Scaling R in the Cloud

with Azure Batch and Docker

Christoph Bodner

Post

AGENDA

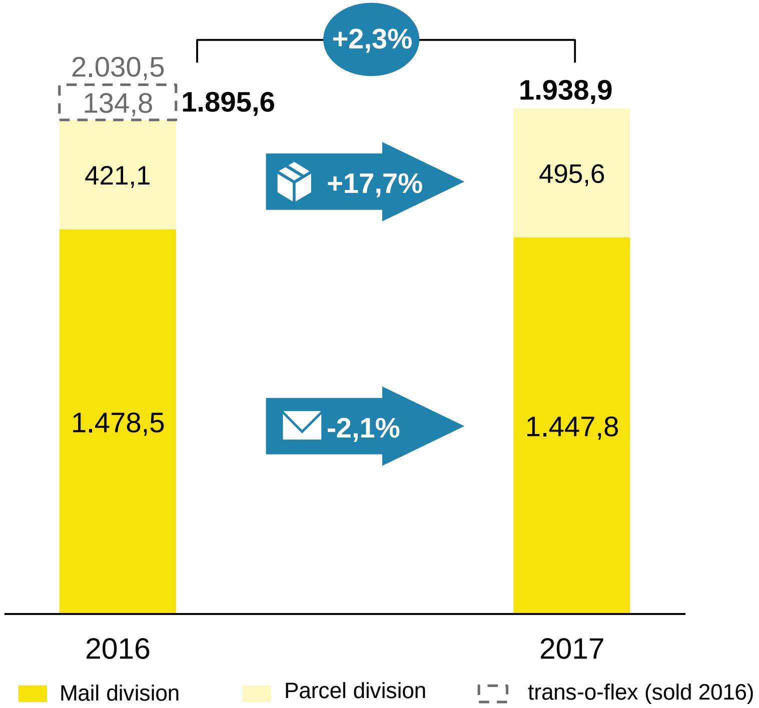
	01	02	03
Topics	Who we are (obligatory marketing stuff…)	Our problems	Our solution
	Data Science@Post AG: - Overview: Post AG - Our team	Deploying R at scale: - Scaling compute resources on demand - Reduce dev-prod disparity	Main components: - Azure Batch - Docker - VSTS

AGENDA

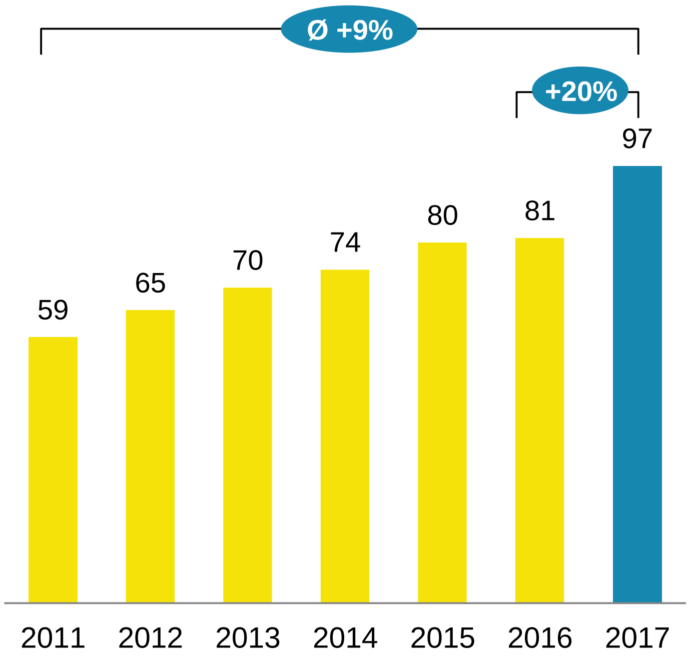
	01	02	03
Topics	Who we are (obligatory marketing stuff…)	Our problems	Our solution
	Data Science@Post AG: - Overview: Post AG - Our team	Deploying R at scale: - Scaling compute resources on demand - Reduce dev-prod disparity	Main components: - Azure Batch - Docker - VSTS

OVERVIEW: POST AG
COMPANY PERFORMANCE & PARCEL VOLUMES OVER TIME

SALES PERFORMANCE
mio EUR



PARCEL VOLUMES OF AUSTRIAN POST
mio parcels



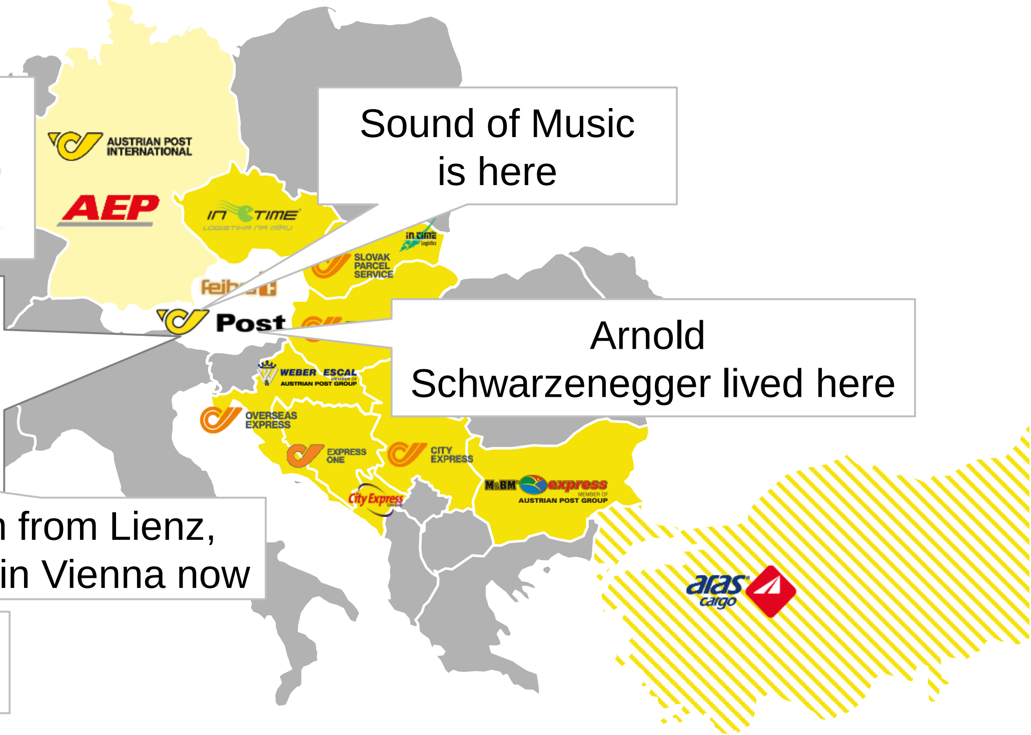
OVERVIEW: POST AG
STRONG PRESENCE IN EASTERN EUROPE

GROWTH FOCUS ON PARCEL & LOGISTICS DIVISION

About 13k inhabitants not including and

I am from Lienz, living in Vienna now

Please pardon my accent☺



Sound of Music is here

Arnold Schwarzenegger lived here

ORGANISATION

DATA SCIENCE IS PART OF BI COMPETENCE CENTER

Lead BI CC
Georg Posan

BI Competence Center

SAP Gurus

Everything related to SAP HANA, BO, etc.

