PULSAR SUMMIT 2020

Connecting millions of users at CHATROULETTE

With Apache Pulsar at the core

Co-presented by:

- Gabriel Volpe
- Tamer Abdulazim

Why we needed Pulsar?

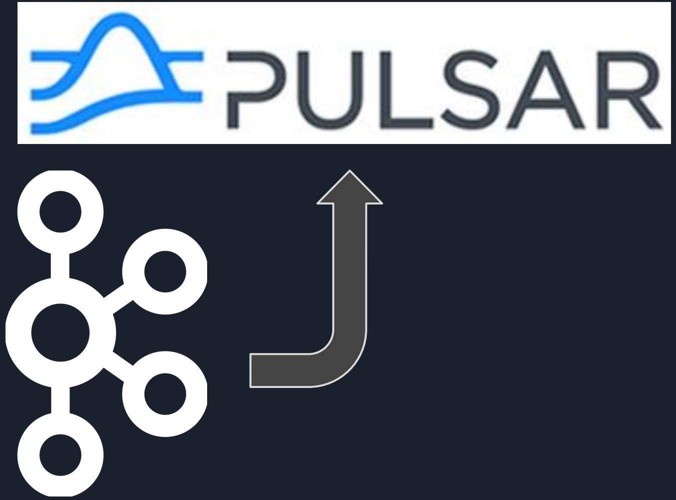
- Single instance Monolithic
- 6+ million daily connections!
- Expanding for conferences and teams
- Unlimited events for analytics

Powered by Chatroulette for Virtual Conferences

CHATROULETTE

From Kafka to Pulsar in 30 min

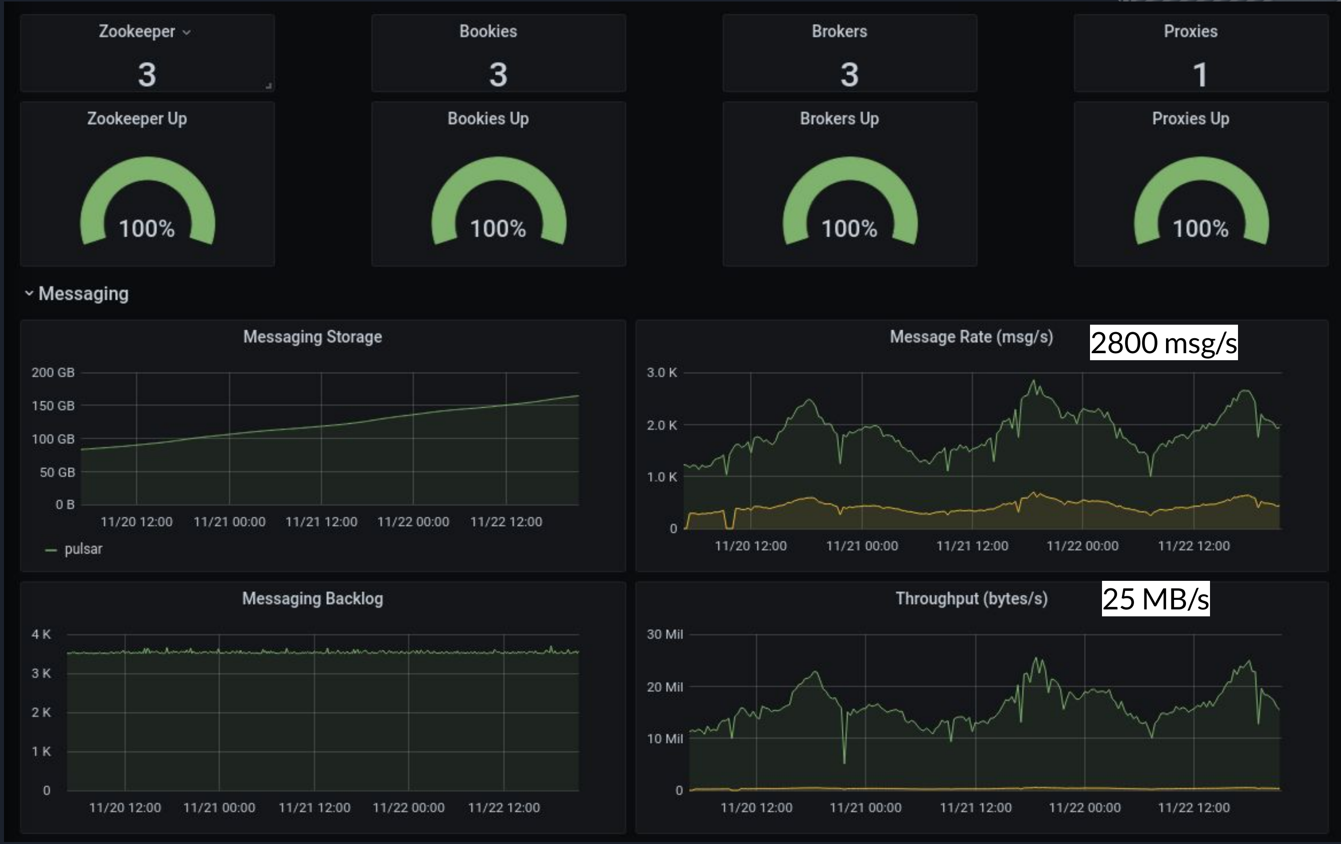
- Low-latency pub/sub with flexible subscription model
- Simpler to operate with K8s on bare metal
- Scalable per-user ephemeral topic
- Pulsar functions
- Tiered storage



Production Environment

Pulsar 2.6.1

- Broker
  - 8-16G RAM
  - 4 Cores
- Bookkeeper
  - 4-8G RAM
  - 2 Cores
- Zookeeper
  - 2G RAM
  - 2 Cores



