

# MLOPS with R: An end-to-end process for building machine learning applications

David Smith  
Cloud Advocate, Microsoft  
@revodavid

# MLOps for R with Azure Machine Learning

David Smith | January 31, 2020



[github.com/revodavid/mlops-r](https://github.com/revodavid/mlops-r)

@revodavid at #nyrconf

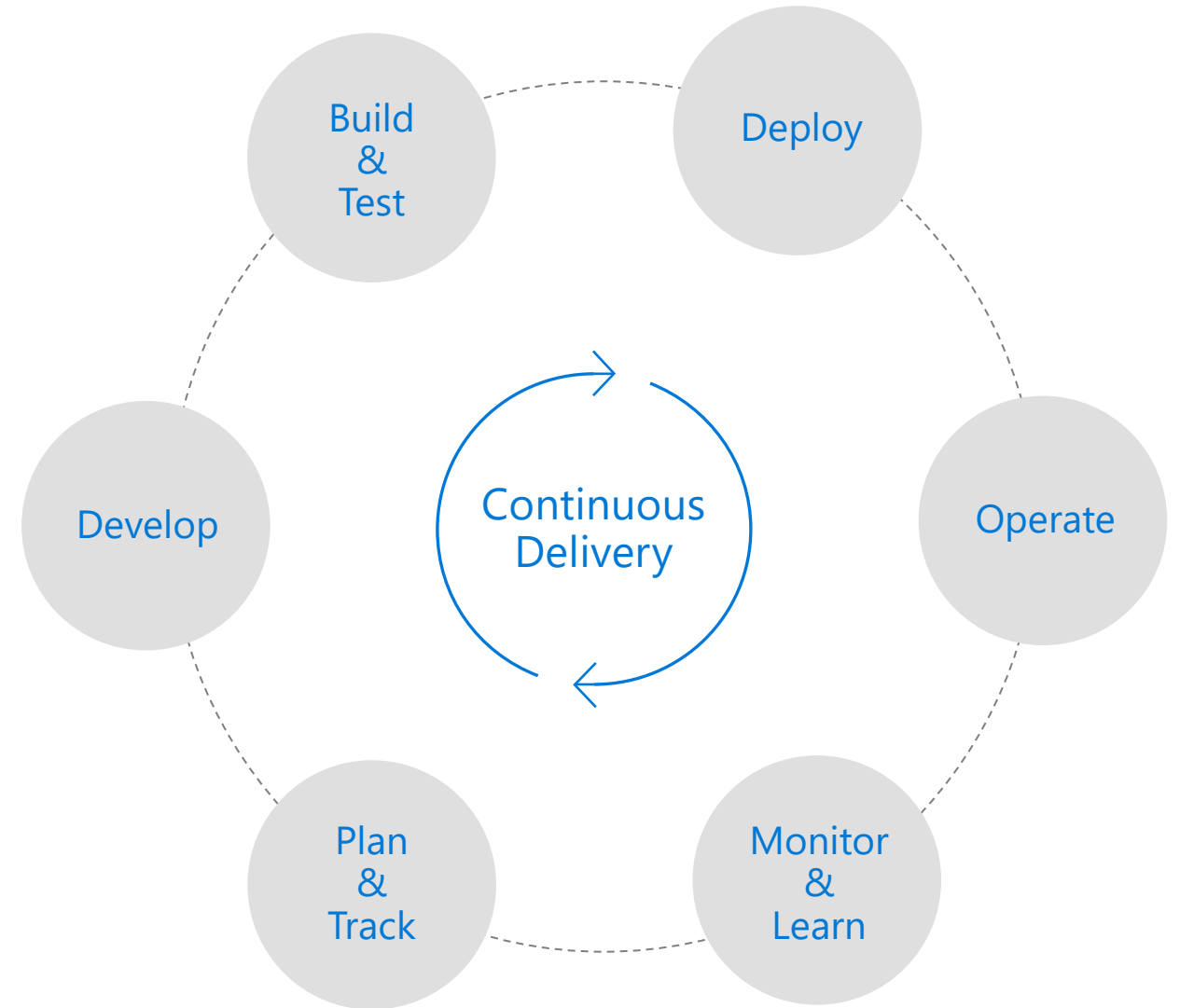
# What is DevOps?

