

avito.tech

Ереван — 2023

Жизнь без kubernetes

Как выживать небольшим проектам

Виталий Лихачев

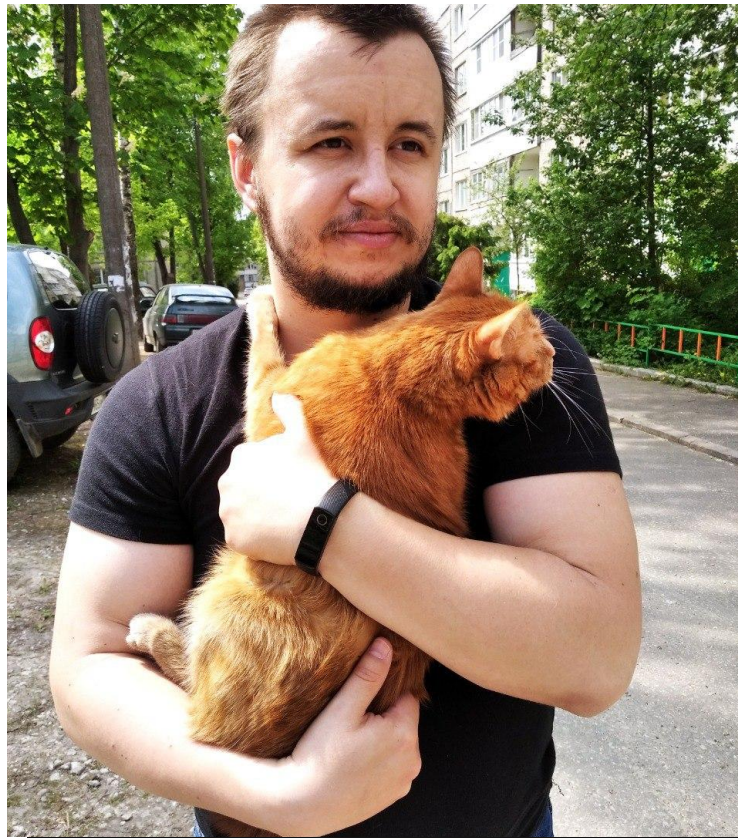
Senior software engineer

Виталий Лихачев

Senior software engineer

Frontend, backend, DevOps, full stack. Работал в стартапах и крупных компаниях, в продукте и на аутсорс.

- ▶ Golang, python (django, drf, flask, fastapi), php (symfony, laravel, Yii2)
- ▶ JS, React, Vue.js
- ▶ PostgreSQL, RabbitMQ, Redis, Kafka
- ▶ Никто не читает списки
- ▶ HaProxy, nginx/openresty+lua/traefik, Linux administration
- ▶ Docker, k8s, terraform, ansible, packer, prometheus, etc.



Что мы хотим от оркестратора?

- ▶ Запуск наших сервисов прозрачно на нескольких нодах
- ▶ Простой способ отслеживания состояния сервисов
- ▶ Коммуникация между сервисами
- ▶ Простой локальный запуск
- ▶ Простота менеджмента сервисов небольшими командами
- ▶ Тегирование нод на основе их возможностей (disk-type=ssd, has-gpu=true)
- ▶ Общее хранилище секретов и конфигурации
- ▶ Метрики/логи

Нельзя просто взять kubernetes?

- ▶ Сложность управления и операционные расходы
- ▶ Тяжелые обновления:
 - Показательный пример — падение reddit (pi day)
- ▶ Крутая кривая обучения

- ▶ Всего пять бинарей (и etcd сверху):
 - kube-api-server
 - kube-scheduler
 - kube-controller-manager
 - kubelet
 - kube-proxy
 - etcd