.net/T-SQL Wizards

SQL Server, PowerBI and MS tech in general

This is us ☺

Analytics team

OUR TEAM
PEOPLE WHO LIKE $\pi z^2 a$ IN EVERY FORM

Christoph Bodner

Lead Data Scientist

Quantitative Finance (WU)
Prev.: KPMG

Thomas Laber

Senior Data Scientist

Business Informatics (TU)
Prev.: Accenture

Martin Blöschl

Junior Data Scientist

Computational Intelligence (TU)

Raphael Pesl

Junior Data Scientist

Mathematics (TU)

AGENDA

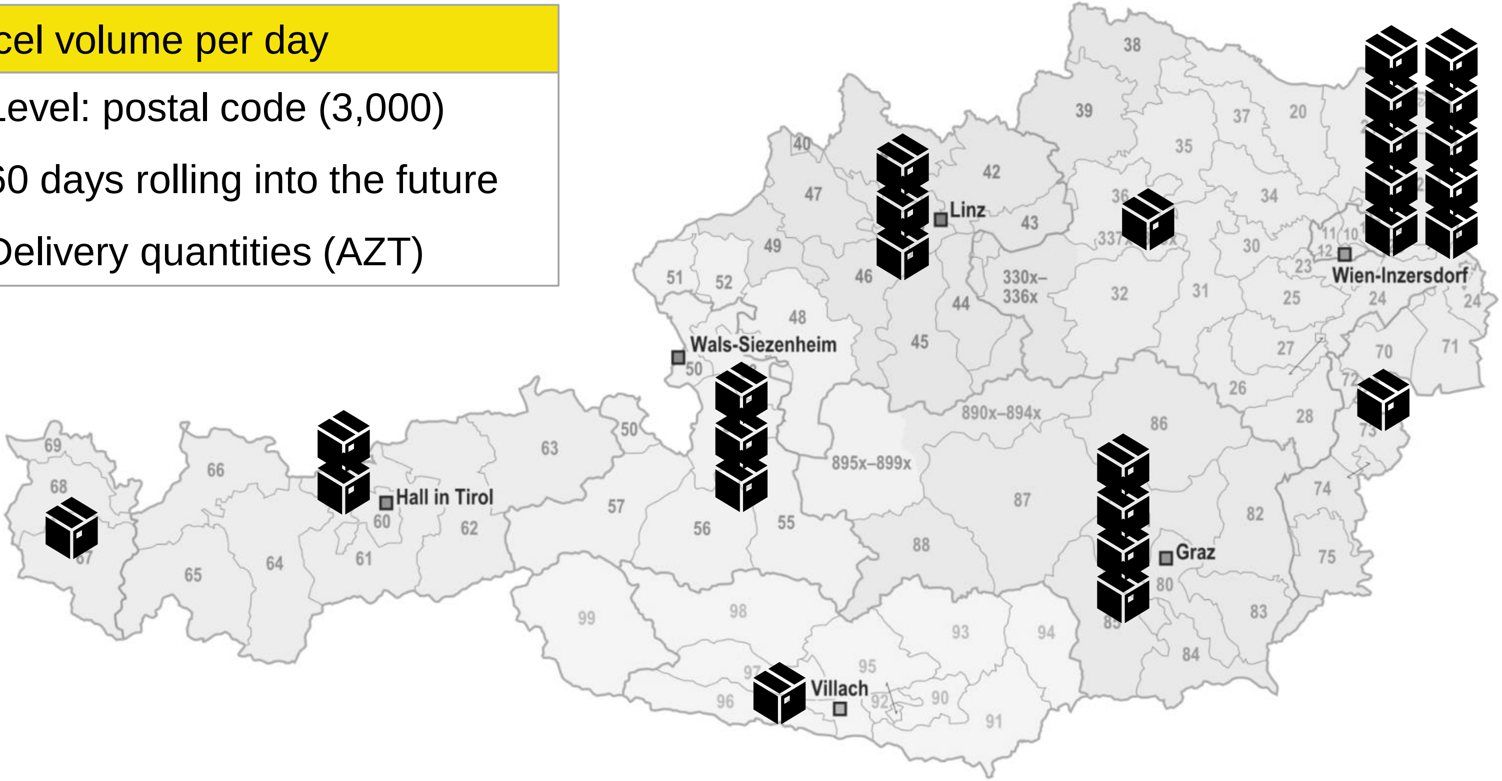
	01	02	03
Topics	Who we are (obligatory marketing stuff…)	Our problems	Our solution
	Data Science@Post AG: - Overview: Post AG - Our team	Deploying R at scale: - Scaling compute resources on demand - Reduce dev-prod disparity	Main components: - Azure Batch - Docker - VSTS

PARCEL VOLUME FORECAST

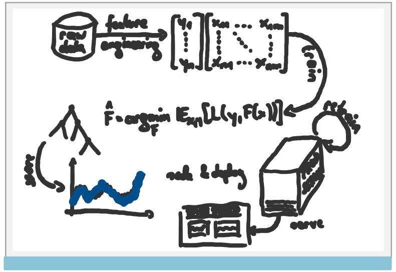
HOW MANY PARCELS WILL WE NEED TO DELIVER IN THE FUTURE?

Parcel volume per day

- Level: postal code (3,000)
- 60 days rolling into the future
- Delivery quantities (AZT)

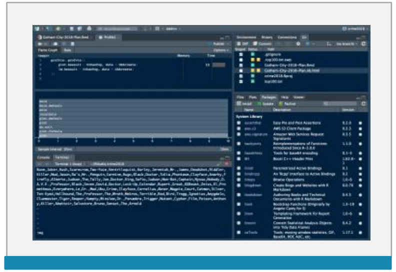


FROM EXPERIMENT TO DEPLOYMENT
OUR JOURNEY



Whiteboard phase

- Draft architecture
- Modeling strategy
- Everything seems easy☺



.gitignore
Gotham-City-2018-Plan.Rmd
Gotham-City-2018-Plan.nb.html
crime2018.Rproj
top100.txt

System Library

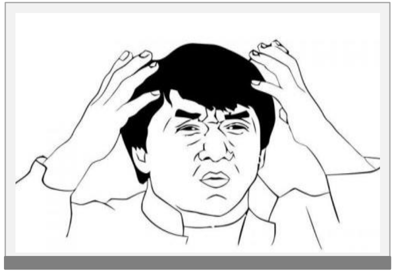
Name	Description	Version
assertthat	Easy Pre and Post Assertions	0.2.0
aws.s3	AWS S3 Client Package	0.3.3
aws.signature	Amazon Web Services Request Signatures	0.3.5
backports	Reimplementations of Functions Introduced Since R-3.0.0	1.1.0
base64enc	Tools for base64 encoding	0.1-3
BH	Boost C++ Header Files	1.62.0-1
bindr	Parametrized Active Bindings	0.1
bindrcpp	An 'Rcpp' Interface to Active Bindings	0.2
bitops	Bitwise Operations	1.0-6
blogdown	Create Blogs and Websites with R Markdown	0.0.78
bookdown	Authoring Books and Technical Documents with R Markdown	0.4.3
boot	Bootstrap Functions (Originally by Angelo Canty for S)	1.3-19
brew	Templating Framework for Report Generation	1.0-6
broom	Convert Statistical Analysis Objects into Tidy Data Frames	0.4.2
caTools	Tools: moving window statistics, GIF, Base64, ROC AUC, etc	1.17.1

Experiment phase

- Build model on sample
- Model improves
- Happy Data Scientists

Deadline phase

- Model training takes too long
- Model gets simplified to speed up training
- Performance suffers



Deployment phase

- Dev != Prod
- Server procurement slow
- Server not powerful enough for training
- Server too powerful for scoring