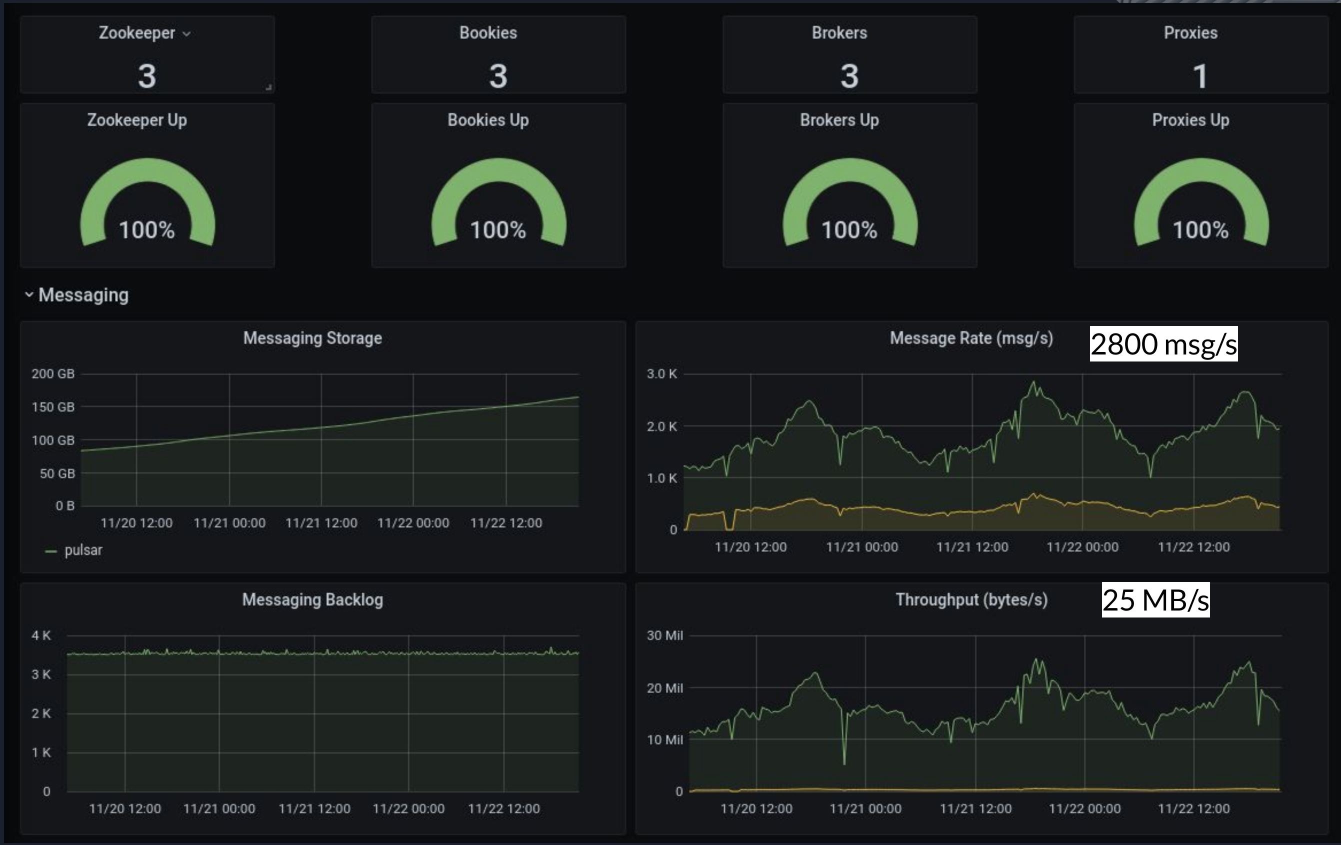
2800 msg/s

25 MB/s

Production Environment

- Kubernetes 1.19
- Istio 1.7
- Ceph 15.2

> 20K ephemeral topic / min



Zookeeper 3

Bookies 3

Brokers 3

Proxies 1

Zookeeper Up 100%

Bookies Up 100%

Brokers Up 100%

Proxies Up 100%

Messaging

2800 msg/s

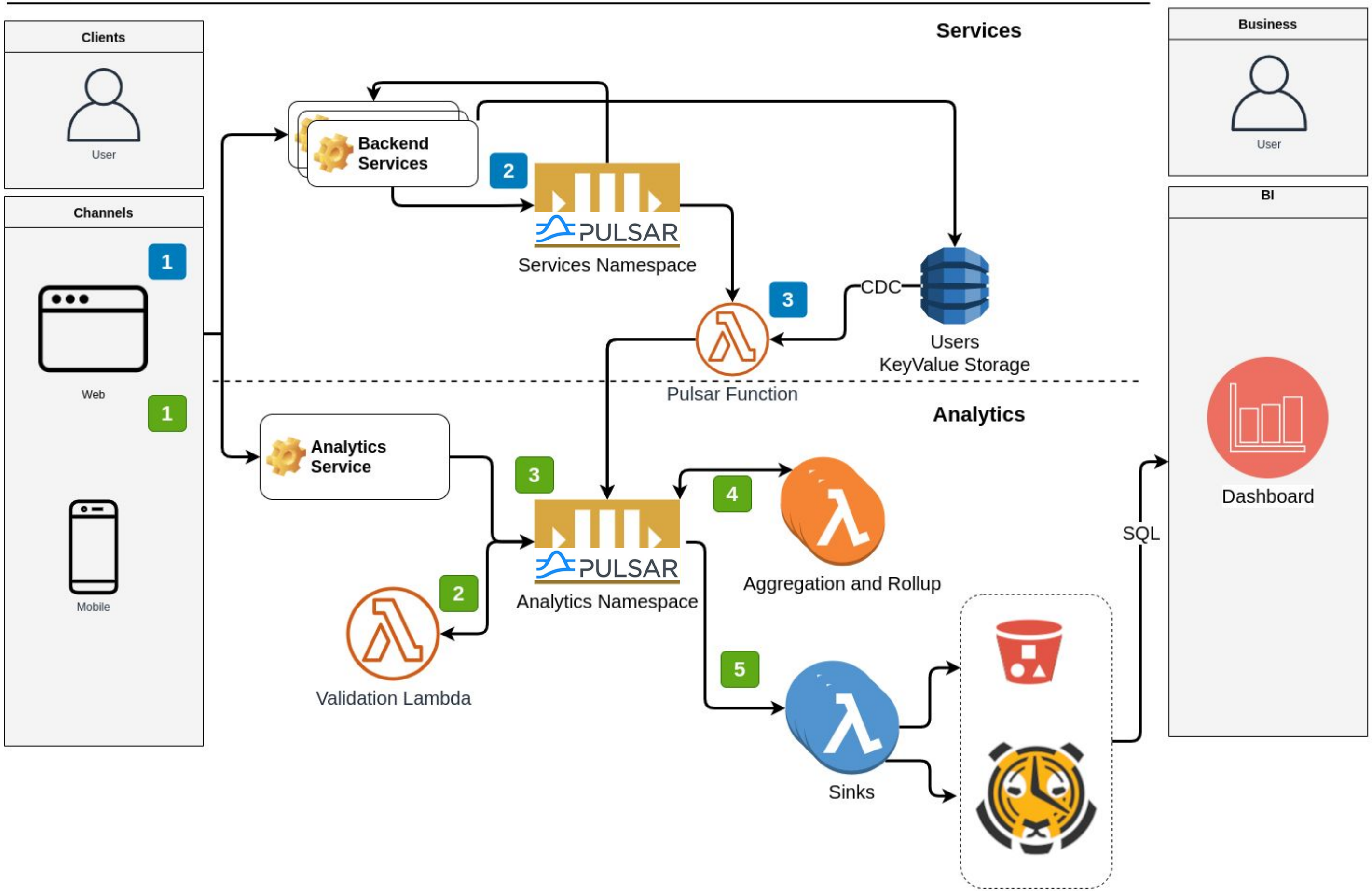
25 MB/s

PULSAR SUMMIT 2020

- Microservices Communication
- Ephemeral Topics
- Compact Topics
- Analytics Pipeline

Basic CR Analytics Pipeline

Work in progress



Lesson Learned

- Ephemeral topics, producer and subscribers scales!