People. Process. Products.



DevOps is the union of **people**, **process**, and **products** to enable continuous delivery of value to your end users. ”

Donovan Brown, Microsoft  
<http://bit.ly/WhatIs-DevOps>



DEVOPS	MLOPS
Manage code (source files)	Manage code (source files) Manage data files, notebooks, Rmd docs
Manage infrastructure (as code)	Manage infrastructure (as code) Manage environments (as code)
Source code control	Source code control Track experiment outcomes Manage data sets
Build executables Builds take hours (mostly) commodity compute	Train models Model training may take weeks or months GPU compute
Manage build versions	Manage model versions Manage reproducible environments
Tests (deterministic) Fix bugs with code	Tests (probabilistic) Fix bugs with code and/or data Model drift / model retraining

# Accident Fatality Probability Estimator

**Occupant Age:**  
16 69 95  
16 24 32 40 48 56 64 72 80 88 95

**Occupant gender:**  
f

**Occupant role:**  
pass

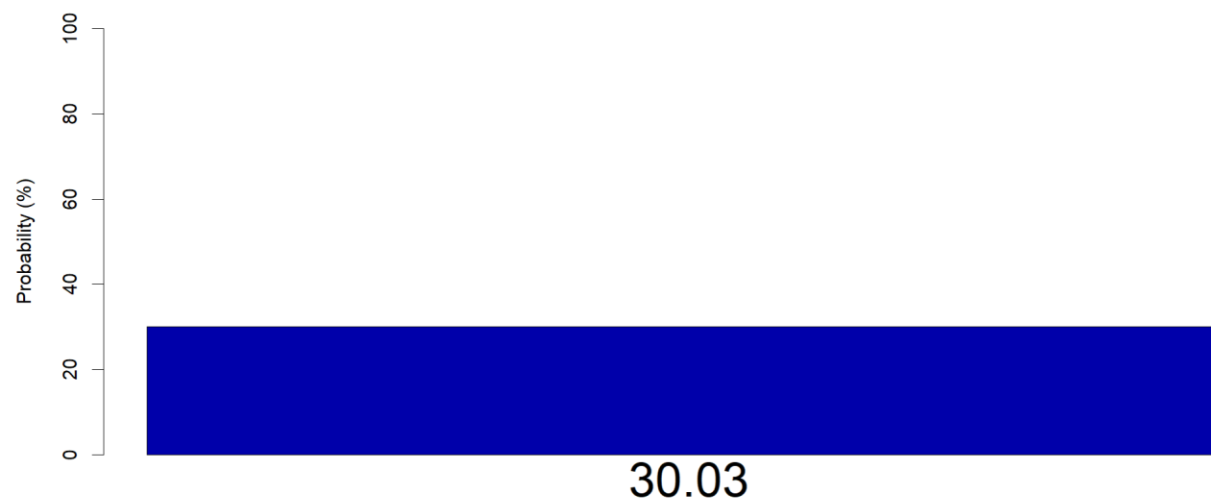
**Vehicle Year:**  
1,955 2,002 2,005  
1,955 1,960 1,965 1,970 1,975 1,980 1,985 1,990 1,995 2,000 2,005

**Seatbelt:**  
belted

**Airbag:**  
none

**Impact speed:**  
40-54

**Collision type:**  
notfrontal



# Azure Machine Learning service

Set of Azure  
Cloud Services



Python & R  
SDKs

---

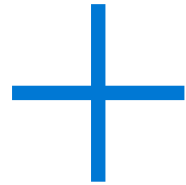
That enables  
you to:

- ✓ Prepare Data
- ✓ Build Models
- ✓ Train Models

- ✓ Manage Models
- ✓ Track Experiments
- ✓ Deploy Models

# Azure Machine Learning service

Set of Azure  
Cloud Services



Python & R  
SDKs



GitHub  
Actions

---

That enables  
you to:

