Serverless Architectures

Andrey Adamovich

@codingandrey

$ whoami

- developer ...
- devops guy
- trainer devchampions.com Informator ...
- speaker Jfokus JEEConf ...
- co-organizer LatCraft the community of bloody good IT folks DevTernity

Let's start!

What is serverless?

Look ma, no servers!

Google | google.com/datacenters

Serverless is meaningless!

Serverless

> *I do not want to care about the servers!*

Serverless may also mean

- Containerless/Dockerless
- Orchestratorless/Kubernetesless

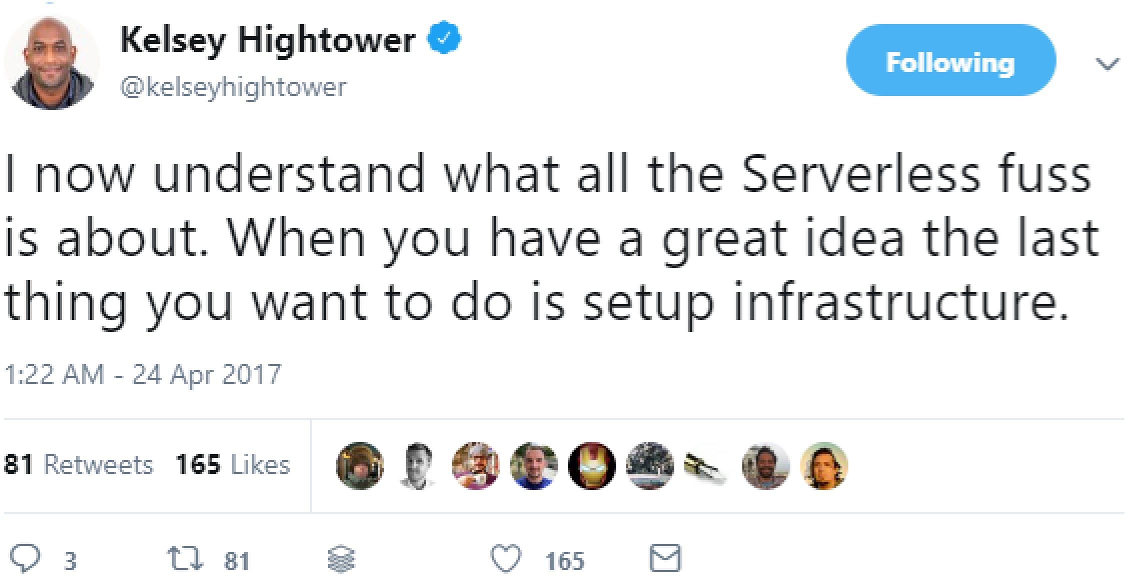
Serverless != Costless

Serverless != FaaS

Serverless != Operationless

Serverless = Servicefull

Idea to production



Kelsey Hightower
@kelseyhightower

I now understand what all the Serverless fuss is about. When you have a great idea the last thing you want to do is setup infrastructure.

