

Making your company cloud-native

The Ticketmatic story

Ruben Vermeersch

Engineer at Ticketmatic

@rubenv



Hi!

Ruben Vermeersch (Ticketmatic)
@rubenv



ticketmatic

Home Explore Pricing Docs Contact

Log In

Powerful ticketing & marketing software

The system of choice by leading venues

"We are very pleased about the flexibility and reliability of Ticketmatic, and about the collaboration with the Ticketmatic team."

Kris Mouchaers
Botanique, Brussels

tm Kies tickets - Cactusfestival x Ruben

Secure https://apps.ticketmatic.com/widgets/cf/addtickets?event=60...



KIES AANTAL TICKETS

Dagticket - vrijdag 7 juli Prijs: € 47,00 + € 2,00 fee	<input type="text" value="0"/> +
Dagticket - zaterdag 8 juli Prijs: € 47,00 + € 2,00 fee	<input type="text" value="0"/> +
Dagticket - zondag 9 juli Prijs: € 47,00 + € 2,00 fee	<input type="text" value="0"/> +
Combiticket - vrijdag 7 / zaterdag 8 juli Prijs: € 80,00 + € 2,00 fee	<input type="text" value="0"/> +
Combiticket - zaterdag 8 / zondag 9 juli Prijs: € 80,00 + € 2,00 fee	<input type="text" value="0"/> +
Combiticket - vrijdag 7 & zondag 9 juli Prijs: € 80,00 + € 2,00 fee	<input type="text" value="0"/> +
Weekendticket - vrijdag 7 / zaterdag 8 / zondag 9 juli Prijs: € 105,00 + € 2,00 fee	<input type="text" value="0"/> +
Campingticket Prijs: € 13,00 + € 2,00 fee	<input type="text" value="0"/> +

◀ Vorige Volgende ▶

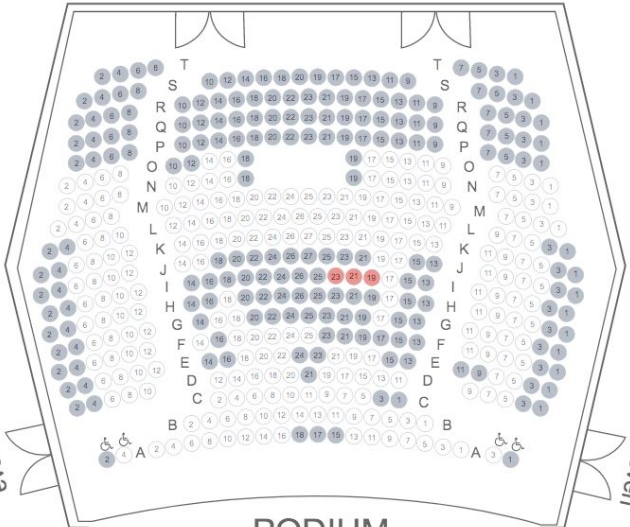
tm Plaatsen - CC Leopoldsburg x Ruben

Secure https://apps.ticketmatic.com/widgets/cc_leopoldsburg/addtic...

Klik op een lege plaats op het zaalplan om de geselecteerde plaatsen te verplaatsen.

?
+
-

even oneven



even oneven

PODIUM

◀ Vorige Volgende ▶

tm Order detail - Ticketmatic x Ruben

Secure https://apps.ticketmatic.com/#/orders/834409

tm CoreOS Fest Theater Contacts Orders Events Settings Help center Support Ruben

< Order #834409 - Web - Rob Doe 17 mei 2017 20:10:38 Actions Log E-mail Print

Tickets (19) List (19) Bundles (3) Add products Add tickets Add bundles

Jeugdtheaterserie
€ 37,50 - 5 tickets

Rob Doe 984984

zo 05/11/2017 15:00 - Het Laagland	Rang 1 - Jeugdtheaterserie XL (volwassene) - Rij 6 - Stoel 11	€ 7,50
vr 29/12/2017 15:30 - TG Graasland	Rang 1 - Jeugdtheaterserie XL (volwassene) - Rij 6 - Stoel 12	€ 7,50
vr 12/01/2018 19:30 - Beumer & Drost	Rang 1 - Jeugdtheaterserie XL (volwassene) - Rij 5 - Stoel 10	€ 7,50
zo 04/03/2018 15:00 - Theater WEI	Rang 1 - Jeugdtheaterserie XL (volwassene) - Rij 5 - Stoel 11	€ 7,50
za 31/03/2018 19:00 - Wie Walvis	Rang 1 - Jeugdtheaterserie XL (volwassene) - Rij 5 - Stoel 12	€ 7,50