- Pulsar function and ephemeral topics needs improvement
- Passing small images payload is better than using object storage at scale (MinIO/Ceph)
- Pulsar window function for analytics scales!
- Running on Pulsar on Istio works
- Using remote Ceph storage for Pulsar persistent volumes
- Tiered storage can solve small files issue in object storage

PULSAR SUMMIT 2020

OSS Pulsar Clients by CHATROULETTE

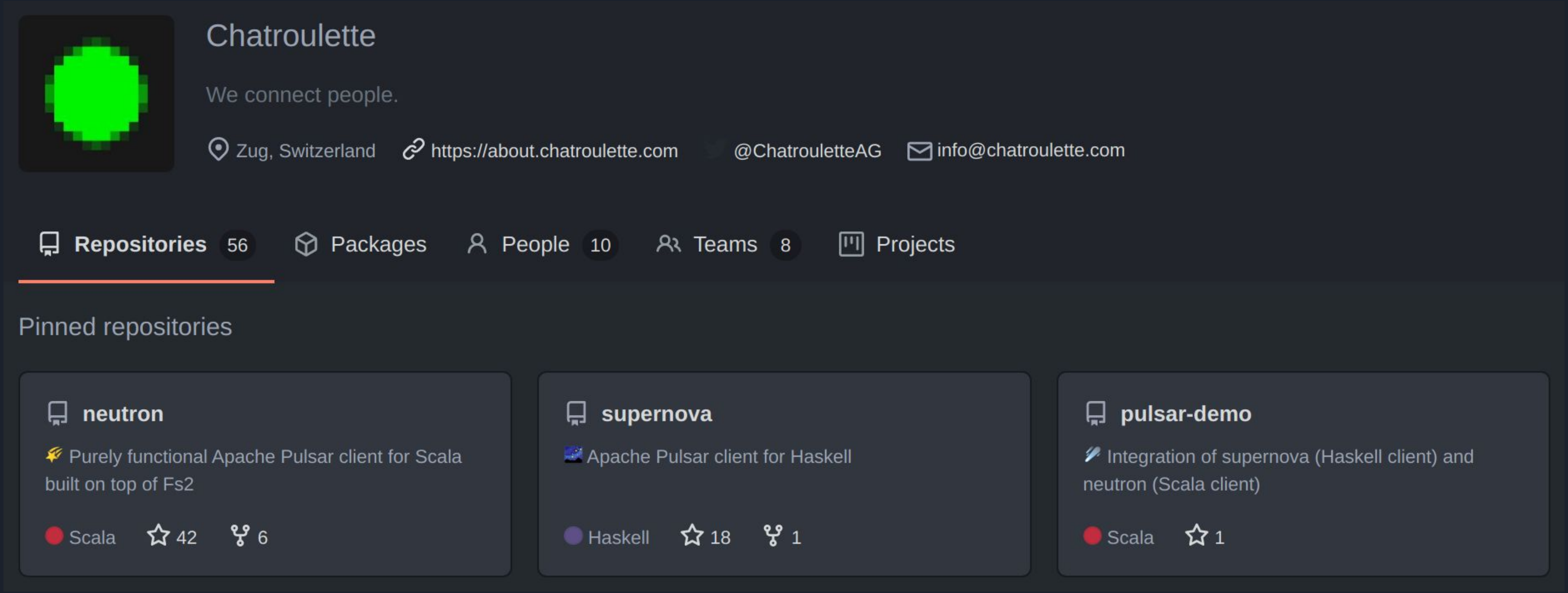
Neutron & Supernova

@volpegabriel87

@gvolpe

PULSAR SUMMIT 2020

https://github.com/cr-org



Chatroulette

We connect people.