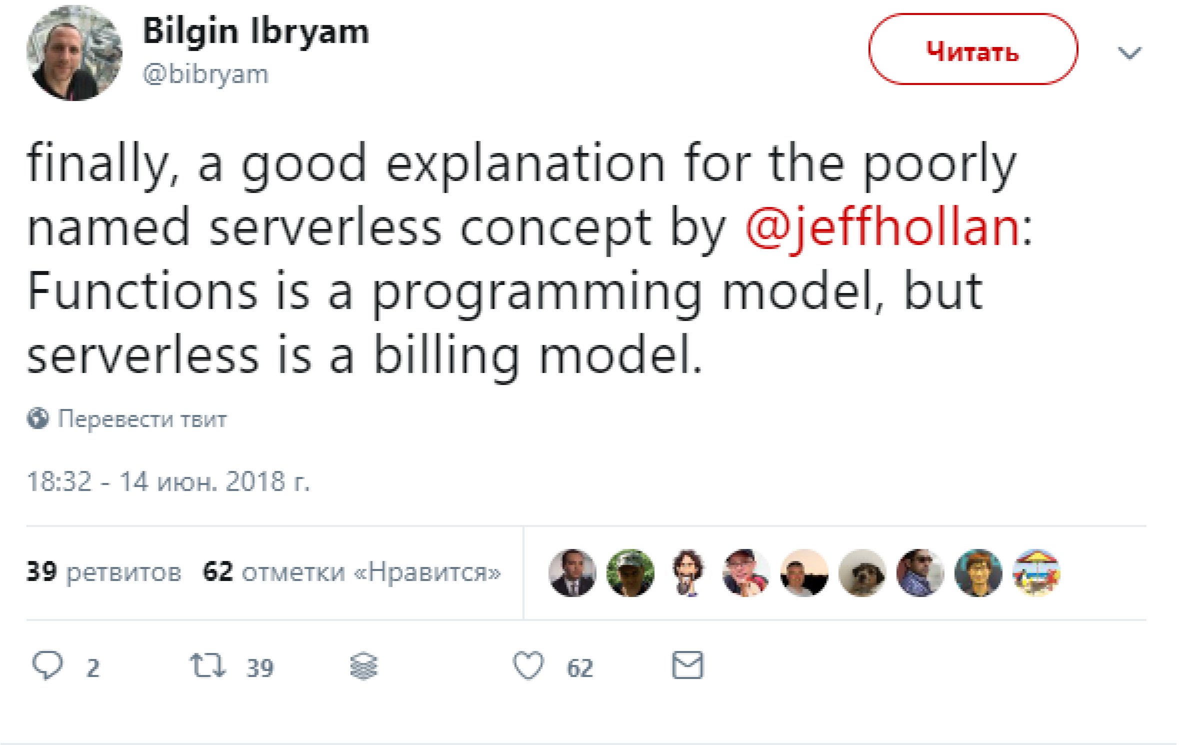
1:22 AM - 24 Apr 2017

81 Retweets 165 Likes

Serverless = IaU

Cheaper than servers

Serverless is a billing model!



Bilgin Ibryam
@bibryam

finally, a good explanation for the poorly named serverless concept by @jeffhollan: Functions is a programming model, but serverless is a billing model.

18:32 - 14 июн. 2018 г.

39 ретвитов 62 отметки «Нравится»

Pay-as-you-go

CapEx vs OpEx

Compute models

- L1: Hardware machines
- L2: Virtual machines and hypervisors
- L3: Containers and orchestrators
- L4: Functions and services

Security issues

- What if my **hardware** is vulnerable? (Meltdown and Spectre)
- What if my **hypervisor** is vulernable?
- What if my **operating system** is vulernable? (Heartbleed)
- What if my **container engine** is vulnerable? (CVE-2019-5736)
- What if my **application** is vulnerable?

Lost professions

FaaS

Just code

```
def handler_name(event, context):
  # implement logic
  return some_value
```

Just code?

Typical questions

- How do I configure DB inside my function?
- How do I add storage into my function?
- How do I install tool X into my function?

You (usually) don't!

Built-in features

- Logging and auditing
- Rate limits
- Auto-scaling
- Security controls
- Multiple versions
- Simplified deployment

Bought-in services

- Database-as-a-service
- Storage-as-a-service
- Messaging-as-a-service
- Function-as-a-service

Evolution

If it were 2005...

- Jetty/Tomcat/Ruby/PHP (on a server)
- MySQL, PostgreSQL, HSQL, Sqlite (on a server)
- File system (on a server)

If it were 2015...

- Jetty/Tomcat/Ruby/PHP (in a container on a server) or PaaS in the cloud
- MySQL, PostgreSQL, HSQL, Sqlite (on a server or in a container on a server) or DaaS in the cloud
- File system (in a volume on a server) or object storage in the cloud

In 2020...

- CDN + FaaS + LB in the cloud
- DaaS in the cloud
- Object storage in the cloud

Benefits

- No baby-sitting with the infrastructure
- Smaller accumulating technical debt
- Minimise operations effort
- Minimise cost of unused resources

FaaS pains

- Memory size
- Archive size
- CPU count
- Cold starts
- Long deployment times
- Vendor lock-in
- Limited runtime verions
- Rate limits on external/internal traffic
- Lots of infrastructure objects to configure

FaaS offerings (generic cloud)

- Google Cloud Plafotm
- Azure
- IBM OpenWhisk
- AWS
- Zeit

FaaS offerings (specialized)

- Cloudflare.com Web Workers (Edge)
- Firebase Functions (BaaS)
- Webtask.io (Chat Bots, Web Hooks)
- Zapier.com (API Integrations)
- Glitch.com (Chat Bots)

Adoption

Every organization that I worked with in the last 3 years runs at least one function in production in the cloud.

Strangler pattern

You can always optimize and migrate slowly!

Not just code

Serverless 1.0 vs Serverless 2.0

Serverless 2.0

- Improved DX (developer experience)
- Support for custom runtimes (with or without Docker)
- Serverless platformas deployable on-prem (e.g. OpenFaaS)
- Local simulation environments
- Integration with Git-based ecosystems
- Automation of infrastructure dependecy management

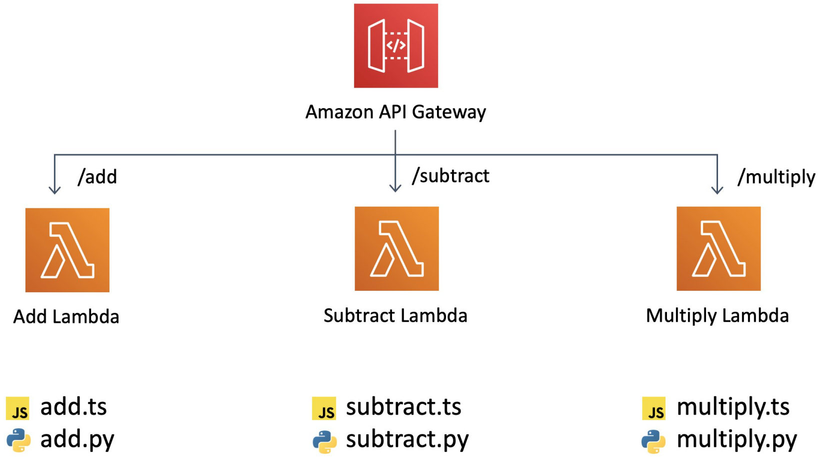
FaaS elements

- Function store
- Gateway (ingress, routing, certificates)
- Security
- Event bus
- GitOps experience

