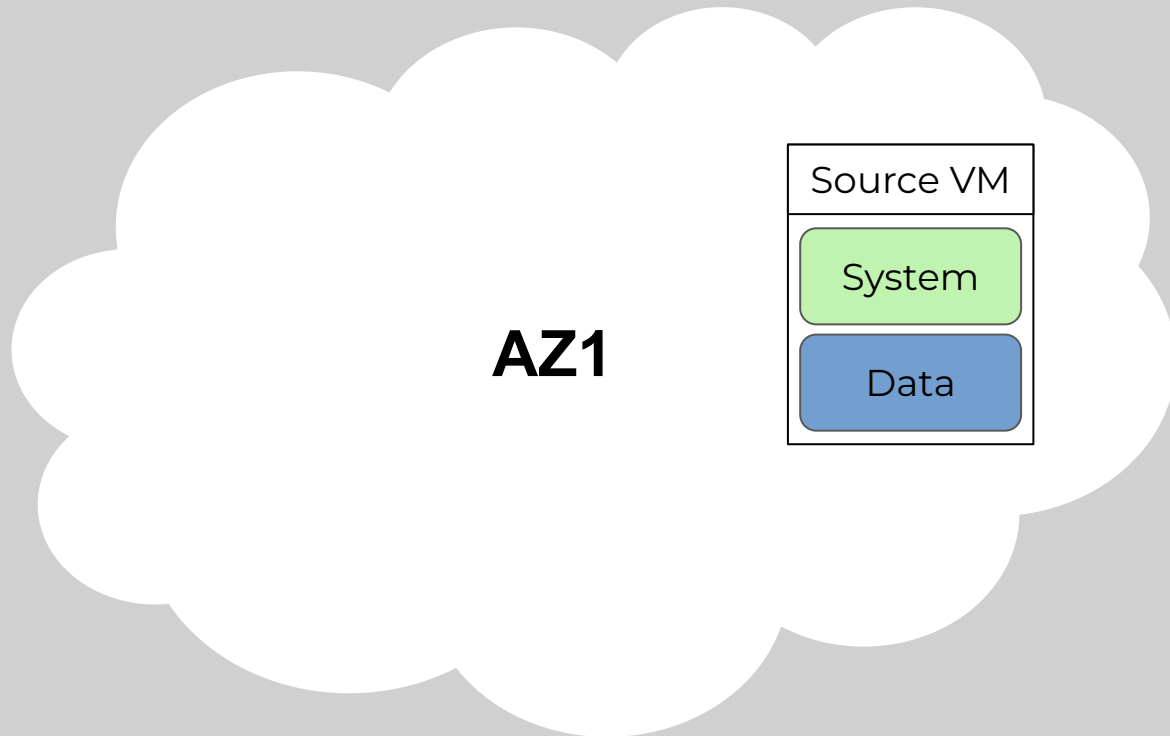


сколько вешать в граммах?

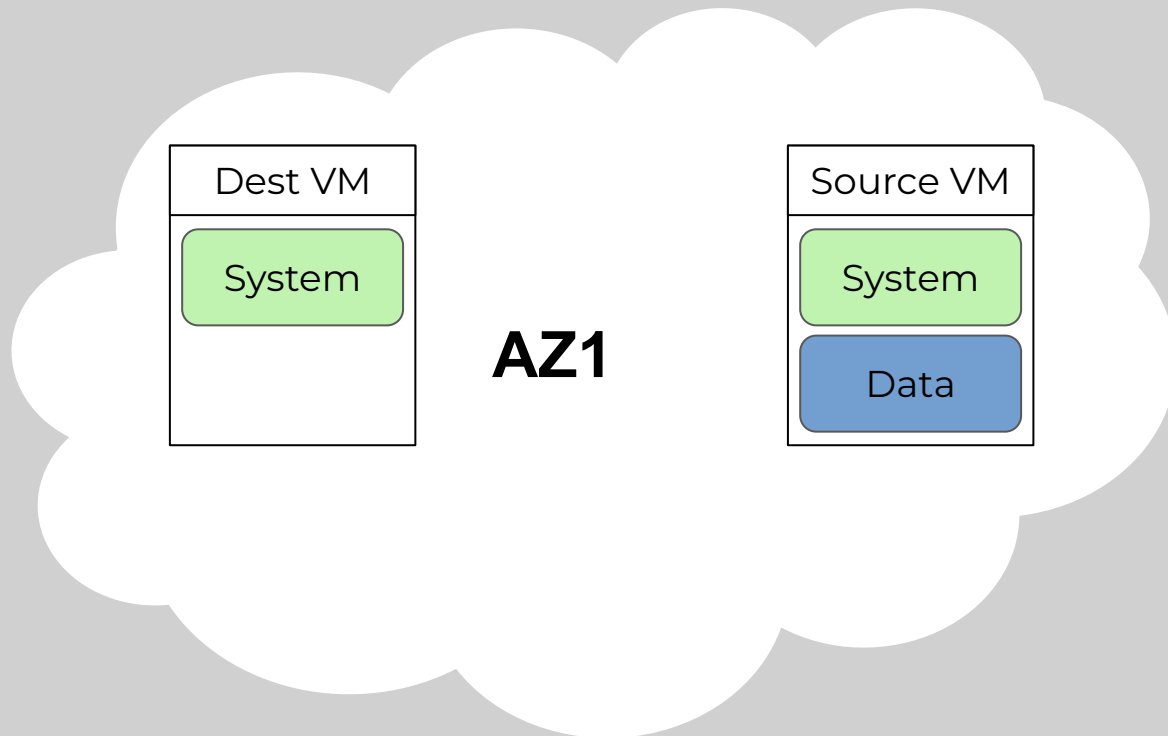
Где мои данные ?



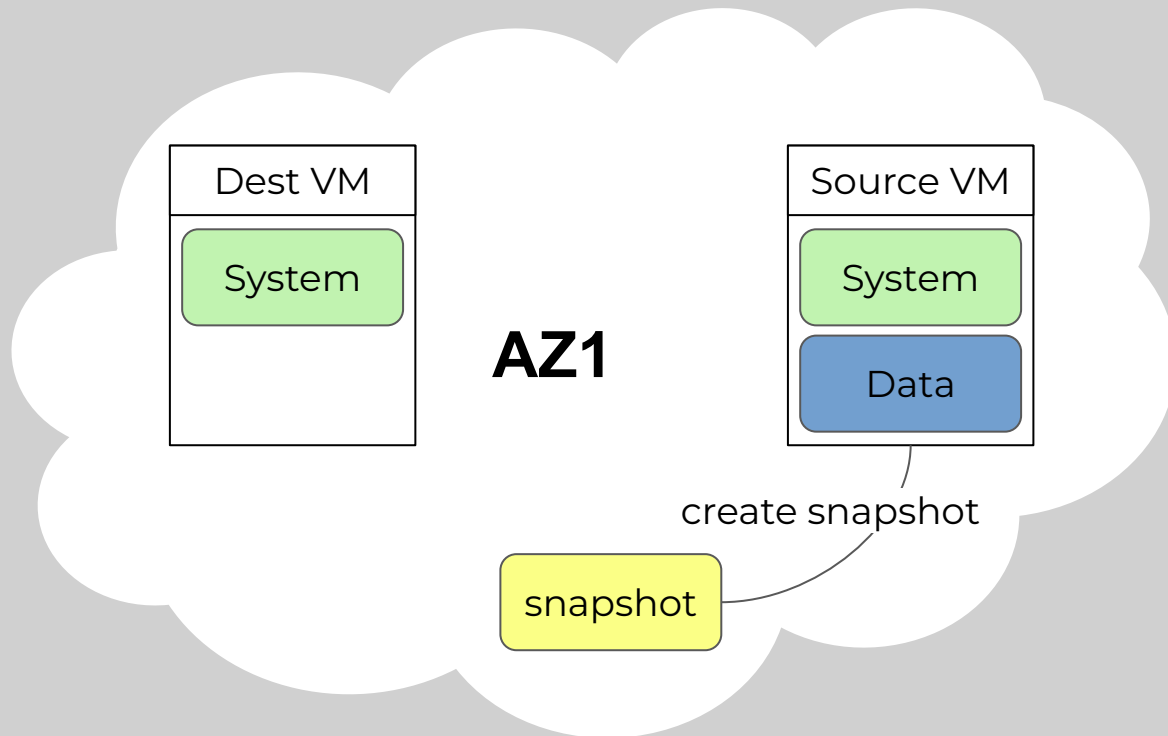
Где мои данные ?