Jeugdtheaterserie
€ 37,50 - 5 tickets

Alice Black 919849


zo 05/11/2017 15:00 - Het Laagland	Rang 1 - Jeugdtheaterserie XL (volwassene) - Rij 6 - Stoel 12	€ 7,50
vr 29/12/2017 15:30 - TG Graasland	Rang 1 - Jeugdtheaterserie XL (volwassene) - Rij 6 - Stoel 11	€ 7,50
vr 12/01/2018 19:30 - Beumer & Drost	Rang 1 - Jeugdtheaterserie XL (volwassene) - Rij 5 - Stoel 11	€ 7,50
zo 04/03/2018 15:00 - Theater WEI	Rang 1 - Jeugdtheaterserie XL (volwassene) - Rij 5 - Stoel 9	€ 7,50
za 31/03/2018 19:00 - Wie Walvis	Rang 1 - Jeugdtheaterserie XL (volwassene) - Rij 5 - Stoel 11	€ 7,50

Jeugdtheaterserie
€ 27,50 - 5 tickets

Ricky Bobby 984983


zo 05/11/2017 15:00 - Het Laagland	Rang 1 - Jeugdtheaterserie XL (kind) - Rij 6 - Stoel 10	€ 5,50
vr 29/12/2017 15:30 - TG Graasland	Rang 1 - Jeugdtheaterserie XL (kind) - Rij 6 - Stoel 10	€ 5,50
vr 12/01/2018 19:30 - Beumer & Drost	Rang 1 - Jeugdtheaterserie XL (kind) - Rij 5 - Stoel 12	€ 5,50
zo 04/03/2018 15:00 - Theater WEI	Rang 1 - Jeugdtheaterserie XL	€ 5,50

Rob Doe id [Edit](#)

 Main street 1
6548 AW Halperdam
Nederland [\(Show on map\)](#)
bobdoe@example.com