Zug, Switzerland https://about.chatroulette.com @ChatrouletteAG info@chatroulette.com

Repositories 56 Packages People 10 Teams 8 Projects

Pinned repositories

neutron
Purely functional Apache Pulsar client for Scala built on top of Fs2
● Scala ☆ 42 6

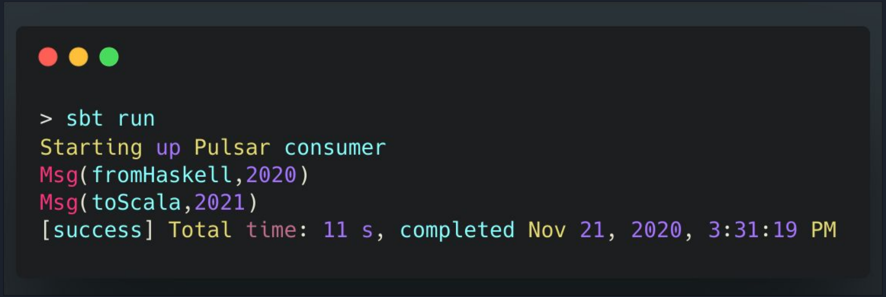
supernova
Apache Pulsar client for Haskell
● Haskell ☆ 18 1

pulsar-demo
Integration of supernova (Haskell client) and neutron (Scala client)
● Scala ☆ 1

@volpegabriel87 @gvolpe

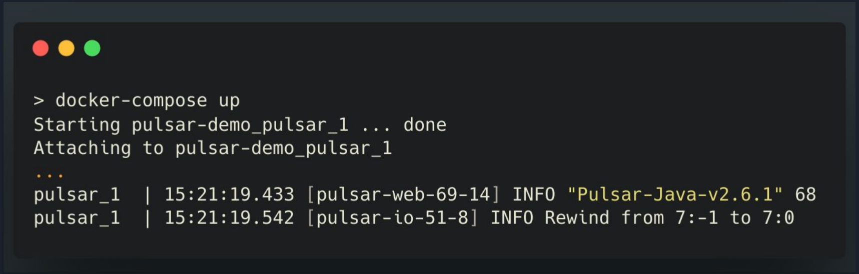
Demonstration

consumer



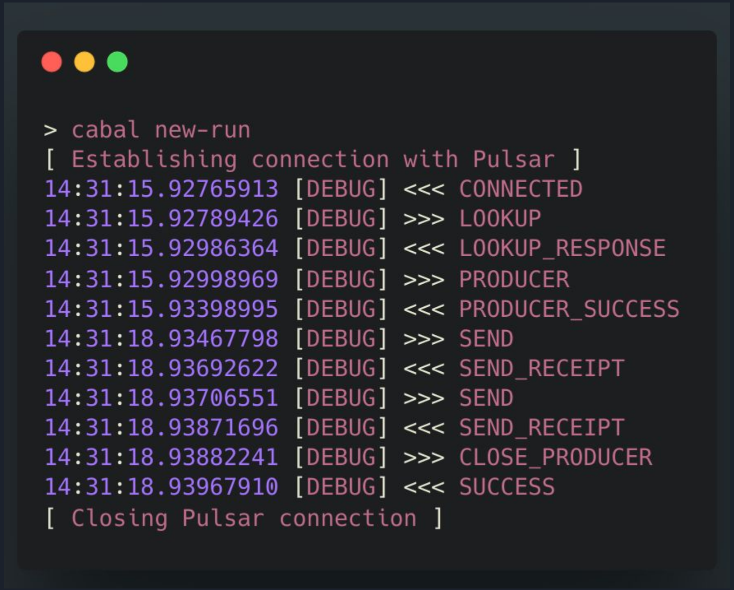
```
> sbt run
Starting up Pulsar consumer
Msg(fromHaskell,2020)
Msg(toScala,2021)
[success] Total time: 11 s, completed Nov 21, 2020, 3:31:19 PM
```

PULSAR



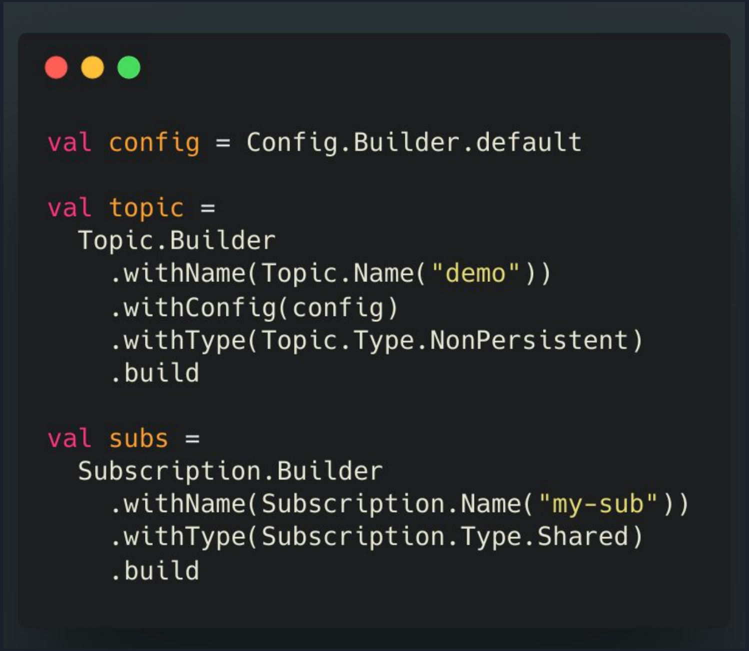
```
> docker-compose up
Starting pulsar-demo_pulsar_1 ... done
Attaching to pulsar-demo_pulsar_1
...
pulsar_1  | 15:21:19.433 [pulsar-web-69-14] INFO "Pulsar-Java-v2.6.1" 68
pulsar_1  | 15:21:19.542 [pulsar-io-51-8] INFO Rewind from 7:-1 to 7:0
```

producer

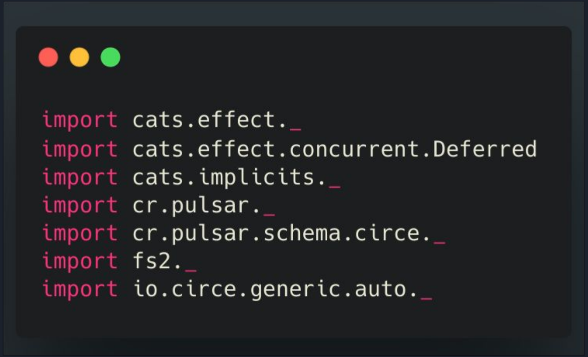


```
> cabal new-run
[ Establishing connection with Pulsar ]
14:31:15.92765913 [DEBUG] <<< CONNECTED
14:31:15.92789426 [DEBUG] >>> LOOKUP
14:31:15.92986364 [DEBUG] <<< LOOKUP_RESPONSE
14:31:15.92998969 [DEBUG] >>> PRODUCER
14:31:15.93398995 [DEBUG] <<< PRODUCER_SUCCESS
14:31:18.93467798 [DEBUG] >>> SEND
14:31:18.93692622 [DEBUG] <<< SEND_RECEIPT
14:31:18.93706551 [DEBUG] >>> SEND
14:31:18.93871696 [DEBUG] <<< SEND_RECEIPT
14:31:18.93882241 [DEBUG] >>> CLOSE_PRODUCER
14:31:18.93967910 [DEBUG] <<< SUCCESS
[ Closing Pulsar connection ]
```