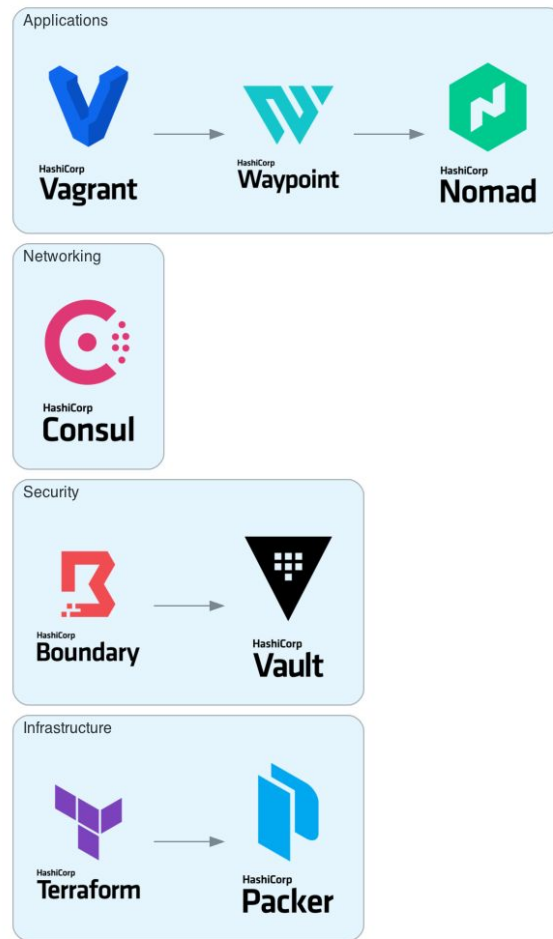
ПЛАН

01. Обзор HashiCorp Stack

02. Концепции Nomad

03. Сценарии использования Nomad

Обзор HashiCorp Stack



ВСЕ НАЧАЛОСЬ В 2012

INFRASTRUCTURE



Terraform

Infrastructure as code



Packer

Machine images

SECURITY



Vault

Identity-based security



Boundary

Secure remote access

NETWORKING



Consul

Multi-cloud service
networking

APPLICATIONS



Nomad

Workload orchestration



Waypoint

App deployment
workflows



Vagrant

Environment workflows

CONSUL

- ▶ Control plane: регистрация сервисов, доступ к сервисам из любой точки кластера
- ▶ Центральный реестр для DNS-записей сервисов
- ▶ Service mesh: sidecar proxies, mTLS
- ▶ Распределённое key/value хранилище



VAULT

- ▶ Управление и хранение секретов
- ▶ Динамические одноразовые секреты
- ▶ Нативная интеграция с nomad



NOMAD

- ▶ Следует философии unix
- ▶ Планировщик и оркестратор
- ▶ Кривая обучения значительно проще k8s
- ▶ Эффективное использование нод с целью их максимальной утилизации
- ▶ Zero downtime deployments: rolling, blue/green, canary
- ▶ Поддержка как контейнеров, так и legacy приложений



Nomad

NOMAD

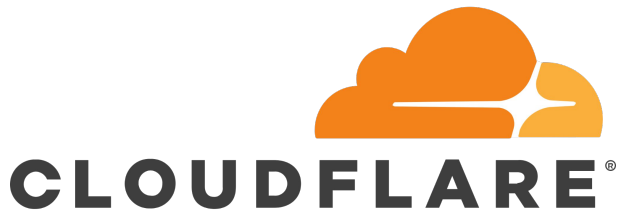
- ▶ Декларативное описание задач
- ▶ Cloud agnostic
- ▶ Container storage interface plugins
- ▶ Container network interface plugins
- ▶ Task drivers: docker, Java Jar файлы, QEMU VMs, ехес для бинарников, etc.
- ▶ ACL
- ▶ Web UI
- ▶ Нативная интеграция с consul и vault



Nomad

КТО ИСПОЛЬЗУЕТ?

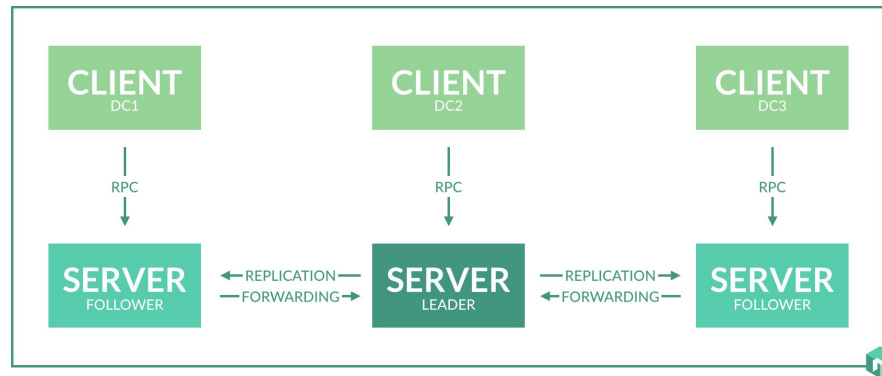
- ▶ Cloudflare
- ▶ Roblox
- ▶ CircleCI
- ▶ PagerDuty
- ▶ eBay