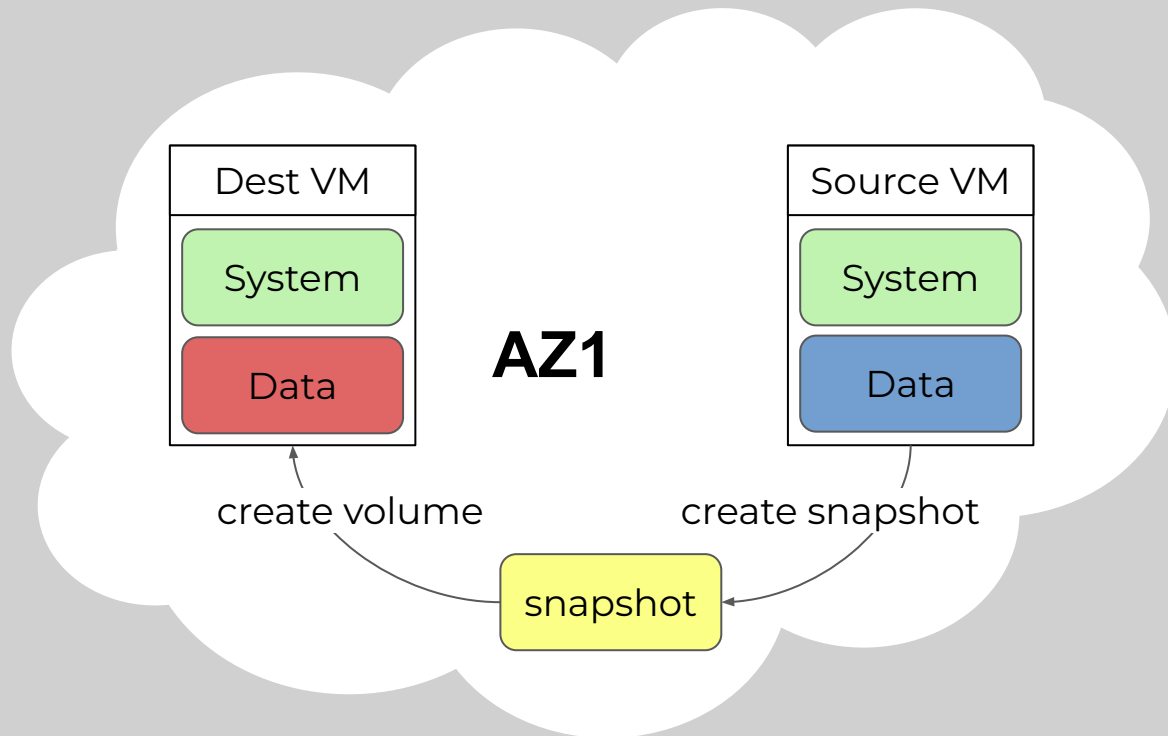
Где мои данные ?



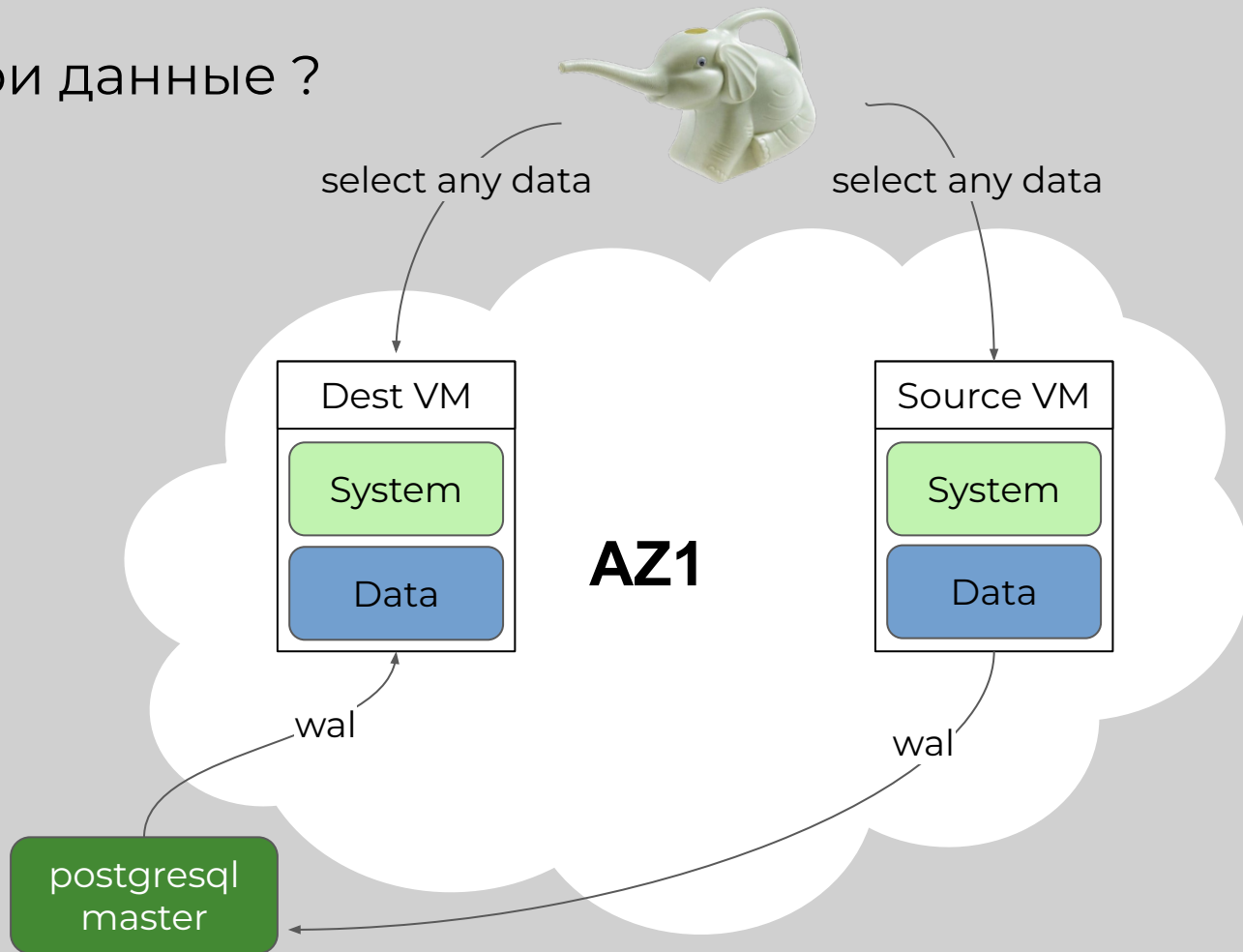
Где мои данные ?



Где мои данные ?



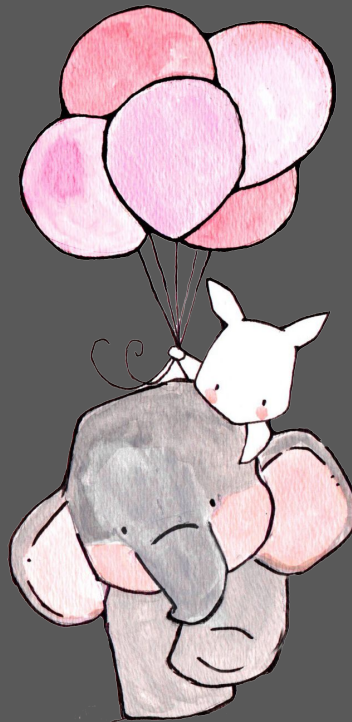
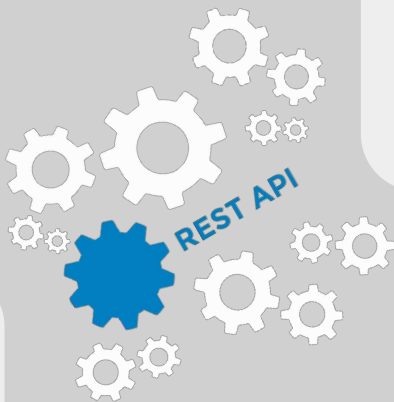
Где мои данные ?



Где мои данные ?