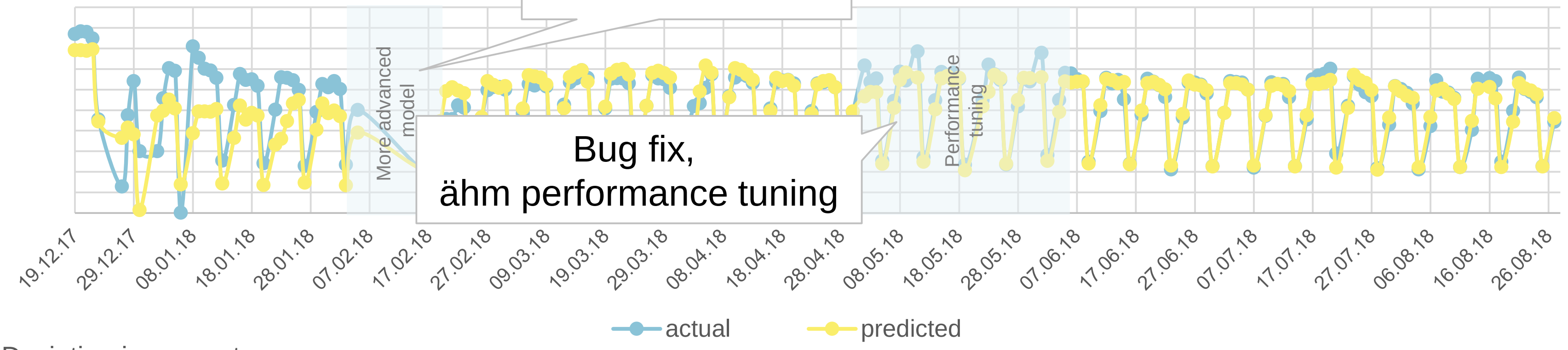
FORECASTING PERFORMANCE

2-DAYS-AHEAD

Performance

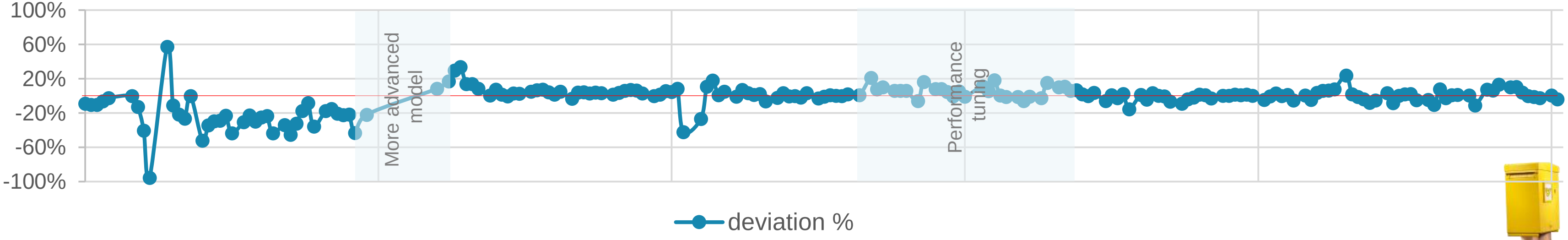
Parcel volumes per day Austria (2 days ahe

Simplified model



Bug fix,
ähm performance tuning

Deviation in percent

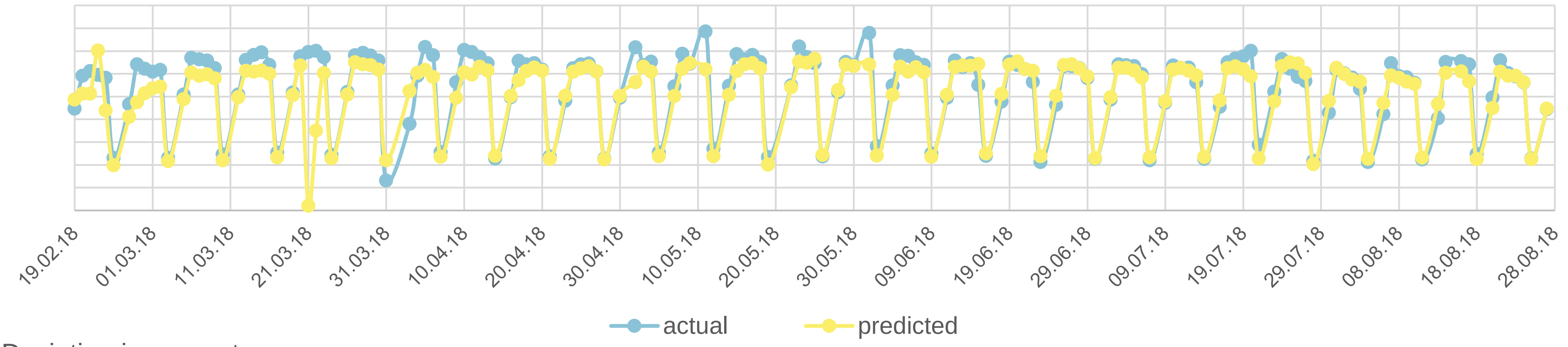


FORECASTING PERFORMANCE

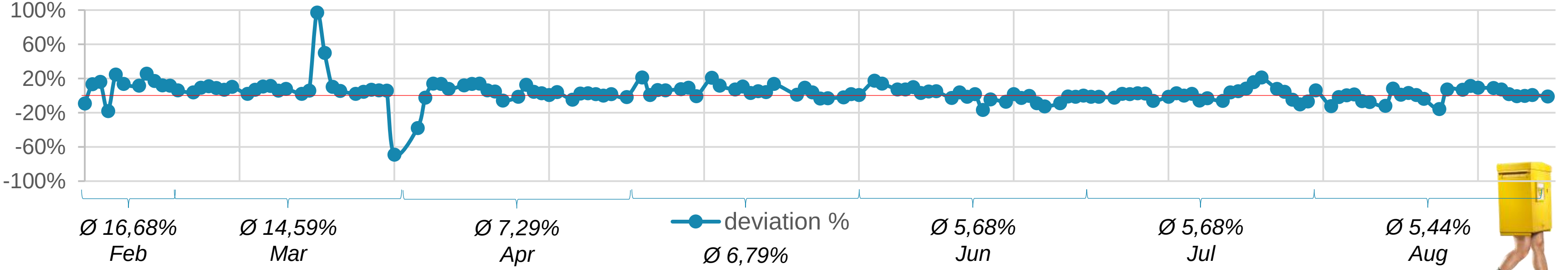
7-DAYS-AHEAD

Performance

Parcel volumes per day Austria (7 days ahead)