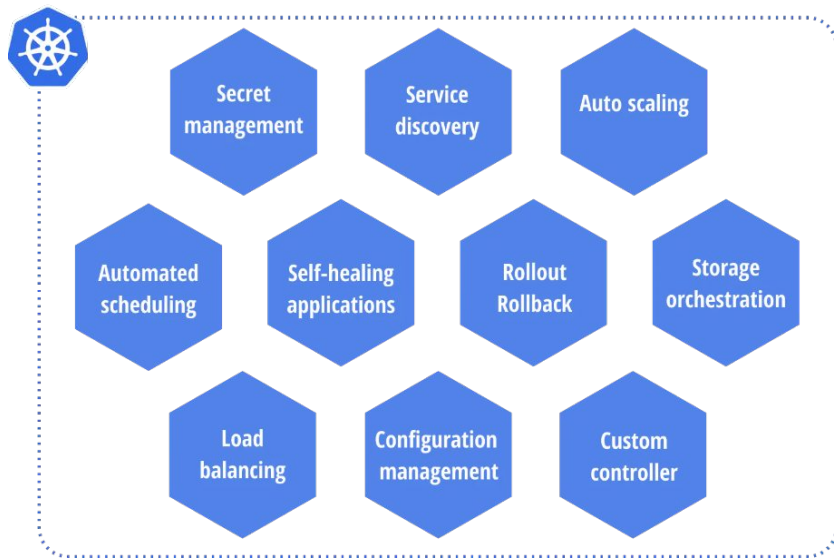
PagerDuty



Концепции Nomad

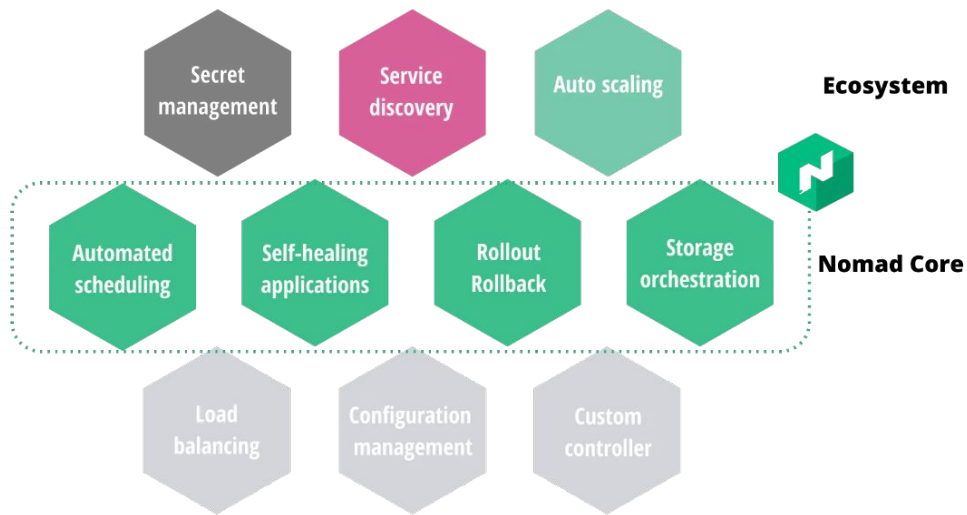


NOMAD VS K8S



► managed

avito.tech



► unmanaged

КОНЦЕПЦИИ

- ▶ Кластеры состоят из узлов, на которых запущен бинарный файл Nomad: как для серверов, так и для клиентов
- ▶ Серверы обеспечивают всю логику (планирование, распределение) для кластера
- ▶ Клиенты регистрируются на серверах и запускают полезную нагрузку
- ▶ Сервер может быть одновременно и клиентом (не рекомендуется для production-окружения)
- ▶ Полезная нагрузка описывается в виде декларативных job-файлов с желаемым состоянием системы
- ▶ Разные драйверы используются Nomad для выполнения задач (docker, exec, etc.)
- ▶ Task — наименьшая логическая единица
- ▶ Task group — набор tasks, которые должны быть запущены на одном клиенте

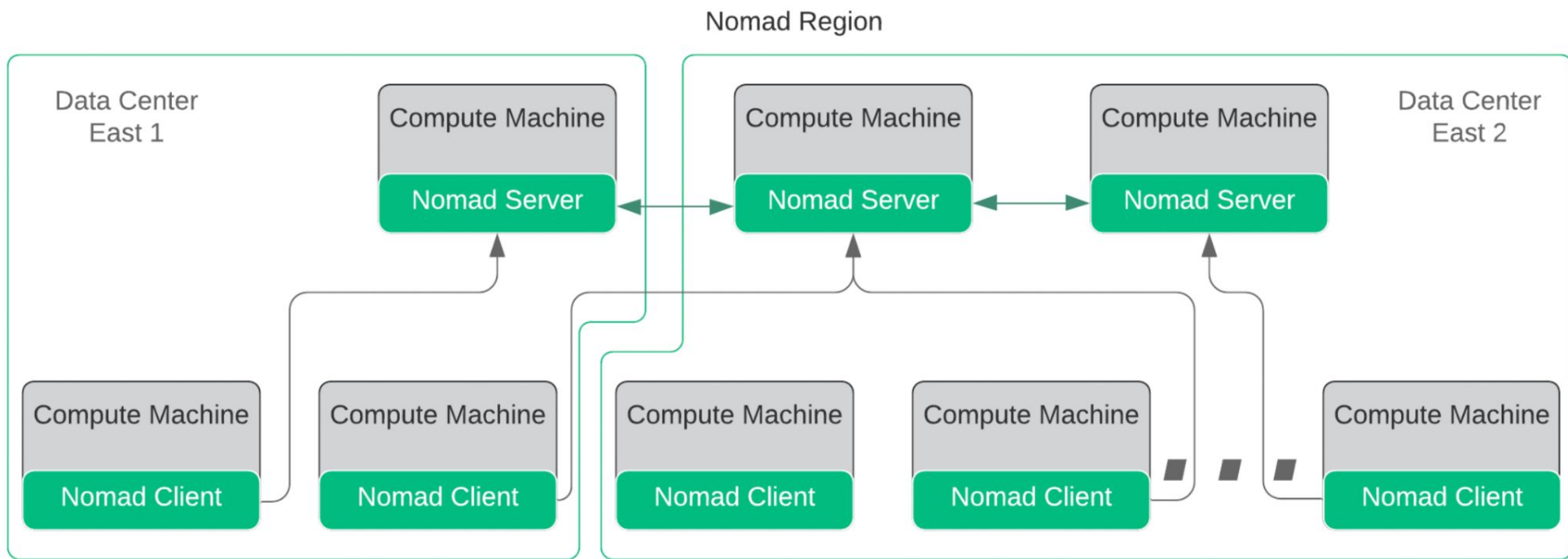
КОНЦЕПЦИИ

- ▶ Allocations: назначение tasks/task groups на клиентов
- ▶ Evaluation: расчет необходимости менять состояние системы (деплой измененной job, изменение состояния клиента)
- ▶ Максимальная утилизация узлов на основе алгоритма bin packing (контролируется spread/affinity опциями)
- ▶ Датацентр — физическая/логическая группировка узлов кластера
- ▶ Кластер может содержать несколько регионов