- ✓ Prepare Data
- ✓ Build Models
- ✓ Train Models

- ✓ Manage Models
- ✓ Track Experiments
- ✓ Deploy Models

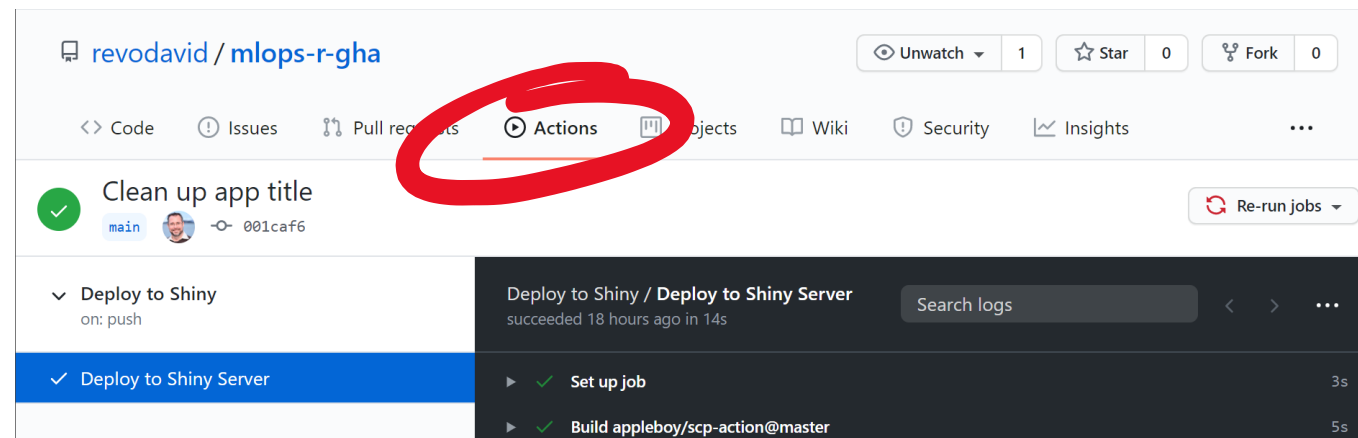
- ✓ Manage Code
- ✓ Collaborate
- ✓ Continuous Integration

# GitHub Actions TL;DR

Create YAML files in `.github/workflows` to define jobs

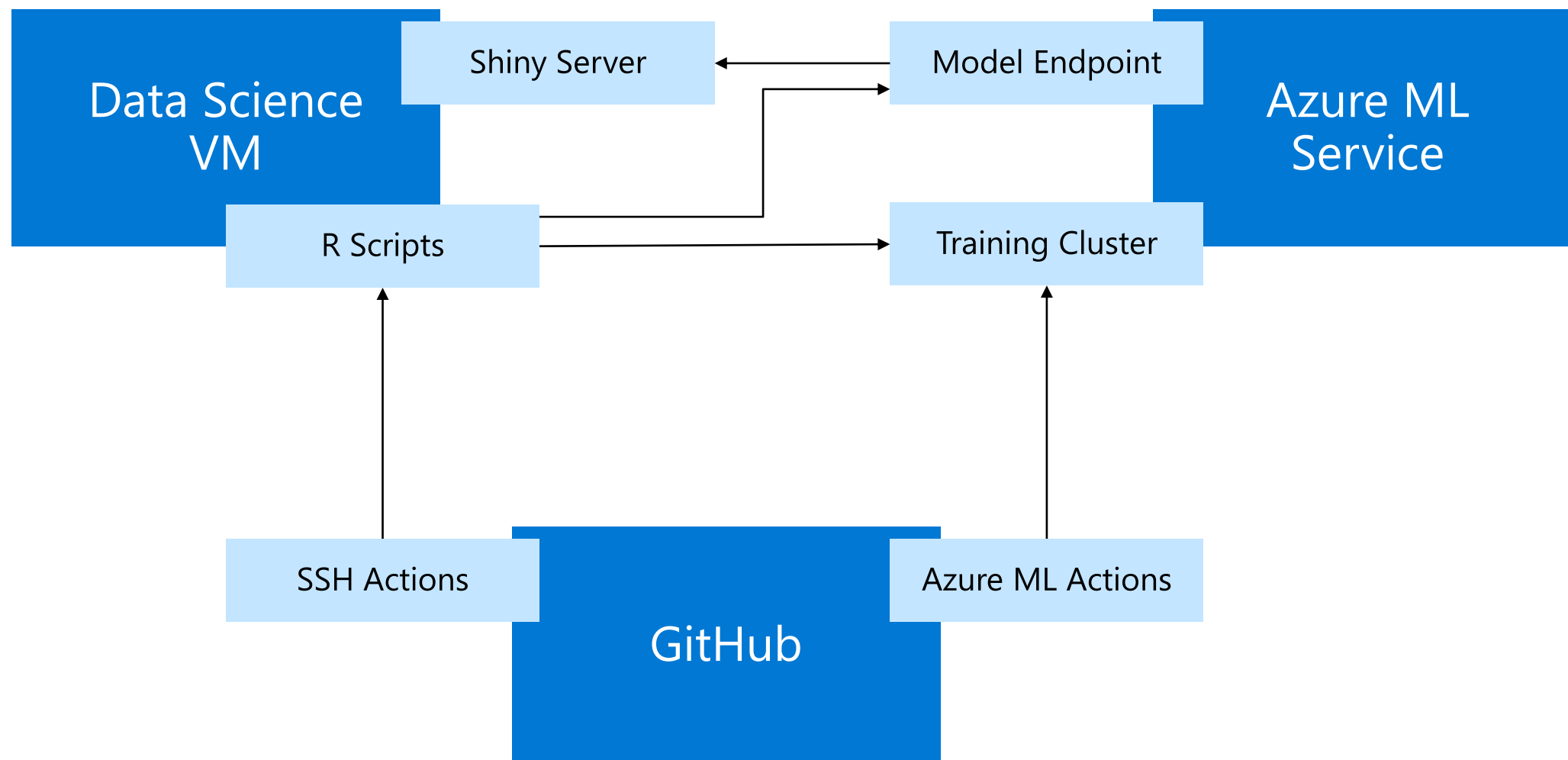
Search GitHub Actions Marketplace for pre-defined templates

