

Building Microservices

The Jet Way

Gad Berger
@gadberger

Jet.com
@JetTechnology

What we'll discuss today



What is a microservice, exactly?



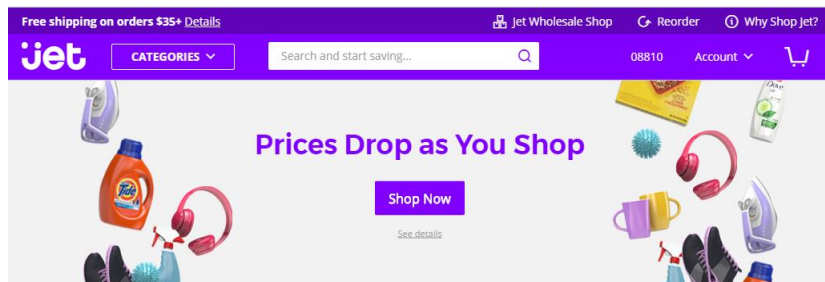
How microservices affect teams



How to write and deploy a microservice



Thoughts on how to handle exceptions



Everyday low prices to start

Prices drop lower as you shop

Fast delivery to your door

2-day delivery on these everyday essentials



Grocery >



Fresh Food >



Household Products >



Health & Beauty >



Baby >



Pet Supplies >

Launched in July of 2015

- “We save you money on the things you already buy”
- Featured: app store & android store
- Over 20k orders per day
- Over 15 Million SKUs
- Walmart ❤️ Jet for \$3.3B in August 2016!

We're hiring!

<http://jet.com/about-us/working-at-jet>

5-year Goal

\$20 Billion in GMV

Do everything at scale

Microservices



tbs HD

tbs HD

Microservice

A service that follows the Single Responsibility Principle.

Perform a single business or system function.

EmailOnOrderShipped

UpdateAccountingDb

Has an input, produces an output.

SendOrderShipmentEmail

Sends Email

NotifyShipmentEmailSent

Benefits

- Scalability
- Distribution of complexity
- Independent releases

Drawbacks

- Latency
- Difficult to troubleshoot
- Increased management overhead

How microservices affect teams

Free to deliver faster, with tighter focus

Settle on a consistent language and pattern

Can introduce some interesting politics

May obfuscate how the system works

What do our services look like?

Define inputs & outputs

```
type Input =  
  | ConfirmationEmailRequest of SendConfirmationEmailRequest  
  
type Output =  
  | ConfirmationEmail of EmailMessage  
  | ConfirmationEmailAlreadySent
```

Define how input transforms to output

```
let handle (input:Input) = async {  
  let confirmationEmail = {  
    Subject = "Thank you"  
    Message = "your order is on its way!"  
    To = user.Email  
  }  
  return <| Some confirmationEmail  
}
```

Define what to do with output

```
let interpret id output =  
  match output with  
  | Some ConfirmationEmailAlreadySent -> async {()} // log it was already sent  
  | Some (ConfirmationEmail(email)) -> async {()} // Send email + save to db
```

Process stream

```
let consume = Kafka.subscribe topic group  
consume (decodeT Input.ConfirmationEmailRequest) handle interpret
```

Consistent
language &
pattern

Handle messages asynchronously



<http://kafka.apache.org>

Topic `order-shipment-email`

Offset	Key	Payload
0	123456789	{ "email" : "order confirmation" }
1	445879613	{ "email" : "order confirmation" }