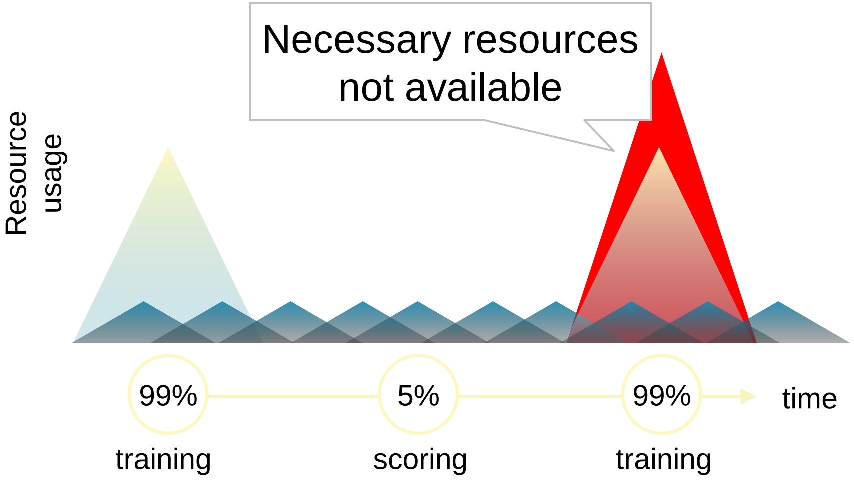
Deviation in percent



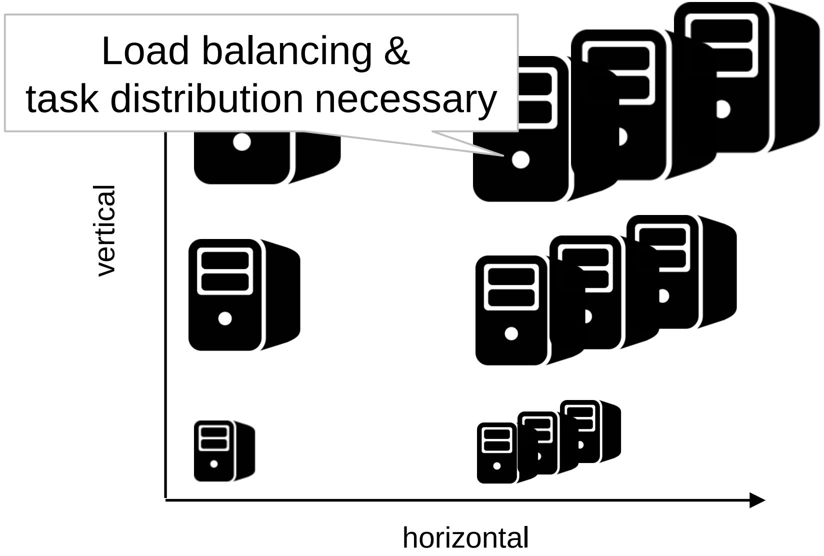
PROBLEMS IN DETAIL

SCALING COMPUTE RESOURCES ON DEMAND

Resource utilisation



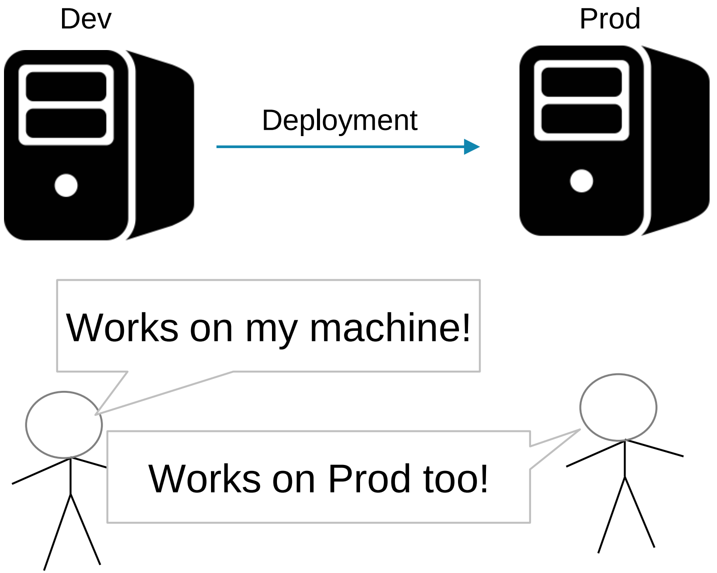
Scaling options



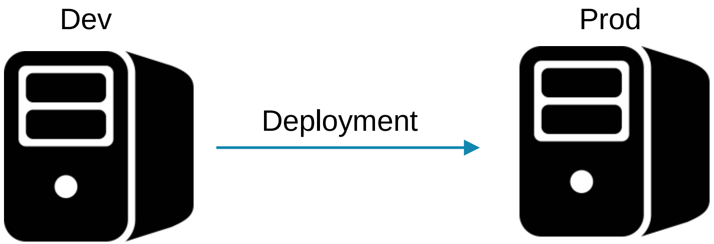
We want to be able to elastically scale **up** and **out** to meet our needs

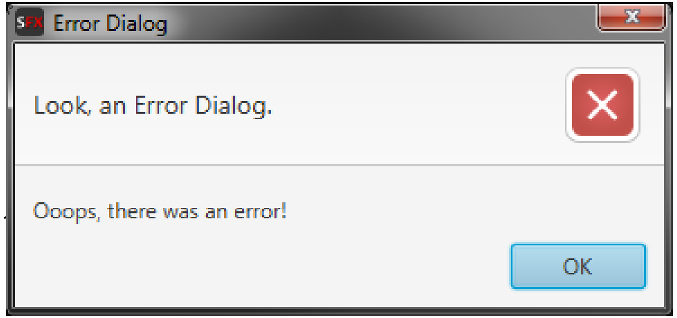
PROBLEMS IN DETAIL
DEV-PROD DISPARITY

Happy Path Deployments



~~Happy Path~~ Deployments





Look, an Error Dialog.

Ooops, there was an error!

Keeping Dev and Prod aligned is hard even with best intentions

HOW CAN WE SOLVE OUR PROBLEMS?

WE HAVE A LONG WISH LIST

Our wish list:

deploy automatically

scale compute

elastically

No vendor lock-in

uniform

architecture

Easy scheduling

eliminate dev-prod disparities

cheap

FPGAs

Easy management

no dependencies

GPUs

programming

language agnostic

Job Management Portal

Solutions we considered:

We have two dominant stacks:

- Microsoft .net + Azure
- SAP

Since we have SAP and .net/Azure support in-house, we looked primarily at these two stacks.

AGENDA

	01	02	03
Topics	Who we are (obligatory marketing stuff…)	Our problems	Our solution
	Data Science@Post AG: - Overview: Post AG - Our team	Deploying R at scale: - Scaling compute resources on demand - Reduce dev-prod disparity	Main components: - Docker - Azure Batch - VSTS

SOLUTIONS WE CONSIDERED

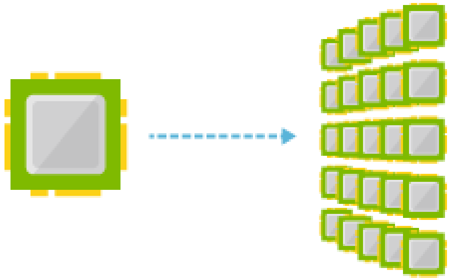
COMPARISON

Also take a look at Azure Databricks for Spark workloads

	Cloud vs. On-Premise Solution			
	Azure ML Studio	Azure ML Workbench	Azure Batch	SAP HANA (PAL)
Supported Languages	R/Python	Python	All	R (Python)
Supports GPUs		Yes	Yes	No
Dev-Prod parity	Easy	Easy	Easy	Hardish
R package available	AzureML	-	doAzureParallel	-
Independent upgrades	partially	partially	Yes	partially
Elastic scaling	-	Yes	Yes	No
Scheduling included	No	No	Yes	Yes
SCORE	◔	◑	◕	◑

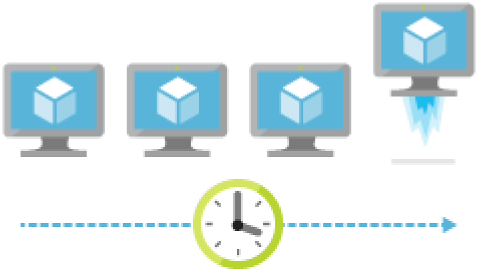
This looks very promising

AZURE BATCH OVERALL WINNER
FLEXIBLE, HIGH VALUE/MONEY AND LOW VENDOR LOCK-IN



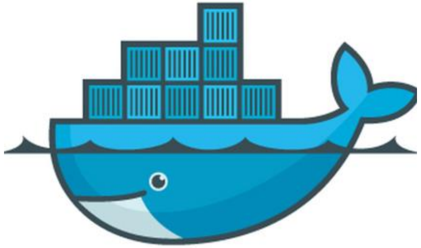
Scale up and out

Specify node sizes and types, e.g. GPU/CPU, RAM and get large discount on low-prio nodes