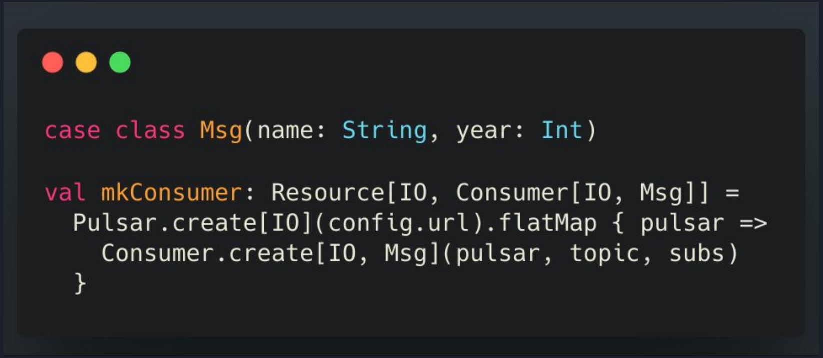
Neutron: Scala client



```scala
val config = Config.Builder.default

val topic =
  Topic.Builder
    .withName(Topic.Name("demo"))
    .withConfig(config)
    .withType(Topic.Type.NonPersistent)
    .build

val subs =
  Subscription.Builder
    .withName(Subscription.Name("my-sub"))
    .withType(Subscription.Type.Shared)
    .build
```

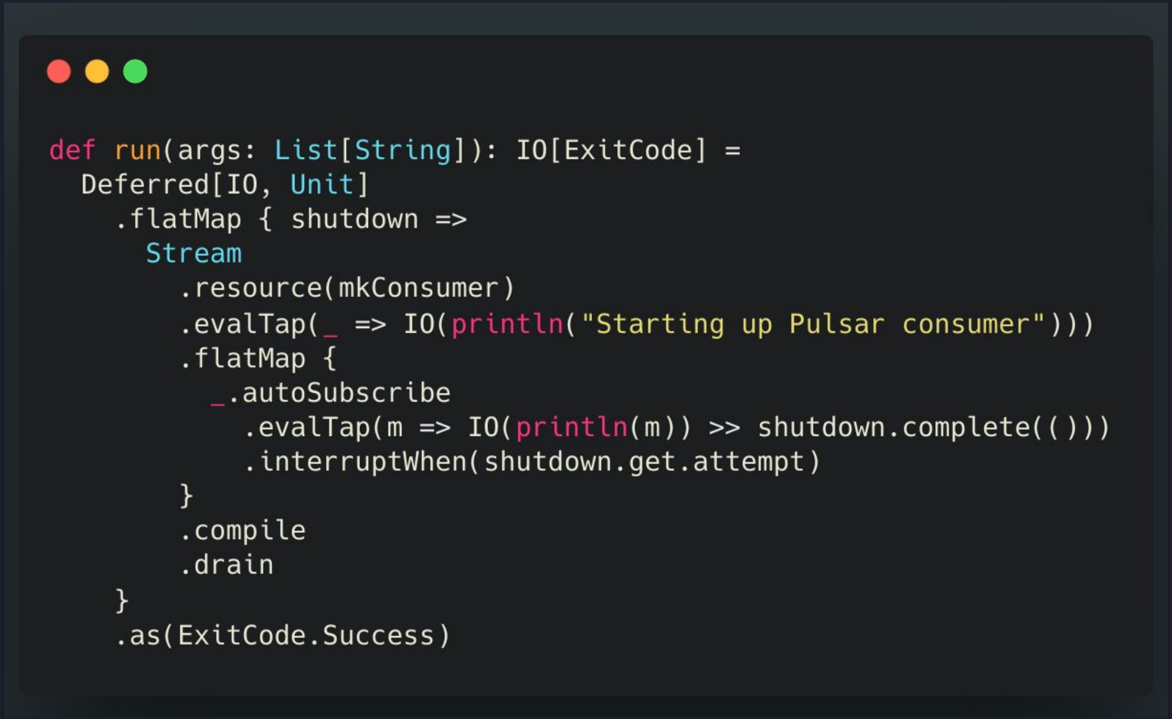


```scala
import cats.effect._
import cats.effect.concurrent.Deferred
import cats.implicits._
import cr.pulsar._
import cr.pulsar.schema.circe._
import fs2._
import io.circe.generic.auto._
```



```scala
case class Msg(name: String, year: Int)

val mkConsumer: Resource[IO, Consumer[IO, Msg]] =
  Pulsar.create[IO](config.url).flatMap { pulsar =>
    Consumer.create[IO, Msg](pulsar, topic, subs)
  }
```

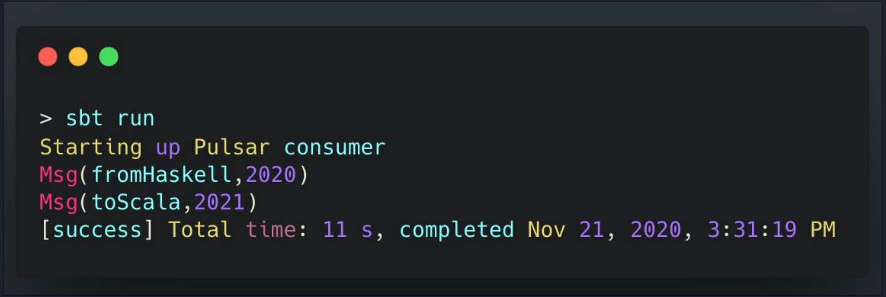
Neutron: Scala client



```scala
def run(args: List[String]): IO[ExitCode] =
  Deferred[IO, Unit]
    .flatMap { shutdown =>
      Stream
        .resource(mkConsumer)
        .evalTap(_ => IO(println("Starting up Pulsar consumer")))
        .flatMap {
          _.autoSubscribe
            .evalTap(m => IO(println(m)) >> shutdown.complete(()))
            .interruptWhen(shutdown.get.attempt)
        }
        .compile
        .drain
    }
    .as(ExitCode.Success)
```