Python вам в помощь



Python: коротко о важном

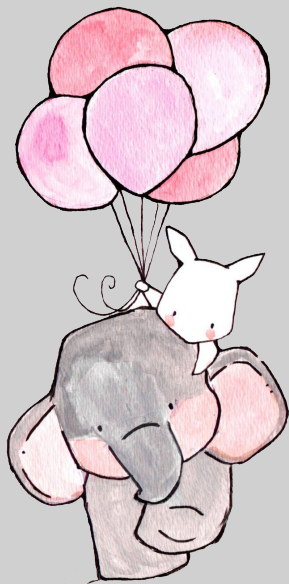
```
111 def attach_volume(self):
112     self.__create_snapshot()
113     self.__get_source_volume_info()
114     self.__create_volume()
115     data = {
116         "volume_id": self.__volume.id
117     }
118     print('Attach volume %s to instance %s' % (self.__volume.id, self.__dest_instance_id))
119     self.__conn.compute.create_volume_attachment(server=self.__dest_instance_id, **data)
120     res = self.__conn.block_store.wait_for_status(self.__volume,
121                                                    status='in-use',
122                                                    failures=['error'],
123                                                    interval=2,
124                                                    wait=60)
125     print('Volume info: %s' % res)
126     self.__delete_snapshot()
---
```


Python вам в помощь

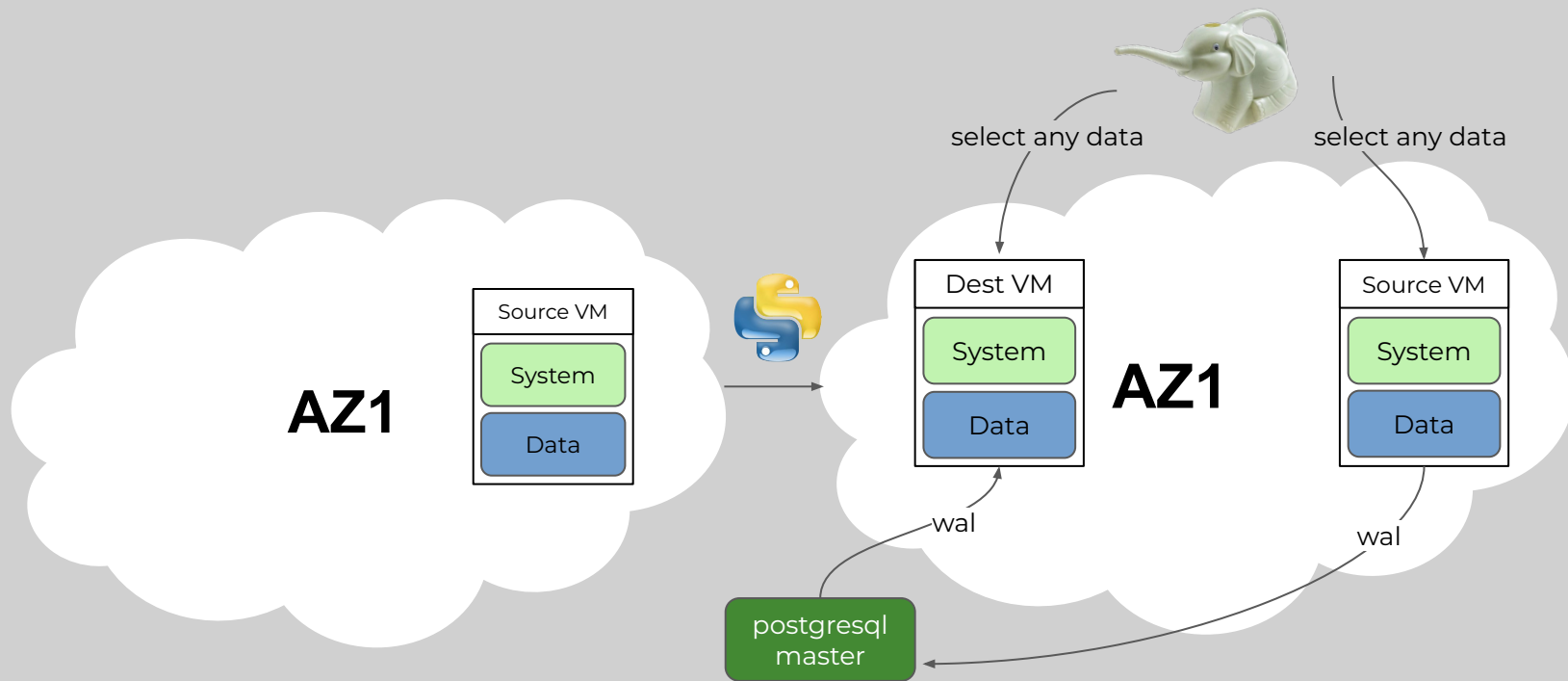


Python вам в помощь

прототип умеет самостоятельно воспроизводиться



Python вам в ПОМОЩЬ



Абразив для образов

Абразив для образов

всё есть **Ansible**



ANSIBLE

Абразив для образов

всё есть **Ansible**



ANSIBLE

Packer заправляет облака



Абразив для образов

всё есть **Ansible**



ANSIBLE

Packer заправляет облака



HashiCorp

Packer

собираем всё вместе



Магия autoscaler **конфигурация**

☐	Name	Status	Specifications	Image	System Disk
	☐ ipa-pg-autoscaler-config	 Bound	c3.3xlarge.4 12 vCPUs 48 GB	webapi-autoscaler-image	Ultra-high I/O 40 GB

Name	ipa-pg-autoscaler-config	ID
Status	 Bound	Created
Image	webapi-autoscaler-image	Login Mode
License Type	None	Key Pair
EIP	--	Flavor selection policy
User Data	View Details	ECS Group

Магия autoscaler **ИНСТАНСЫ С ДАННЫМИ**

ECS Information

ID	f63752c8-3e64-4d06-9ec6-33c68014df2c
Name	ipa-pg-autoscaler-config-68VKJK93 
Region	ru-moscow
AZ	AZ1
Specifications	Dedicated general-purpose c3.3xlarge.4 12 vCPUs 48 GiB

Магия autoscaler **ИНСТАНСЫ С ДАННЫМИ**

ECS Information

ID	f63752c8-3e64-4d06-9ec6-33c68014df2c
Name	ipa-pg-autoscaler-config-68VKJK93 
Region	ru-moscow
AZ	AZ1
Specifications	Dedicated general-purpose c3.3xlarge.4 12 vCPUs 48 GiB

Running

Monitoring

 Monitoring

▼ Disks

System Disk

[ipa-pg-autoscale...](#) Ultra-high I/O | 40 GiB

Data Disk

[ipa-pg-autoscale...](#) Ultra-high I/O | 400 GiB

Магия autoscaler **pg_isready!**

```
volkov@ipa-pg-autoscaler-config-68vkjk93:~$ sudo pg_isready
/var/run/postgresql:5432 - accepting connections
volkov@ipa-pg-autoscaler-config-68vkjk93:~$ df -h
```

Filesystem	Size	Used	Avail	Use%	Mounted on
udev	24G	0	24G	0%	/dev
tmpfs	4.8G	392M	4.4G	9%	/run
/dev/vda1	40G	3.0G	35G	8%	/
tmpfs	24G	28K	24G	1%	/dev/shm
tmpfs	5.0M	0	5.0M	0%	/run/lock
tmpfs	24G	0	24G	0%	/sys/fs/cgroup
/dev/vdb	394G	262G	115G	70%	/var/lib/postgresql
tmpfs	4.8G	0	4.8G	0%	/run/user/1004

```
volkov@ipa-pg-autoscaler-config-68vkjk93:~$ █
```



Магия autoscaler **scale up/down**

Add AS Policy

Enable

Disable

Delete

You can add 8 more policies.

☐ Name 	Status	Policy Type 	Monitored Object
☐ ipa-pg-autoscaler-up-policy	 Disabled	Alarm	ipa-pg-autoscaler-group 47178ecf-21fa-4508-bf...
☐ ipa-pg-autoscaler-down-policy	 Disabled	Alarm	ipa-pg-autoscaler-group 47178ecf-21fa-4508-bf...

Магия autoscaler **scale up/down**

Disable

Delete

You can add 8 more policies.

	Status	Policy Type 	Monitored Object	Trigger Condition 
	 Disabled	Alarm	ipa-pg-autoscaler-group 47178ecf-21fa-4508-bf...	CPU Usage Avg. $\geq$ 17%. A scaling acti
	 Disabled	Alarm	ipa-pg-autoscaler-group 47178ecf-21fa-4508-bf...	CPU Usage Avg. $\leq$ 13%. A scaling acti

Магия autoscaler **scale up/down**

ed You can add 8 more policies.

	Policy Type 	Monitored Object	Trigger Condition 
ed	Alarm	ipa-pg-autoscaler-group 47178ecf-21fa-4508-bf...	CPU Usage Avg. >= 17%. A scaling action will be triggered...
ed	Alarm	ipa-pg-autoscaler-group 47178ecf-21fa-4508-bf...	CPU Usage Avg. <= 13%. A scaling action will be triggered...

Магия autoscaler **scale up/down**

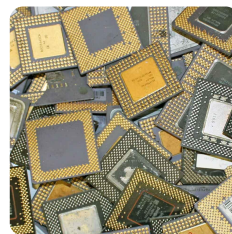
ed You can add 8 more policies.

	Policy Type 	Monitored Object	Trigger Condition 
ed	Alarm	ipa-pg-autoscaler-group 47178ecf-21fa-4508-bf...	CPU Usage Avg. $\geq 17\%$. A scaling action will be triggered...
ed	Alarm	ipa-pg-autoscaler-group 47178ecf-21fa-4508-bf...	CPU Usage Avg. $\leq 13\%$. A scaling action will be triggered...

Магия autoscaler **scale up/down**

е You can add 8 more policies.

	Policy Type 	Monitored Object	Trigger Condition 
ed	Alarm	ipa-pg-autoscaler-group 47178ecf-21fa-4508-bf...	CPU Usage Avg. $\geq 17\%$. A scaling action will be triggered...
ed	Alarm	ipa-pg-autoscaler-group 47178ecf-21fa-4508-bf...	CPU Usage Avg. $\leq 13\%$. A scaling action will be triggered...