Push, then monitor workflows from the Actions tab

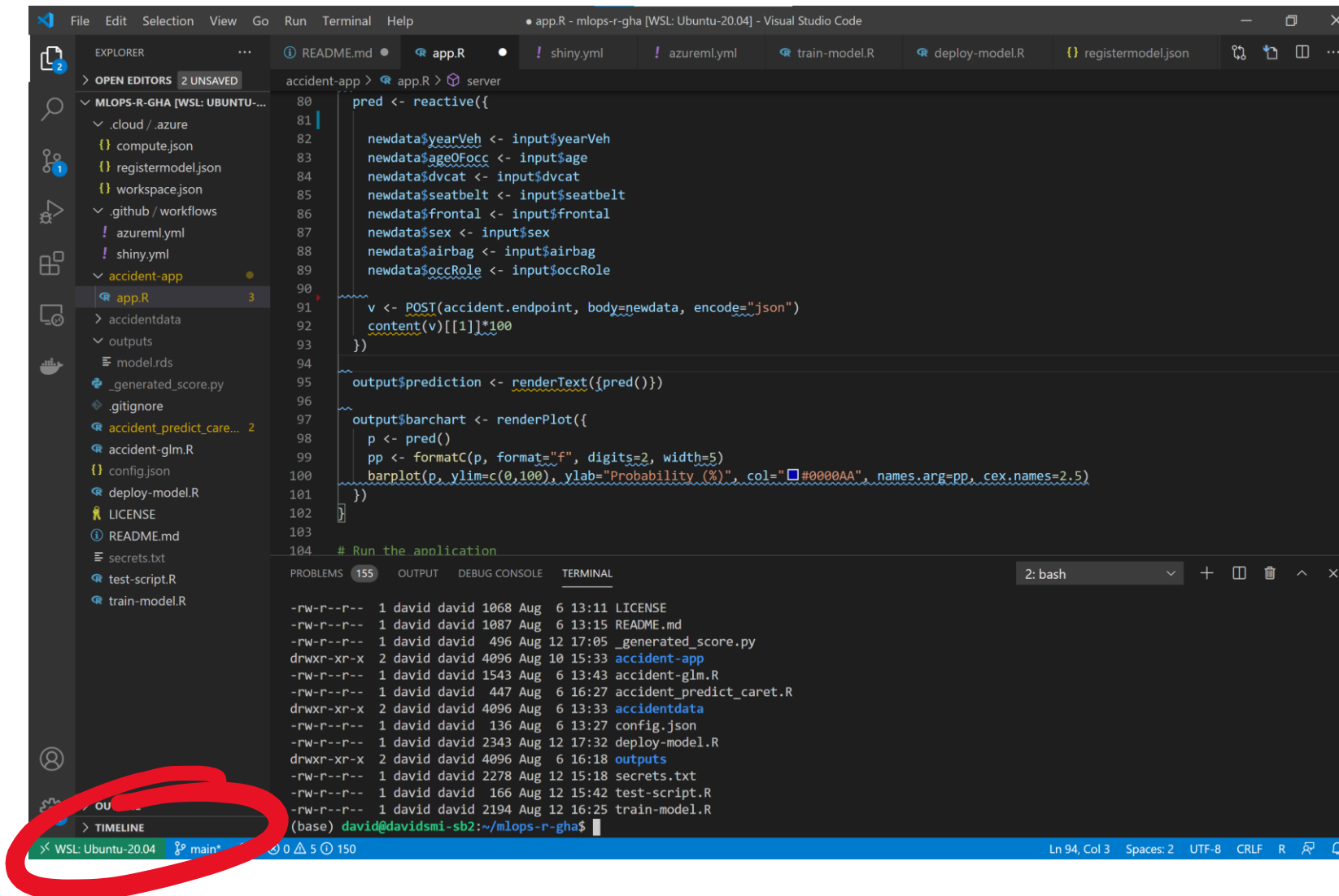




# "Accident" Shiny Application Architecture



# Dev environment: VS Code + WSL



The screenshot shows the Visual Studio Code interface with a WSL (Windows Subsystem for Linux) environment. The Explorer panel on the left shows the file structure of the 'accident-app' project. The main editor displays the 'app.R' file, which contains R code for a web application using the 'reactive' framework. The code defines input variables for vehicle details and accident data, sends a POST request to an endpoint, and renders the results as text and a bar chart. The terminal at the bottom shows the output of the application, including file permissions and the current directory path.

```
pred <- reactive({
  newdata$yearVeh <- input$yearVeh
  newdata$ageOfOcc <- input$age
  newdata$dvcac <- input$dvcac
  newdata$seatbelt <- input$seatbelt
  newdata$frontal <- input$frontal
  newdata$sex <- input$sex
  newdata$airbag <- input$airbag
  newdata$occRole <- input$occRole

  v <- POST(accident.endpoint, body=newdata, encode="json")
  content(v)[[1]]*100
})

output$prediction <- renderText({pred()})

output$barchart <- renderPlot({
  p <- pred()
  pp <- formatC(p, format="f", digits=2, width=5)
  barplot(p, ylim=c(0,100), ylab="Probability (%)", col="#0000AA", names.arg=pp, cex.names=2.5)
})

# Run the application
```

Terminal output:

```
(base) david@davidsmi-sb2:~/mlops-r-gha$
-rw-r--r-- 1 david david 1068 Aug 6 13:11 LICENSE
-rw-r--r-- 1 david david 1087 Aug 6 13:15 README.md
-rw-r--r-- 1 david david 496 Aug 12 17:05 _generated_score.py
drwxr-xr-x 2 david david 4096 Aug 10 15:33 accident-app
-rw-r--r-- 1 david david 1543 Aug 6 13:43 accident-glm.R
-rw-r--r-- 1 david david 447 Aug 6 16:27 accident_predict_caret.R
drwxr-xr-x 2 david david 4096 Aug 6 13:33 accidentdata
-rw-r--r-- 1 david david 136 Aug 6 13:27 config.json
-rw-r--r-- 1 david david 2343 Aug 12 17:32 deploy-model.R
drwxr-xr-x 2 david david 4096 Aug 6 16:18 outputs
-rw-r--r-- 1 david david 2278 Aug 12 15:18 secrets.txt
-rw-r--r-- 1 david david 166 Aug 12 15:42 test-script.R
-rw-r--r-- 1 david david 2194 Aug 12 16:25 train-model.R
```