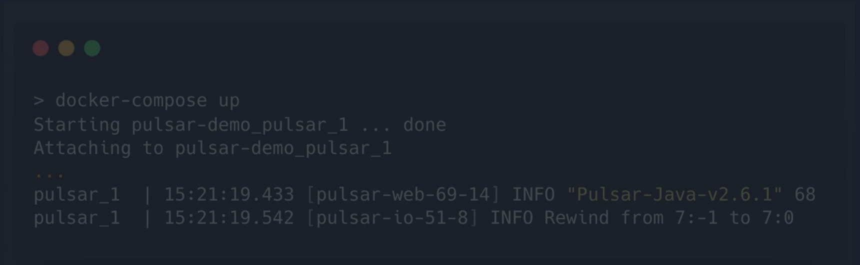
Demonstration

consumer



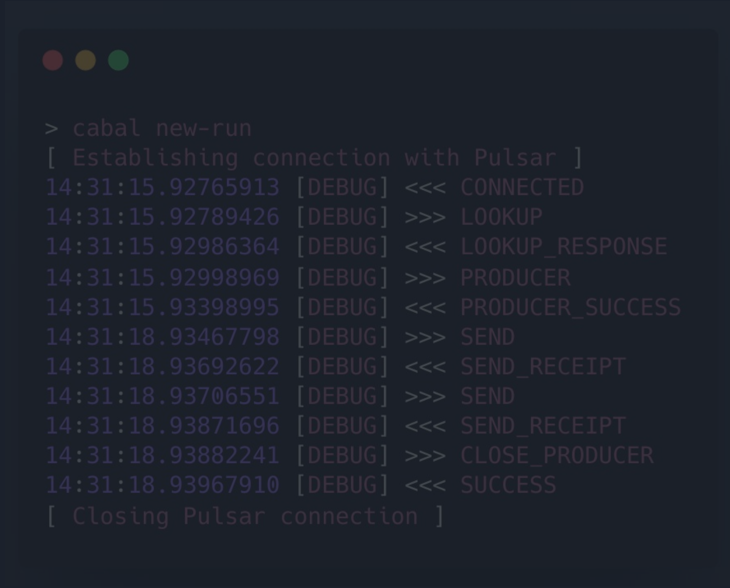
```
> sbt run
Starting up Pulsar consumer
Msg(fromHaskell,2020)
Msg(toScala,2021)
[success] Total time: 11 s, completed Nov 21, 2020, 3:31:19 PM
```

PULSAR



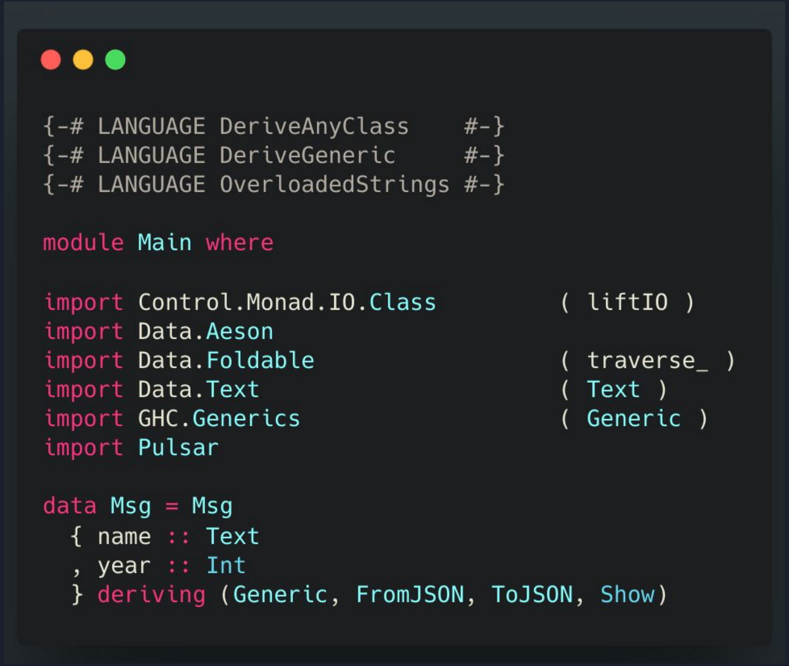
```
> docker-compose up
Starting pulsar-demo_pulsar_1 ... done
Attaching to pulsar-demo_pulsar_1
...
pulsar_1  | 15:21:19.433 [pulsar-web-69-14] INFO "Pulsar-Java-v2.6.1" 68
pulsar_1  | 15:21:19.542 [pulsar-io-51-8] INFO Rewind from 7:-1 to 7:0
```