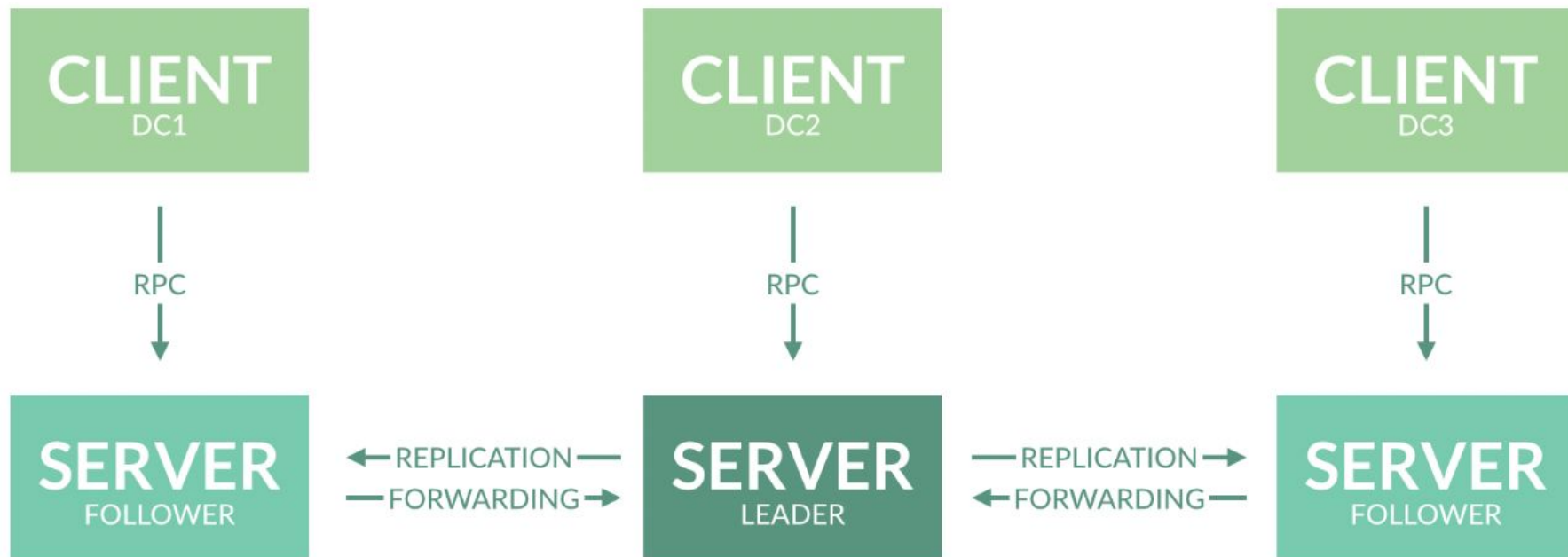
ПЛАНИРОВЩИК

- ▶ Поиск подходящих узлов для запуска job
- ▶ Ранжирование узлов на основе критериев, заданных в описании job, и на основе занятых на узле ресурсов
- ▶ Создание allocation plan
- ▶ Отправка allocation plan на выполнение

АРХИТЕКТУРА КЛАСТЕРА



СЕТЕВАЯ СВЯЗНОСТЬ



JOB

Декларативная спецификация желаемого состояния системы
в формате HashiCorp configuration language (HCL).

ФОРМАТ ОПИСАНИЯ JOB

```
job "example" {  
  datacenters = ["dc1"]  
  type = "system" # service/system/batch  
}
```

- ▶ Service: web service
- ▶ Batch: отчёты, cron jobs
- ▶ System: logging, monitoring, sidecars (k8s DaemonSet)

TASK GROUP

```
job "example" {  
  datacenters = ["dc1"]  
  group "cache" {  
    task "redis" {  
      driver = "docker"  
    }  
    task "web" {  
      driver = "docker"  
    }  
  }  
}
```

TASK DRIVER

- ▶ Docker
- ▶ Isolated fork/exec driver
- ▶ Java

```
task "example" {  
    driver = "exec"  
  
    config {  
        command = "name-of-my-binary"  
    }  
  
    artifact {  
        source =  
        "https://internal.file.server/name-of-my-binary"  
        options {  
            checksum = "sha256:abd123445ds4555555555"  
        }  
    }  
}
```

RESOURCES

```
job "example" {  
  group "cache"  
  {  
    task "example" {  
      resources {  
        cpu      = 500  
        memory   = 256  
        device "nvidia/gpu" {  
          count = 2  
        }  
      }  
    }  
  }  
}
```


BRIDGED NETWORK

```
job "example" {  
  datacenters = ["dc1"]  
  group "cache" {  
    network {  
      mode = "bridge"  
      port "redis_port" {  
        static = 6379  
        to      = 6379  
      }  
    }  
  }  
  task "redis" {  
    driver = "docker"  
    // {{ env "NOMAD_PORT_redis_port" }}  
  }  
}
```