Payment & delivery

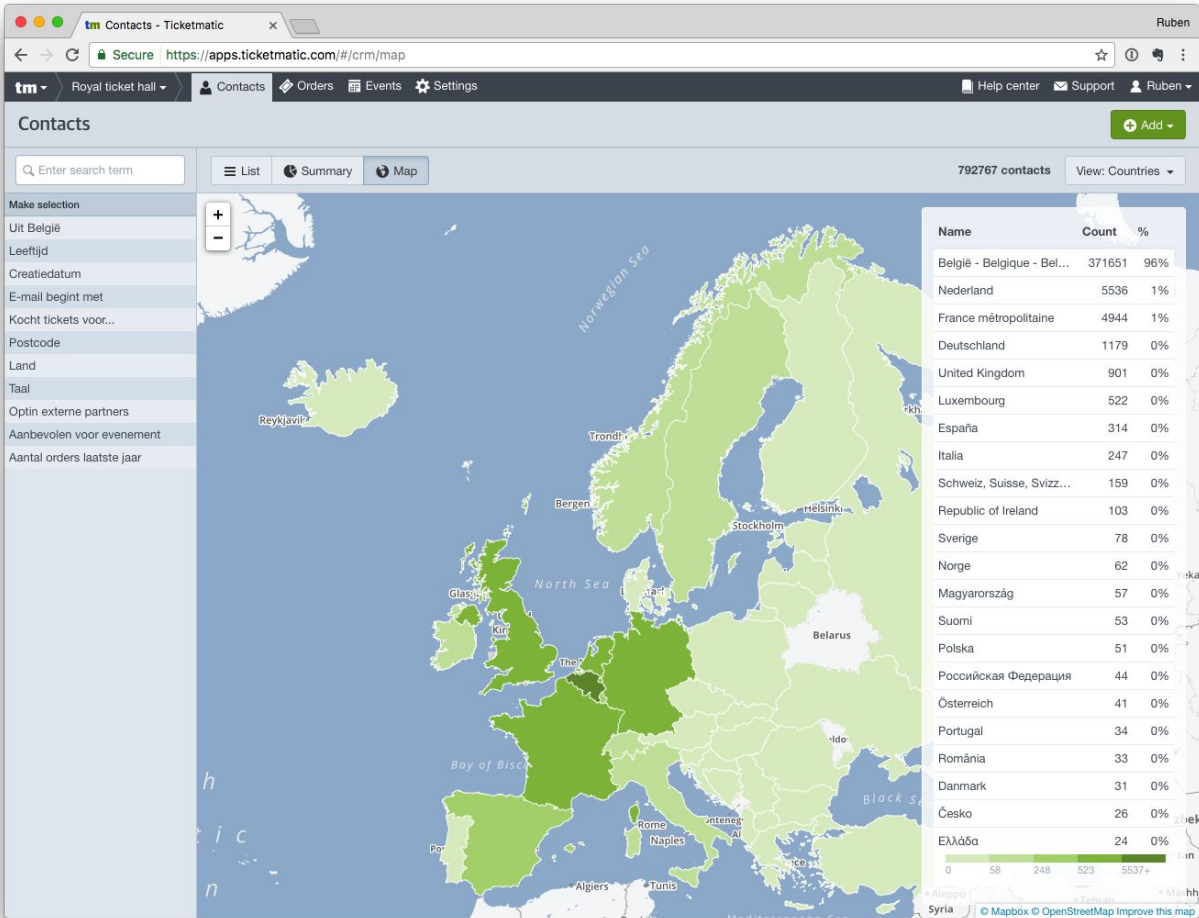
Payment	iDeal
Delivery	E-ticket(s)
Delivery status	Delivered

 **€ 217,00 paid**

Total	€ 217,00
Already paid	€ 217,00

Custom fields [Edit](#)

Opmerking



tm Seating plan - Ticketmatic

Secure <https://apps.ticketmatic.com/#/settings/seatplans/10495/zone/1>

tm Toneelhuis Contacts Orders Events Settings Translate Help center Support Ruben

Edit seating plan: Bourla TM3 Published In use Save

Layout
 Seats
 Ranks
 Priorities
 Seat description
 Lock templates

● (407 - 44%)
 ● (85 - 9%)
 ● (304 - 33%)
 ● (0 - 0%)
 ● (0 - 0%)
 ● (135 - 15%)
 ● (0 - 0%)
 ● (0 - 0%)

Today

How we went cloud-native (Kubernetes)