Магия Autoscaler
ручной труд?

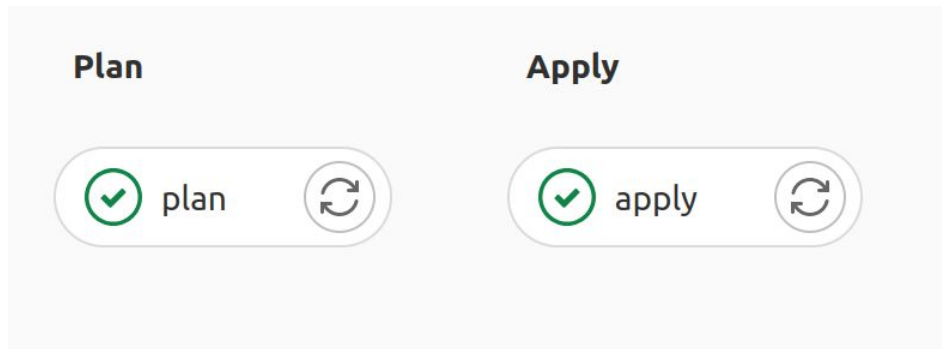


Магия Autoscaler **Terraform**

```
terraform {  
  required_providers {  
    sbercloud = {  
      source = "sbercloud-terraform/sbercloud"  
      version = "1.9.0"  
    }  
  }  
}
```



Магия Autoscaler **горшочек, вари!**



Балансер?

< | ipa-pg-autoscaler-lb → Running

Basic Information

Listeners

Backend Server Groups

Monitoring

Access Logs

Tags

Add Backend Server Group

ipa-pg-autoscale... | TCP →



Removal Protection



Add

Modify Weight

Remove

Available servers: 10

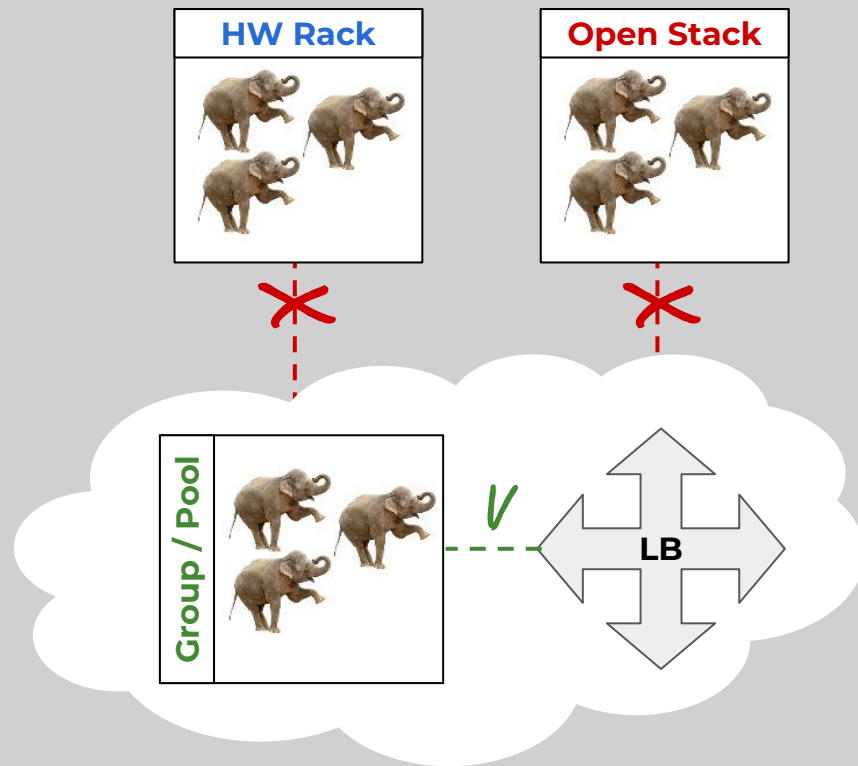
☐	Name	Status
☐	ipa-pg-autoscaler-config-WQTFU02W	→ Running
☐	ipa-pg-autoscaler-config-Q3IMXWT0	→ Running
☐	ipa-pg-autoscaler-config-SB38F71E	→ Running
☐	ipa-pg-autoscaler-config-86RJ46XP	→ Running
☐	ipa-pg-autoscaler-config-U1FDMWJD	→ Running