SERVICE DISCOVERY

```
job "example" {  
  datacenters = ["dc1"]  
  group "cache" {  
    task "redis" {  
      service {  
        tags = ["my-tag"]  
        port = "db"  
        check {  
          type      = "tcp"  
          port      = "db"  
          interval  = "10s"  
          timeout   = "2s"  
        }  
      }  
    }  
  }  
}
```

CONSUL TEMPLATE

```
template {  
  data = <<EOH  
  ---  
    bind_port:  {{ env "NOMAD_PORT_db" }}  
    scratch_dir: {{ env "NOMAD_TASK_DIR" }}  
    node_id:     {{ env "node.unique.id" }}  
    service_key: {{ key "service/my-key" }}  
  EOH  
  destination = "local/file.yml"  
}
```

SERVER/CLIENT CONFIG

```
server {  
  enabled = true  
  bootstrap_expect = 3  
  server_join {  
    retry_join = ["nomad1", "nomad2"]  
  }  
}
```

```
client {  
  enabled = true  
  servers = ["nomad-server-1"]  
}
```

CLUSTERING OPTIONS

- ▶ Manual (IP/DNS)
- ▶ Чepeз consul
- ▶ Cloud auto-join (AWS, Azure, GCP Tags)

```
server_join {  
  retry_join = ["provider=aws tag_key=..."]  
}
```

```
// 169.254.169.254
```

```
consul {  
  address =  
  "127.0.0.1:8500"  
  server_service_name = "nomad"  
  client_service_name =  
  "nomad-client"  
  auto_advertise      = true  
  server_auto_join    = true  
  client_auto_join    = true  
}
```

СЦЕНАРИИ ИСПОЛЬЗОВАНИЯ

CLI

```
nomad server members  
nomad node status  
nomad job plan|run job.hcl  
nomad job status  
nomad eval status  
nomad alloc status  
nomad alloc logs <allocid>  
...
```

AUTOSCALING

- ▶ Horizontal app autoscaling: scale up/down task groups (enterprise only)
- ▶ Horizontal cluster autoscaling: plugins for AWS, GCP, Azure, Digital Ocean, etc.

HORIZONTAL CLUSTER AUTOSCALING

```
scaling "cluster_policy" {  
  enabled = true  
  min     = 1  
  max     = 2  
  
  policy {  
    cooldown           = "2m"  
    evaluation_interval = "1m"  
  
    check "cpu_allocated_percentage" {  
      source = "prometheus"  
      query  =  
"sum(nomad_client_allocated_cpu{node_class=\"hashistack\"}*100/(nomad_client_unallocated_cpu{node_class=\"hashistack\"}+nomad_client_allocated_cpu{node_class=\"hashistack\"}))/count(nomad_client_allocated_cpu{node_class=\"hashistack\"})"  
  
      strategy "target-value" {  
        target = 70  
      }  
    }  
  }  
}
```

МИГРАЦИЯ НАГРУЗКИ

```
migrate {  
  max_parallel = 2  
  health_check = "checks"  
  min_healthy_time = "15s"  
  healthy_deadline = "5m"  
}
```

```
nomad node drain -enable -yes 46f1
```

```
Alloc "5b4d6db5-3fcb-eb7d-0415-23eefcd78b6a"  
marked for migration  
Alloc "56f770c0-f8aa-4565-086d-01faa977f82d"  
marked for migration
```

```
nomad run webapp.nomad.hcl  
==> Monitoring evaluation "5129bc74"  
      Evaluation triggered by job "webapp"  
      Allocation "5b4d6db5" created: node "46f1c6c4", group  
      "webapp"  
      Allocation "670a715f" created: node "f7476465", group  
      "webapp"  
      Allocation "56f770c0" created: node "46f1c6c4", group "webapp"  
      ...  
      Evaluation status changed: "pending" -> "complete"  
==> Evaluation "5129bc74" finished with status "complete"
```

SERVICE MESH

```
# curl -L -o cni-plugins.tgz  
"https://github.com/containernetworking/plugins/releases/download/v1.3.0/cni-plug  
ins-linux-$( [ $(uname -m) = aarch64 ] && echo arm64 || echo amd64)"-v1.3.0.tgz  
# mkdir -p /opt/cni/bin  
# sudo tar -C /opt/cni/bin -xzf cni-plugins.tgz  
  
# consul agent -dev  
# nomad agent -dev-connect
```

Services 3 total

<input type="text" value="Search"/>		Search Across ▼	Head
✓	consul		
	1 instance		
✓	nomad-client		
✓	Registered via Nomad	1 instance	http
✓	nomad		
✓	Registered via Nomad	3 instances	http, rpc, serf

