Development productivity in a post-serverless world

Domenico Musto

CHRONOMICS

Once upon a time

Domenico Musto

domenico.musto@gmail.com

@mimmozzo

Principal Engineer at Chronomics

Things I love

- Distributed systems
- Event driven architectures
- Cloud native
- Serverless
- Continuous delivery

Things I hate

- Regular expressions
- IoC container

CHRONOMICS

Agenda

- From servers to serverless
- Infrastructure as code

From servers to serverless

From servers to serverless

From servers to serverless

Devs

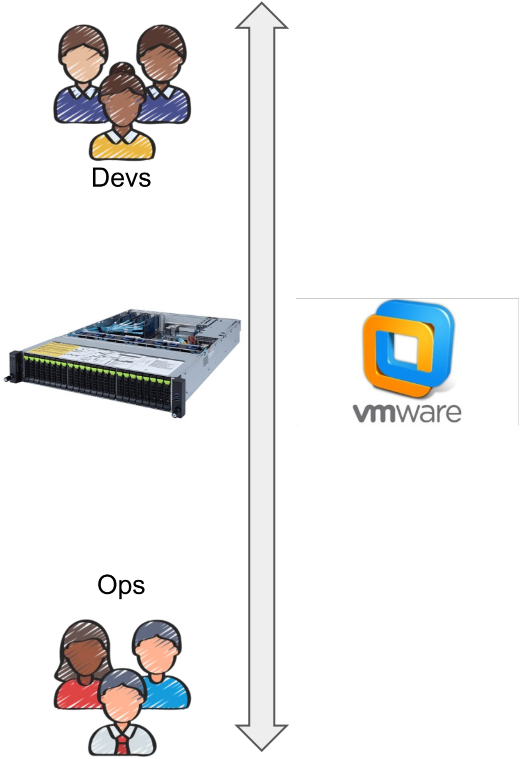
Ops

From servers to serverless

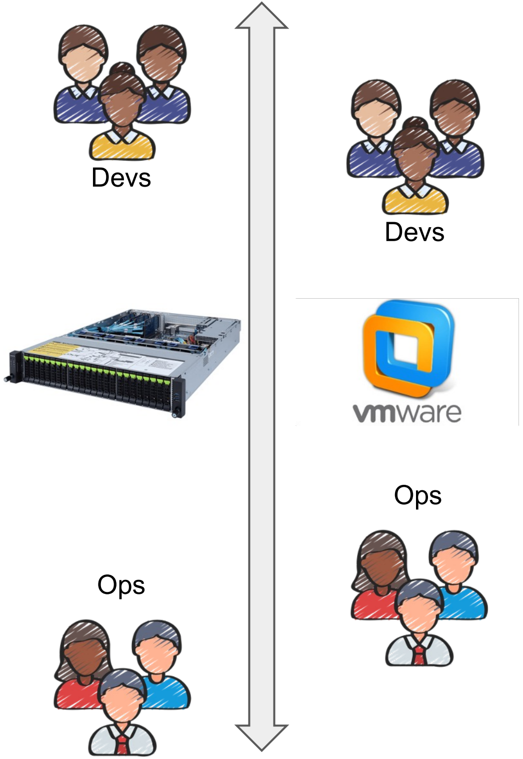
Devs

Ops

From servers to serverless



From servers to serverless



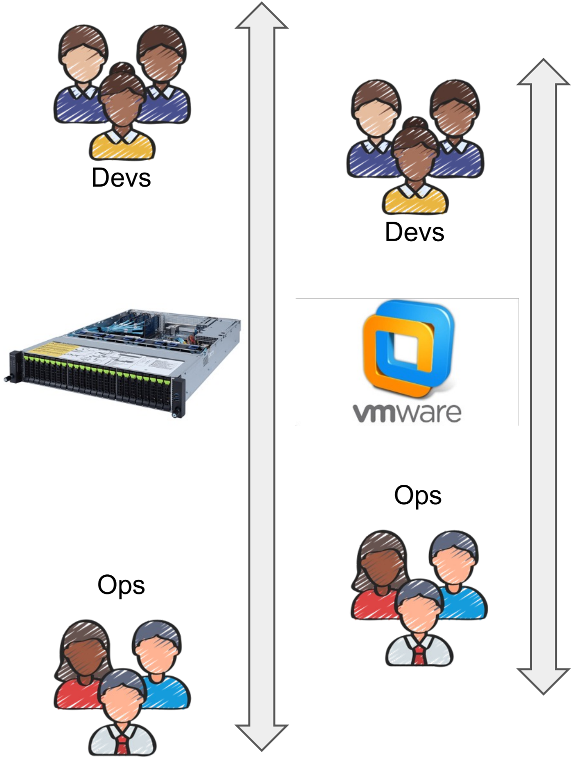
Devs

Ops

Devs

Ops

From servers to serverless



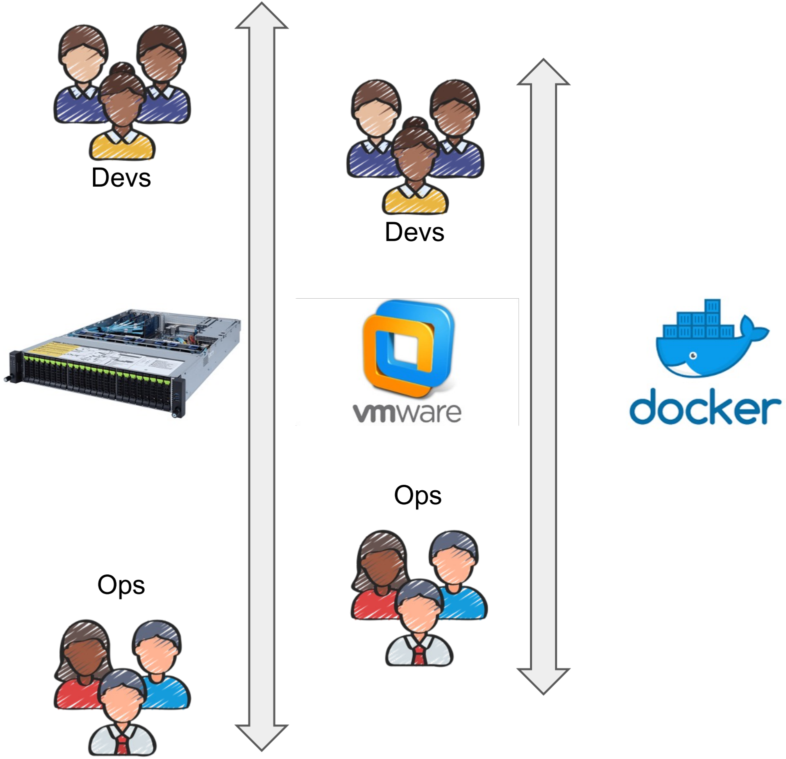
Devs

Ops

Devs

Ops

From servers to serverless



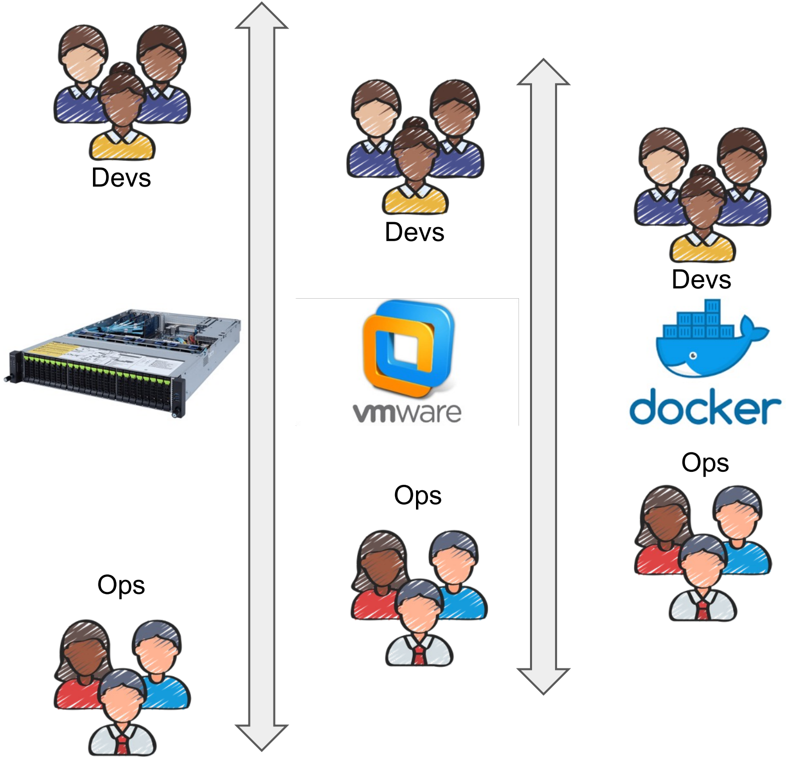
Devs

Ops

Devs

Ops

From servers to serverless



Devs