5

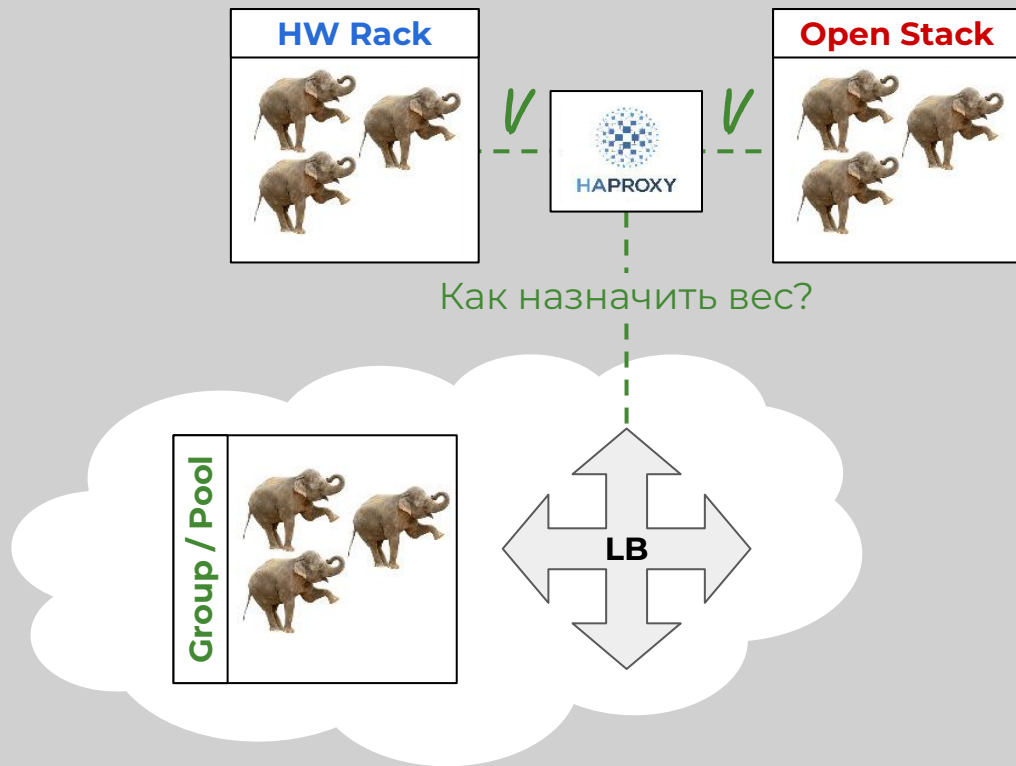
Total Records: 10

< 1 2 >

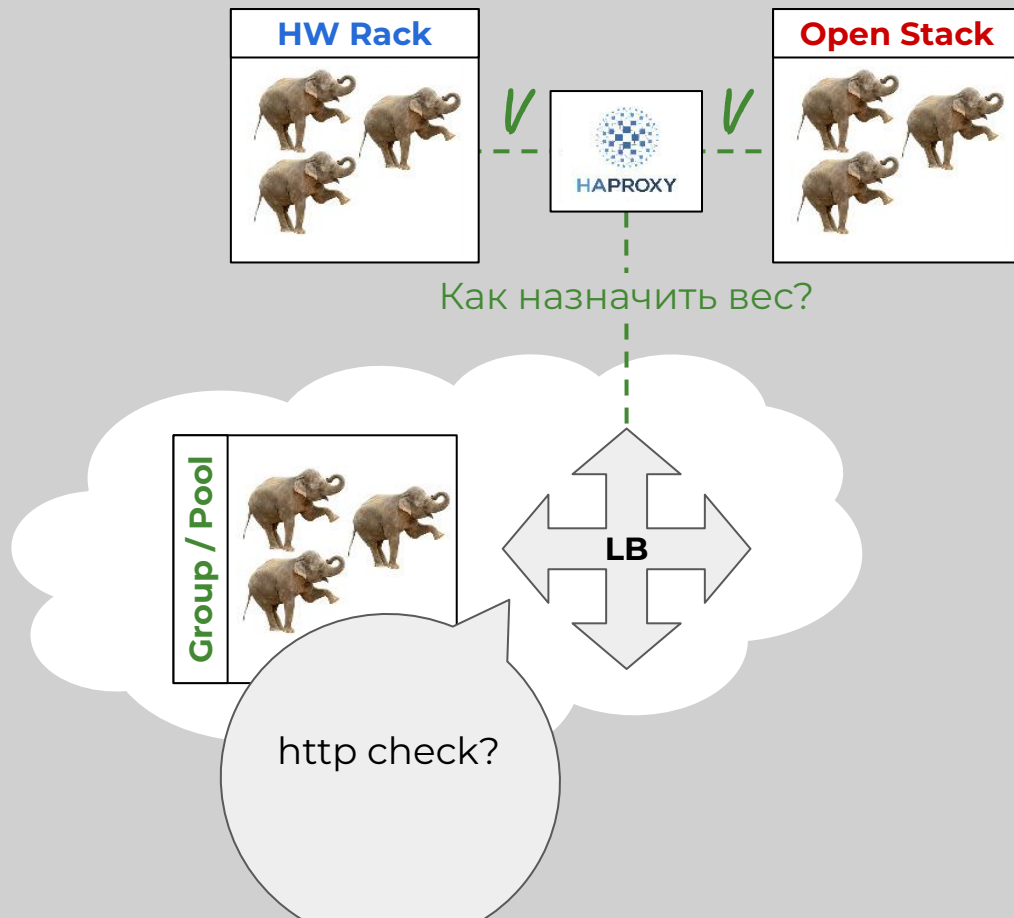
Балансер
что с ним не так



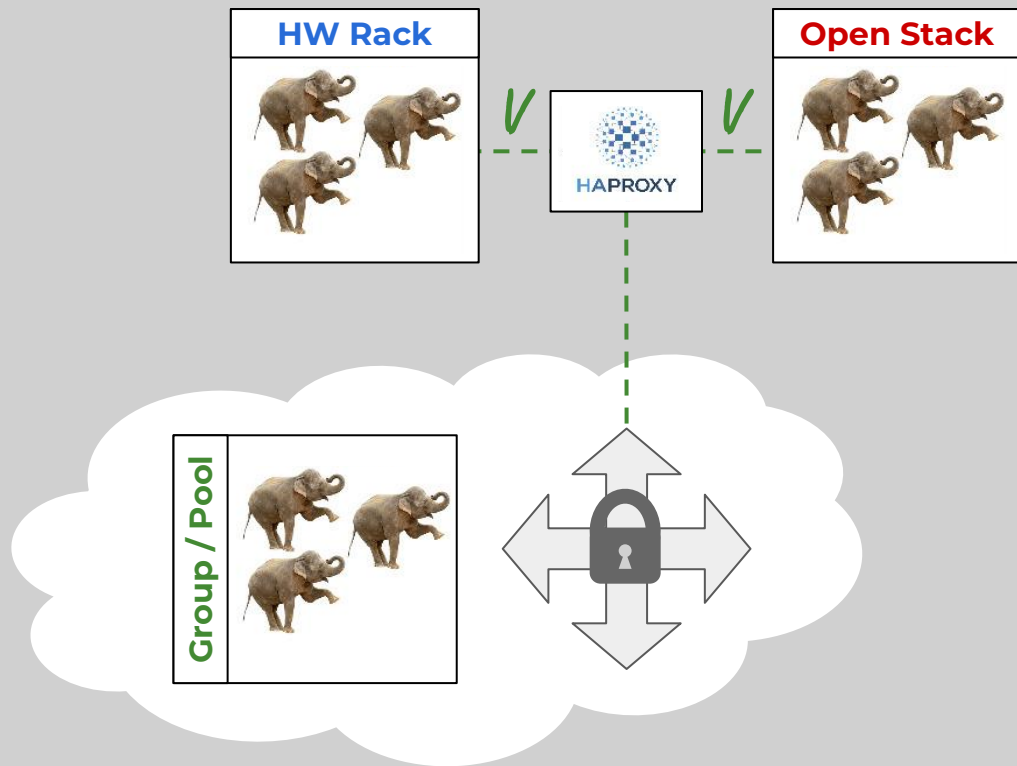
Балансер
что с ним не так



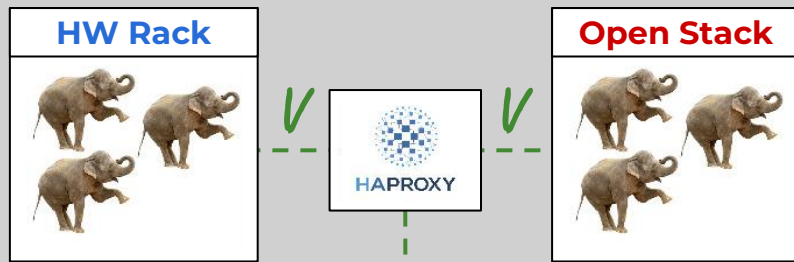
Балансер
что с ним не так



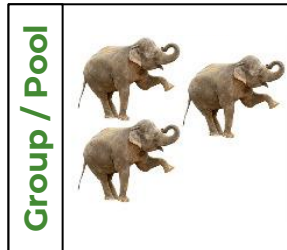
Балансер
что с ним не так



Балансер
что с ним не так



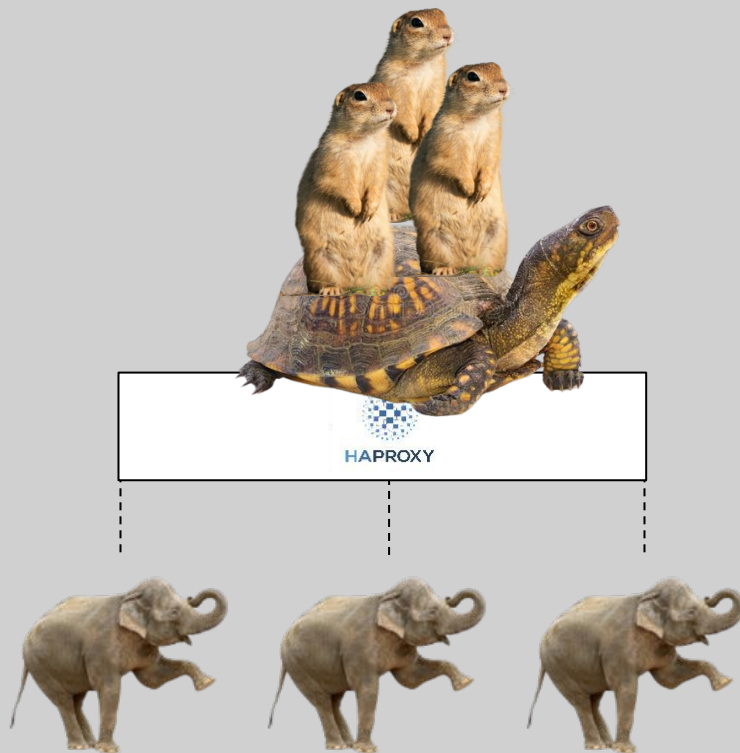
Правда где-то тут?



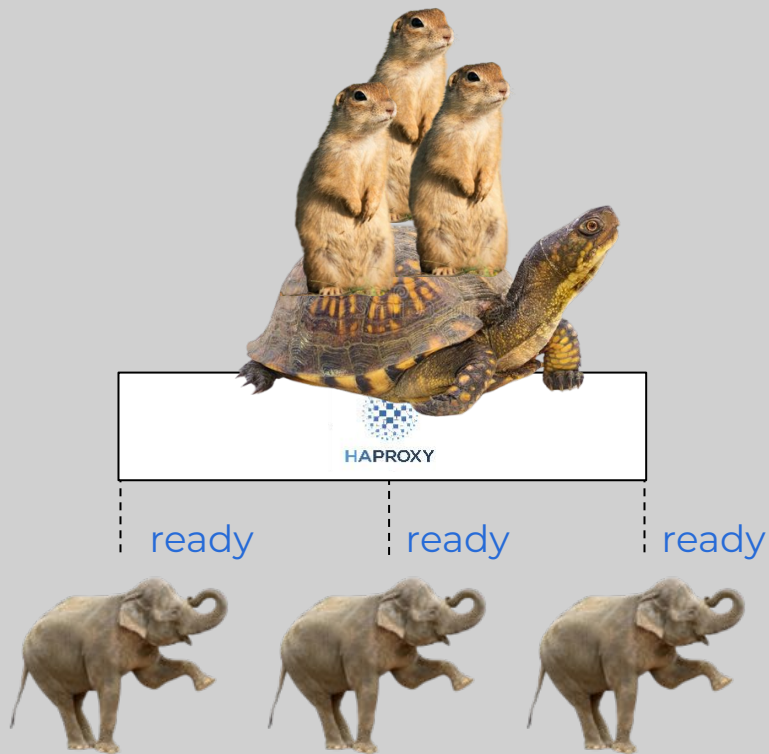


Облака – это хорошо,
а где мой haproxу?