Scheduling integrated

Specify job schedule and resize pool based on number of outstanding tasks



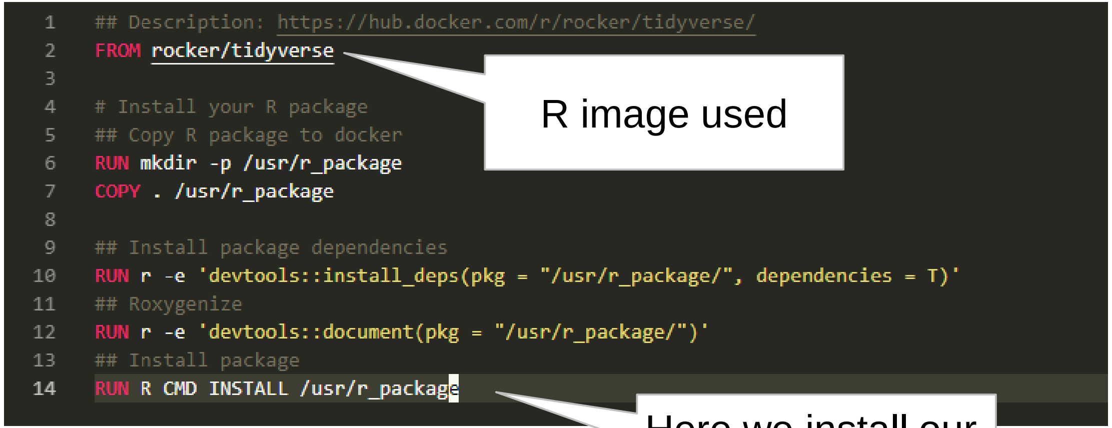
Docker support

Dev and Prod parity
Fits into CD pipeline

Source images: Microsoft and Docker homepage

DOCKER BRIEFLY EXPLAINED
BUILD ONCE – RUN ANYWHERE

Example Dockerfile

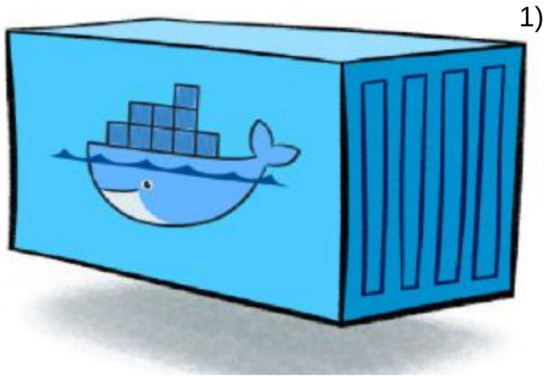


```
## Description: https://hub.docker.com/r/rocker/tidyverse/
FROM rocker/tidyverse

# Install your R package
## Copy R package to docker
RUN mkdir -p /usr/r_package
COPY . /usr/r_package

## Install package dependencies
RUN r -e 'devtools::install_deps(pkg = "/usr/r_package/", dependencies = T)'
## Roxygenize
RUN r -e 'devtools::document(pkg = "/usr/r_package/")'
## Install package
RUN R CMD INSTALL /usr/r_package
```

R image used

Here we install our package

Build Container



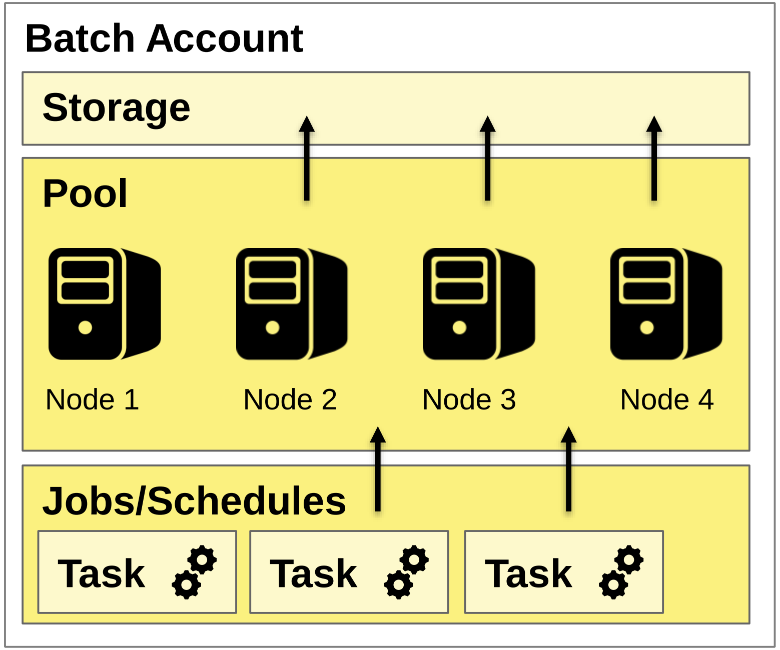
Container can be deployed to docker runtime in production: Dev = Prod (mostly☺)

Source image: 1) https://stephenafamo.com/blog/understanding-docker-containers/

AZURE BATCH

CONCEPTS

Components overview



Description

Pool ……… Definition of compute resources
- Node number/type
- Container settings
- Autoscaling
- Task number per node

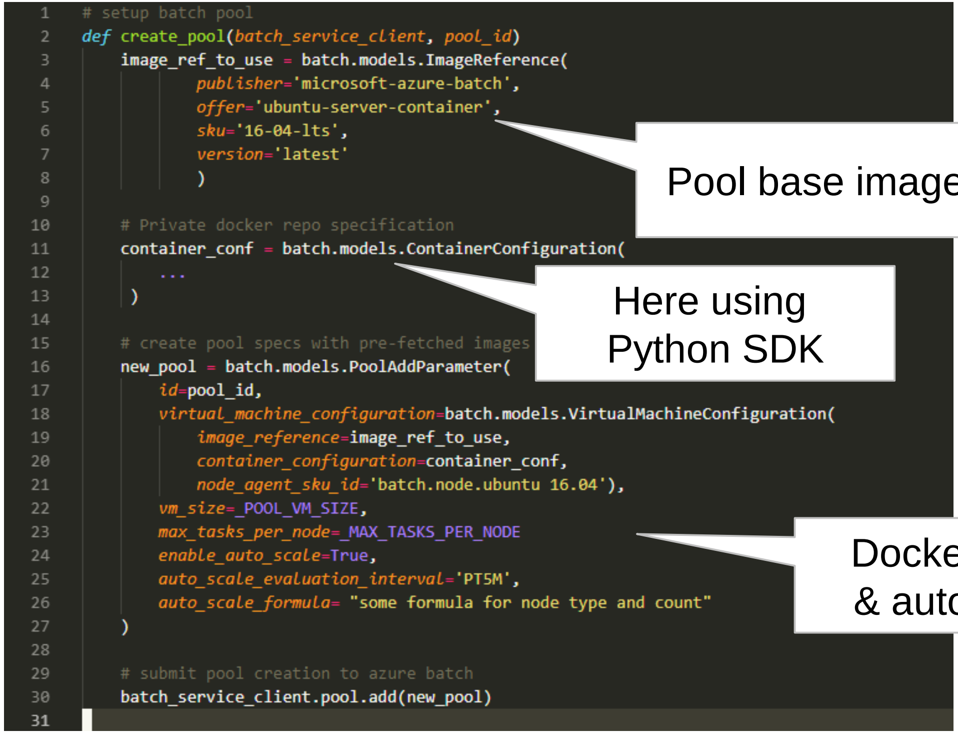
Schedule … Recurring job for a pool
- Job specification