producer

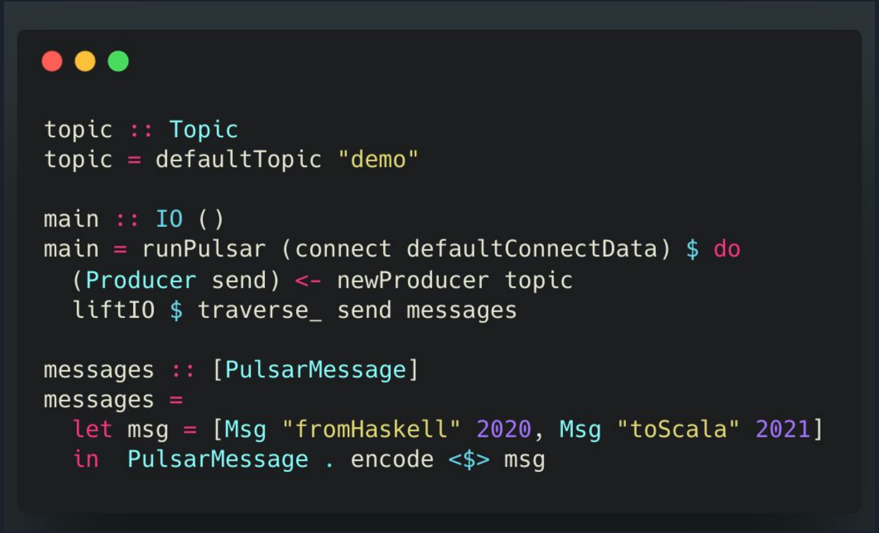


```
> cabal new-run
[ Establishing connection with Pulsar ]
14:31:15.92765913 [DEBUG] <<< CONNECTED
14:31:15.92789426 [DEBUG] >>> LOOKUP
14:31:15.92986364 [DEBUG] <<< LOOKUP_RESPONSE
14:31:15.92998969 [DEBUG] >>> PRODUCER
14:31:15.93398995 [DEBUG] <<< PRODUCER_SUCCESS
14:31:18.93467798 [DEBUG] >>> SEND
14:31:18.93692622 [DEBUG] <<< SEND_RECEIPT
14:31:18.93706551 [DEBUG] >>> SEND
14:31:18.93871696 [DEBUG] <<< SEND_RECEIPT
14:31:18.93882241 [DEBUG] >>> CLOSE_PRODUCER
14:31:18.93967910 [DEBUG] <<< SUCCESS
[ Closing Pulsar connection ]
```

Supernova: Haskell client



```haskell
{-# LANGUAGE DeriveAnyClass    #-}
{-# LANGUAGE DeriveGeneric     #-}
{-# LANGUAGE OverloadedStrings #-}

module Main where

import Control.Monad.IO.Class         ( liftIO )
import Data.Aeson
import Data.Foldable                  ( traverse_ )
import Data.Text                      ( Text )
import GHC.Generics                   ( Generic )
import Pulsar

data Msg = Msg
  { name :: Text
  , year :: Int
  } deriving (Generic, FromJSON, ToJSON, Show)
```

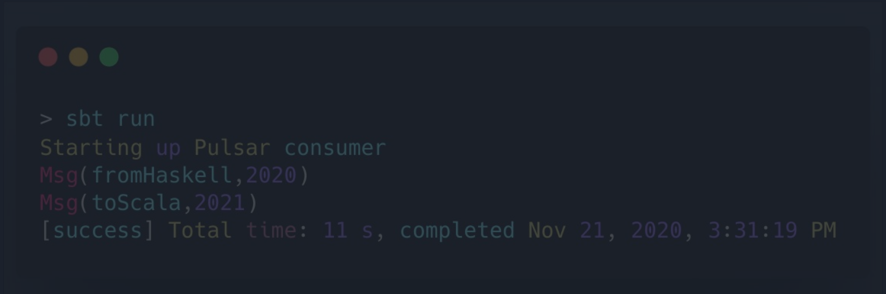


```haskell
topic :: Topic
topic = defaultTopic "demo"

main :: IO ()
main = runPulsar (connect defaultConnectData) $ do
  (Producer send) <- newProducer topic
  liftIO $ traverse_ send messages

messages :: [PulsarMessage]
messages =
  let msg = [Msg "fromHaskell" 2020, Msg "toScala" 2021]
  in  PulsarMessage . encode <$> msg
```

PULSAR SUMMIT 2020

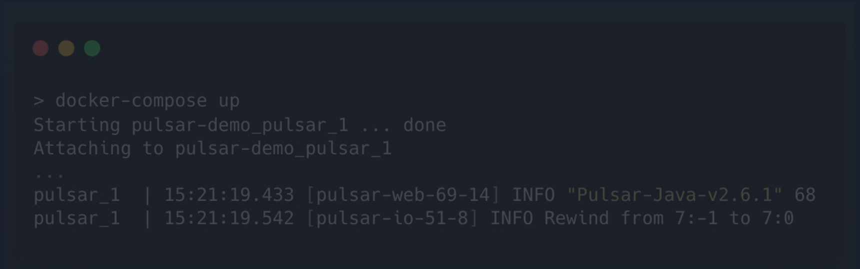
Demonstration

consumer



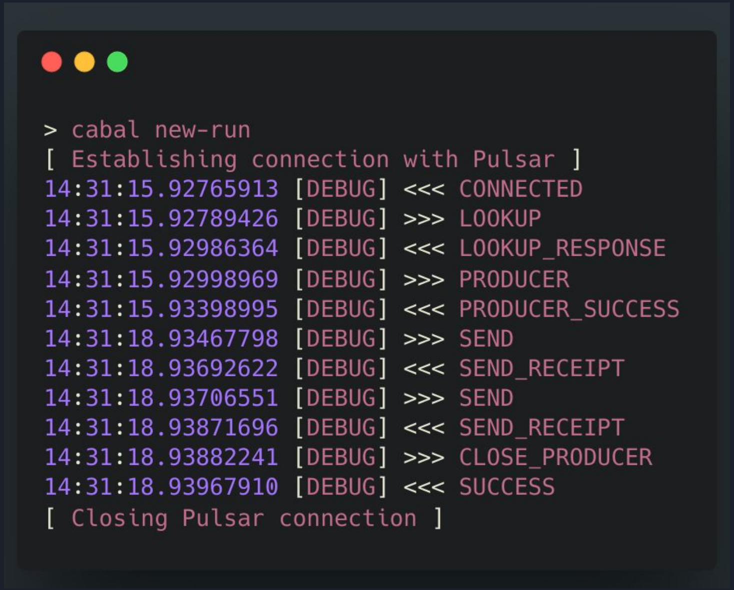
```
> sbt run
Starting up Pulsar consumer
Msg(fromHaskell,2020)
Msg(toScala,2021)
[success] Total time: 11 s, completed Nov 21, 2020, 3:31:19 PM
```

