Depend on Docker

Get IT done with Docker on Azure

Microsoft Azure
+ AI Conference

CO-PRODUCED BY
Microsoft & DEVintersection

BAKER HUGHES
a GE company

Alex Iankoulski

Principal Software Architect

Baker Hughes, a GE Company
https://www.linkedin.com/in/alex-iankoulski/

BAKER HUGHES
a GE company

Mahadevan Balasubramaniam

Principal Data Scientist

Baker Hughes, a GE Company
https://www.linkedin.com/in/bmdevan/

Arun Subramaniyan

VP Data Science & Analytics

Baker Hughes, a GE Company
https://www.linkedin.com/in/arunsubramaniyan/

Microsoft Azure
+AI Conference
CO-PRODUCED BY
Microsoft & DEVintersection

Some problems we care about

Every 2 Seconds

A GE Jet engine takes off in the world

~30%

Of the world's power is produced by a GE turbine

We are the first

Fullstream O&G Company

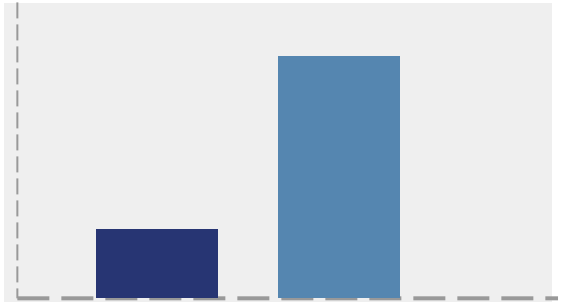
Exploration, Production, Transportation, Operations

We need to model mission critical systems accurately

Scale and complexity of industrial data

O&G	Aviation	Inspections	ERP Systems
Drilling data 0.3GB/well/day	**A flight from NY to London** ~1 TB	**Ultrasound: tubes** 1.2TB/8 hours	**Transactional data**
Wireline data 5GB/well/day	**Atmospheric data** 0.1GB/day	**Corrosion monitoring** 1GB/site/year	**Logistics reports**
Fiber optic data 0.1GB/well/day		**Pipeline inspection** 1.5TB/600 km	**Maintenance logs**
ESP monitoring 0.4GB/well/day			**FSE reports**
Seismic data 500GB/survey			**Inventory reports**

10 - 100x More volume
100 - 1000x More velocity

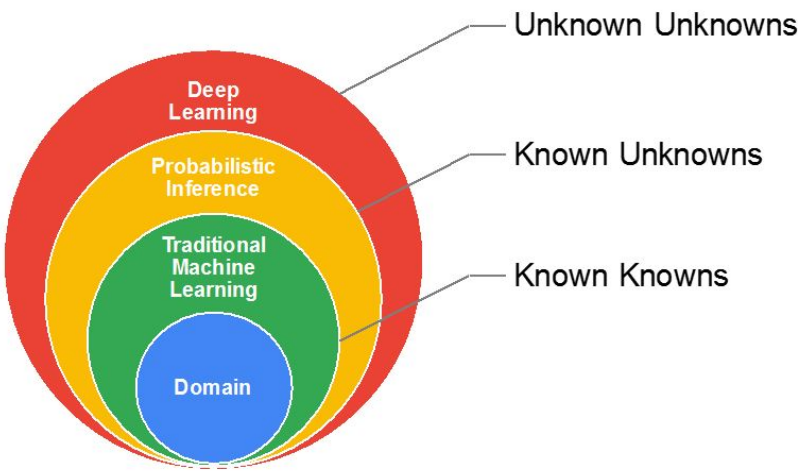


ERP Systems | Industrial

Hundreds of assets, multiple data silos

Digital Twins

A live up-to-date digital representation of an asset, system, or process.



Hybrid models

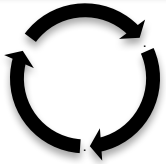
Physics-based — Known Knowns

+

Probabilistics — Known Unknowns

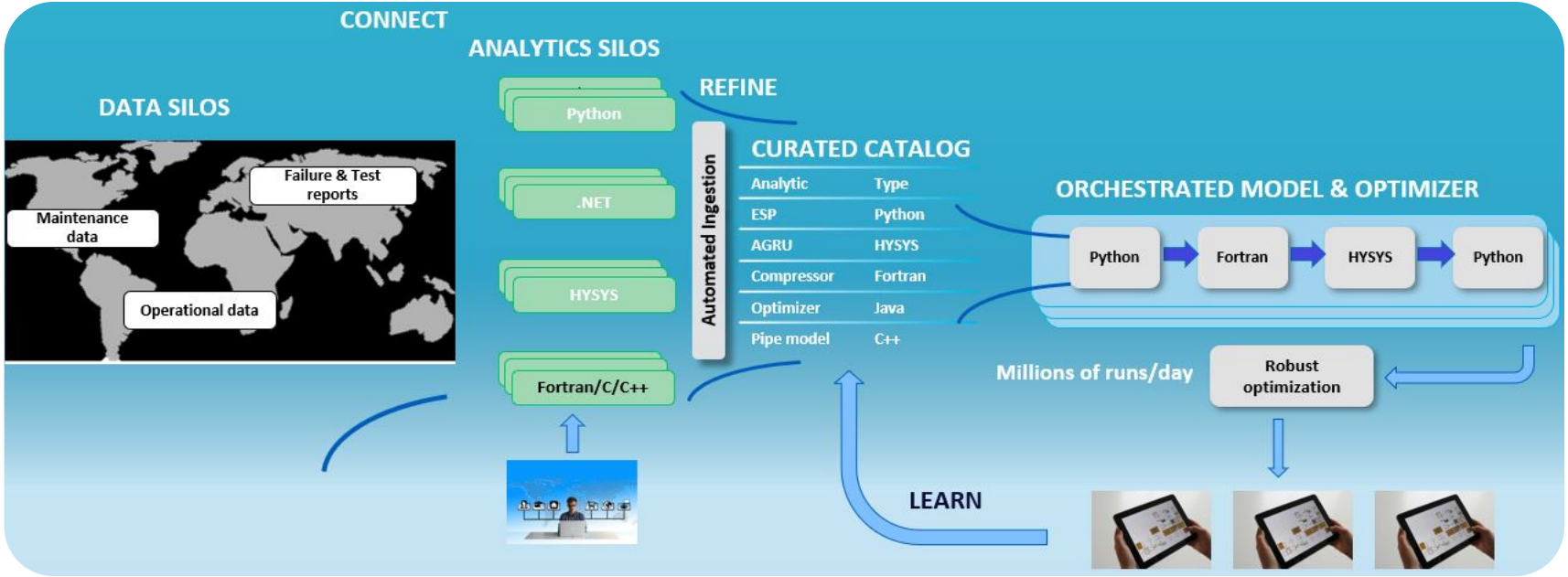
+

Deep learning (AI) — Unknown Unknowns



- Continuously tuned
- Scalable
- Adaptable

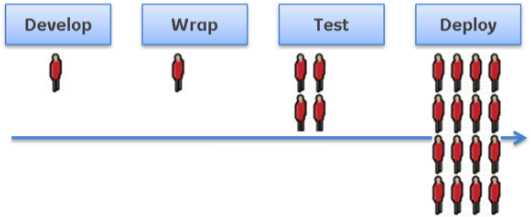
Connecting Data & Analytics



SILOS OF DATA/ANALYTICS + FEDERATION + SCALE + CONSTANT LEARNING = SCALABLE COMPUTING

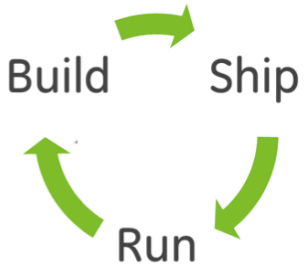
Democratize analytics through containers and cloud

Pre-Container/Cloud



- Production deployment: 6-12 months
- New developer setup: 1 week
- Quarterly releases

Post-Container/Cloud



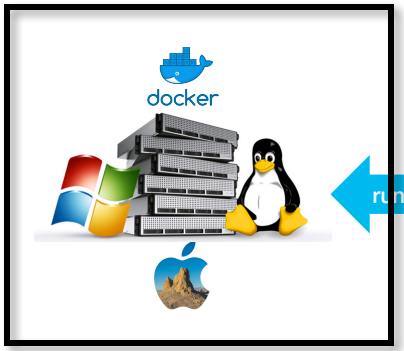
- Production: ~ 1 hour
- New developer setup: 1 hour
- Continuous releases