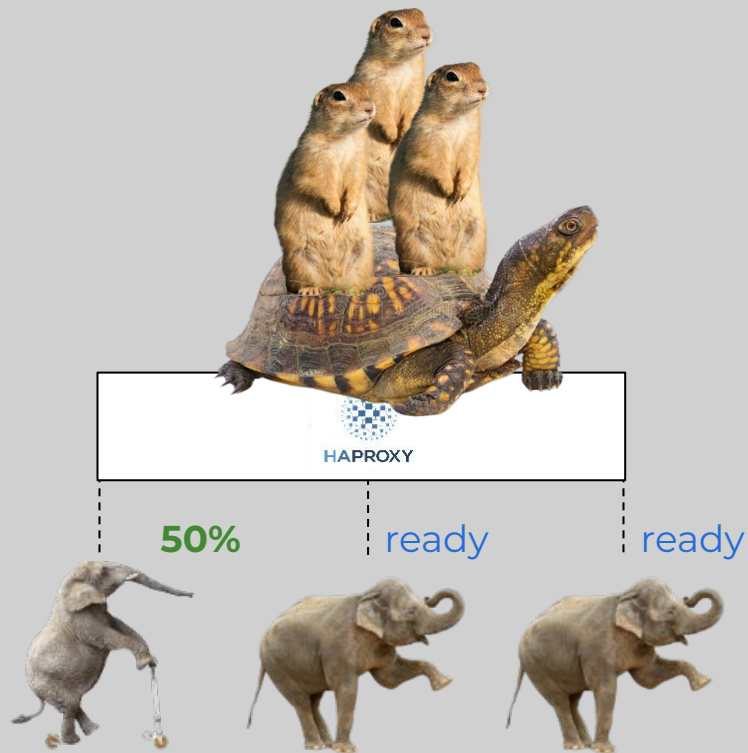
Облака – это хорошо, а где мой haproxy?



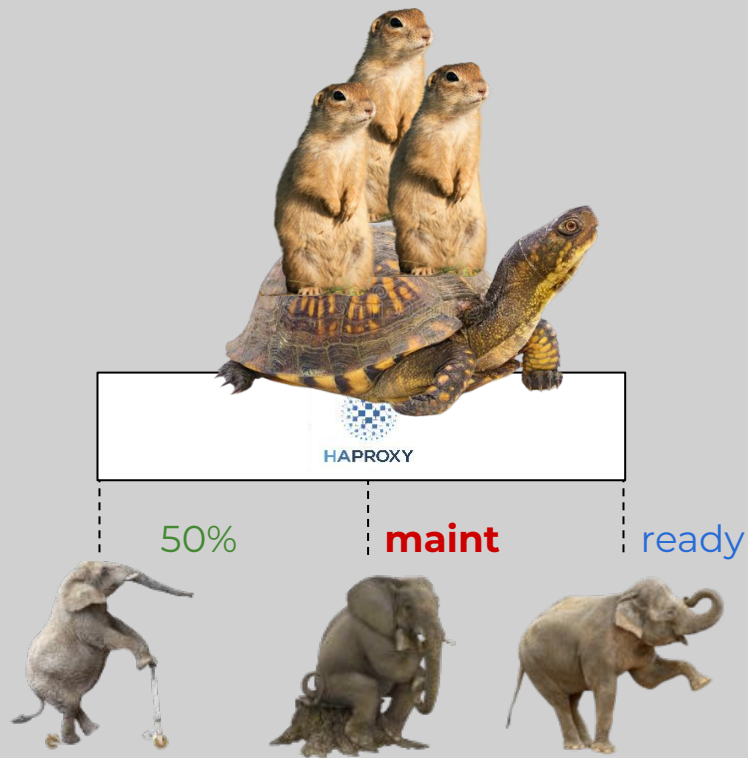
Облака – это хорошо, а где мой haproxy?



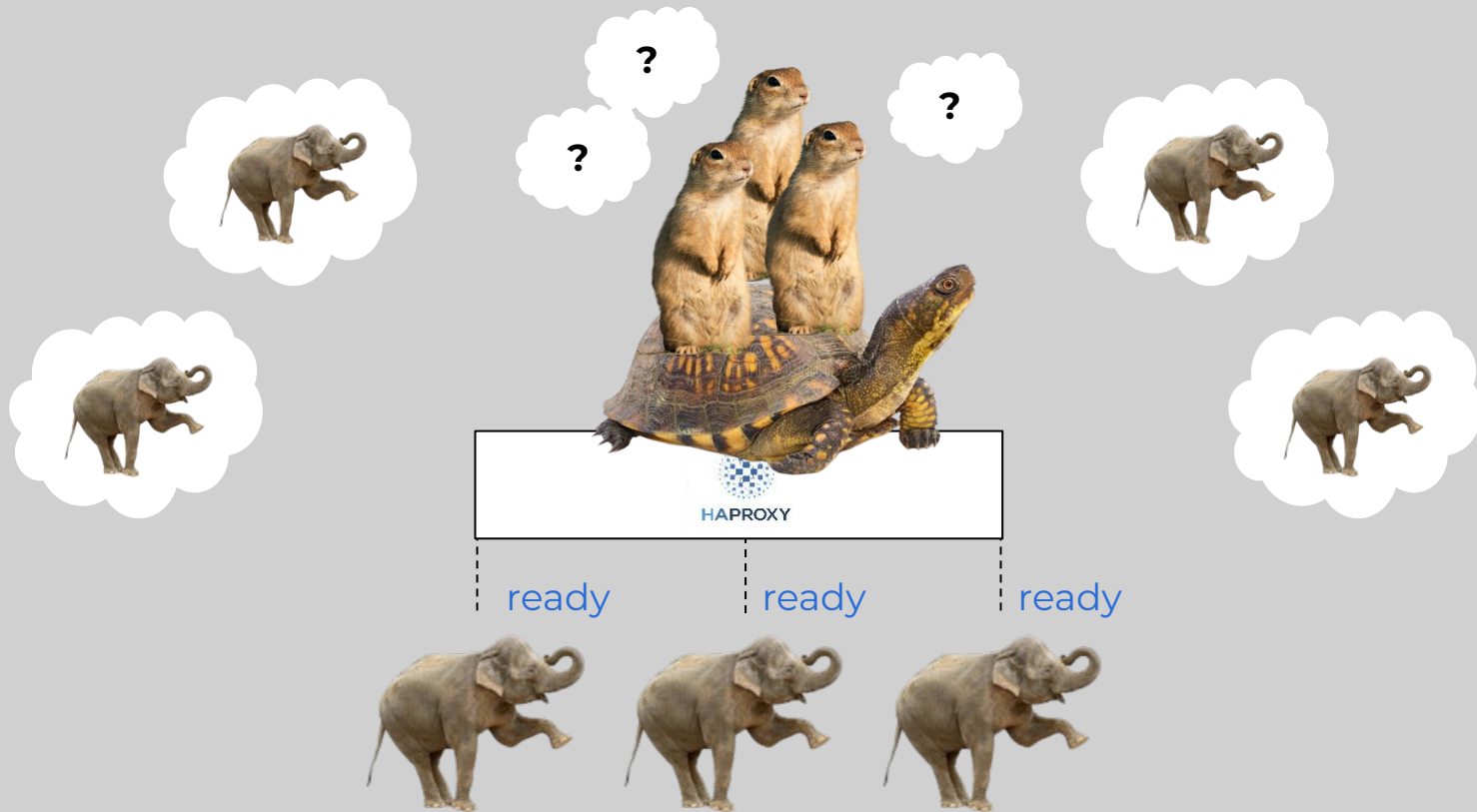
Облака – это хорошо, а где мой haproxy?



Облака – это хорошо, а где мой haproxy?



Облака – это хорошо, а где мой haproxy?





Haproxy Auto Discovery?

HAProxy. Всё придумано до нас?

Documentation for HAProxy Enterprise 2.5r1

This is the latest version of HAProxy Enterprise

More ▾



› API › Data Plane API

Data Plane API

The Data Plane API is a service that runs alongside HAProxy Enterprise and lets you configure the load balancer using HTTP RESTful commands. This feature benefits a number of use cases that are suited for a dynamically-generated configuration.

HAProxy. А это вы можете!

Documentation for HAProxy Enterprise 2.5r1

This is the latest version of HAProxy Enterprise

More ▾

 > [API](#) > [Data Plane API](#) > [Installation](#)

Installation

You can install the Data Plane API for either [HAProxy Enterprise](#) or [HAProxy \(Community version\)](#).

HAProxy Community

Install the Data Plane API with the community version of HAProxy.

HAProxy Enterprise

Install the Data Plane API with HAProxy Enterprise.

Нарпроху. Делай как мы!

Create the Data Plane API configuration file,
`/etc/haproxy/dataplaneapi.hcl` and add the following:

```
dataplaneapi {  
    host = "0.0.0.0"  
    port = 5555  
  
    user "admin" {  
        insecure = true  
        password = "adminpwd"  
    }  
  
    transaction {  
        transaction_dir = "/tmp/haproxy"  
    }  
}  
  
haproxy {  
    config_file = "/etc/haproxy/haproxy.cfg"  
    haproxy_bin = "/usr/sbin/haproxy"  
  
    reload {  
        reload_cmd = "service haproxy reload"  
        restart_cmd = "service haproxy restart"  
        reload_delay = "5"  
    }  
}
```

Нарпроху. Делай как мы!