PULSAR



```
> docker-compose up
Starting pulsar-demo_pulsar_1 ... done
Attaching to pulsar-demo_pulsar_1
...
pulsar_1  | 15:21:19.433 [pulsar-web-69-14] INFO "Pulsar-Java-v2.6.1" 68
pulsar_1  | 15:21:19.542 [pulsar-io-51-8] INFO Rewind from 7:-1 to 7:0
```

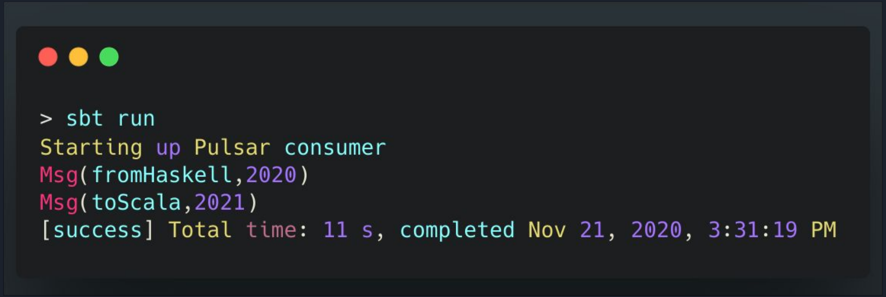
producer



```
> cabal new-run
[ Establishing connection with Pulsar ]
14:31:15.92765913 [DEBUG] <<< CONNECTED
14:31:15.92789426 [DEBUG] >>> LOOKUP
14:31:15.92986364 [DEBUG] <<< LOOKUP_RESPONSE
14:31:15.92998969 [DEBUG] >>> PRODUCER
14:31:15.93398995 [DEBUG] <<< PRODUCER_SUCCESS
14:31:18.93467798 [DEBUG] >>> SEND
14:31:18.93692622 [DEBUG] <<< SEND_RECEIPT
14:31:18.93706551 [DEBUG] >>> SEND
14:31:18.93871696 [DEBUG] <<< SEND_RECEIPT
14:31:18.93882241 [DEBUG] >>> CLOSE_PRODUCER
14:31:18.93967910 [DEBUG] <<< SUCCESS
[ Closing Pulsar connection ]
```

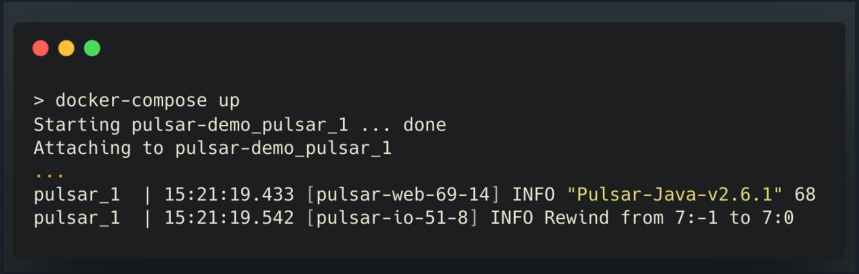
Demonstration

consumer



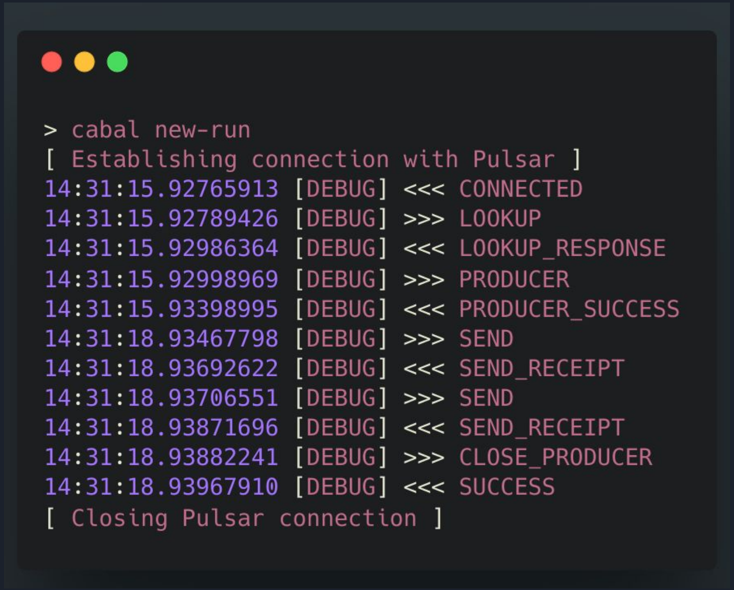
```
> sbt run
Starting up Pulsar consumer
Msg(fromHaskell,2020)
Msg(toScala,2021)
[success] Total time: 11 s, completed Nov 21, 2020, 3:31:19 PM
```

PULSAR



```
> docker-compose up
Starting pulsar-demo_pulsar_1 ... done
Attaching to pulsar-demo_pulsar_1
...
pulsar_1  | 15:21:19.433 [pulsar-web-69-14] INFO "Pulsar-Java-v2.6.1" 68
pulsar_1  | 15:21:19.542 [pulsar-io-51-8] INFO Rewind from 7:-1 to 7:0
```

producer



```
> cabal new-run
[ Establishing connection with Pulsar ]
14:31:15.92765913 [DEBUG] <<< CONNECTED
14:31:15.92789426 [DEBUG] >>> LOOKUP
14:31:15.92986364 [DEBUG] <<< LOOKUP_RESPONSE
14:31:15.92998969 [DEBUG] >>> PRODUCER
14:31:15.93398995 [DEBUG] <<< PRODUCER_SUCCESS
14:31:18.93467798 [DEBUG] >>> SEND
14:31:18.93692622 [DEBUG] <<< SEND_RECEIPT
14:31:18.93706551 [DEBUG] >>> SEND
14:31:18.93871696 [DEBUG] <<< SEND_RECEIPT
14:31:18.93882241 [DEBUG] >>> CLOSE_PRODUCER
14:31:18.93967910 [DEBUG] <<< SUCCESS
[ Closing Pulsar connection ]
```

THANKS!

谢谢！

@volpegabriel87

@gvolpe