Kubernetes allows polyglot auto-scaling and "self"-healing at scale

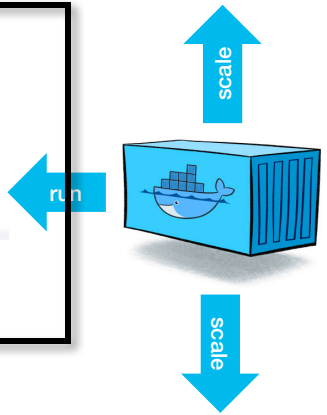
Scale up & down at will

- Ship and Run anywhere

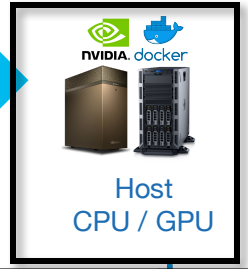
Demo 1
Analytics in Different OS



Operating System



Cluster
CPU / GPU



Host
CPU / GPU

Edge / IoT Device
CPU / GPU

Platform

Demo 4
Scale Up / Down
Containers

Demo 2
Leverage
GPU's

Demo 3
Encrypted
Transfer

Orchestrator

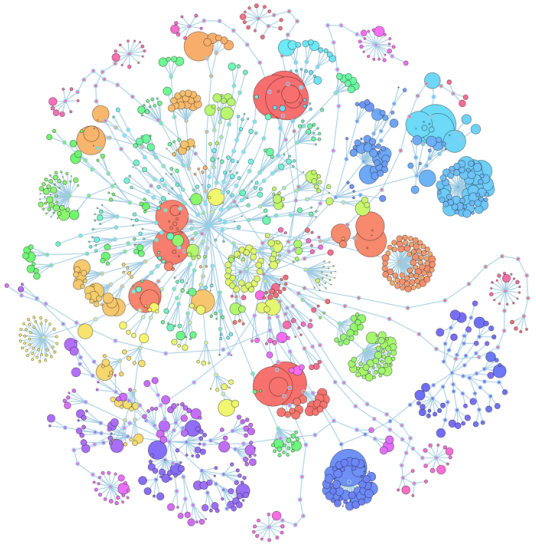
Infrastructure

Microsoft Azure
+AI Conference
CO-PRODUCED BY
Microsoft & DEVintersection

Depend on Docker (DOD)

- A philosophy – don't install anything, build a container instead
- A type of project – needs only access to source and Docker
- Open source – https://github.com/bhgedigital/depend-on-docker
- Easy to use – create a new DoD project with a single command
- Flexible – defaults provided, but customizable
- Universal – works with Linux and Windows containers