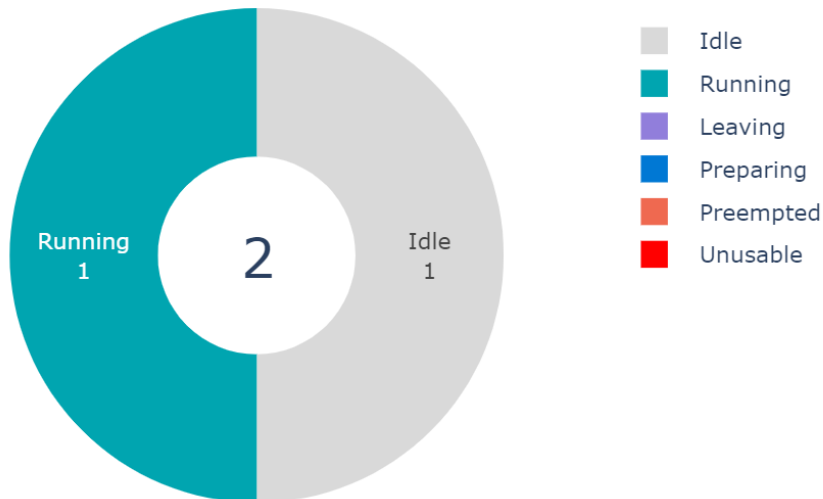
# Create 2-node training cluster

azureml.yml

```
# Connect or Create a Compute Target in Azure Machine Learning
- name: Connect/Create Azure Machine Learning Compute Target
  id: aml_compute_training
  uses: Azure/aml-compute@v1
  with:
    azure_credentials: ${ secrets.AZURE_CREDENTIALS }
```

compute.json

```
{
  "name": "rcluster",
  "compute_type": "amlcluster",
  "min_nodes": 0,
  "max_nodes": 4,
  "idle_seconds_before_scaledown": 600
}
```



- Unused nodes de-allocate automatically
- Increase idle timeout when developing

# Train model

## azureml.yml

```
# Connect to the Shiny VM to train the model
- name: Train model
  uses: JimCronqvist/action-ssh@master
  env:
    AZURE_CREDENTIALS: '${{ secrets.AZURE_CREDENTIALS}}'
  with:
    hosts: '${{ secrets.SHINYUSERNAME }}@${{ secrets.SHINYHOST }}'
    privateKey: '${{ secrets.SHINYKEY }}'
    command: |
      cd mlops-r-gha
      export AZURE_CREDENTIALS
      Rscript train-model.R
```

## train-model.R

```
est <- estimator(source_directory=".",
                 entry_script = "accident-glm.R",
                 script_params = list("--data_folder" = ds$path(target_path),
                                     compute_target = compute_target)
run <- submit_experiment(exp, est)
```