Patterns

Single-purpose function

The Single Purpose Function

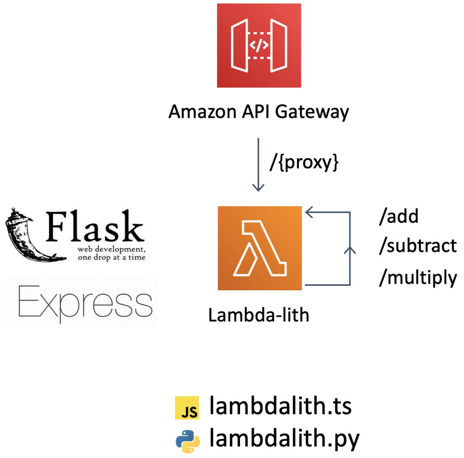


Single-purpose function

Small function that does one thing, does it only and does it well!

Lambdalith

The Lambda-lith



Lambdalith

Function implements several features, most likely responds to various endpoints. It may be a fully-blown web server.

Function vs Service

Function vs Service

F: Fast to start, invoked on demand, uses limited resources, is not guarateed to preserve state.

S: Always there (at least one instance), may have some long-living internal cache, background proceses, usually takes longer to start.

Function vs Container

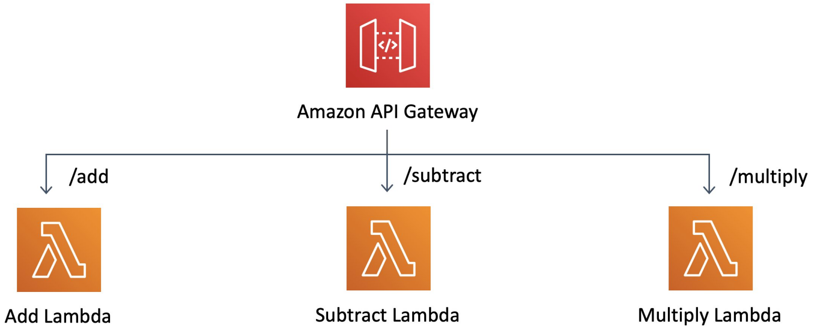
AWS Fargate

On-prem/Custom

- OpenFaaS
- Kubeless
- KNative

Fat lambda

The Fat Lambda



JS fatlambda.ts

fatlambda.py

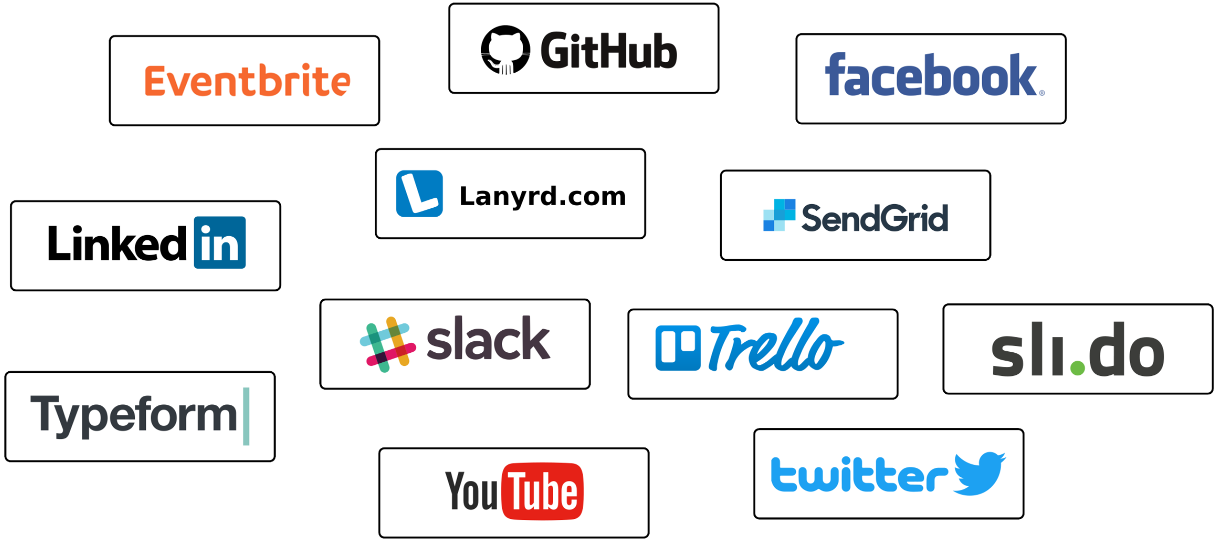
Fat lambda

One code base, many (deployed) functions.