DOD Example - SEMTK



Example: semtk-opensource
Code Flower

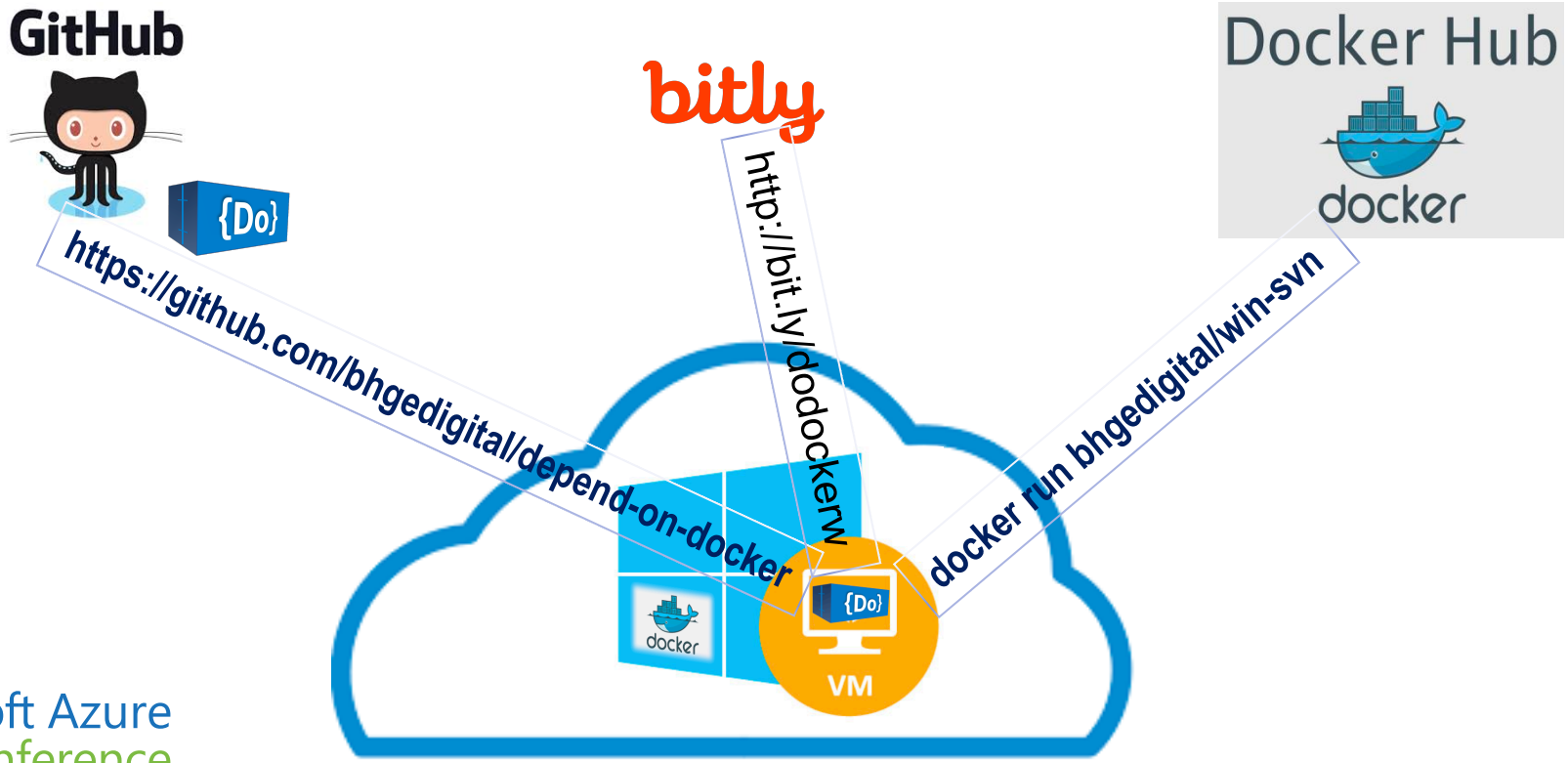
https://github.com/ge-semtk/semtk

CONTAINERIZED

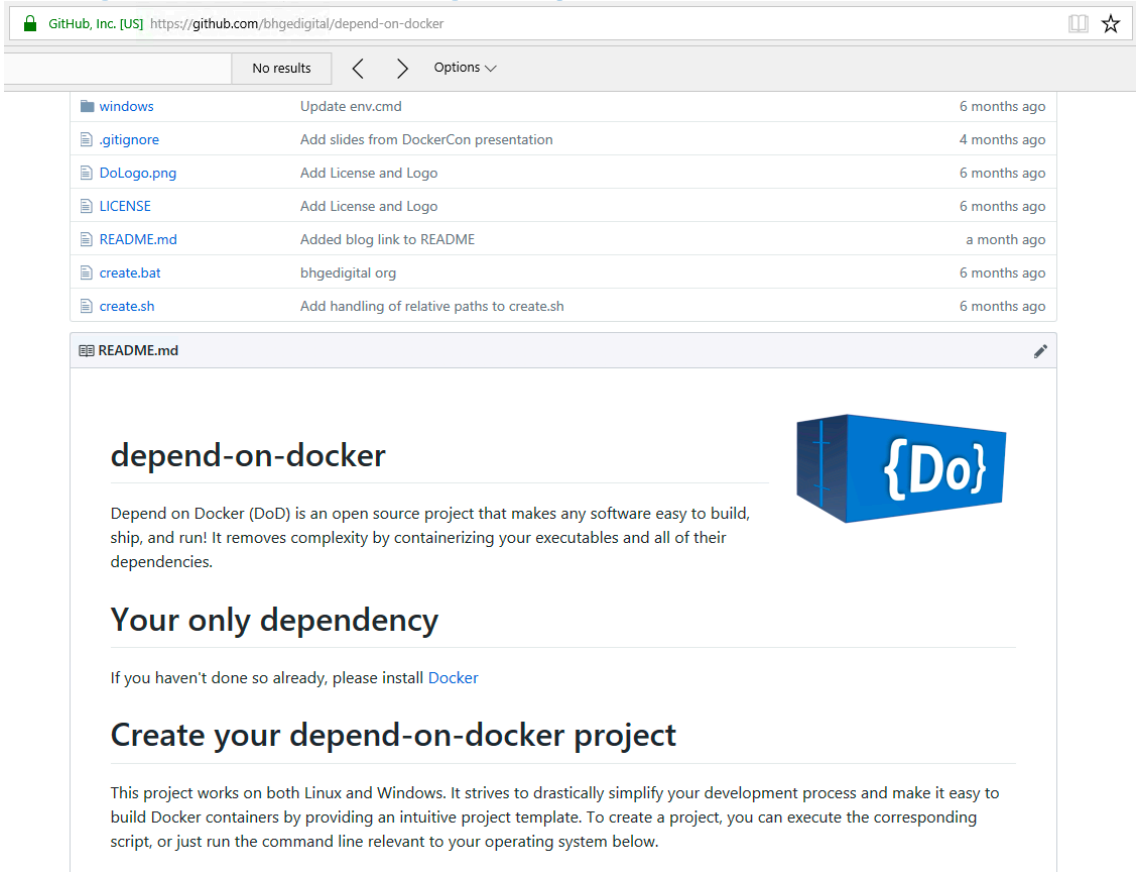
Containerized application
{dependencies encapsulated,
complexity reduced}

```
iankouls@1403111-C02T82RGGTFM:~/Projects/2018/github.com/iankoulski/semtk
> ./compose.sh up -d
Creating network "host_semtknet" with driver "bridge"
Creating host_semtk-sparqlgraph-results_1 ... done
Creating host_semtk-nodegroup-store_1     ... done
Creating host_semtk-ingestion_1           ... done
Creating host_semtk-hive_1                ... done
Creating host_semtk-sparqldb_1            ... done
Creating host_semtk-nodegroup_1           ... done
Creating host_semtk-dispatch_1            ... done
Creating host_semtk-nodegroup-execution_1 ... done
Creating host_semtk-sparql-query_1        ... done
Creating host_semtk-sparqlgraph-status_1  ... done
Creating host_semtk-sparqlgraph-web_1     ... done
Creating host_semtk-ontology-info_1       ... done
iankouls@1403111-C02T82RGGTFM:~/Projects/2018/github.com/iankoulski/semtk
> ./compose.sh ps
            Name                          Command               State                        Ports
-------------------------------------------------------------------------------------------------------------------------
host_semtk-dispatch_1              /bin/sh -c cd /service; ls ...   Up      0.0.0.0:12053->8080/tcp
host_semtk-hive_1                  /bin/sh -c cd /service; ls ...   Up      0.0.0.0:12055->8080/tcp
host_semtk-ingestion_1             /bin/sh -c cd /service; ls ...   Up      0.0.0.0:12091->8080/tcp
host_semtk-nodegroup-execution_1   /bin/sh -c cd /service; ls ...   Up      0.0.0.0:12058->8080/tcp
host_semtk-nodegroup-store_1       /bin/sh -c cd /service; ls ...   Up      0.0.0.0:12056->8080/tcp
host_semtk-nodegroup_1             /bin/sh -c cd /service; ls ...   Up      0.0.0.0:12059->8080/tcp
host_semtk-ontology-info_1         /bin/sh -c cd /service; ls ...   Up      0.0.0.0:12057->8080/tcp
host_semtk-sparql-query_1          /bin/sh -c cd /service; ls ...   Up      0.0.0.0:12050->8080/tcp
host_semtk-sparqldb_1              /bin/sh -c /startup.sh           Up      0.0.0.0:1111->1111/tcp, 0.0.0.0:2420->8890/tcp
host_semtk-sparqlgraph-results_1   /bin/sh -c cd /service; ls ...   Up      0.0.0.0:12052->8080/tcp
host_semtk-sparqlgraph-status_1    /bin/sh -c cd /service; ls ...   Up      0.0.0.0:12051->8080/tcp
host_semtk-sparqlgraph-web_1       /bin/sh -c /usr/local/tomc ...   Up      0.0.0.0:8860->8080/tcp
iankouls@1403111-C02T82RGGTFM:~/Projects/2018/github.com/iankoulski/semtk
>
```

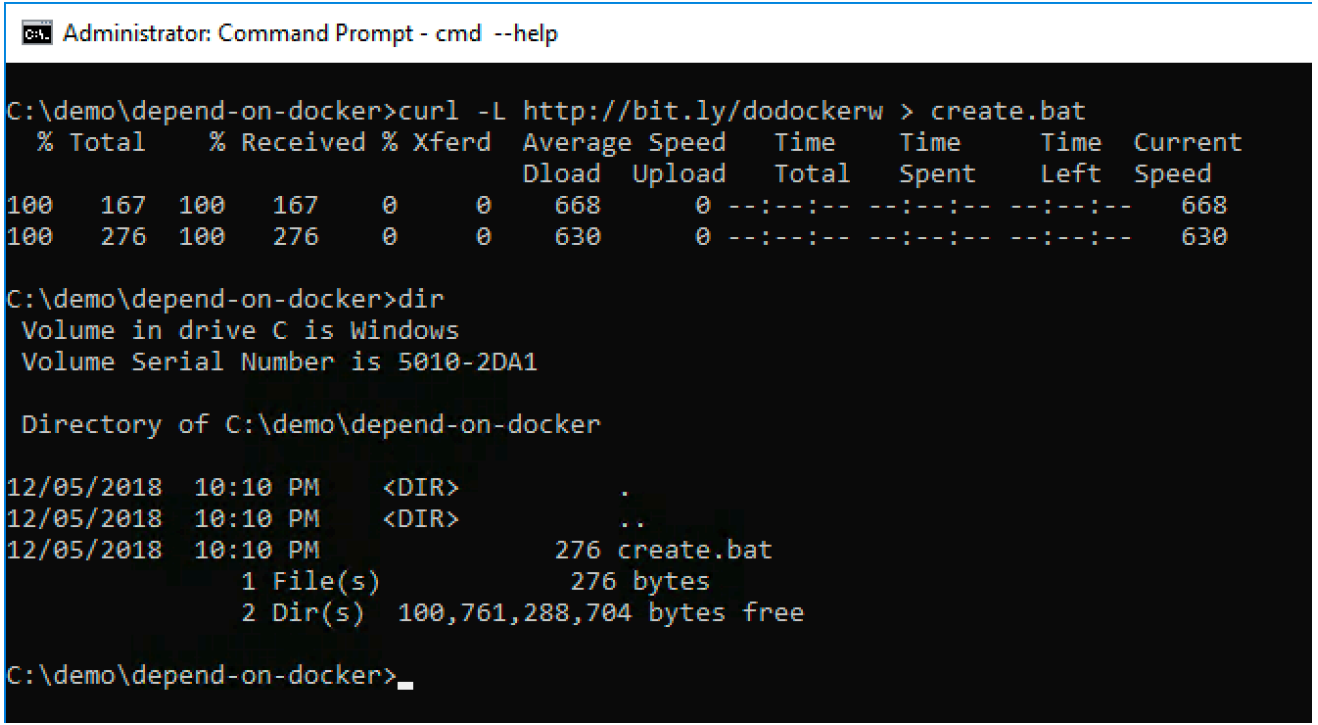
Demo 1 – Depend on Docker Project



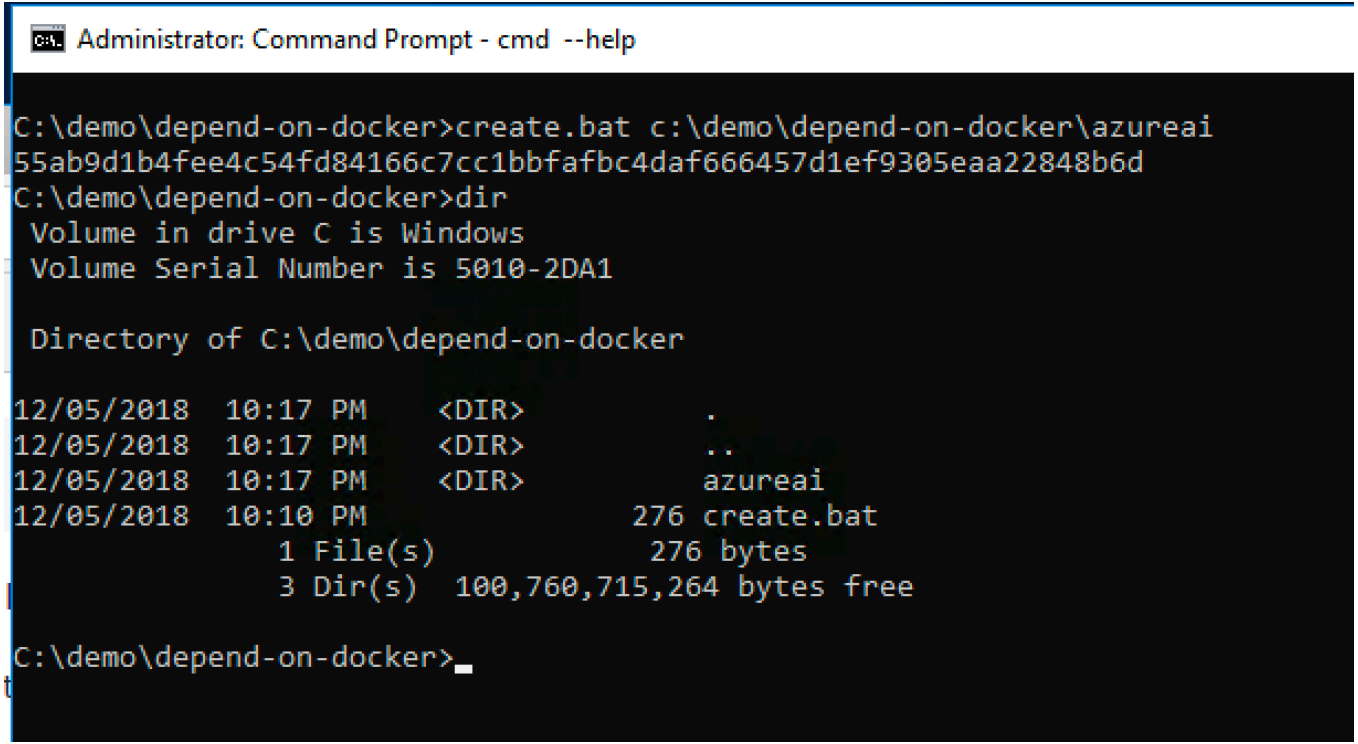
Demo 1: Step 1 – View project on GitHub (copy command)