Start the Data Plane API:

```
$ sudo dataplaneapi -f /etc/haproxy/dataplaneapi.hcl
```

Verify that the API is running properly by calling the *info* function:

```
$ curl -X GET --user admin:adminpwd http://localhost:5555/v2/info  
  
{"api":{"build_date":"0001-01-01T00:00:00.000Z","version":"v2.2.0 68bd22b"},"system":{}}
```

HAProxy. Делай лучше нас!

Add a `program` section to your HAProxy configuration:

```
global
    master-worker

program api
    command dataplaneapi -f /etc/haproxy/dataplaneapi.hcl
    no option start-on-reload
```

The `no option start-on-reload` line avoids restarting the Data Plane API each time that HAProxy reloads.

NOTE

For this to work, you must run HAProxy in `master-worker` [↗](#) mode by adding the `master-worker` directive to your `global` section (or by adding the `-W` command-line argument)

Нарроху. Делай лучше нас!

```
$ sudo systemctl restart haproxy  
$ sudo systemctl status haproxy
```

Main PID: 1274 (haproxy)

Tasks: 6

Memory: 5.5M

CPU: 2.838s

CGroup: /system.slice/haproxy.service

1274 /usr/sbin/haproxy -Ws -f /etc/haproxy/haproxy.cfg -p /run/haproxy.pid -sf 2768 -x /run/haproxy/admin.sock

2768 /usr/sbin/haproxy -Ws -f /etc/haproxy/haproxy.cfg -p /run/haproxy.pid -sf 2662 -x /run/haproxy/admin.sock

2830 /usr/local/bin/dataplane-api --host 0.0.0.0 --port 5555 -b /usr/local/sbin/haproxy -c

/etc/haproxy/haproxy.cfg -d 5 -r systemctl reload haproxy -u controller

2831 /usr/sbin/haproxy -Ws -f /etc/haproxy/haproxy.cfg -p /run/haproxy.pid -sf 2768 -x /run/haproxy/admin.sock

Haproxy. Auto Discovery?

haproxytech / dataplaneapi Public

Watch 20

[Code](#) [Issues](#) 29 [Pull requests](#) 5 [Actions](#) [Projects](#) [Security](#) [Insights](#)

master [dataplaneapi / discovery / README.md](#)

Go to file

...

 **rmaticevic** DOC/MINOR: added consul service discovery documentation ✓

Latest commit 79b223c on 16 Apr 2021 [History](#)

2 contributors  

9 lines (5 sloc) 168 Bytes

[<>](#) [File](#) [Raw](#) [Blame](#) [Edit](#) [More](#)

Data Plane API Service Discovery

AWS

Check the AWS Service Discovery [README](#).

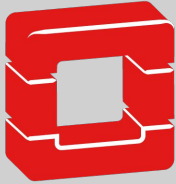
CONSUL

Check the CONSUL Service Discovery [README](#).

Haproxy. Auto Discovery?



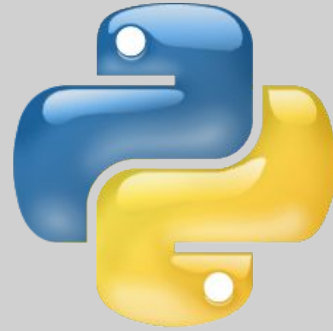
CLOUD



openstack



REST API



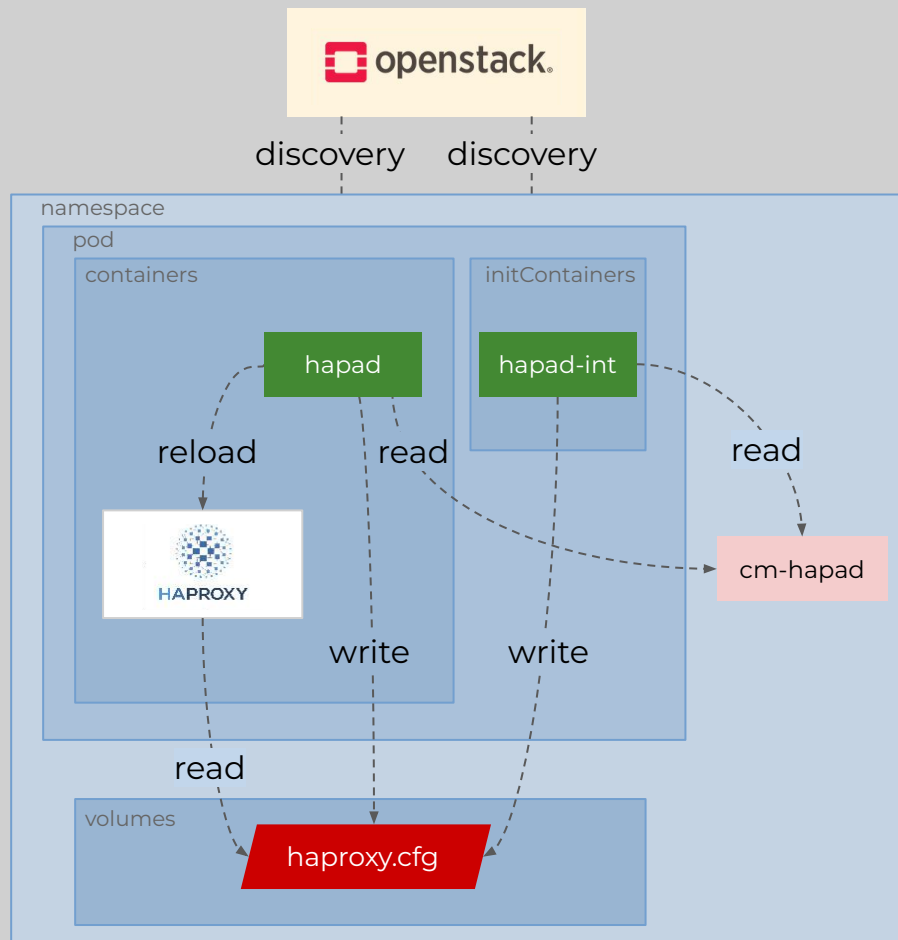
Напроху. 20 строк о самом-самом.

```
12 def load_backends(self, _name={'name': 'ipa'}):
13     """ Выберем IP адреса виртуальных машин с regex по имени """
14     nodes=[]
15     try:
16         servers=self.__conn.compute.servers(details='true',**_name)
17         for server in servers:
18             for address in server.addresses:
19                 addr = server.addresses[address]
20                 for item in addr:
21                     nodes.append({"name":"{0}".format(item['addr'])})
22         self.__previos_nodes=nodes
23     except Exception as ex:
24         nodes = self.__previos_nodes
25         print(f"Error on fetch openstack instances: {ex}")
26         print(f"Saved previous list: {nodes}")
27     return nodes
```

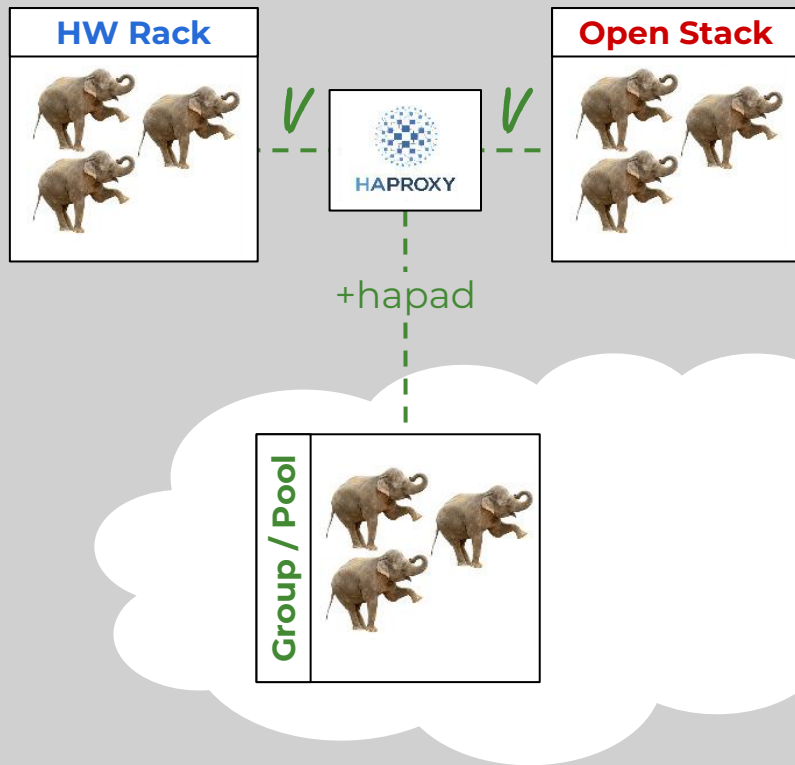
Нaproxy. 20 строк о самом-самом.

```
90 def __reload_haproxy(self):
91     """
92     Пишем хапрокси в сокет для перезагрузки конфигурации
93     """
94     try:
95         s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
96         s.connect((self.__haproxy_addr, self.__haproxy_port))
97         s.send('reload\n'.encode())
98         print("Haproxy reloaded")
99     except Exception as ex:
100         print(f"Haproxy reload error: {ex}")
101     finally:
102         s.close()
```