Job ……….. Contains pool and task specs
- Job manager task

Task ………. Work specification
- Command line
- upload options

AZURE BATCH
SETUP OVERVIEW

Pool configuration



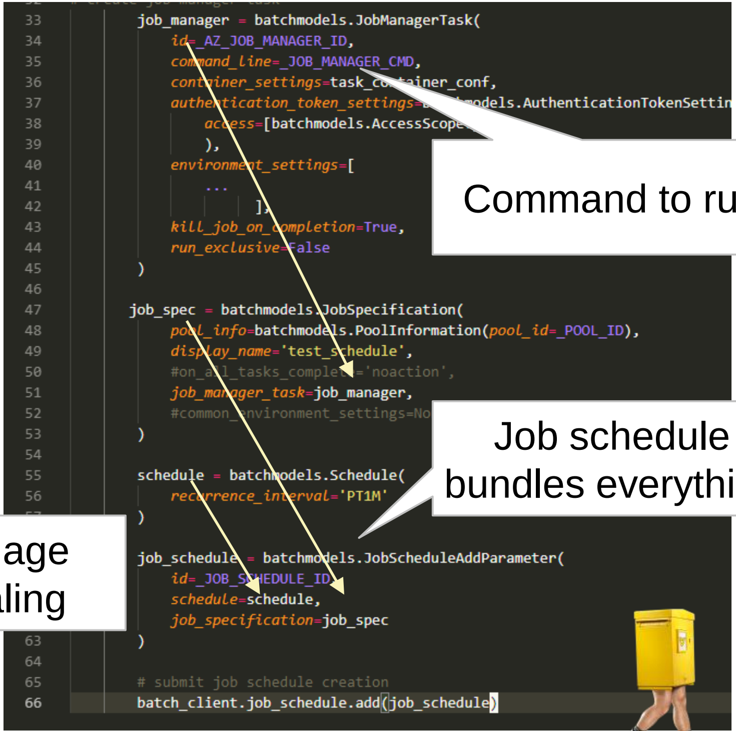
```
# setup batch pool
def create_pool(batch_service_client, pool_id):
    image_ref_to_use = batch.models.ImageReference(
            publisher='microsoft-azure-batch',
            offer='ubuntu-server-container',
            sku='16-04-lts',
            version='latest'
            )

    # Private docker repo specification
    container_conf = batch.models.ContainerConfiguration(
        ...
     )

    # create pool specs with pre-fetched images
    new_pool = batch.models.PoolAddParameter(
        id=pool_id,
        virtual_machine_configuration=batch.models.VirtualMachineConfiguration(
            image_reference=image_ref_to_use,
            container_configuration=container_conf,
            node_agent_sku_id='batch.node.ubuntu 16.04'),
        vm_size=_POOL_VM_SIZE,
        max_tasks_per_node=_MAX_TASKS_PER_NODE
        enable_auto_scale=True,
        auto_scale_evaluation_interval='PT5M',
        auto_scale_formula= "some formula for node type and count"
    )

    # submit pool creation to azure batch
    batch_service_client.pool.add(new_pool)
```

Pool base image

Here using Python SDK

Docker image & autoscaling

Job Manager Task

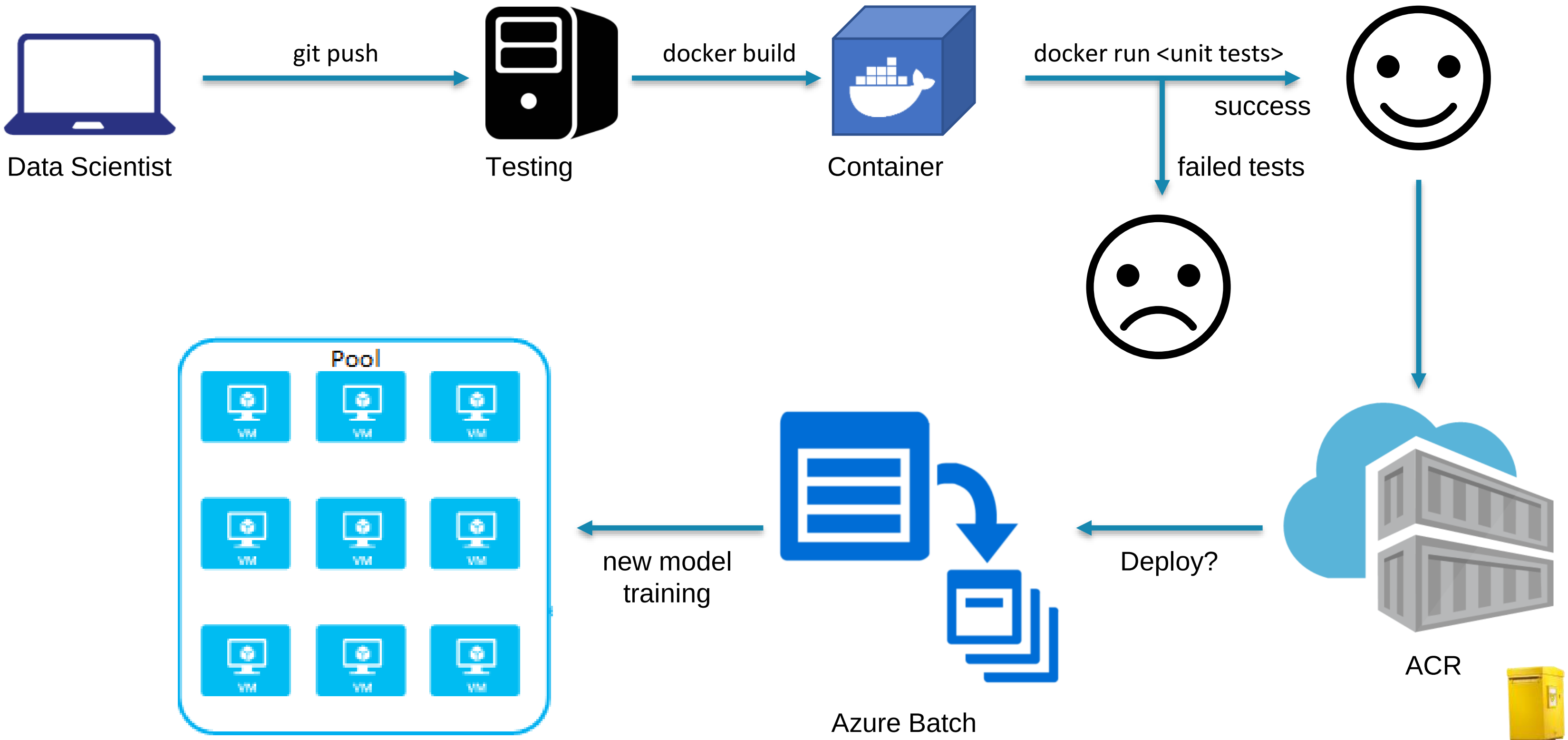


```
        job_manager = batchmodels.JobManagerTask(
            id=_AZ_JOB_MANAGER_ID,
            command_line=_JOB_MANAGER_CMD,
            container_settings=task_container_conf,
            authentication_token_settings=batchmodels.AuthenticationTokenSettin
                access=[batchmodels.AccessScope.
                ),
            environment_settings=[
                ...
                      ],
            kill_job_on_completion=True,
            run_exclusive=False
        )

       job_spec = batchmodels.JobSpecification(
            pool_info=batchmodels.PoolInformation(pool_id=_POOL_ID),
            display_name='test_schedule',
            #on_all_tasks_complete='noaction',
            job_manager_task=job_manager,
            #common_environment_settings=No
        )

        schedule = batchmodels.Schedule(
            recurrence_interval='PT1M'
        )

        job_schedule = batchmodels.JobScheduleAddParameter(
            id=_JOB_SCHEDULE_ID,
            schedule=schedule,
            job_specification=job_spec
        )

        # submit job schedule creation
        batch_client.job_schedule.add(job_schedule)
```