Demo 1: Step 2 – Execute command to get create.bat

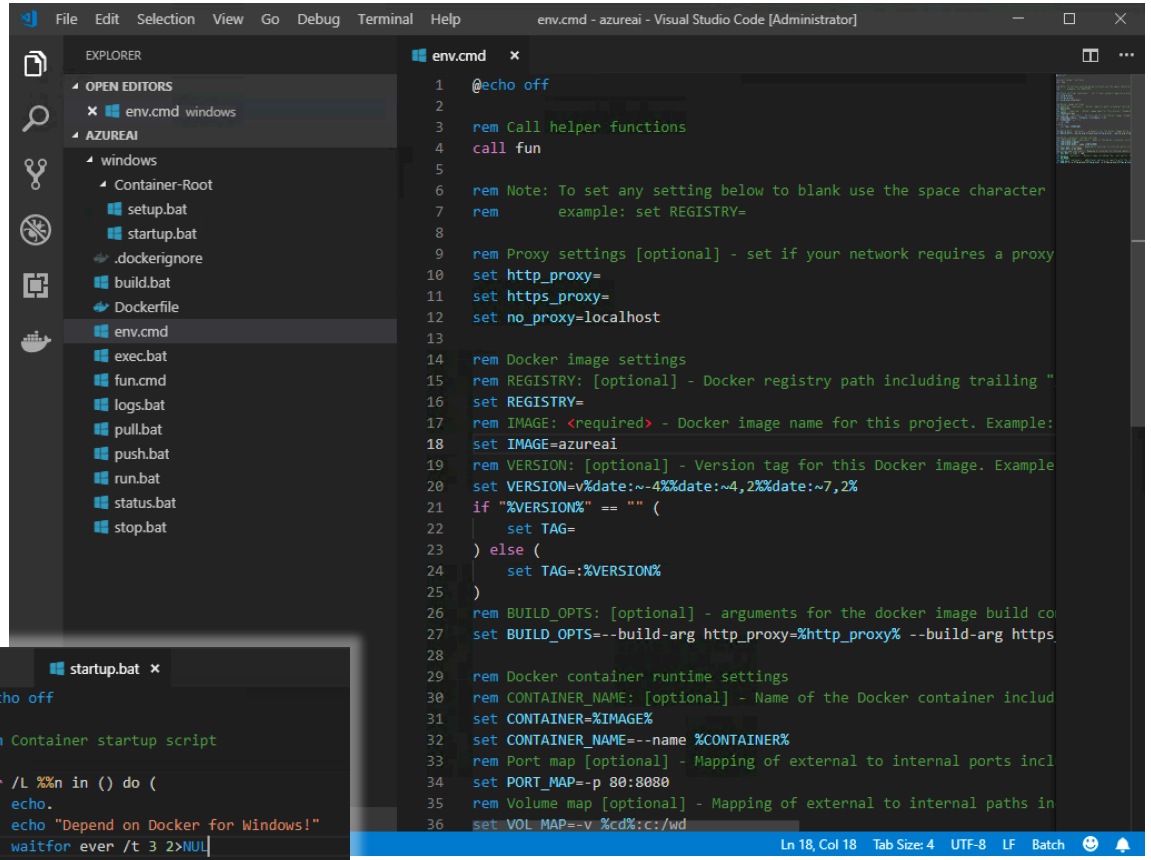


Demo 1: Step 3 – Execute create.bat [new_project_path]