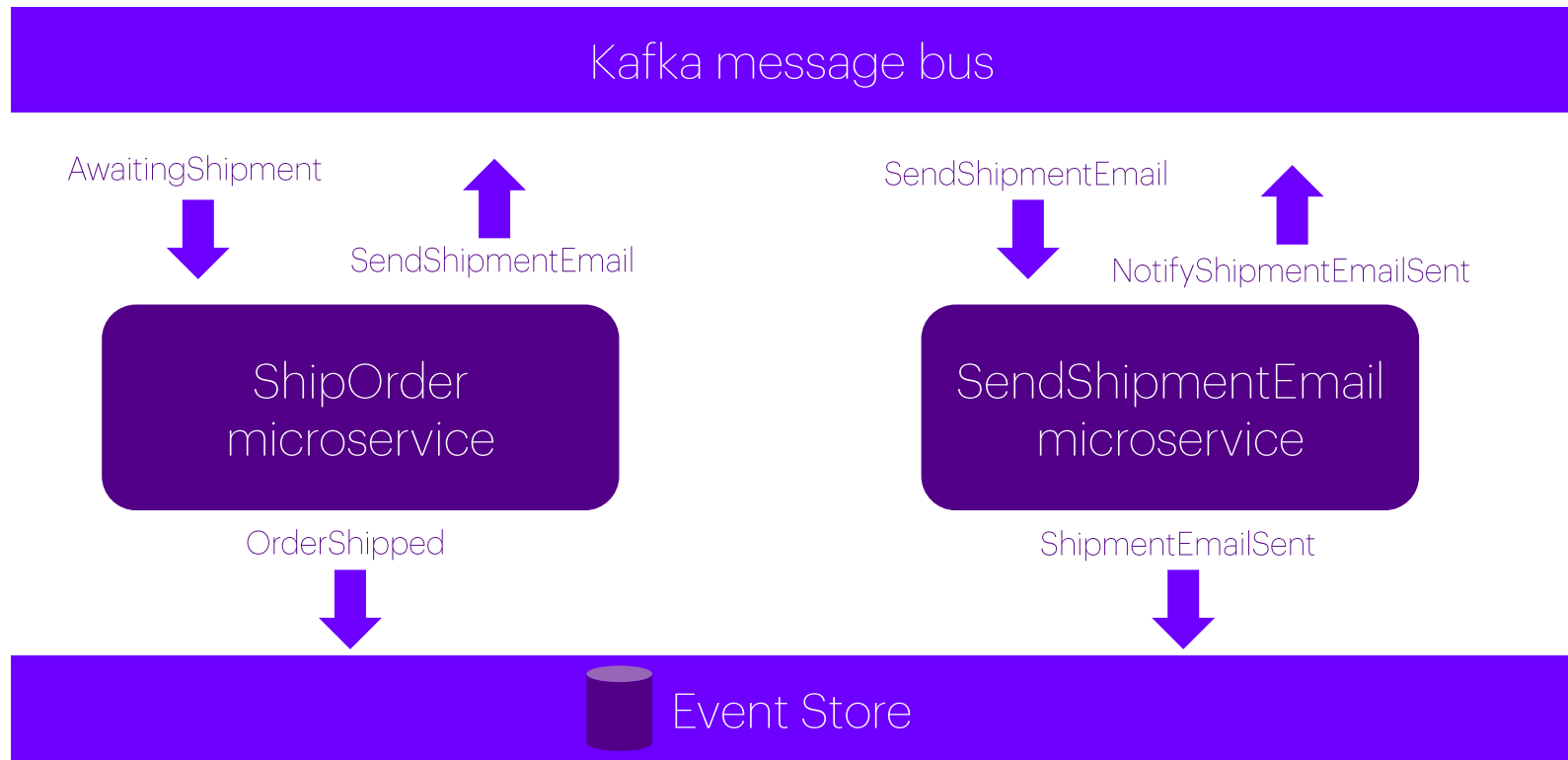


Event Store
<https://geteventstore.com>

Stream `CustomerOrder-123456789`

Event	Type	Payload
0	OrderConfirmed	{ "orderId" : 123, "total" : 100 }
1	OrderShipped	{ "when" : justnow, "to" : "awesome customer" }