Devs

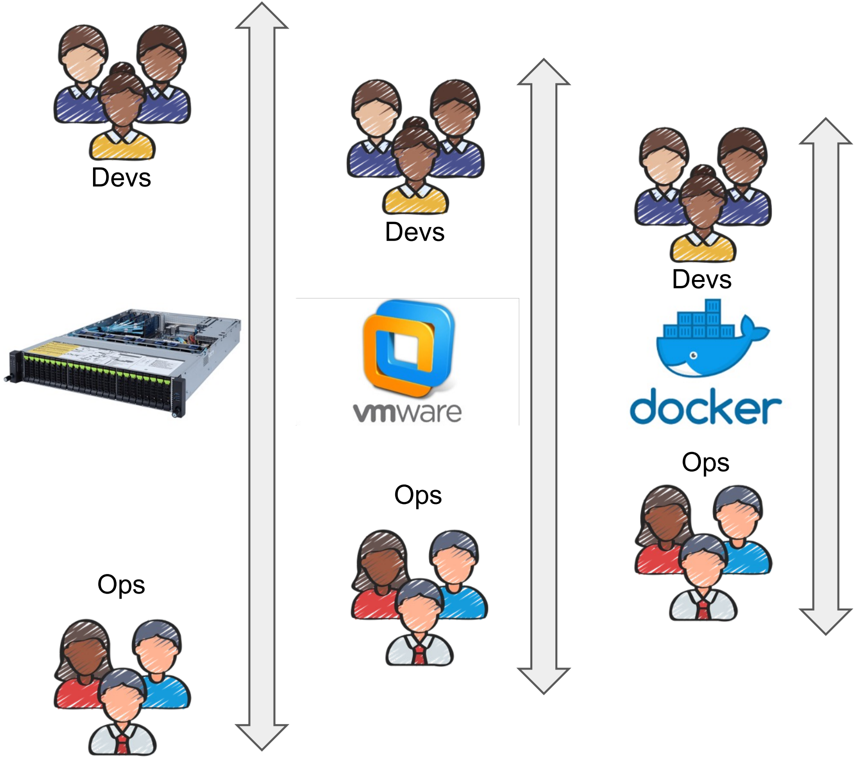
Devs

Ops

Ops

Ops

From servers to serverless



Devs

Ops

Devs

vmware

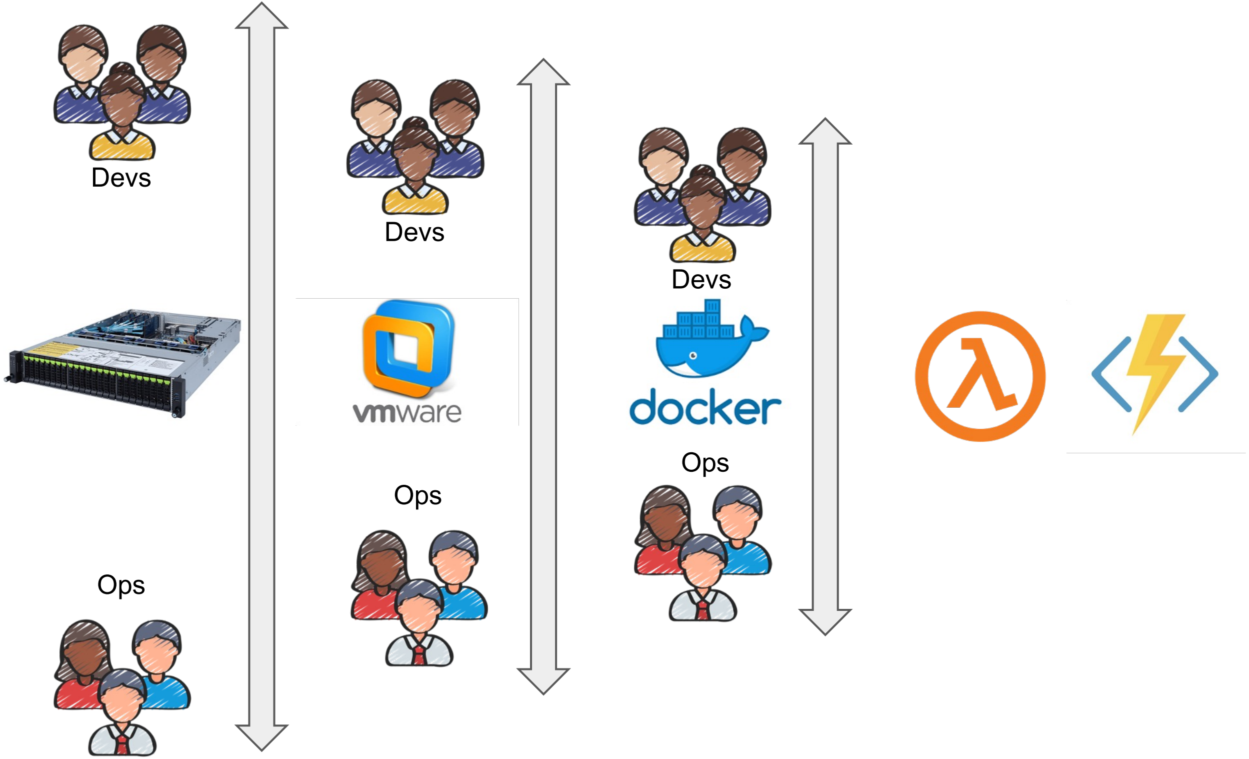
Ops

Devs

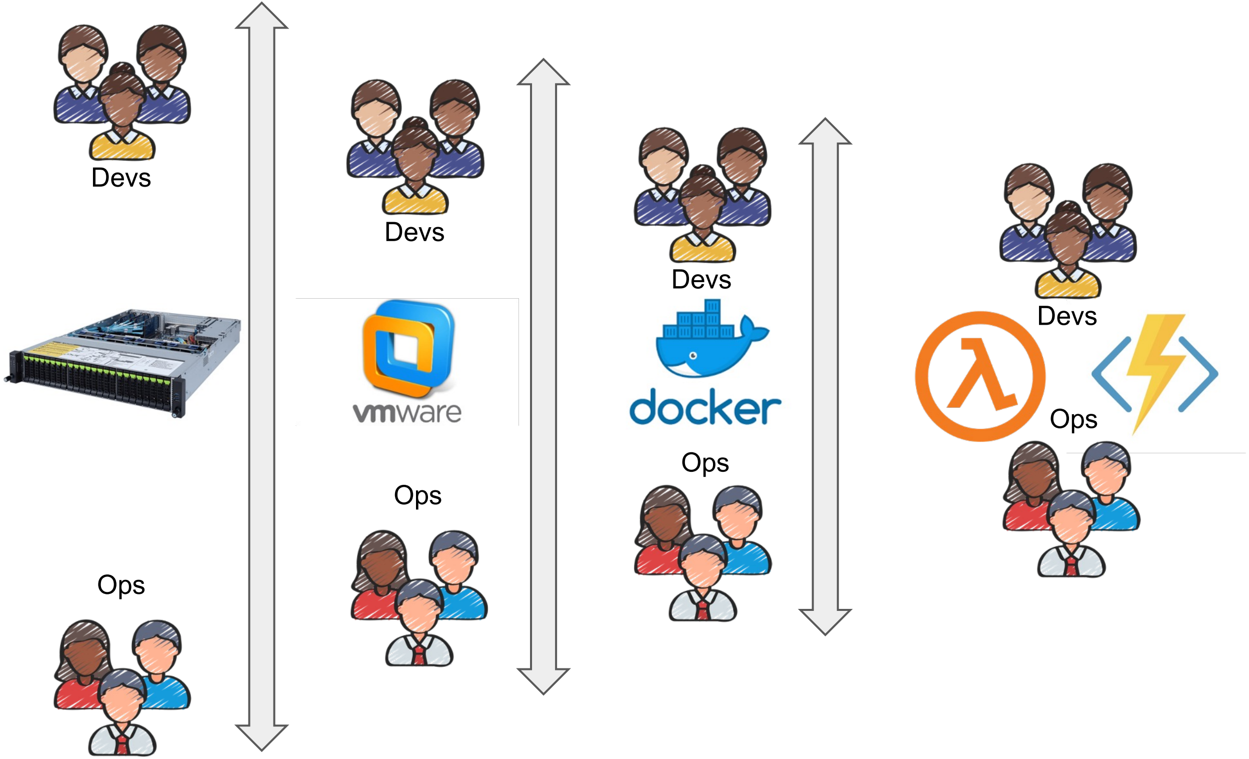
docker

Ops

From servers to serverless



From servers to serverless



Devs

Ops

Devs

Ops

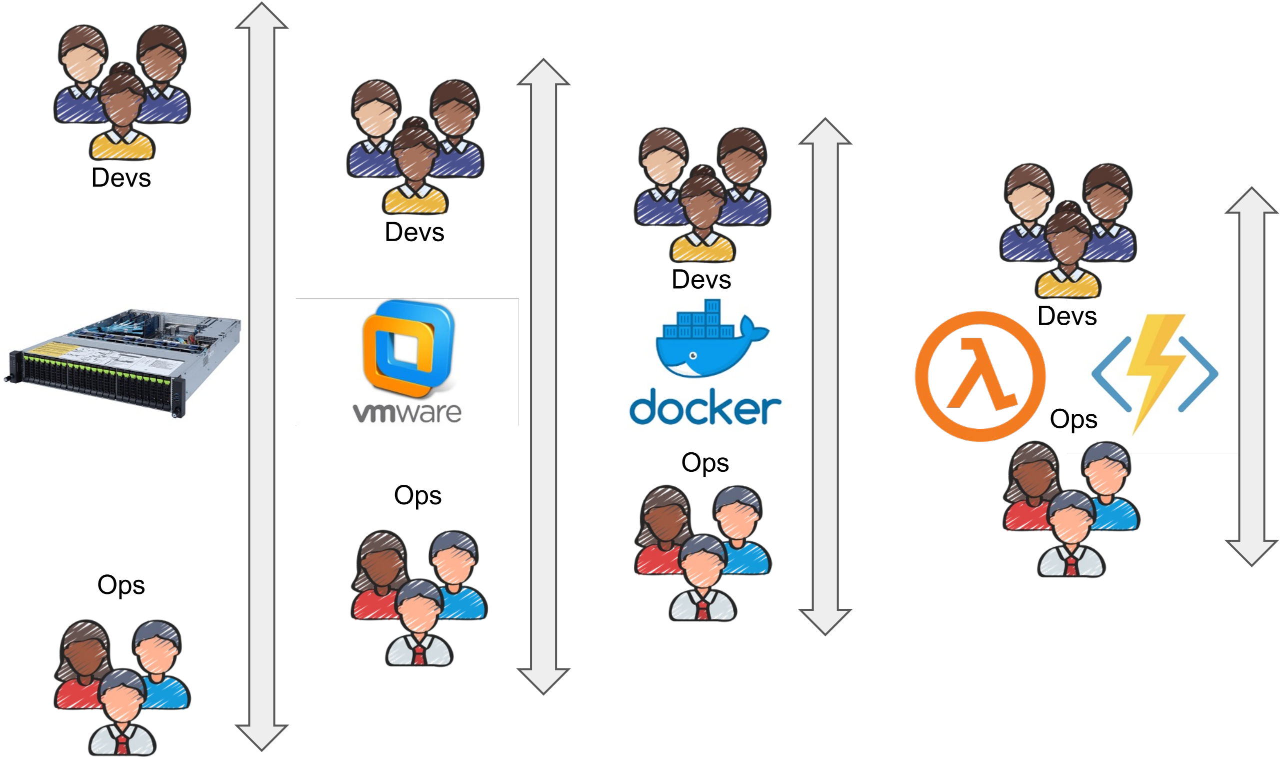
Devs

Ops

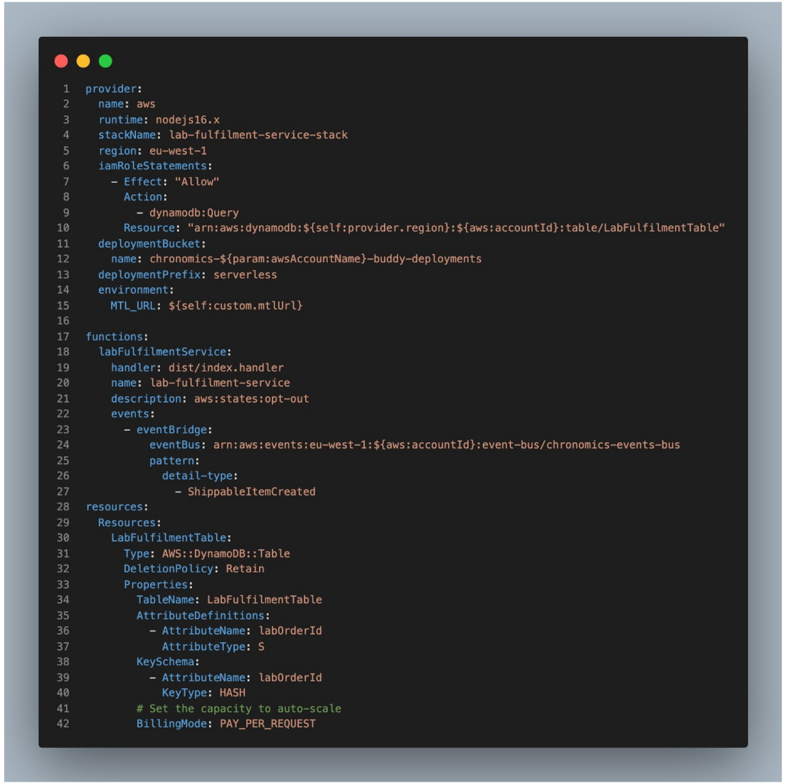
Devs

Ops

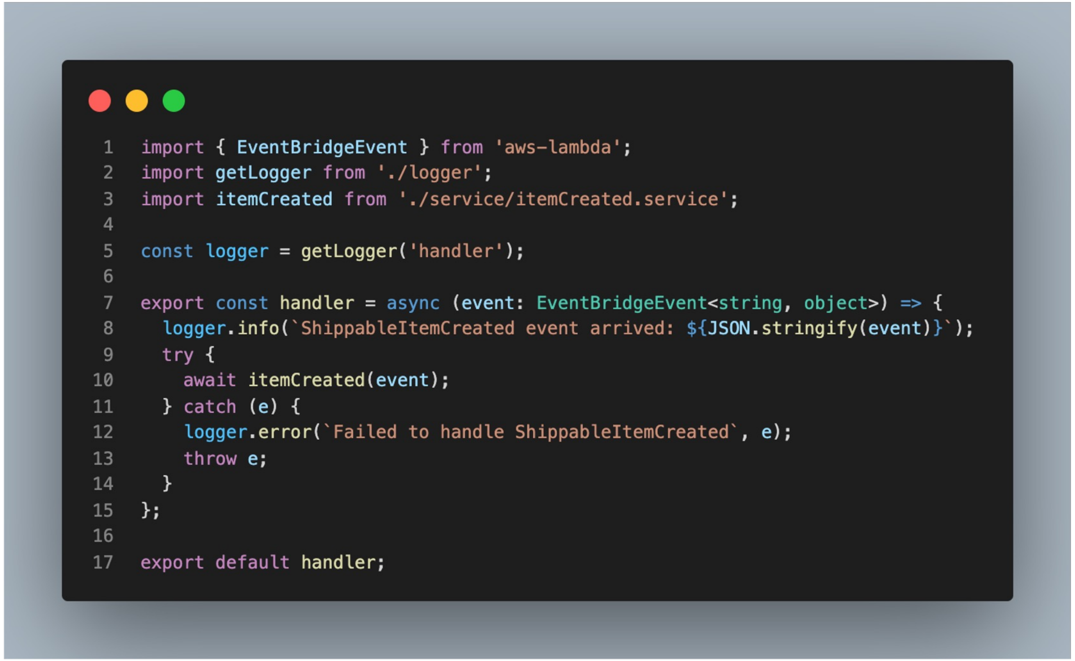
From servers to serverless