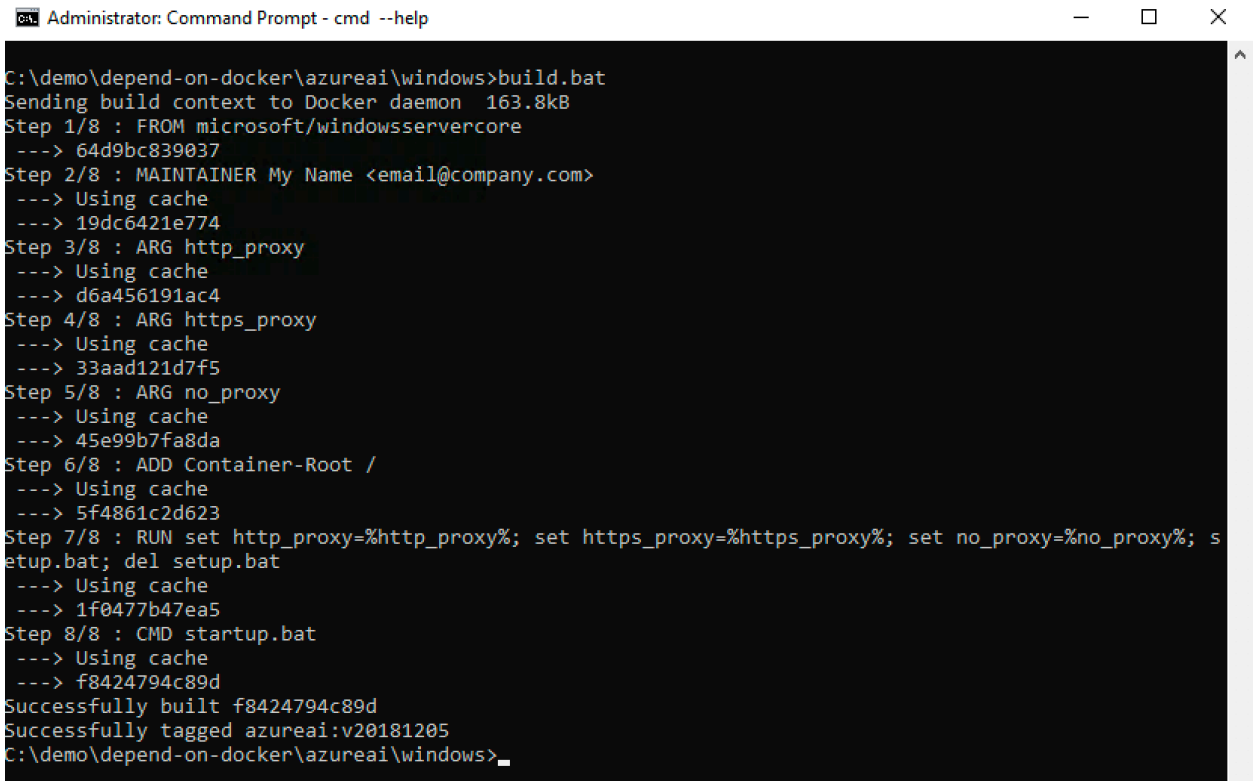


Demo 1: Step 4 – Customize your project

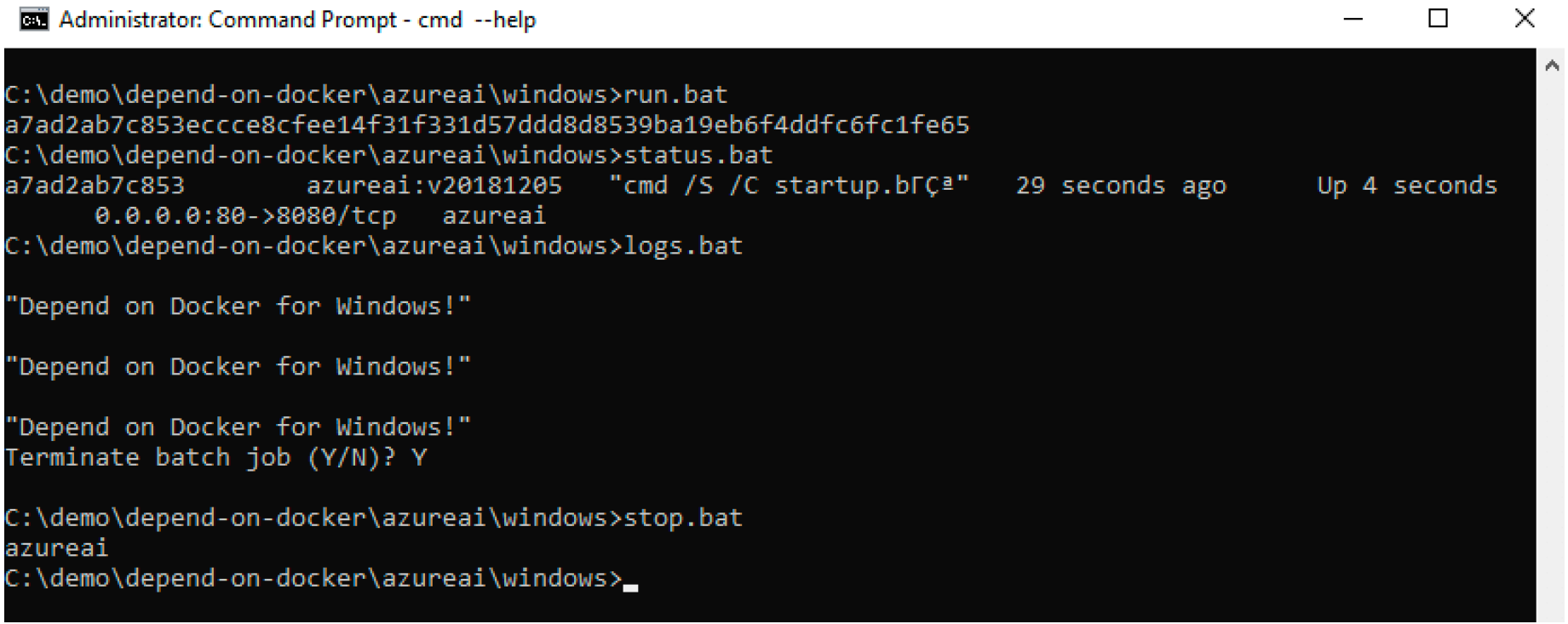
- **Dockerfile** – change base image and author info as needed
- **env.cmd** – image and container settings
- **setup.bat** – add commands to run while image is being built
- **startup.bat** – add commands to run when the container starts up, Sample →



Demo 1: Step 5 – Build your project

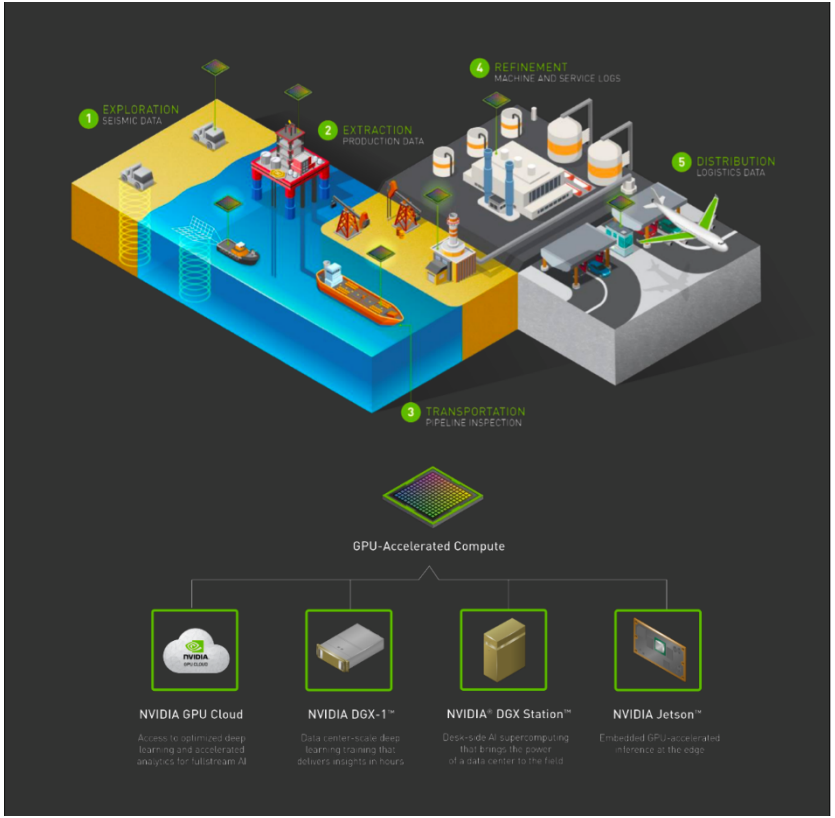


Demo 1: Step 6 – run.bat, status.bat, logs.bat, stop.bat



AI Impacts Oil & Gas End to End

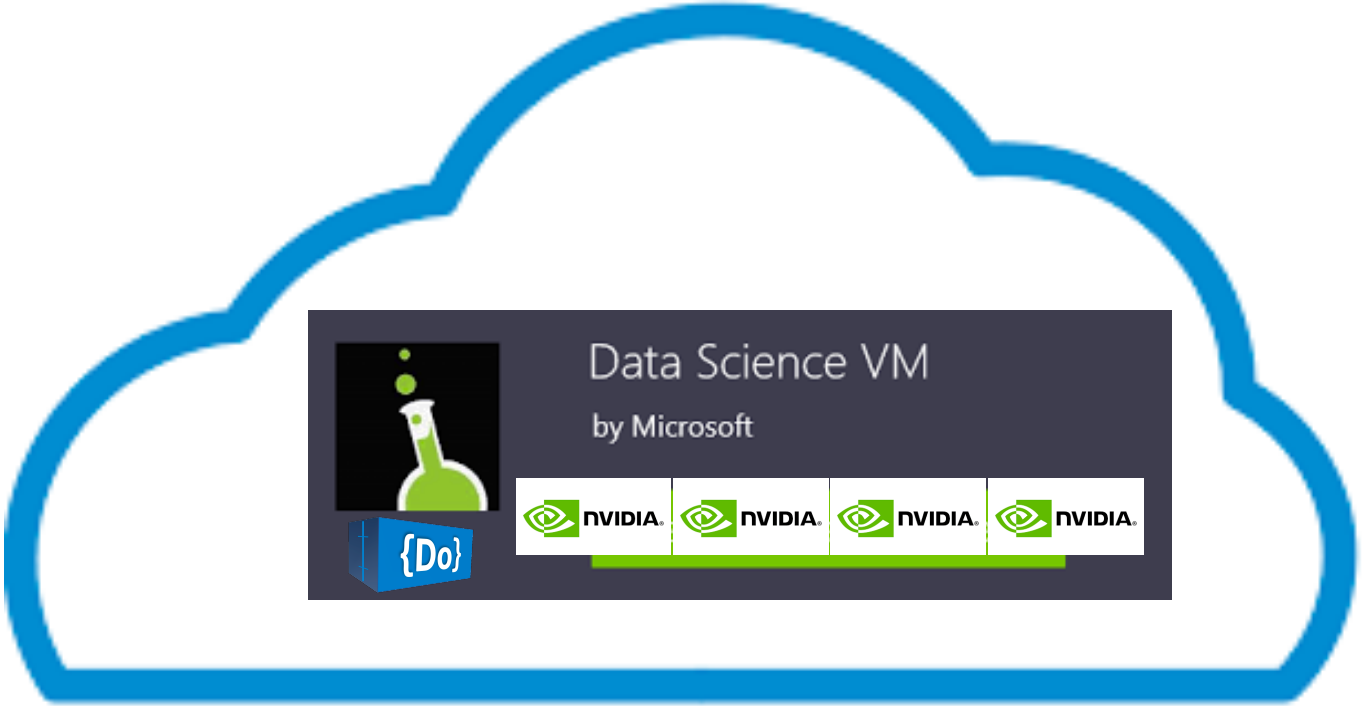
Robust Fault Prediction
Automated Inspection
Reservoir Characterization
Seismic Interpretation
Automated Drilling
Production Optimization