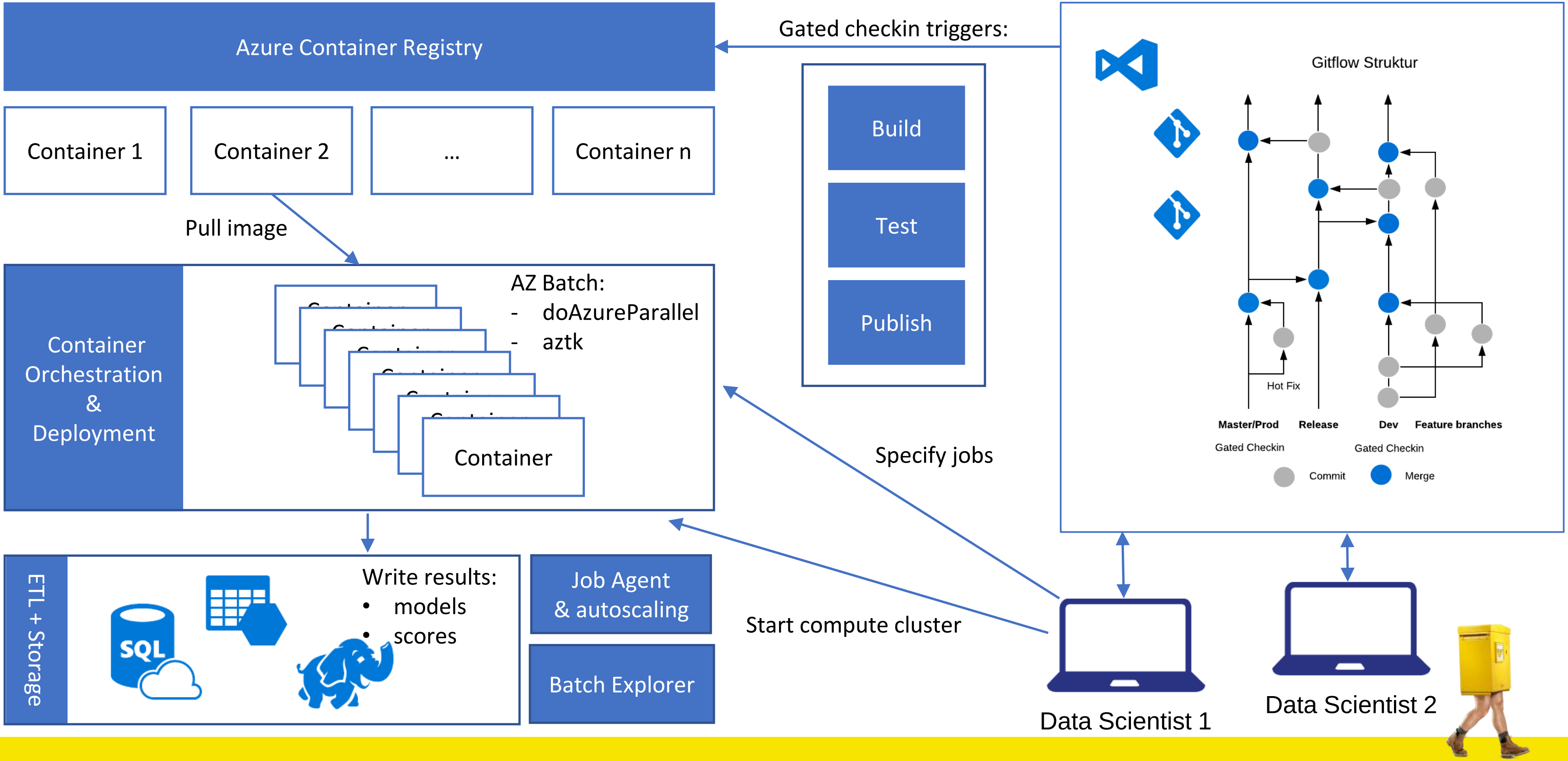
Command to run

Job schedule bundles everything

VSTS
CONTINUOUS DELIVERY PIPELINE (SIMPLIFIED)



OUR STACK
OVERVIEW



Azure Container Registry

Container 1

Container 2

...

Container n

Pull image

Container Orchestration & Deployment

AZ Batch:
- doAzureParallel
- aztk

Container

Write results:
- models
- scores

ETL + Storage

Job Agent & autoscaling

Batch Explorer

Gated checkin triggers:

Build

Test

Publish

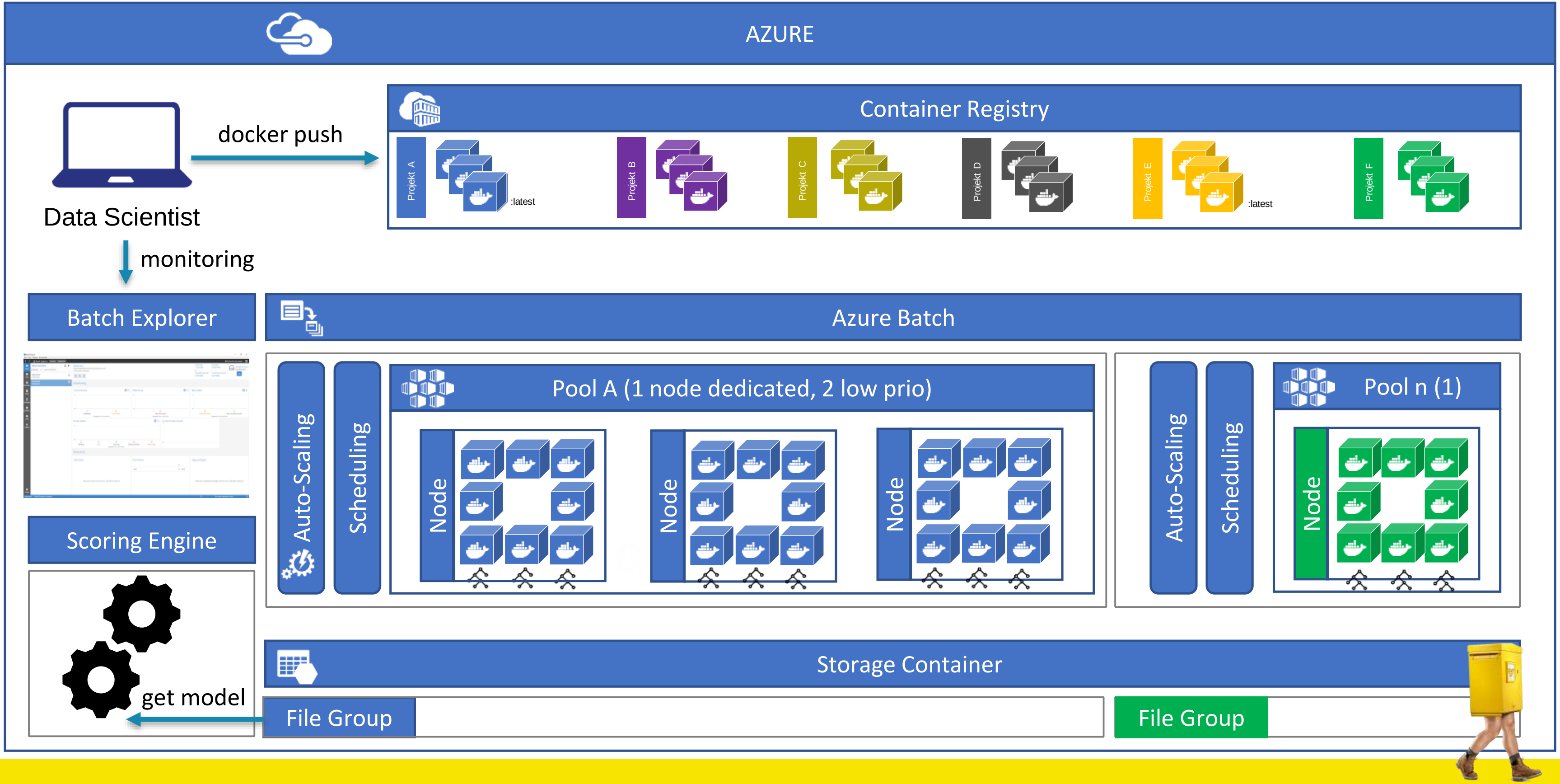
Specify jobs

Start compute cluster

Data Scientist 1

Data Scientist 2

OUR STACK
WE ARE CURRENTLY BUILDING A HPC ENVIRONMENT



AZURE

Data Scientist

docker push

monitoring

Container Registry

Projekt A :latest

Projekt B

Projekt C

Projekt D

Projekt E :latest

Projekt F

Batch Explorer

Azure Batch

Auto-Scaling

Scheduling

Pool A (1 node dedicated, 2 low prio)

Node

Node

Node

Auto-Scaling

Scheduling

Pool n (1)

Node

Scoring Engine

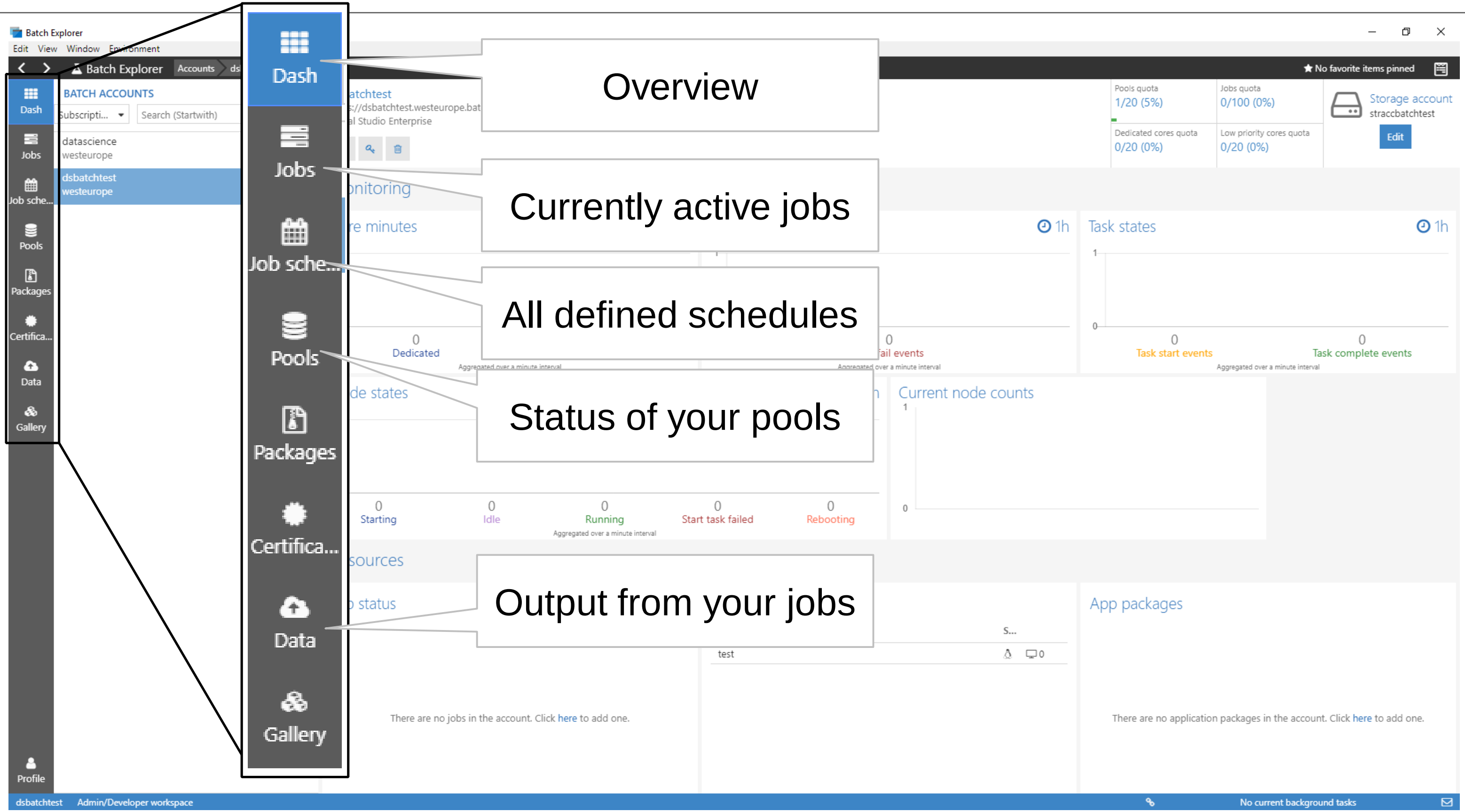
get model

Storage Container

File Group

File Group

MONITORING BATCH JOBS
WITH BATCH EXPLORER



There are no jobs in the account. Click here to add one.

There are no application packages in the account. Click here to add one.

Overview

Currently active jobs

All defined schedules

Status of your pools

Output from your jobs

OPEN ISSUES

SECURITY & EVALUATING DIFFERENT REAL-TIME SCORING OPTIONS

Security

- ~~Secret management with KeyVault~~
- ~~Azure Active Directory integration~~
- ~~VNET integration of pools~~
- Docker security best practices audit/check
- Disable public endpoints

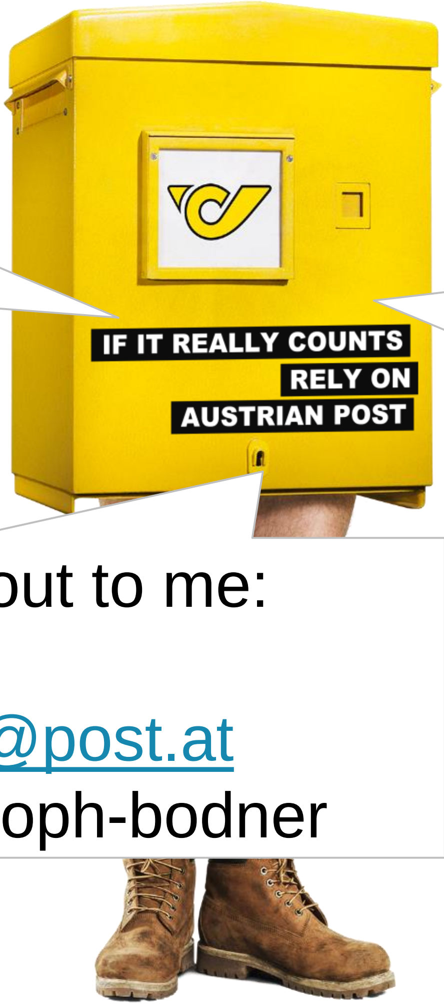
Real-time scoring options

- Using AzureML Studio
- Using Azure Model Management
- Kubernetes Cluster
- Azure Functions
- Azure Container Instances + Logic Apps
- …

What we are currently looking into

A bit of work is still open, but we plan to have everything wrapped up end of September

Thank you for your attention!



Any questions?

Feel free to reach out to me:

✉ christoph.bodner@post.at

in linkedin.com/in/christoph-bodner