Experiments tracked with source scripts and recorded metrics

Control execution with command line parameters

Most packages pre-loaded, custom packages supported

# accident

Edit table Refresh Reset view Add chart

Add filter

☐ Include child runs

Run status

**0**  
Running

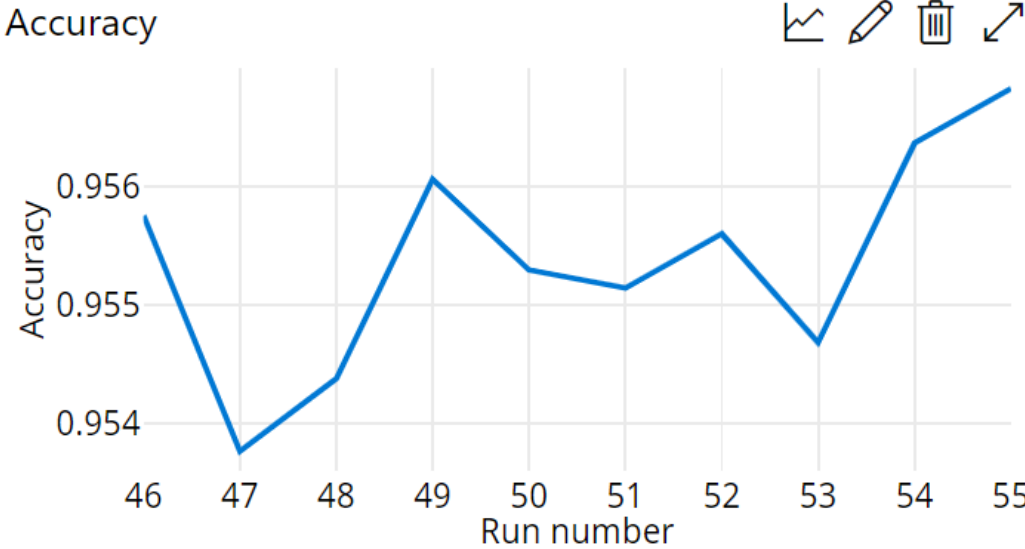
**3**  
Failed

**52**  
Completed

**0**  
Other

☐ Show only selected rows (10 selected )

	Run	Run ID	Status	Submitted time	Duration	Submitted by	Compute target
	Run 55	accide...	Completed	Aug 13, 2020 3:52 PM	4m 40s	Service principal	rcluster



# Deploy Model as REST endpoint

deploy-model.yml

```
inference_config <- inference_config(  
  entry_script = "accident_predict_caret.R",  