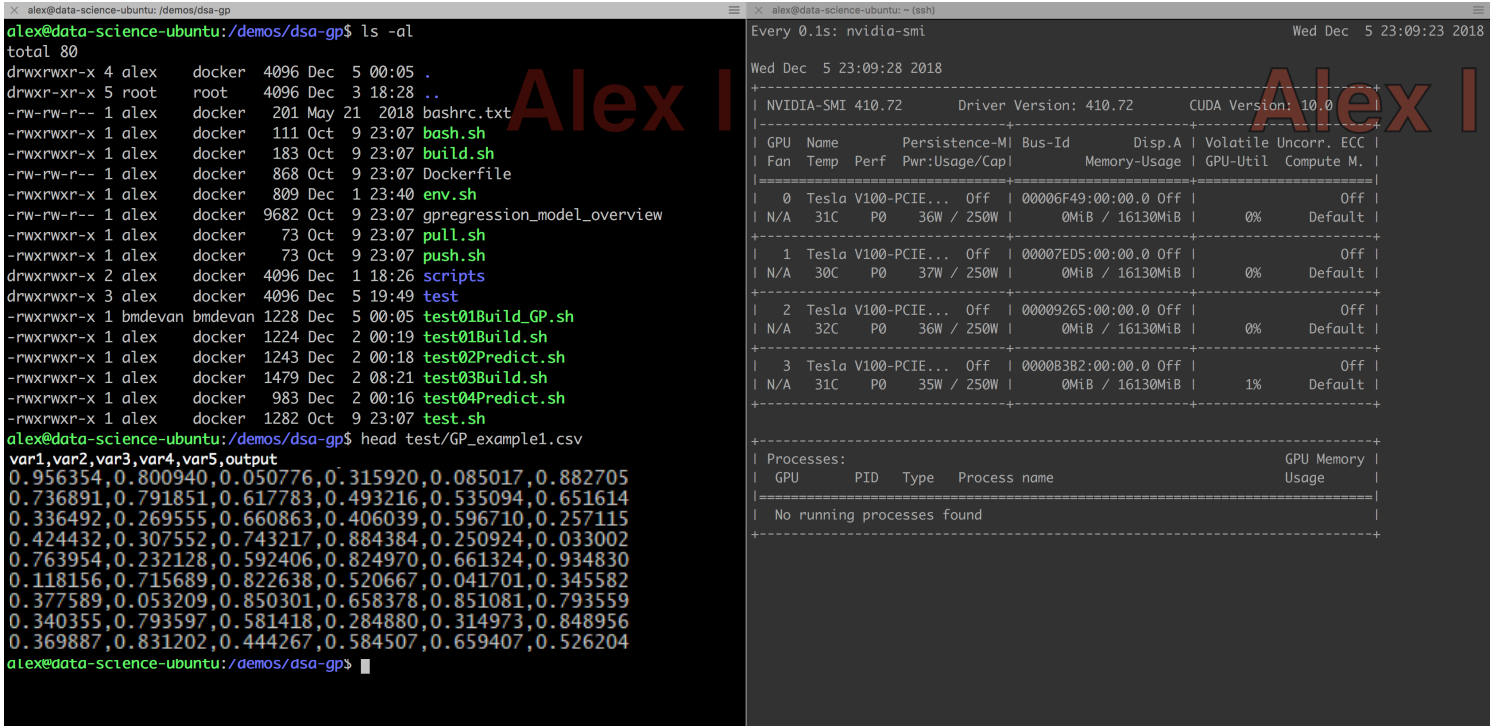
GPU acceleration essential to realize business outcomes @ Scale

Microsoft Azure
+AI Conference
CO-PRODUCED BY
Microsoft & DEVintersection

Demo 2 – Depend on Docker Time Series Model on NVIDIA Tesla V100 GPU



Demo 2: Step 1 – Show Input Datafile and GPUs



watch –n0.1 nvidia-smi

Demo 2: Step 2 – Build model and see GPU0 load

```
Saving model
Retrieving variable names
['var1', 'var2', 'var3', 'var4', 'var5', 'output', '
2std', 'mean+2std']
(10000, 10)
Data frame with input data generated
Local time: Wed Dec 05 23:19:26 2018
Gaussian Process Regression model run has started.
Optimization terminated with:
  Message: b'CONVERGENCE: REL_REDUCTION_OF_F_<=_FAC
  Objective function value: 10051.865514
  Number of iterations: 23
  Number of functions evaluations: 24
Model run executed
Model details successfully recorded!
Model trends successfully recorded!
Model metrics successfully recorded!
Model parameters successfully recorded!
Input data with predictions successfully recorded!
Gaussian Process Regression model has finished!
alex@data-science-ubuntu:/demos/dsa-gp$
```

alex@data-science-ubuntu: /demos/dsa-gp

```
alex@data-science-ubuntu:/demos/dsa-gp$ ./test01Build_GP.sh

Testing GP Regression Build ...

Running on GPU # 0

nvidia-docker run -it --name dsa-gp --rm=true -e CUDA_VISIBLE_DEVICES=0 --privileg
ed -v /demos/dsa-gp/test:/hostfiles aidocker.azurecr.io/dod/dsa-gp:CURRENT-SNAPSHO
T python3 /localadm/scripts/GPregression.py /hostfiles/testInputGP.json
(10000, 5)
(10000, 1)
Starting
Centering output
Initializing model
2018-12-05 23:19:26.175848: I tensorflow/core/platform/cpu_feature_guard.cc:141] Y
our CPU supports instructions that this TensorFlow binary was not compiled to use:
 AVX2 FMA
2018-12-05 23:19:37.301660: I tensorflow/core/common_runtime/gpu/gpu_device.cc:140
5] Found device 0 with properties:
name: Tesla V100-PCIE-16GB major: 7 minor: 0 memoryClockRate(GHz): 1.38
pciBusID: 6f49:00:00.0
totalMemory: 15.75GiB freeMemory: 15.34GiB
2018-12-05 23:19:37.301719: I tensorflow/core/common_runtime/gpu/gpu_device.cc:148
4] Adding visible gpu devices: 0
2018-12-05 23:19:46.810942: I tensorflow/core/common_runtime/gpu/gpu_device.cc:965
] Device interconnect StreamExecutor with strength 1 edge matrix:
2018-12-05 23:19:46.810997: I tensorflow/core/common_runtime/gpu/gpu_device.cc:971
]      0
2018-12-05 23:19:46.811015: I tensorflow/core/common_runtime/gpu/gpu_device.cc:984
] 0:   N
2018-12-05 23:19:46.811315: I tensorflow/core/common_runtime/gpu/gpu_device.cc:109
7] Created TensorFlow device (/job:localhost/replica:0/task:0/device:GPU:0 with 14
844 MB memory) -> physical GPU (device: 0, name: Tesla V100-PCIE-16GB, pci bus id:
 6f49:00:00.0, compute capability: 7.0)
/usr/local/GPFlow/gpflow/densities.py:89: UserWarning: Shape of x must be 2D at co
mputation.
  warnings.warn('Shape of x must be 2D at computation.')
Training
2018-12-05 23:20:00.610499: I tensorflow/core/kernels/cuda_solvers.cc:159] Creatin
g CudaSolver handles for stream 0x57e1410
```


alex@data-science-ubuntu: ~ (ssh)

```
Every 0.1s: nvidia-smi                                  Wed Dec  5 23:20:16 2018

Wed Dec  5 23:20:20 2018
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 410.72       Driver Version: 410.72       CUDA Version: 10.0     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|===============================+======================+======================|
|   0  Tesla V100-PCIE...  Off  | 00006F49:00:00.0 Off |                  Off |
| N/A   44C    P0   231W / 250W |  15448MiB / 16130MiB |    100%      Default |
+-------------------------------+----------------------+----------------------+
|   1  Tesla V100-PCIE...  Off  | 00007ED5:00:00.0 Off |                  Off |
| N/A   30C    P0    37W / 250W |      0MiB / 16130MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+
|   2  Tesla V100-PCIE...  Off  | 00009265:00:00.0 Off |                  Off |
| N/A   32C    P0    36W / 250W |      0MiB / 16130MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+
|   3  Tesla V100-PCIE...  Off  | 0000B3B2:00:00.0 Off |                  Off |
| N/A   32C    P0    35W / 250W |      0MiB / 16130MiB |      5%      Default |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                       GPU Memory |
|  GPU       PID   Type   Process name                             Usage      |
|=============================================================================|
|    0     72960      C   python3                                 15437MiB |
+-----------------------------------------------------------------------------+
```