SERVICE MESH

```
group "ingress" {
  network {
    mode = "bridge"
    port "http" {
      static = 9002 # 1 <---
      to      = 9002 # 2 <---
    }
  }

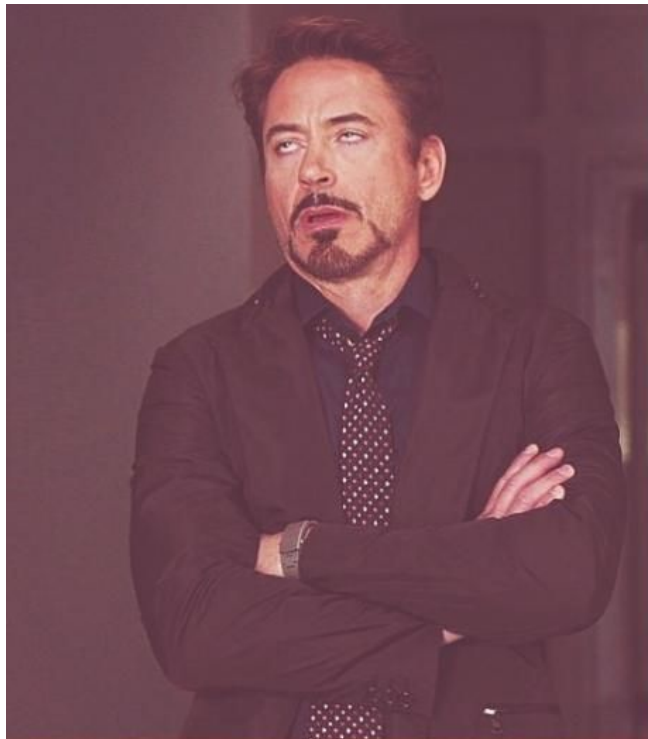
  service {
    name = "http-echo-nginx"
    port = "http" # 3 <---
    connect {
      sidecar_service {
        proxy {
          upstreams {
            # 4 <---
            destination_name = "http-echo-service"
            local_bind_port = 8080
          }
        }
      }
    }
  }
}
```

```
task "nginx" {
  driver = "docker"
  config {
    image = "nginx:1.20"
    volumes =
["local/default.conf:/etc/nginx/conf.d/default.conf"]
  }

  template {
    change_mode = "restart"
    destination = "local/default.conf"
    data        = <<EOH
      upstream app {
        server {{ env
"NOMAD_UPSTREAM_ADDR_http-echo-service" }};
      }
      server {
        listen {{ env "NOMAD_PORT_http" }};
        location / {
          proxy_pass http://app;
        }
      }
    EOH
  }
}
```

SERVICE MESH

```
group "api" {  
  count = 2  
  network {  
    mode = "bridge"  
  }  
  
  service {  
    name = "http-echo-service"  
    port = "5678"  
    connect {  
      sidecar_service {}  
    }  
  }  
  
  task "web" {  
    driver = "docker"  
    config {  
      image = "hashicorp/http-echo"  
      args = ["-text", "'Hello, nomad  
${NOMAD_ALLOC_INDEX}'"]  
    }  
  }  
}
```



SERVICE MESH

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
36b6025c8200	hashicorp/http-echo	"/http-echo -text '!'..."	About a minute ago	Up About a minute		web-1beaea82-3b47-b8cc-01f4-8b28de928925
22f07d726253	hashicorp/http-echo	"/http-echo -text '!'..."	About a minute ago	Up About a minute		web-a02c8212-a95a-139b-ee2d-2d9bac6f68bf
5b8b37a34de2	nginx:1.20	"/docker-entrypoint..."	About a minute ago	Up About a minute		nginx-1df49757-74ac-f1a7-3105-735642cb0cea
09976888ef2e	envoyproxy/envoy:v1.25.1	"/docker-entrypoint..."	About a minute ago	Up About a minute		connect-proxy-http-echo-service-a02c8212-a95a-139b-ee2d-2d9bac6f68bf
2d9bac6f68bf	envoyproxy/envoy:v1.25.1	"/docker-entrypoint..."	About a minute ago	Up About a minute		connect-proxy-http-echo-service-1beaea82-3b47-b8cc-01f4-8b28de928925
2876221bcee3	envoyproxy/envoy:v1.25.1	"/docker-entrypoint..."	About a minute ago	Up About a minute		connect-proxy-http-echo-service-1beaea82-3b47-b8cc-01f4-8b28de928925
8b28de928925	envoyproxy/envoy:v1.25.1	"/docker-entrypoint..."	About a minute ago	Up About a minute		connect-proxy-http-echo-nginx-1df49757-74ac-f1a7-3105-735642cb0cea
dcfe86a64de6	envoyproxy/envoy:v1.25.1	"/docker-entrypoint..."	About a minute ago	Up About a minute		connect-proxy-http-echo-nginx-1df49757-74ac-f1a7-3105-735642cb0cea
5642cb0cea	gcr.io/google_containers/pause-amd64:3.1	"/pause"	About a minute ago	Up About a minute		nomad_init_1df49757-74ac-f1a7-3105-735642cb0cea
62fbb99aed8e	gcr.io/google_containers/pause-amd64:3.1	"/pause"	About a minute ago	Up About a minute		nomad_init_a02c8212-a95a-139b-ee2d-2d9bac6f68bf
c060eed27d0a	gcr.io/google_containers/pause-amd64:3.1	"/pause"	About a minute ago	Up About a minute		nomad_init_a02c8212-a95a-139b-ee2d-2d9bac6f68bf
efa97fc641f0	gcr.io/google_containers/pause-amd64:3.1	"/pause"	About a minute ago	Up About a minute		nomad_init_1beaea82-3b47-b8cc-01f4-8b28de928925