  source_directory = ".",  
  environment = r_env)  
  
aci_config <- aci_webservice_deployment_config(cpu_cores = 1, memory_gb = 0.5)  
  
aci_service <- deploy_model(ws,  
  'accidents-gha',  
  list(model),  
  inference_config,  
  aci_config)  
wait_for_deployment(aci_service, show_output = TRUE)
```

Name	Description	Created on	Created by	Updated on
<a href="#">accidents-gha</a>	--	August 13, 2020 4:01 PM	d2fbafa0-336d-4ae2-83da-7c7df...	August 13, 2020 4:01 PM

# Integrate model into Shiny app

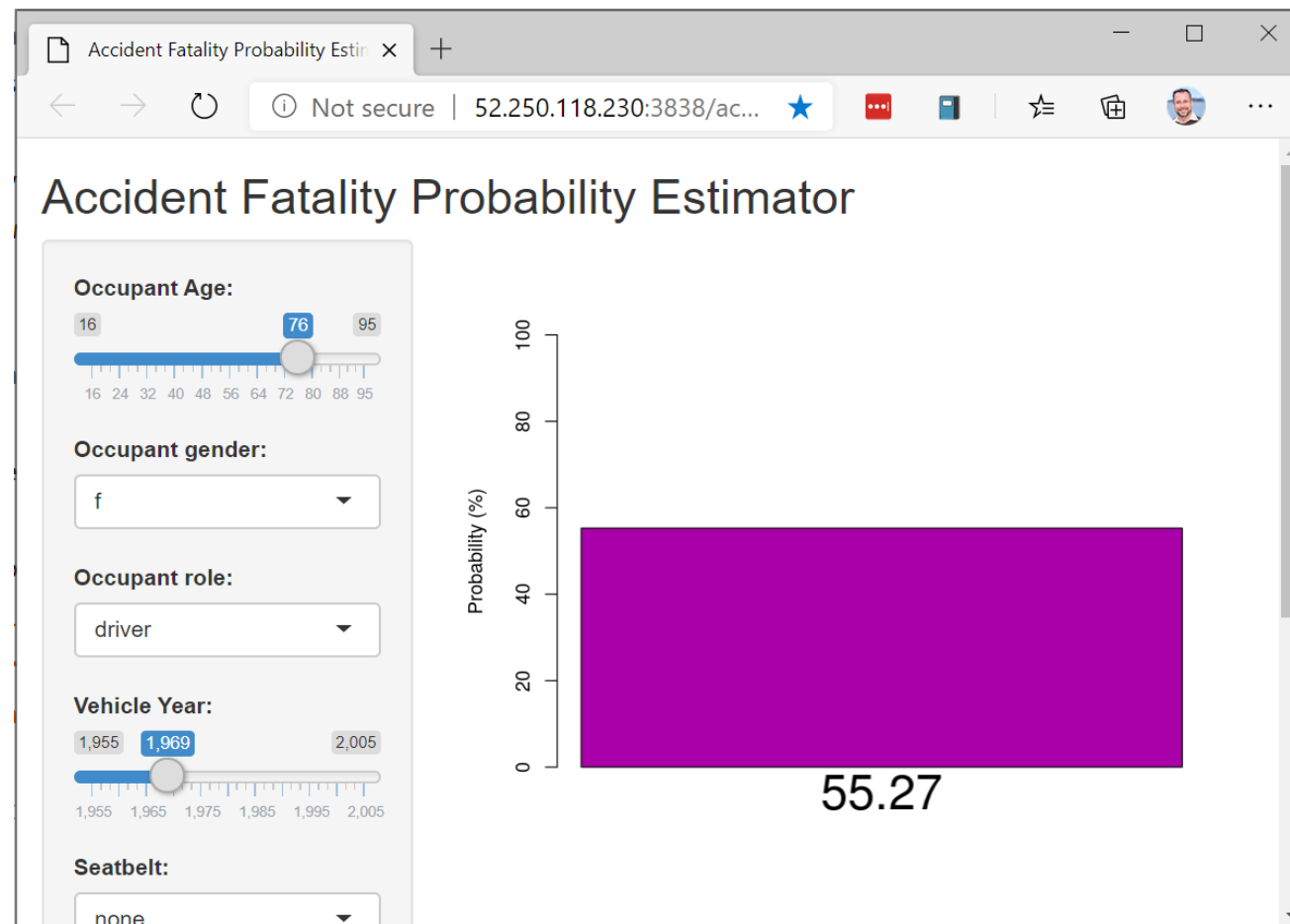
app.R

```
library(httr)
v <- POST(accident.endpoint,
          body=input,
          encode="json")
```

```
pred <- content(v)[[1]]
```