Microsoft Azure + AI Conference

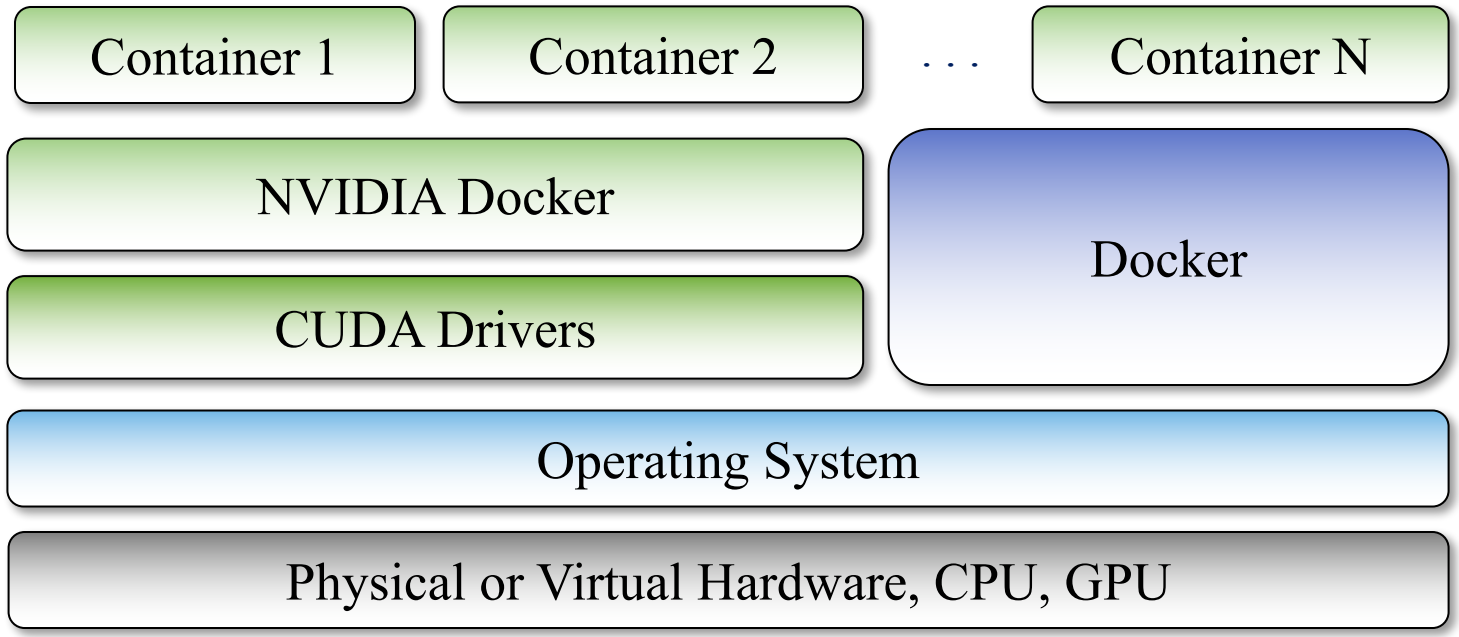
CO-PRODUCED BY

Microsoft & DEVintersection

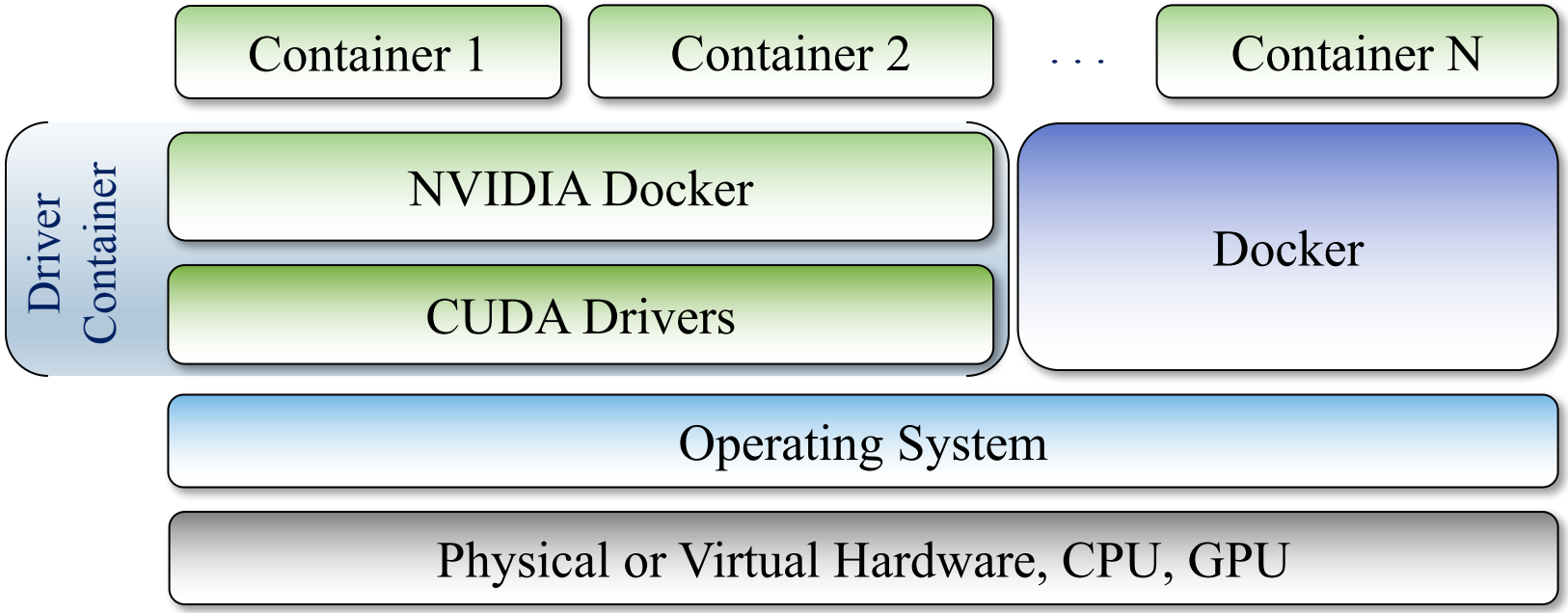
Demo 2: Step 3 – Show predictions

```
alex@data-science-ubuntu:/demos/dsa-gp$ head test/inputDataPredictions.csv
var1,var2,var3,var4,var5,output,predicted_mean,std,mean - 2*std,mean+2*std
0.956354,0.800940,0.050776,0.315920,0.085017,0.882705,0.892235,0.078100,0.736035,1.048434
0.736891,0.791851,0.617783,0.493216,0.535094,0.651614,0.656932,0.015587,0.625758,0.688107
0.336492,0.269555,0.660863,0.406039,0.596710,0.257115,0.265201,0.026873,0.211455,0.318946
0.424432,0.307552,0.743217,0.884384,0.250924,0.033002,0.041741,0.038647,-0.035552,0.119035
0.763954,0.232128,0.592406,0.824970,0.661324,0.934830,0.936326,0.055599,0.825128,1.047523
0.118156,0.715689,0.822638,0.520667,0.041701,0.345582,0.349199,0.018316,0.312566,0.385831
0.377589,0.053209,0.850301,0.658378,0.851081,0.793559,0.801760,0.092743,0.616274,0.987246
0.340355,0.793597,0.581418,0.284880,0.314973,0.848956,0.854729,0.031081,0.792566,0.916891
0.369887,0.831202,0.444267,0.584507,0.659407,0.526204,0.532105,0.084864,0.362377,0.701833
```