How messages flow through microservices



Deployment and Lifecycle

Jenkins - <https://jenkins.io>

Build

Test

Deploy

Nomad - <https://www.nomadproject.io>

Execution
time

Deployment

Configuration

Restarts

Versioning

Scaling

Availability

Subsystem
grouping

Consul - <https://www.consul.io>

Service
discovery

Configuration
+ Secrets

[SERVICES](#)[NODES](#)[KEY/VALUE](#)[ACL](#)[QA ▾](#)[EXPAND](#)

spectrum-blue-eventstore-cluster

6 passing

spectrum-graph-api-resolver

1 passing

spectrum-green-eventstore-cluster

2 passing

spectrum-replicator-checkpoint-writer

1 passing

spectrum-replicator-cluster-to-proj

1 passing

spectrum-superman-listener

2 passing

spectrum-task-command-processor

1 passing

spectrum-task-command-processor

TAGS

No tags


NODES

q2-gog-nmdd-vm2 10.140.1.16

1 passing

Serf Health Status serfHealth

passing


Jenkins

[Gad Berger](#) | [log out](#)


[Jenkins](#) > [Spectrum](#) >
 [ENABLE AUTO REFRESH](#)

[Up](#)
[Status](#)
[Configure](#)
[New Item](#)
[Delete Folder](#)
[People](#)
[Build History](#)
[Edit View](#)
[Project Relationship](#)
[Check File Fingerprint](#)
[Credentials](#)
[Move](#)
[Job Config History](#)

Build Queue

No builds in the queue.



























Build Executor Status



Spectrum

[add description](#)

[All](#)
[Builds](#)
[Deploy-Dev](#)
[Deploy-Prod](#)
[Deploy-QA](#)
[Pipelines](#)
[Releases](#)
[Spectrum](#)
[+](#)

S	W	Categorized - Job ↓	Last Success	Last Failure	Last Duration
+		 .. » Spectrum - API	10 days - #12	21 days - #9	39 sec
-		 .. » Spectrum - Microservices	3 days 11 hr - #46	12 days - #14	5.7 sec
		 spectrum-nomad-services-build	3 days 11 hr - #47	N/A	3 min 14 sec 
		 spectrum-nomad-services-deploy-dev-all	3 days 11 hr - #46	N/A	5.7 sec 
		 spectrum-nomad-services-deploy-dev-single	7 days 12 hr - #69	12 days - #67	2.5 sec 
		 spectrum-nomad-services-deploy-qa-all	3 days 11 hr - #3	N/A	5.6 sec 
		 spectrum-nomad-services-deploy-qa-single	12 days - #15	12 days - #14	4.9 sec 
		 spectrum-nomad-services-release	3 days 11 hr - #8	N/A	3 min 18 sec 
+		 .. » Spectrum - Replicator	7 days 5 hr - #6	10 days - #10	5.1 sec
+		 .. » Spectrum - Website	N/A	6 hr 47 min - #5	7.9 sec

Make sure that you have great logs

Microservices introduce accidental complexity

Distributed systems need actual distribution to flush out bugs

QA environment should be a copy of Production

Log status to a centralized view

Use correlation IDs

Splunk App Search & Reporting Messages Settings Activity **Eng**

Search Pivot Reports Alerts Dashboards

Search & Reporting

Save As Close

Index=qa-gambit | "INFO" | 41dff12f7334534ace7f00291e50c7

Last 1 day 🔍

124 events (9/20/16 2:30:00 AM to 9/21/16 2:30:00 AM) No Event Sampling 📅 Job ⌵ ⌶ ⌷ ⌨️ 👤 Smart Mode ▾