shiny.yml

```
- name: copy files via ssh key
  uses: appleboy/scp-action@master
  with:
    host: ${ secrets.SHINYHOST }
    username: ${ secrets.SHINYUSERNAME }
    key: ${ secrets.SHINYKEY }
    source: "accident-app/app.R"
    target: "~"
```



# Demo: Actions in action



# Costs per day\*

\*\$US. Cost vary by region and service. Example for illustration.

GitHub Actions	\$0.00
Azure ML service	\$0.00 (Studio, logging, orchestration)
Training cluster	\$0.10 (as needed)
Scoring endpoint	\$1.67 (Azure Container Instances)
Shiny Server	\$3.59 (Data Science VM, 4CPU 14Gb)
<b>TOTAL</b>	<b>\$5.36</b>

Azure subscription with \$200 in free credits: **[aka.ms/AML-NYR](https://aka.ms/AML-NYR)**

# Thank you!

David Smith  
Cloud Advocate, Microsoft  
@revodavid



Slides and links:  
**[github.com/revodavid/mlops-r-gha](https://github.com/revodavid/mlops-r-gha)**

# Azure ML service SDK for R

Open-source R package for use with CRAN R: **azuremlsdk**

- Create Workspaces, Experiments, Compute, Models, and other artifacts with R commands
- Use any R function/package (and track requirements for deployment)
- HyperDrive support: smart hyperparameter search with parallel compute
- Publish models as web services (in Azure or your own infra)
- Trigger training / deployment pipelines from CI/CD services

# Try it yourself!

1. Fork [github.com/revodavid/mlops-r-gha](https://github.com/revodavid/mlops-r-gha)
2. Deploy Shiny Server VM
3. Add secrets
4. Trigger Actions
  - Update training script / data file (retrain model)
  - Update app.R (redploy Shiny app)



## GitHub Actions

 <https://azure.com/pipelines>

Free **GH** **ACTIONS** for  
publ **ACTIONS** across  
Up **ACTIONS** across  
Windows, Linux and macOS

[github.com/revodavid/mlops-r-gha](https://github.com/revodavid/mlops-r-gha)



## Azure ML Service

 <https://azure.com/ml>

**Free** workspaces, experiments,  
model registry

Standard Azure rates for compute  
and deployment **or** use your own  
servers for free

@revodavid at #nyrconf