Where we come from

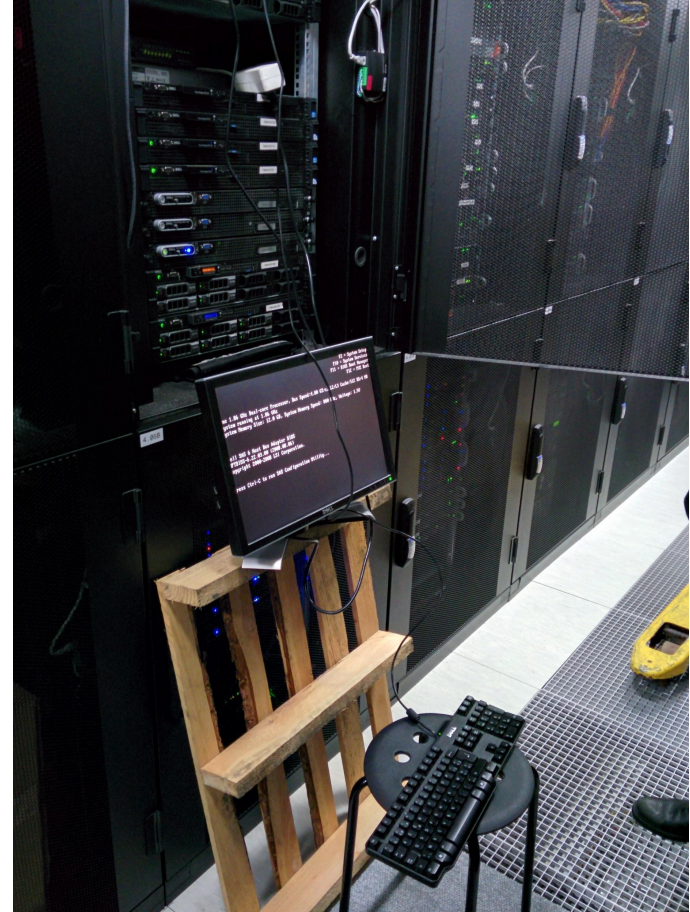
Own servers

Traditional web application architecture

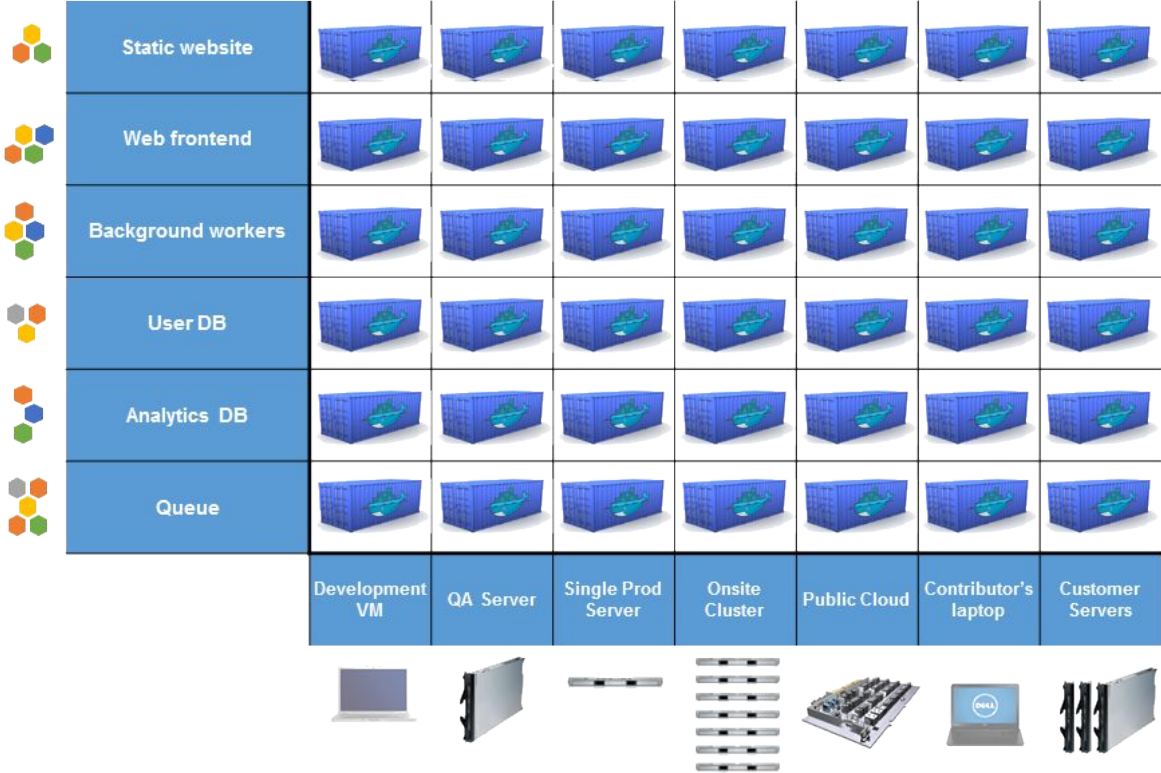
Management pain

→ deployment pain

→ multiple environments



Docker to the rescue!