Events (124) Patterns Statistics Visualization

Format Timeline ▾ — Zoom Out + Zoom to Selection × Deselect

1 hour per column

List ▾ Format ▾ 20 Per Page ▾

< Prev 1 2 3 4 5 6 7 Next >

#	Time	Event
>	9/21/16 1:03:55.548 AM	2016-09-21 01:03:55.548 [INFO] SqlProjection\Writing checkout entry for order = 41dff12f7334534ace7f00291e50c7 host: q2-gammish source=C:\data\current\Log\SqProjection.log sourcetype=nlog_format
Selected Fields	9/21/16 1:03:55.470 AM	2016-09-21 01:03:55.470 [INFO] OrderStatusPublisher(Handle): Indexing orders in batch - [(id = 41dff12f7334534ace7f00291e50c7; short_id = "387891391121";)] host: q2-gammish source=C:\data\current\Log\OrderStatusPublisher.log sourcetype=nlog_format
Interesting Fields	9/21/16 1:03:55.142 AM	2016-09-21 01:03:55.142 [INFO] OrderStatusPublisher(Handle): Indexing orders in batch - [(id = 41dff12f7334534ace7f00291e50c7; short_id = "387891391121";), (id = 41dff12f7334534ace7f00291e50c7; short_id = "387891391121";)] host: q2-gammish source=C:\data\current\Log\OrderStatusPublisher.log sourcetype=nlog_format
🔍 Hide Fields	9/21/16 1:03:55.126 AM	2016-09-21 01:03:55.126 [INFO] Gambit_Sql.Orders.fs[ID: upsert_order=41dff12f7334534ace7f00291e50c7] Took 00:00:00.0506405 host: q2-gammish source=C:\data\current\Log\SqGambit.Sql.Orders.fs.log sourcetype=nlog_format
🔍 All Fields	9/21/16 1:03:55.126 AM	2016-09-21 01:03:55.126 [INFO] ThePredealer(Order 41dff12f7334534ace7f00291e50c7) blocked from deal for one or more reasons! host: q2-gammish source=C:\data\current\Log\ThePredealer.log sourcetype=nlog_format
	9/21/16 1:03:54.985 AM	2016-09-21 01:03:54.985 [INFO] FraudCheckStarter\Fraud check score 0.1068648178 - result: PASSED FRAUD CHECK for order (id = 41dff12f7334534ace7f00291e50c7; short_id = "387891391121";). host: q2-gammish source=C:\data\current\Log\FraudCheckStarter.log sourcetype=nlog_format

Be prepared for network faults

Networked systems will fail

Wrap network calls in retry logic

Avoid too much backpressure

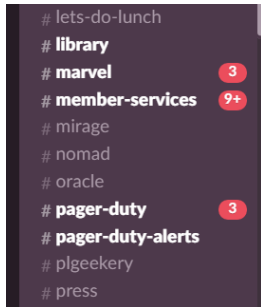
Infinite retries could seize up a system

Instead use dead letter queues and circuit breakers

Chaos test your services

What happens when your service crashes?

```
let rec retry attempts (timeout:int) (f:Async<'a>) = async {
    try
        return! f
    with
        | _ as ex ->
            if attempts <= 0 then raise ex
            do! Async.Sleep(timeout)
            return! retry (attempts - 1) timeout f
}
```



Incredibles BOT 10:42 AM

You can't! It's impossible! I'm far too busy, so ask me now before I again become sane. - Incredibles is alive in (qa)



Shield BOT 11:04 AM

Not all heroes are super. - Shield is alive in (qa)



Shield BOT 12:21 PM

Not all heroes are super. - Shield is alive in (qa)



Gambit BOT 12:28 PM

Bonjour Cherie! At 9/13/2016 4:28:26 PM, Gambit is alive in qa environment, instance id workerRole_IN_0, running #Build# 1.0.1589 Rev# 52f29d96556db6b31404a157cff2dca0974e9cbb on machine - q2-gammshp0 with IPs - ["10.107.0.37"]

In summary

Single
Responsibility
Principle

Consistent
language &
pattern

Message
Bus

Logging

Prepare
for
Network
Faults

For more information - @gadberger

Microservices

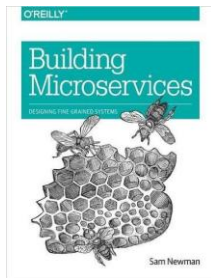
martinfowler.com

microservices.io

Distributed Messaging

kafka.apache.org

Books



Building Microservices
by Sam Newman

Event Sourcing

geteventstore.com

tech.jet.com - Event Sourcing is Awesome!



j

jet