Services 5 total

Search	Search Across	Health Status	Service Type	Sort
consul	1 instance			
nomad-client	Registered via Nomad 1 instance http			
http-echo-nginx	Registered via Nomad 1 instance in service mesh with proxy			
nomad	Registered via Nomad 3 instances http, rpc, serf			
http-echo-service	Registered via Nomad 2 instances in service mesh with proxy			

SERVICE MESH

```
$ curl localhost:9002  
'Hello, nomad 0'
```

```
$ curl localhost:9002  
'Hello, nomad 1'
```

```
# docker inspect nginx-1df49757-74ac-f1a7-3105-735642cb0cea  
env: NOMAD_ADDR_http=192.168.111.34:9002
```

```
# docker exec -it nginx-1df49757-74ac-f1a7-3105-735642cb0cea grep 172 /etc/hosts  
172.26.64.28 62fbb99aed8e
```



SERVICE MESH

```
# iptables-save
```

```
-A POSTROUTING -s 172.26.64.28/32 -m comment --comment "name: \"nomad\" id: \"1df...\"" -j CNI-758...
```

```
-A CNI-758... ! -d 224.0.0.0/4 -m comment --comment "name: \"nomad\" id: \"1df...\"" -j MASQUERADE
```

```
-A CNI-DN-758... -p tcp -m tcp --dport 9002 -j DNAT --to-destination 172.26.64.28:9002
```

```
-A PREROUTING -m addrtype --dst-type LOCAL -j CNI-HOSTPORT-DNAT
```

```
-A CNI-HOSTPORT-DNAT -p tcp -m comment --comment "dnat name: \"nomad\" id: \"1df...\"" -m multiport  
--dports 9002 -j CNI-DN-758...
```


HE XOY SERVICE MESH

```
template {  
  change_mode = "restart"  
  destination = "local/default.conf"  
  data        = <<EOH  
    upstream app {  
      {{ range nomadService "http-echo-service" }}  
        server {{ .Address }}:{{ .Port }};  
      {{ end }}  
    }  
  
    server {  
      listen {{ env "NOMAD_PORT_http" }};  
  
      location / {  
        proxy_pass http://app;  
      }  
    }  
  }  
EOH  
}
```

ROLLING UPDATES

```
update {  
    max_parallel      = 2  
    min_healthy_time = "30s"  
    healthy_deadline = "10m"  
}
```

ROLLBACK UPDATES

```
$ nomad job status echo
```

```
ID           = echo
```

```
Name        = echo
```

```
...
```

Allocations

ID	Node ID	Task Group	Version	Desired	Status	Created	Modified
2347201c	97f7a131	api	14	run	running	2m56s ago	2m13s ago

```
...
```

```
$ nomad job revert echo 13
```

SIDECAR

```
$ nomad job run -var  
LOKI_URL=http://loki:3100/api/prom/push \  
service-mesh-sidecar.hcl
```

```
group "ingress" {  
  ...  
  task "nginx" {  
    driver = "docker"  
    leader = true # <---  
  }  
  ...  
  
  task "promtail" {  
    driver = "docker"  
    service {  
      name = "promtail"  
    }  
    config {  
      image = "grafana/promtail:master"  
      args = [  
        "-config.file",  
        "local/config.yaml",  
      ]  
    }  
  }  
}
```

SIDECAR

```
$ nomad job run -var  
LOKI_URL=http://loki:3100/api/prom/push \  
service-mesh-sidecar.hcl
```

```
template {  
    data = <<EOH  
server:  
    http_listen_port: 9080  
    grpc_listen_port: 0  
  
positions:  
    filename: /tmp/positions.yaml  
  
client:  
    url: ${var.LOKI_URL}  
  
scrape_configs:  
- job_name: system  
  static_configs:  
- targets:  
    - localhost  
  labels:  
    job: nginx  
    __path__: /alloc/logs/nginx*  
EOH  
  
    destination = "local/config.yaml"  
  }  
}  
}
```

HOST VOLUME

```
# Часть конфигурации клиента
host_volume "mysql" {
  path    = "/opt/mysql/data"
  read_only = false
}
```

```
job "mysql-server" {
  datacenters = ["dc1"]
  type        = "service"

  group "mysql-server" {
    count = 1

    volume "mysql" {
      type        = "host"
      read_only   = false
      source       = "mysql" # <---
    }

    task "mysql-server" {
      driver = "docker"

      volume_mount {
        volume        = "mysql"
        destination   = "/var/lib/mysql"
        read_only     = false
      }

      env = {
        "MYSQL_ROOT_PASSWORD" = "password"
      }
    }
  }

  ...
}
```

WEB UI

Jobs

Name	Status	Type	Node Pool	Priority	Groups	Summary
<u>echo</u>	RUNNING	service	default	50	2	<div><div></div></div>
Per page	25					
						1 - 1 of 1

WEB UI

OverviewDefinitionVersionsDeploymentsAllocationsEvaluationsServices

Status: Healthy

CurrentHistorical

5/5 Allocations Running

5 Running

0 Pending

0 Failed

0 Lost

0 Unplaced

Replaced Allocations

0 Rescheduled

0 Restarted

Versions

v13

5

[Latest Deployment →](#)

Successful

5/5 Healthy

[Allocation History ▾](#)