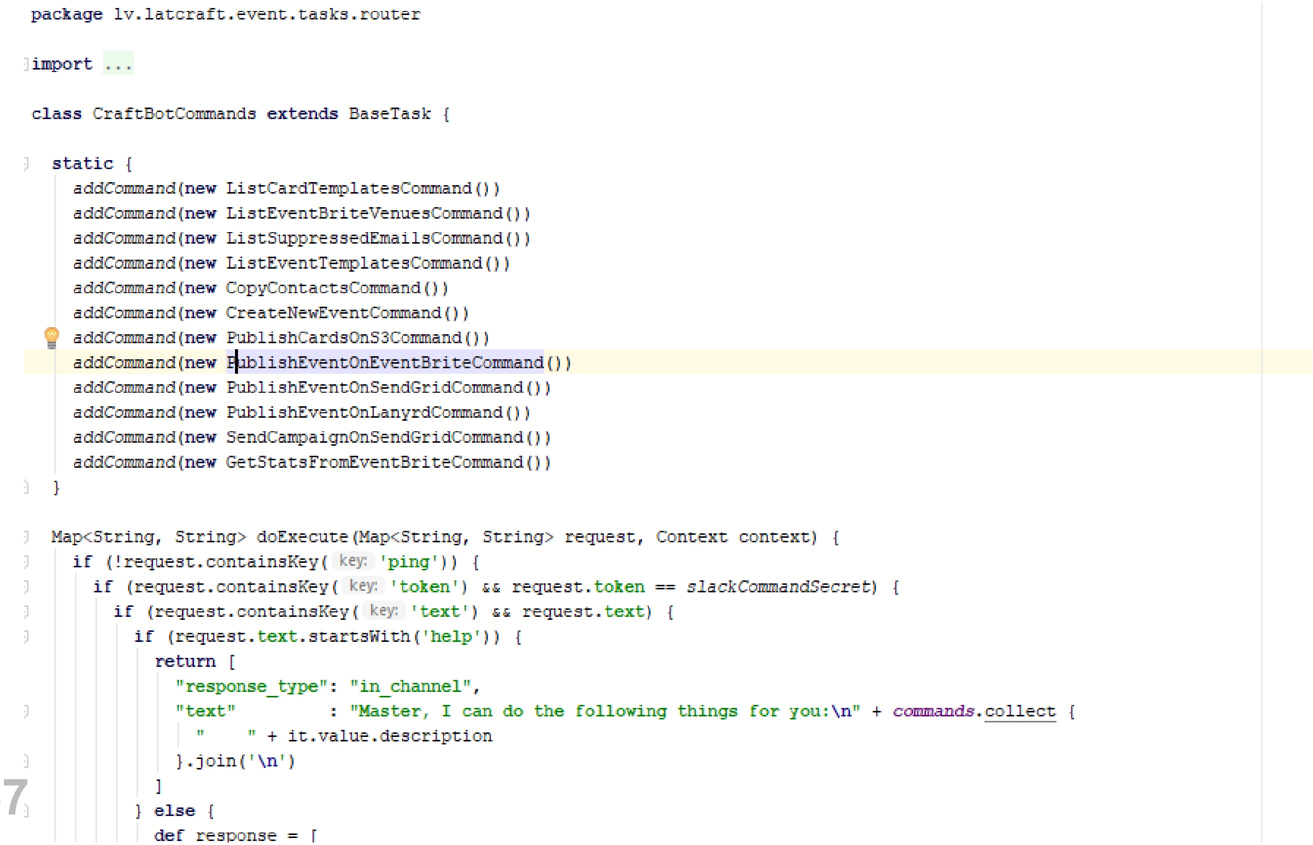
Routing lambda

Function is calling other functions (through event bus or through direct calls).

Craftbot

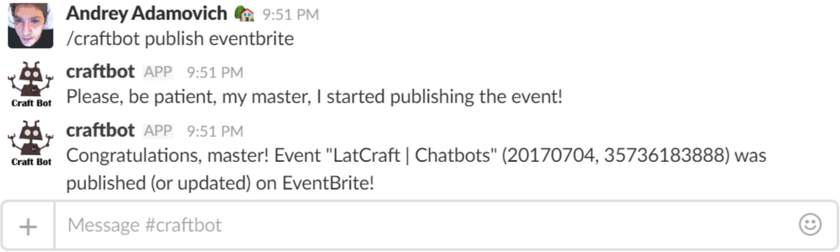

Eventbrite
GitHub
facebook
Lanyrd.com
Linked in
SendGrid
slack
Trello
sli.do
Typeform
You Tube
twitter