Haproxy Auto Discovery



Haproxy Auto Discovery



Haproxy

Auto Discovery

replica																												
	Queue			Session rate			Sessions						Bytes				Denied		Errors		Warnings		Server					
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In		Out		Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck
10.77.223.143	0	0	-	0	6		4	6	500	15 909	15 909	36s	16 482 432 620	385 156 532 538			0		0	0	0	0	11d23h UP	L7OK/0 in 3ms	100/100	Y	-	
10.77.223.46	0	0	-	0	3		2	6	500	15 882	15 882	44s	16 303 828 110	381 015 116 561			0		0	0	0	0	11d23h UP	L7OK/0 in 2ms	100/100	Y	-	
10.77.223.42	0	0	-	0	3		3	6	500	15 822	15 822	20s	16 086 833 992	377 384 942 229			0		0	0	0	0	11d23h UP	L7OK/0 in 3ms	100/100	Y	-	
10.77.223.3	0	0	-	0	3		4	6	500	15 991	15 991	37s	16 189 194 199	377 933 898 426			0		0	0	0	0	11d23h UP	L7OK/0 in 3ms	100/100	Y	-	
10.77.223.7	0	0	-	0	3		3	6	500	15 966	15 966	15s	16 316 617 414	380 822 933 619			0		0	0	0	0	11d23h UP	L7OK/0 in 2ms	100/100	Y	-	
10.77.223.83	0	0	-	0	2		3	6	500	15 754	15 754	40s	16 500 836 551	385 468 957 967			0		0	0	0	0	11d23h UP	L7OK/0 in 3ms	100/100	Y	-	
10.77.223.80	0	0	-	0	2		3	6	500	15 724	15 724	1m4s	16 045 902 182	374 967 467 972			0		0	0	0	0	11d23h UP	L7OK/0 in 2ms	100/100	Y	-	
10.77.223.175	0	0	-	0	2		4	6	500	15 894	15 894	1m4s	16 362 633 384	383 988 336 474			0		0	0	0	0	11d23h UP	L7OK/0 in 3ms	100/100	Y	-	
10.77.223.200	0	0	-	0	3		3	6	500	15 726	15 726	47s	16 445 976 844	387 578 499 350			0		0	0	0	0	11d23h UP	L7OK/0 in 2ms	100/100	Y	-	
10.77.223.155	0	0	-	0	2		3	6	500	15 744	15 744	1m13s	16 298 239 365	381 654 725 126			0		0	0	0	0	11d23h UP	L7OK/0 in 2ms	100/100	Y	-	
s1db.webapi.os-m9.hw	0	0	-	0	3		3	6	500	15 260	15 260	1m7s	15 781 002 386	353 919 106 714			0		0	0	0	0	11d23h UP	L7OK/0 in 4ms	100/100	Y	-	
s2db.webapi.os-m9.hw	0	0	-	0	2		0	6	500	4 680	4 680	8d3h	4 594 475 896	105 687 419 831			0		0	0	0	0	8d3h MAINT		100/100	Y	-	
s6db.webapi.sc-m9.hw	0	0	-	0	2		3	6	500	15 882	15 882	44s	16 390 934 393	381 602 005 342			0		0	0	0	0	11d23h UP	L7OK/0 in 2ms	100/100	Y	-	
s7db.webapi.sc-m9.hw	0	0	-	0	3		3	6	500	16 087	16 087	1m7s	15 573 743 406	372 346 694 252			0		0	0	0	0	11d23h UP	L7OK/0 in 1ms	100/100	Y	-	
s8db.webapi.sc-m9.hw	0	0	-	1	2		3	6	500	15 964	15 964	2s	16 334 526 871	383 545 325 604			0		0	0	0	0	11d23h UP	L7OK/0 in 2ms	100/100	Y	-	
s3db.webapi.m9.hw	0	0	-	0	3		6	12	500	33 118	33 118	50s	36 550 464 633	854 152 113 519			0		0	0	0	0	11d23h UP	L7OK/0 in 3ms	200/200	Y	-	
s4db.webapi.m9.hw	0	0	-	0	3		6	12	500	33 408	33 408	45s	37 197 720 786	866 264 295 215			0		0	0	0	0	11d23h UP	L7OK/0 in 2ms	200/200	Y	-	

А метрики-то где?



А метрики-то где?

VMagent + openstack_sd_config!!!



VICTORIA
METRICS

А метрики-то где?

VMagent + openstack_sd_config!!!

Label ==> metadata



```
11 relabel_configs:
12   - source_labels: [__meta_openstack_instance_status]
13     action: keep
14     regex: ACTIVE
15   - source_labels: [__meta_openstack_tag_{{ index }}_prometheus_port]
16     regex: (\d+)
17     action: keep
18   - source_labels: [__meta_openstack_instance_name]
19     target_label: instance
20   - source_labels: [__address__, __meta_openstack_tag_{{ index }}_prometheus_port]
21     action: replace
22     regex: ([^:]+)(?::\d+)?;(\d+)
23     replacement: $1:$2
24     target_label: __address__
25   - source_labels: [__meta_openstack_tag_{{ index }}_prometheus_path]
26     action: replace
27     target_label: __metrics_path__
28     regex: (.+)
29   - action: labelmap
30     regex: __meta_openstack_tag_{{ index }}_label_(.+)
31   - action: labelmap
32     regex: __meta_openstack_tag_label_(.+)
33
```


А метрики-то где?

VMagent + openstack_sd_config!!!

Label ==> metadata

Terraform тоже в курсе!



```
20 as_default_metadata = {  
21     label_team = "webapi"  
22     label_environment = "production"  
23     label_dc = "m9"  
24     label_job = "webapi-postgresql"  
25     "0_label_type" = "node"  
26     "0_prometheus_port" = "9100"  
27     "1_label_type" = "postgresql"  
28     "1_prometheus_port" = "9187"  
29 }
```

Инстансы изнутри и снаружи

Start

Stop

Restart

More ▾

Tag: project=ipa

✕

Add filter

☐ Name/ID	AZ ⌵	Status ⌵	Specifications/Image	IP Address
☐ ipa-pg-autoscaler-config-Z6GUWOYC 75d81dd3-1f4d-4227-8fce-e28f42faff72	AZ1	➡ Running	12 vCPUs 48 GiB c3.3xlarge.4 webapi-autoscaler-image	10.77.223.143 (Private IP)

Endpoint	State	Labels	Scrapes
http://10.77.223.143:9100/metrics (response)	UP	{dc="m9", environment="production", instance="ipa-pg-autoscaler-config-Z6GUWOYC", job="webapi-postgresql", team="webapi", type="node"} {__address__="10.77.223.143:0", __meta_openstack_address_pool="doublegis_sbercloud_vpc_subnet_2", __meta_openstack_instance_flavor="c3.3xlarge.4", __meta_openstack_instance_id="75d81dd3-1f4d-4227-8fce-e28f42faff72", __meta_openstack_instance_name="ipa-pg-autoscaler-config-Z6GUWOYC", __meta_openstack_instance_status="ACTIVE", __meta_openstack_private_ip="10.77.223.143", __meta_openstack_project_id="0aeca931b100268f2f3fc017b105993a", __meta_openstack_tag_0_label_type="node", __meta_openstack_tag_0_prometheus_port="9100", __meta_openstack_tag_1_label_type="postgresql", __meta_openstack_tag_1_prometheus_port="9187", __meta_openstack_tag_label_dc="m9", __meta_openstack_tag_label_environment="production", __meta_openstack_tag_label_job="webapi-postgresql", __meta_openstack_tag_label_team="webapi", __meta_openstack_tag_project="ipa", __meta_openstack_user_id="0aeca931c680268f4f41c017d61284c2", __metrics_path__="/metrics", __scheme__="http", __scrape_interval__="15s", __scrape_timeout__="15s", job="sbercloud_0"}	63992

Выше, Быстрее, Сильнее

Автоматическое определение AZ

Автоматическое создание DNS в локальной зоне

Метрики при срабатывании политик autoscaler

Опциональные параметры на срабатывание autoscaler

Выводы

Оперативно реализовали возможность резко вырасти
и не платить за простой

Не оказались в ситуации когда вендорлок – это приговор

Незначительно расширили стэк применяемых технологий

DataPlaneApi



Terraform provider



POSTGRESQL

в гетерогенной инфраструктуре



2ГИС

Волков Николай



@rmrfthm