Or not quite...

Deployment?

Multiple machines?

Orchestration?

Just a packaging format, still all manual operations!

Or not quite...



Our story

Experiences, lessons learned, what you can do

Get the cluster running

We ♥  CoreOS

Now what?

Confusion is normal!

Infrastructure is only part of the story

What about...

Operations?
Development?

Operations

Kubectl is great

But super low-level

Like building an airplane with just a screwdriver

Where's the workflow?

Operational difficulties

Dev → QA → Production

Reliable, repeatable, no room for errors

Handle multiple environments with one set of configuration

kube-appdeploy

Manifest templates + variables

<https://github.com/rubenv/kube-appdeploy>

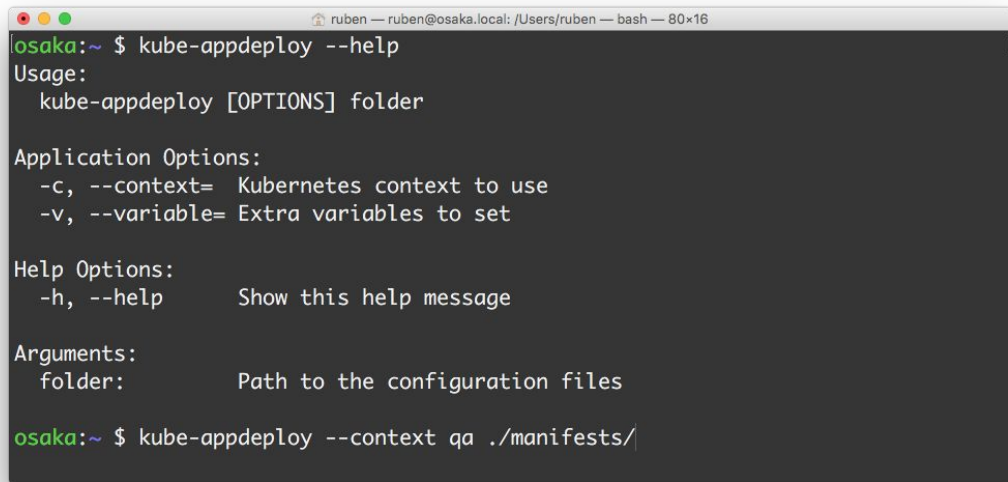
Simple example

```
ruben — ruben@osaka.local: /Users/ruben/Documents/Google Drive/CoreOS Fest Talk/manifests — Python · nvim simple.yaml — 80×24
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: someservice
spec:
  template:
    metadata:
      labels:
        role: someservice
    spec:
      containers:
      - name: someservice
        image: ...
        env:
        - name: ENVIRONMENT
          value: {{ .Variables.environment }}
~
~
~
~
~
simple.yaml 10,5 All
"simple.yaml" 16L, 310C
```

Conditional deployments

```
ruben — ruben@osaka.local: /Users/ruben/Documents/Google Drive/CoreOS Fest Talk/manifests — Python • nvim conditional.yaml — 80x24
{{ if .Variables.internal_env }}
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: fakeaws
spec:
  template:
    metadata:
      labels:
        role: fakeaws
    spec:
      containers:
      - name: fake-dynamodb
        image: docker.io/peopleperhour/dynamodb
        ports:
        - name: dynamo
          containerPort: 8000
{{ end }}
~
~
~
~
conditional.yaml [R0] 5,1 All
```

Run it

A terminal window with a dark background and light text. The window title is "ruben — ruben@osaka.local: /Users/ruben — bash — 80x16". The prompt is "osaka:~". The user has entered "kube-appdeploy --help". The output shows the usage and options for the command. The prompt is "osaka:~" again, and the user has entered "kube-appdeploy --context qa ./manifests/".

```
osaka:~ $ kube-appdeploy --help
Usage:
  kube-appdeploy [OPTIONS] folder

Application Options:
  -c, --context= Kubernetes context to use
  -v, --variable= Extra variables to set

Help Options:
  -h, --help      Show this help message

Arguments:
  folder:          Path to the configuration files

osaka:~ $ kube-appdeploy --context qa ./manifests/
```

Loads variables from **variables.yaml** and/or from CLI

But can be used as a library with a variety of sources

What did we gain?