Routing function



```
package lv.latcraft.event.tasks.router

import ...

class CraftBotCommands extends BaseTask {

  static {
    addCommand(new ListCardTemplatesCommand())
    addCommand(new ListEventBriteVenuesCommand())
    addCommand(new ListSuppressedEmailsCommand())
    addCommand(new ListEventTemplatesCommand())
    addCommand(new CopyContactsCommand())
    addCommand(new CreateNewEventCommand())
    addCommand(new PublishCardsOnS3Command())
    addCommand(new PublishEventOnEventBriteCommand())
    addCommand(new PublishEventOnSendGridCommand())
    addCommand(new PublishEventOnLanyrdCommand())
    addCommand(new SendCampaignOnSendGridCommand())
    addCommand(new GetStatsFromEventBriteCommand())
  }

  Map<String, String> doExecute(Map<String, String> request, Context context) {
    if (!request.containsKey( key: 'ping')) {
      if (request.containsKey( key: 'token') && request.token == slackCommandSecret) {
        if (request.containsKey( key: 'text') && request.text) {
          if (request.text.startsWith('help')) {
            return [
              "response_type": "in_channel",
              "text"         : "Master, I can do the following things for you:\n" + commands.collect {
                "    " + it.value.description
              }.join('\n')
            ]
          } else {
            def response = [
```

Callback channel

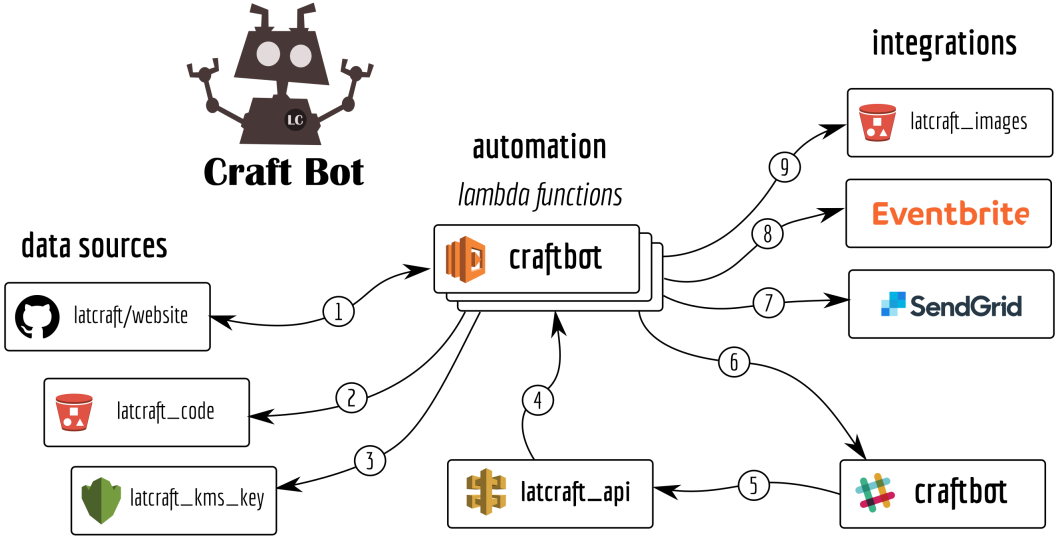


Andrey Adamovich 9:51 PM
/craftbot publish eventbrite

craftbot APP 9:51 PM
Please, be patient, my master, I started publishing the event!