From servers to serverless



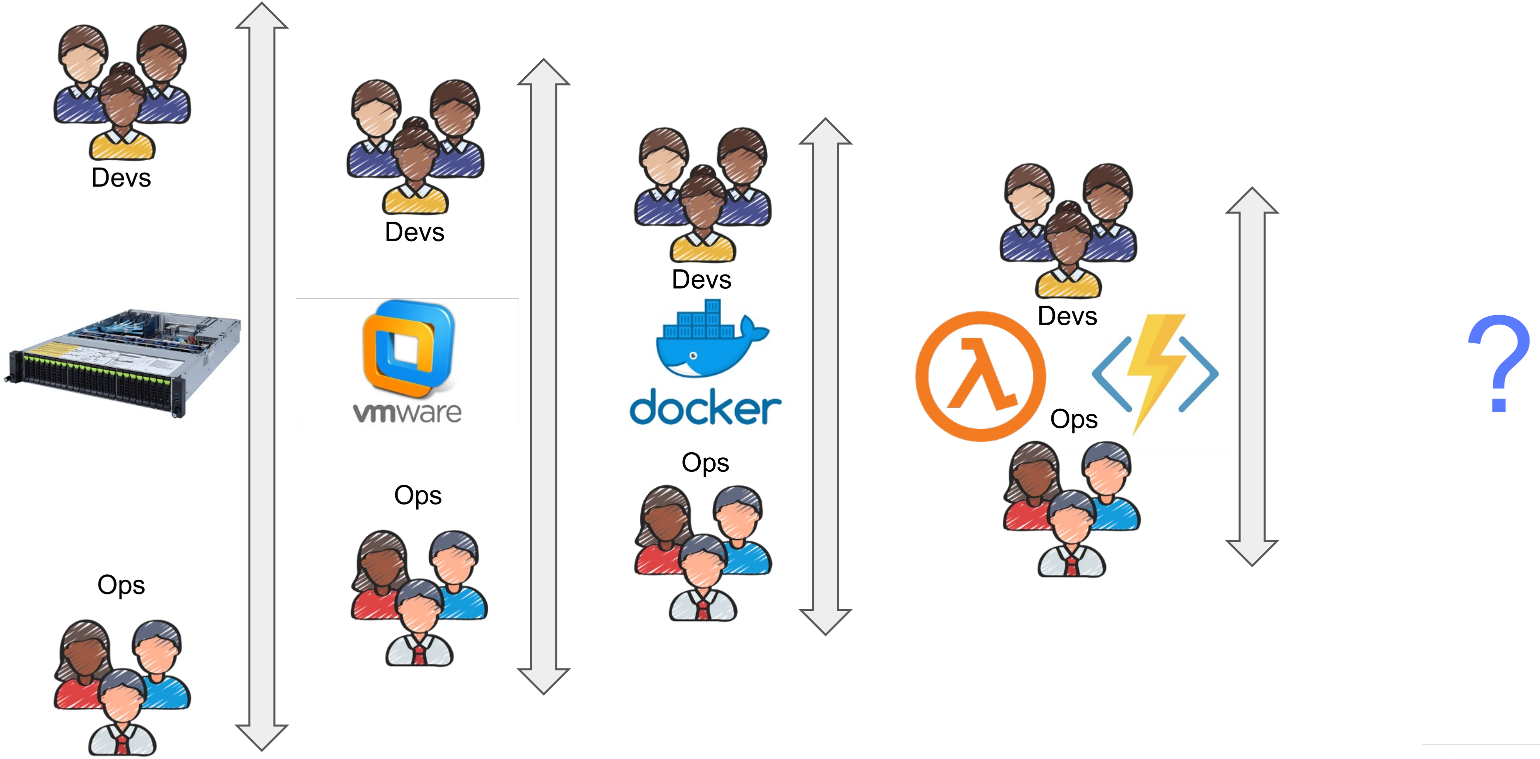
```yaml
provider:
  name: aws
  runtime: nodejs16.x
  stackName: lab-fulfilment-service-stack
  region: eu-west-1
  iamRoleStatements:
    - Effect: "Allow"
      Action:
        - dynamodb:Query
      Resource: "arn:aws:dynamodb:${self:provider.region}:${aws:accountId}:table/LabFulfilmentTable"
  deploymentBucket:
    name: chronomics-${param:awsAccountName}-buddy-deployments
  deploymentPrefix: serverless
  environment:
    MTL_URL: ${self:custom.mtlUrl}

functions:
  labFulfilmentService:
    handler: dist/index.handler
    name: lab-fulfilment-service
    description: aws:states:opt-out
    events:
      - eventBridge:
          eventBus: arn:aws:events:eu-west-1:${aws:accountId}:event-bus/chronomics-events-bus
          pattern:
            detail-type:
              - ShippableItemCreated
resources:
  Resources:
    LabFulfilmentTable:
      Type: AWS::DynamoDB::Table
      DeletionPolicy: Retain
      Properties:
        TableName: LabFulfilmentTable
        AttributeDefinitions:
          - AttributeName: labOrderId
            AttributeType: S
        KeySchema:
          - AttributeName: labOrderId
            KeyType: HASH
        # Set the capacity to auto-scale
        BillingMode: PAY_PER_REQUEST
```



```typescript
import { EventBridgeEvent } from 'aws-lambda';
import getLogger from './logger';
import itemCreated from './service/itemCreated.service';

const logger = getLogger('handler');

export const handler = async (event: EventBridgeEvent<string, object>) => {
  logger.info(`ShippableItemCreated event arrived: ${JSON.stringify(event)}`);
  try {
    await itemCreated(event);
  } catch (e) {
    logger.error(`Failed to handle ShippableItemCreated`, e);
    throw e;
  }
};

export default handler;
```

CHRONOMICS

From servers to serverless



Devs

Ops

Devs

Ops

Devs

Ops

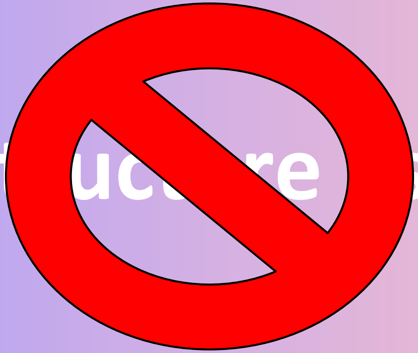
Devs

Ops

?

Infrastructure as code

Infrastructure as code



Systems over infrastructure and application

Agenda

- From servers to serverless
- ~~Infrastructure as code~~
- Platform as code
- Infrastructure from code

Platform as code

CHRONOMICS

Platform as code

- You code the platform over infrastructure or application
- Single language
- Single CI/CD pipeline

Infrastructure from code

Infrastructure from code

- Application driven systems
- Infrastructure as a sub-resource
- Decoupled from any cloud provider

Q&A

domenico.musto@gmail.com

@mimmozzo