One set of manifests for the whole product
Takes the fat finger out of the loop

Kubernetes Helm

Similar in concept, much more extensive

Probably the way forward

<https://github.com/kubernetes/helm>

Pod presets

Inject configuration into all pods

Moves all variability into one place

Doesn't help you with conditional deployments

<https://kubernetes.io/docs/tasks/inject-data-application/podpreset/>

Progress!

Initial focus of Kubernetes project was ground-up infrastructure

Now making progress on the operational side

Development

Don't make developers
miserable

Theory differs from reality

You should build your services in isolation

Usually doesn't work out that way

Need the whole environment

It's all in Docker containers

Horrible dev cycle

Livereload for cloud services?

Towards a better development environment

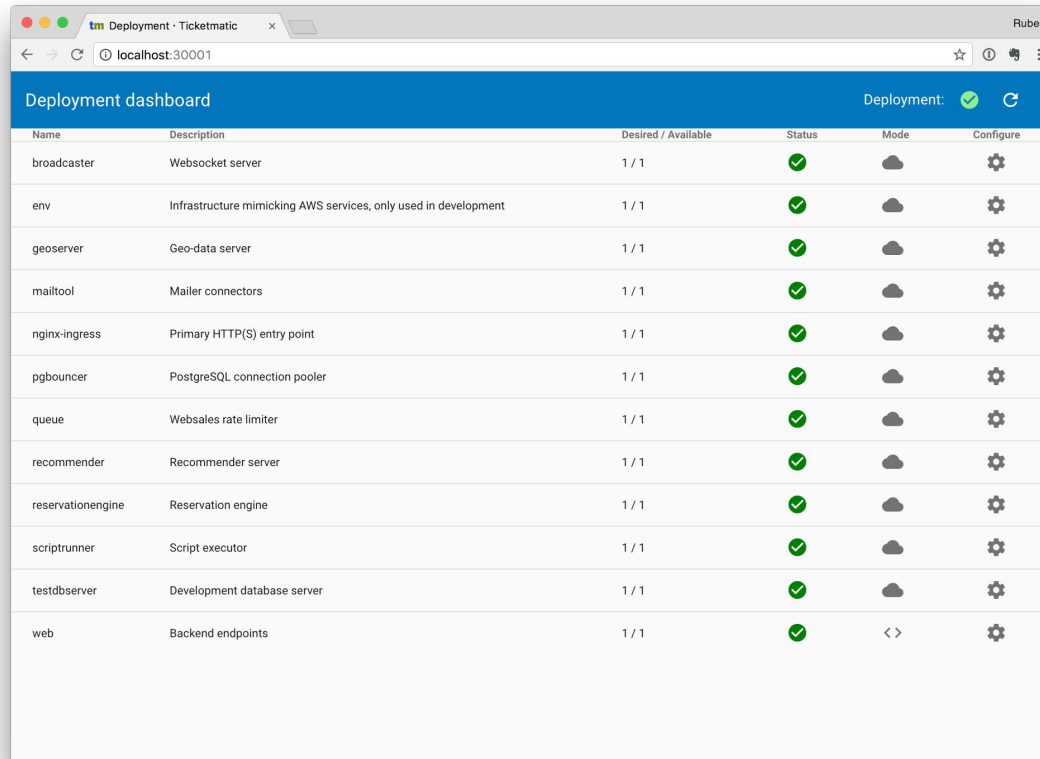
Equivalent of `git pull` and `start`.

Without extra overhead for those who don't care.

Developers + Ticketmatic

devmatic

Only what is needed



The screenshot shows a web browser window with the URL `localhost:30001`. The page title is "Deployment dashboard" and the user is identified as "Ruben". The dashboard displays a table of services with their deployment status. All services are shown as "Desired / Available" (1 / 1) and "Status" (green checkmark).