Task Groups

Name ↓	Count	Allocation Status	Volume	Reserved CPU	Reserved Memory	Reserved Disk
api	4	<div><div></div></div>		350 MHz	428 MiB	300 MiB
ingress	1	<div><div></div></div>		400 MHz	460 MiB	300 MiB

Recent Allocations

Show Tasks

ID	Task Group	Created	Modified	Status	Version	Client	Volume	CPU	Memory
d9c88232	api	Aug 05 16:50:34 +0400	10 minutes ago	<div><div></div></div> running	13	97f7a131		<div><div></div></div>	<div><div></div></div>
d16d8ba8	api	Aug 05 14:52:01 +0400	10 minutes ago	<div><div></div></div> running	13	97f7a131		<div><div></div></div>	<div><div></div></div>
9d966e18	api	Aug 05 14:52:01 +0400	10 minutes ago	<div><div></div></div> running	13	97f7a131		<div><div></div></div>	<div><div></div></div>
5b4569e9	api	Aug 05 16:50:34 +0400	10 minutes ago	<div><div></div></div> running	13	97f7a131		<div><div></div></div>	<div><div></div></div>
4322c5fe	ingress	Aug 05 16:58:17 +0400	10 minutes ago	<div><div></div></div> running	13	97f7a131		<div><div></div></div>	<div><div></div></div>

Overview	Definition	Versions	Deployments	Allocations	Evaluations	Services
d16d8ba8	api	Aug 05 14:52:01 +0400	12 minutes ago	■ running	13	97f7a131
/ connect-proxy-http... View Logs						
/ web View Logs						
9d966e18	api	Aug 05 14:52:01 +0400	12 minutes ago	■ running	13	97f7a131
/ connect-proxy-http... View Logs						
/ web View Logs						
5b4569e9	api	Aug 05 16:50:34 +0400	12 minutes ago	■ running	13	97f7a131
/ connect-proxy-http... View Logs						
/ web View Logs						
4322c5fe	ingress	Aug 05 16:58:17 +0400	12 minutes ago	■ running	13	97f7a131
/ connect-proxy-http... View Logs						
/ nginx View Logs						
/ promtail View Logs						
9687474e	ingress	Aug 05 16:57:20 +0400	12 minutes ago	■ complete	12	97f7a131

nginx ■ Running[Go to Task page](#)

Recent Events

Time	Type	Description
Aug 05, '23 16:58:28 +0400	Started	Task started by client

stdout

stderr

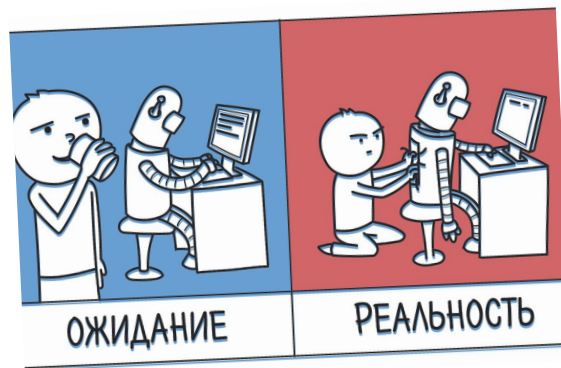
☐ Word Wrap

Head

Tail



```
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
10-listen-on-ipv6-by-default.sh: info: /etc/nginx/conf.d/default.conf differs from the packaged version
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
172.26.64.1 - - [05/Aug/2023:13:10:43 +0000] "GET / HTTP/1.1" 200 17 "-" "curl/8.2.1" "-"
172.26.64.1 - - [05/Aug/2023:13:10:44 +0000] "GET / HTTP/1.1" 200 17 "-" "curl/8.2.1" "-"
172.26.64.1 - - [05/Aug/2023:13:10:44 +0000] "GET / HTTP/1.1" 200 17 "-" "curl/8.2.1" "-"
172.26.64.1 - - [05/Aug/2023:13:10:44 +0000] "GET / HTTP/1.1" 200 17 "-" "curl/8.2.1" "-"
172.26.64.1 - - [05/Aug/2023:13:10:45 +0000] "GET / HTTP/1.1" 200 17 "-" "curl/8.2.1" "-"
172.26.64.1 - - [05/Aug/2023:13:10:45 +0000] "GET / HTTP/1.1" 200 17 "-" "curl/8.2.1" "-"
172.26.64.1 - - [05/Aug/2023:13:10:45 +0000] "GET / HTTP/1.1" 200 17 "-" "curl/8.2.1" "-"
172.26.64.1 - - [05/Aug/2023:13:10:45 +0000] "GET / HTTP/1.1" 200 17 "-" "curl/8.2.1" "-"
172.26.64.1 - - [05/Aug/2023:13:10:46 +0000] "GET / HTTP/1.1" 200 17 "-" "curl/8.2.1" "-"
172.26.64.1 - - [05/Aug/2023:13:10:47 +0000] "GET / HTTP/1.1" 200 17 "-" "curl/8.2.1" "-"
```



Инфраструктура не должна влиять
на архитектуру приложения.
P.S. It depends.

avito.tech

Ереван — 2023

Виталий Лихачев

Senior software engineer



linkedin.com/in/makeitgit



[makeitgit](https://t.me/makeitgit)



Технологическая
сингулярность
приближается.