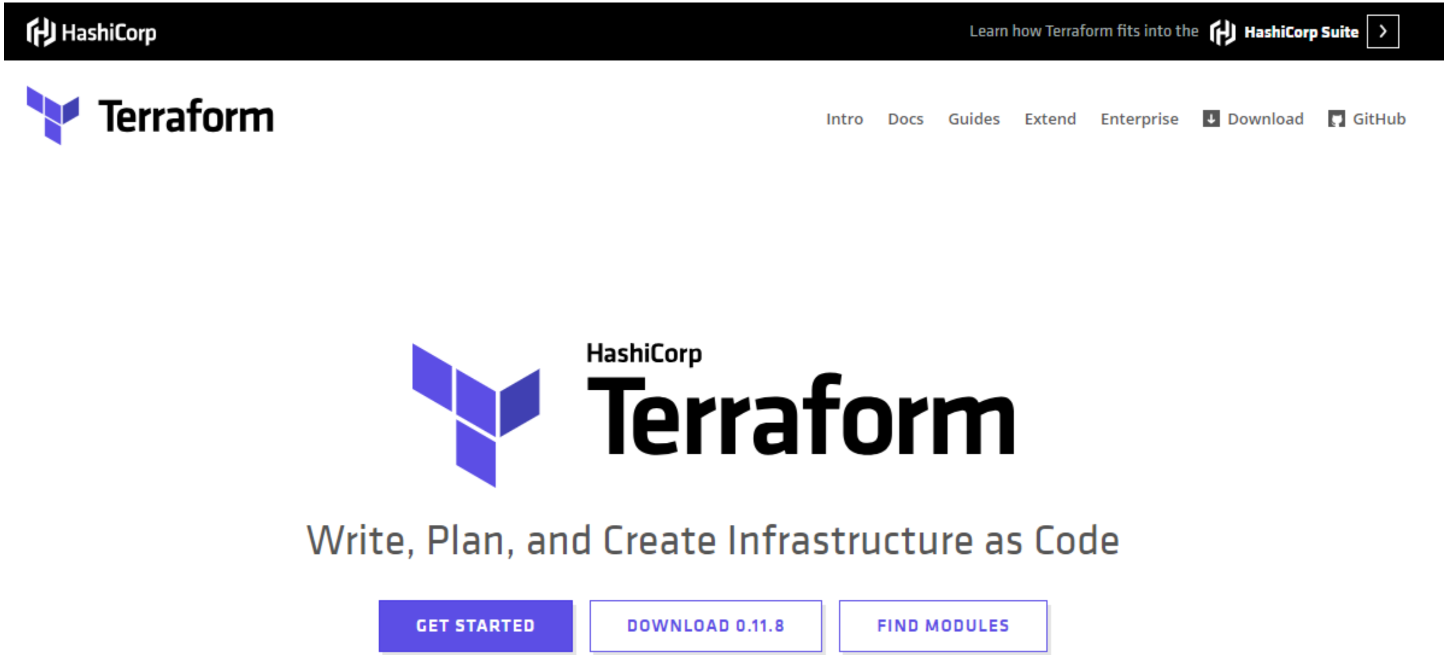
craftbot APP 9:51 PM
Congratulations, master! Event "LatCraft | Chatbots" (20170704, 35736183888) was published (or updated) on EventBrite!

Final setup



1. read/write event data
2. get code and configuration
3. decrypt integration secrets
4. invoke lambda functions
5. send bot commands
6. send statuses and logs
7. send invitation and reminders
8. publish event on eventbrite
9. upload event cards and templates

Terraform



HashiCorp

Learn how Terraform fits into the HashiCorp Suite

Terraform

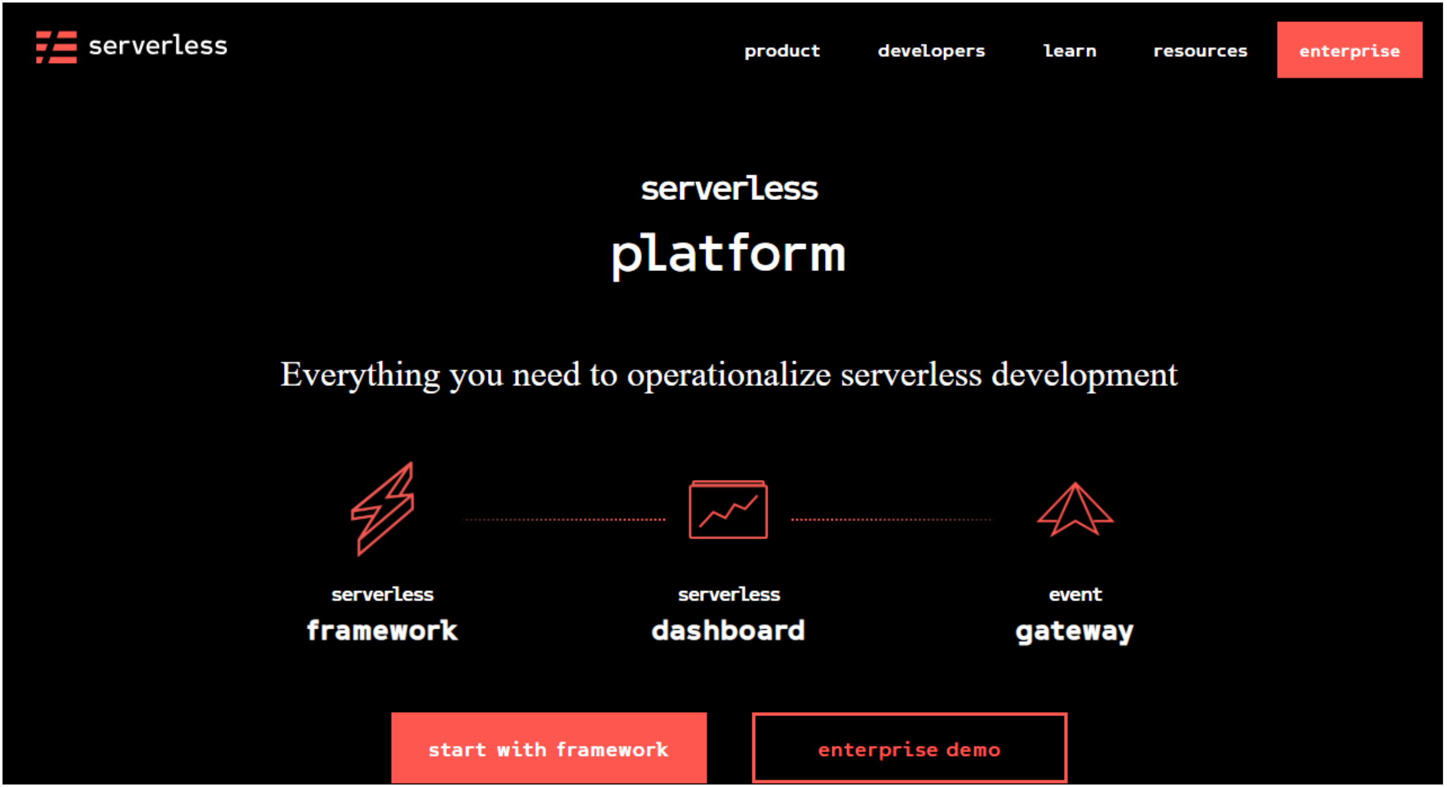
Intro Docs Guides Extend Enterprise Download GitHub