Name	Description	Desired / Available	Status	Mode	Configure
broadcaster	Websocket server	1 / 1	✓	☁	⚙
env	Infrastructure mimicking AWS services, only used in development	1 / 1	✓	☁	⚙
geoserver	Geo-data server	1 / 1	✓	☁	⚙
mailtool	Mailer connectors	1 / 1	✓	☁	⚙
nginx-ingress	Primary HTTP(S) entry point	1 / 1	✓	☁	⚙
pgbouncer	PostgreSQL connection pooler	1 / 1	✓	☁	⚙
queue	Websales rate limiter	1 / 1	✓	☁	⚙
recommender	Recommender server	1 / 1	✓	☁	⚙
reservationengine	Reservation engine	1 / 1	✓	☁	⚙
scriptrunner	Script executor	1 / 1	✓	☁	⚙
testdbserver	Development database server	1 / 1	✓	☁	⚙
web	Backend endpoints	1 / 1	✓	<>	⚙

Logs

The screenshot shows a web browser window with the Ticketmatic deployment dashboard. The dashboard lists various services, and the 'pgbouncer' service is selected. A modal window displays the logs for the 'pgbouncer-504666810-xtvqd' pod.

Deployment dashboard Deployment: ✔ ↻

Name	Description	Desired / Available	Status	Mode	Configure
broadcaster	Websocket server	1 / 1	✔	☁	⚙
env					⚙
geoserver					⚙
mailtool					⚙
nginx-ingress					⚙
pgbouncer					⚙
queue					⚙
recommender					⚙
reservationengine					⚙
scriptrunner					⚙
testdbserver					⚙
web					⚙

pgbouncer

Pods:

pgbouncer-504666810-xtvqd (Running)

Name	Status	Image	Restart Count	Mode	Reload	Console
pgbouncer	✔ Ready	eu.gcr.io/tm-docker/pgbouncer-build-201704251202	0	☁		🖥

Logs: ✕ 📄 ⬇ ☰

```
pgbouncer-504666810-xtvqd: 2017/05/24 16:35:49 dial tcp: i/o timeout
pgbouncer-504666810-xtvqd: 2017/05/24 16:36:09 dial tcp 10.3.0.228:5432: i/o timeout
pgbouncer-504666810-xtvqd: 2017-05-24 16:36:11.167 13 LOG file descriptor limit: 1048576 (H:1048576), max_client
pgbouncer-504666810-xtvqd: 2017-05-24 16:36:11.186 13 LOG listening on 0.0.0.0:5432
pgbouncer-504666810-xtvqd: 2017-05-24 16:36:11.186 13 LOG listening on *:5432
pgbouncer-504666810-xtvqd: 2017-05-24 16:36:11.195 13 LOG listening on unix:/tmp/.s.PGSQL.5432
pgbouncer-504666810-xtvqd: 2017-05-24 16:36:11.195 13 LOG process up: pgbouncer 1.6.1, libevent 2.0.22-stable (ep
pgbouncer-504666810-xtvqd: 2017-05-24 16:37:11.186 13 LOG Stats: 9 req/s, in 3068 b/s, out 73146 b/s,query 9713
pgbouncer-504666810-xtvqd: 2017-05-24 16:38:11.187 13 LOG Stats: 44 req/s, in 4597 b/s, out 25222 b/s,query 527
pgbouncer-504666810-xtvqd: 2017-05-24 16:39:11.188 13 LOG Stats: 0 req/s, in 44 b/s, out 14 b/s,query 7644 us
```

Shell

The screenshot shows a web browser window with the URL `localhost:30001`. The page title is "Deployment dashboard" and the user is logged in as "Ruben". The dashboard displays a table of services with columns for Name, Description, Desired / Available, Status, Mode, and Configure. A modal window titled "Console: web" is open, showing the command `kubectl --namespace=ticketmatic exec web-3503662786-072t5 -c web -ti sh`.