HashiCorp Terraform

Write, Plan, and Create Infrastructure as Code

GET STARTED DOWNLOAD 0.11.8 FIND MODULES

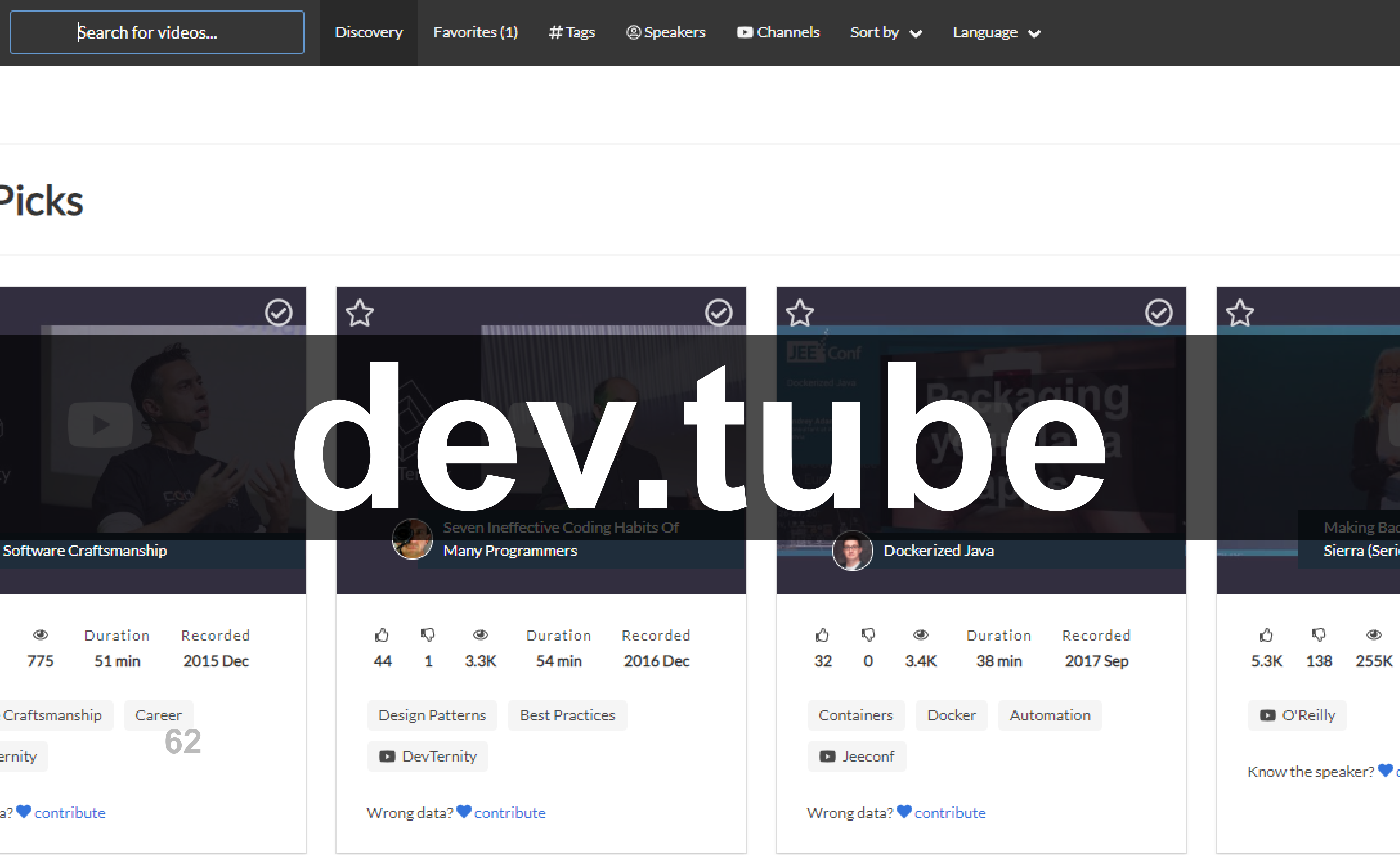
Serverless



serverless

platform

Everything you need to operationalize serverless development



dev.tube

Search for videos...

Discovery | Favorites (1) | # Tags | Speakers | Channels | Sort by | Language

Picks

Software Craftsmanship

👁	Duration	Recorded
775	51 min	2015 Dec

Craftsmanship | Career

ernity

a? 💙 contribute

Seven Ineffective Coding Habits Of Many Programmers

👍	👎	👁	Duration	Recorded
44	1	3.3K	54 min	2016 Dec

Design Patterns | Best Practices

DevTernity

Wrong data? 💙 contribute

Dockerized Java

👍	👎	👁	Duration	Recorded
32	0	3.4K	38 min	2017 Sep

Containers | Docker | Automation

Jeeconf

Wrong data? 💙 contribute

Making Bac

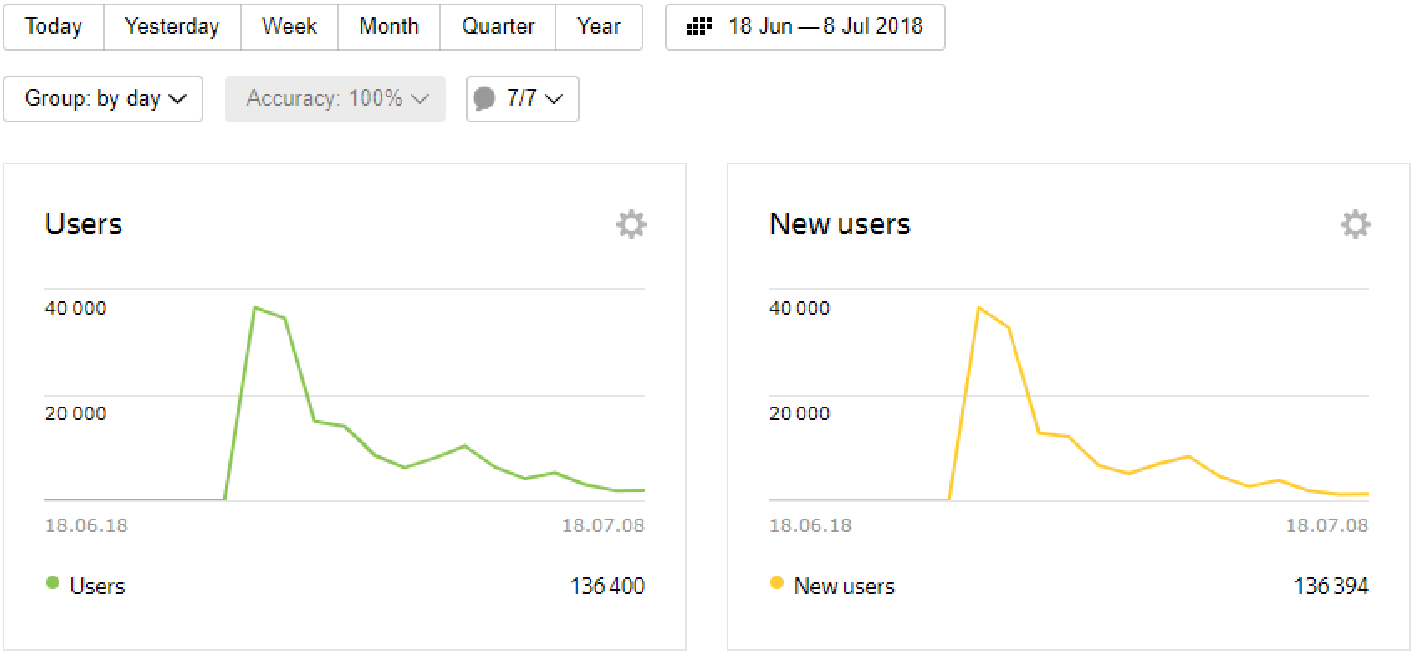
Sierra (Seri

👍	👎	👁
5.3K	138	255K

O'Reilly

Know the speaker? 💙 c

Peak time



18 Jun — 8 Jul 2018

Group: by day

Accuracy: 100%

● Users 136 400

● New users 136 394

Node.js

GCP

Price optimization

- Data is inside a function! HA database attached to public internet traffic is expensive!
- Load balancer is expensive! f1.micro with nginx was cheaper.
- Indexer runs once a day with some heuristics to not over-use Youtube quota.

Conclusion

- Serverless is a platform! PaaS is reborn!
- Serverless is not about lack of servers, it is about focusing on a higher level abstraction!
- Serverless is about bringing value faster by resuing services!
- Servicefull!

Thank you!

$ ping me

@codingandrey

github.com/aadamovich

lv.linkedin.com/in/andreyadamovich

extremeautomation.io

andrey@aestasit.com

Links & References

Projects

- https://serverless.com/
- https://kubeless.io/
- https://fnproject.io/
- https://openfaas.com/
- https://fission.io/
- https://openwhisk.apache.org/
- https://www.iron.io/