Name	Description	Desired / Available	Status	Mode	Configure
broadcaster	Websocket server	1 / 1	✓	☁	⚙
env	Infrastructure mimicking AWS services, only used in development	1 / 1	✓	☁	⚙
geoserver	Geo-data server	1 / 1	✓	☁	⚙
mailtool	Mailer connectors	1 / 1	✓	☁	⚙
nginx-ingress	Primary HTTP(S) entry point	1 / 1	✓	☁	⚙
pgbouncer	PostgreSQL connection pooler	1 / 1	✓	☁	⚙
queue	Websales rate limiter	1 / 1	✓	☁	⚙
recommender	Recommender service	1 / 1	✓	☁	⚙
reservationengine	Reservation engine	1 / 1	✓	☁	⚙
scriptrunner	Script executor	1 / 1	✓	☁	⚙
testdbserver	Development database server	1 / 1	✓	☁	⚙
web	Backend endpoints	1 / 1	✓	<>	⚙

```
kubectl --namespace=ticketmatic exec web-3503662786-072t5 -c web -ti sh
```

Source mode

The screenshot shows a web browser window titled "Deployment · Ticketmatic" with the URL "localhost:30001". The page displays a "Deployment dashboard" with a table of services. A modal window is open over the "queue" service, titled "Configuration: queue / queue". The modal contains the following text:

< > Using source code version

Run in development mode

This container contains a Go project

You must run `go get -t git.ticketmatic.com/tm/queue/...` before this container will work.

SAVE

Name	Description	Desired / Available	Status	Mode	Configure
broadcaster	Websocket server	1 / 1	✓	☁	⚙
env	Infrastructure mimicking AWS services, only used in development	1 / 1	✓	☁	⚙
geoserver	Geo-data server	1 / 1	✓	☁	⚙
mailtool	Mailer con			☁	⚙
nginx-ingress	Primary H			☁	⚙
pgbouncer	PostgreSQ			☁	⚙
queue	Websales			☁	⚙
recommender	Recomm			☁	⚙
reservationengine	Reservati			☁	⚙
scriptrunner	Script exe			☁	⚙
testdbserver	Development database server	1 / 1	✓	☁	⚙
web	Backend endpoints	1 / 1	✓	< >	⚙

How it works: deploy-manager

Dashboard = kube-appdeploy + Kubernetes client-go

Self-driving deployment

How it works: source mode

```
ruben — ruben@osaka.local: /Users/ruben/Documents/Google Drive/CoreOS Fest Talk/manifests — nvim devmode.yaml — 80x24
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: queue
  annotations:
    description: "Websales rate limiter"
  dev: |
    {
      "queue": {
        "mode": "go",
        "goPath": "git.ticketmatic.com/tm/queue/..."
      }
    }
spec:
  template:
    metadata:
      labels:
        role: queue
    spec:
      containers:
        - name: queue
devmode.yaml 1,1 Top
```

How it works: source mode

```
template:
  metadata:
    labels:
      role: queue
  spec:
    containers:
      - name: queue
      {{- if ismode . "queue" "queue" "image" }}
        image: {{ .Variables.images.queue }}
      {{ else }}
        image: docker.io/rubenv/gorerun:latest
        command:
          - gorerun
          - -pkg
          - git.ticketmatic.com/tm/queue
          - /go/src/git.ticketmatic.com/tm/queue/bin/queue/main.go
        volumeMounts:
          - name: go
            mountPath: /go/src
      {{ end }}
    env:
      - name: TICKETMATIC_ENV
```

devmode.yaml 34,1 32%

Results

Trouble-free uptake

Big gains in productivity due to self-driving deployment

But...

We shouldn't have to build this

Rounding up

Want to go cloud native?

Think about all your stakeholders!

Invest in great processes and experiences

Realize that it's normal to be confused

Been a hell of a ride!

Don't let this discourage you

Still the most enjoyable platform we've used

As a project / community

Working from the ground up is fantastic

Undefined quickly becomes confusing...

Or worse: repeated efforts

What can we standardise upon?

Thanks!

Ruben Vermeersch
@rubenv

Slides (soon): <https://rocketeer.be/>
Always looking for great people at Ticketmatic!