GPU setup – a challenge at any scale

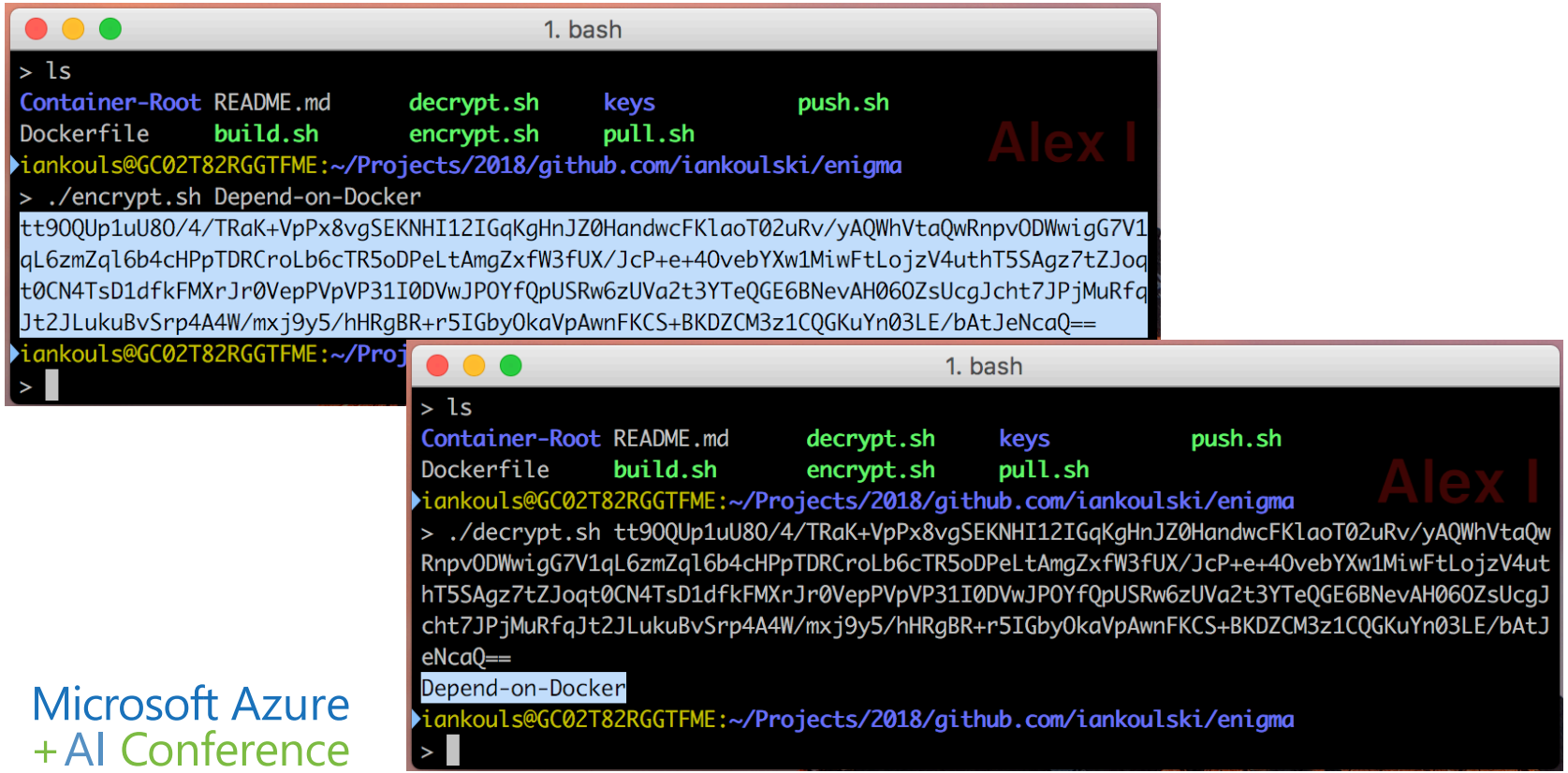


GPU setup – a challenge at any scale



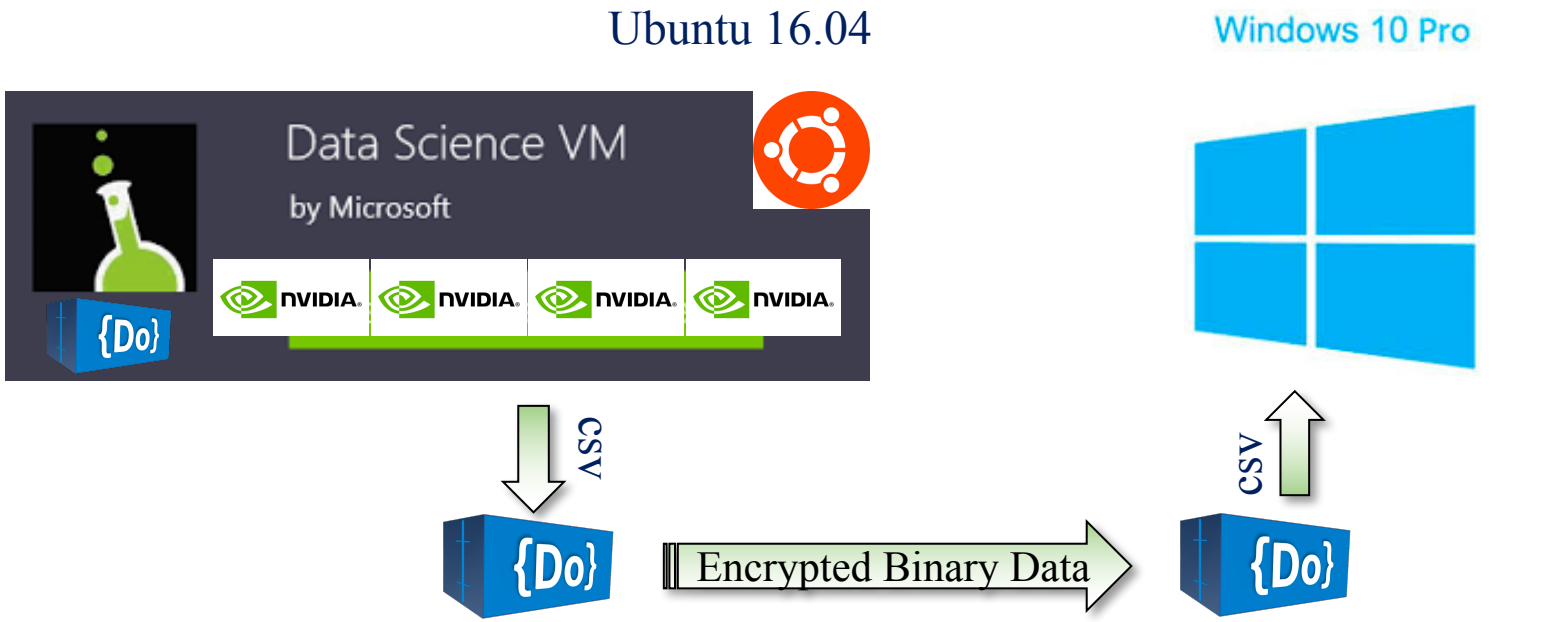
Depend-on-Docker Example
Enigma Project

https://github.com/bhgedigital/enigma



Microsoft Azure
+AI Conference
CO-PRODUCED BY
Microsoft & DEVintersection

Demo 3 – Data Security with Depend on Docker Enigma Container

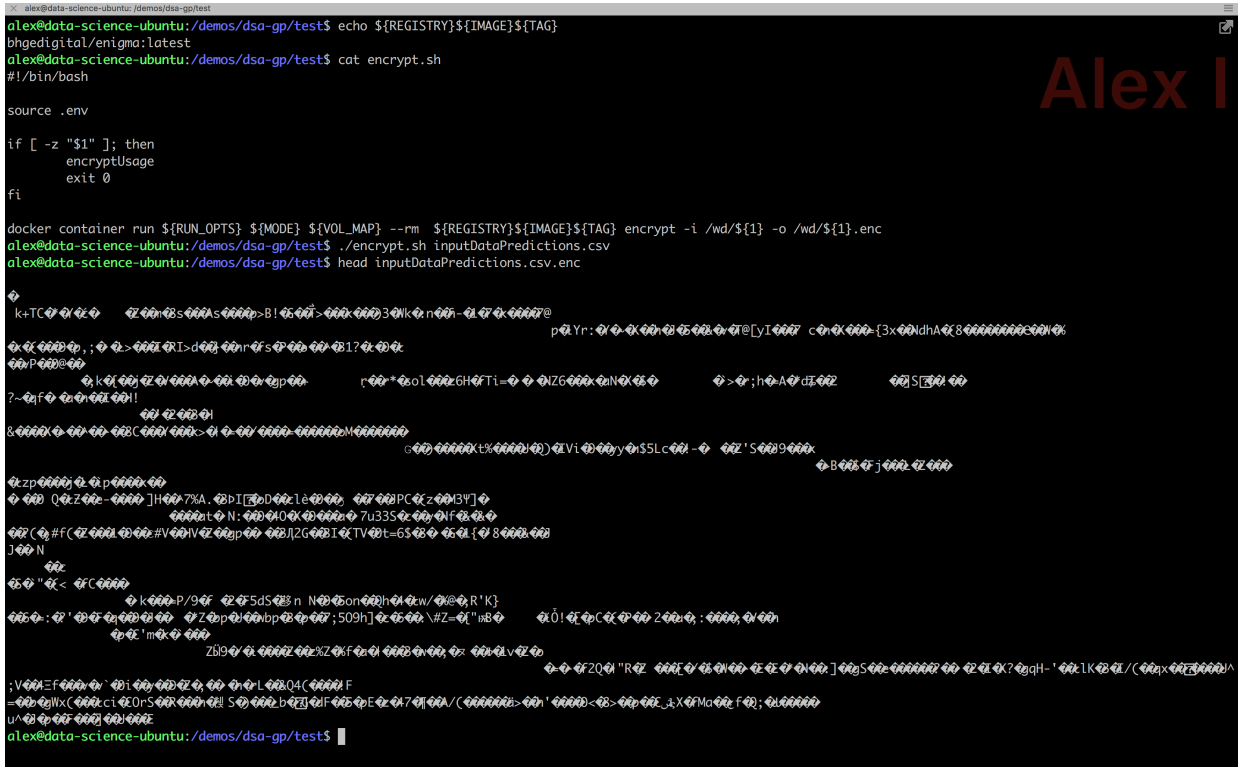


Demo 3: Step 1 – Encrypt predictions on Linux

Linux enigma image →

encrypt predictions csv file →

First few lines of encrypted content →



copy encrypted file to shared folder →

```
alex@data-science-ubuntu:/demos/dsa-gp/test$ cp -vf inputDataPredictions.csv.enc /z/encrypted/
'inputDataPredictions.csv.enc' -> '/z/encrypted/inputDataPredictions.csv.enc'
alex@data-science-ubuntu:/demos/dsa-gp/test$
```

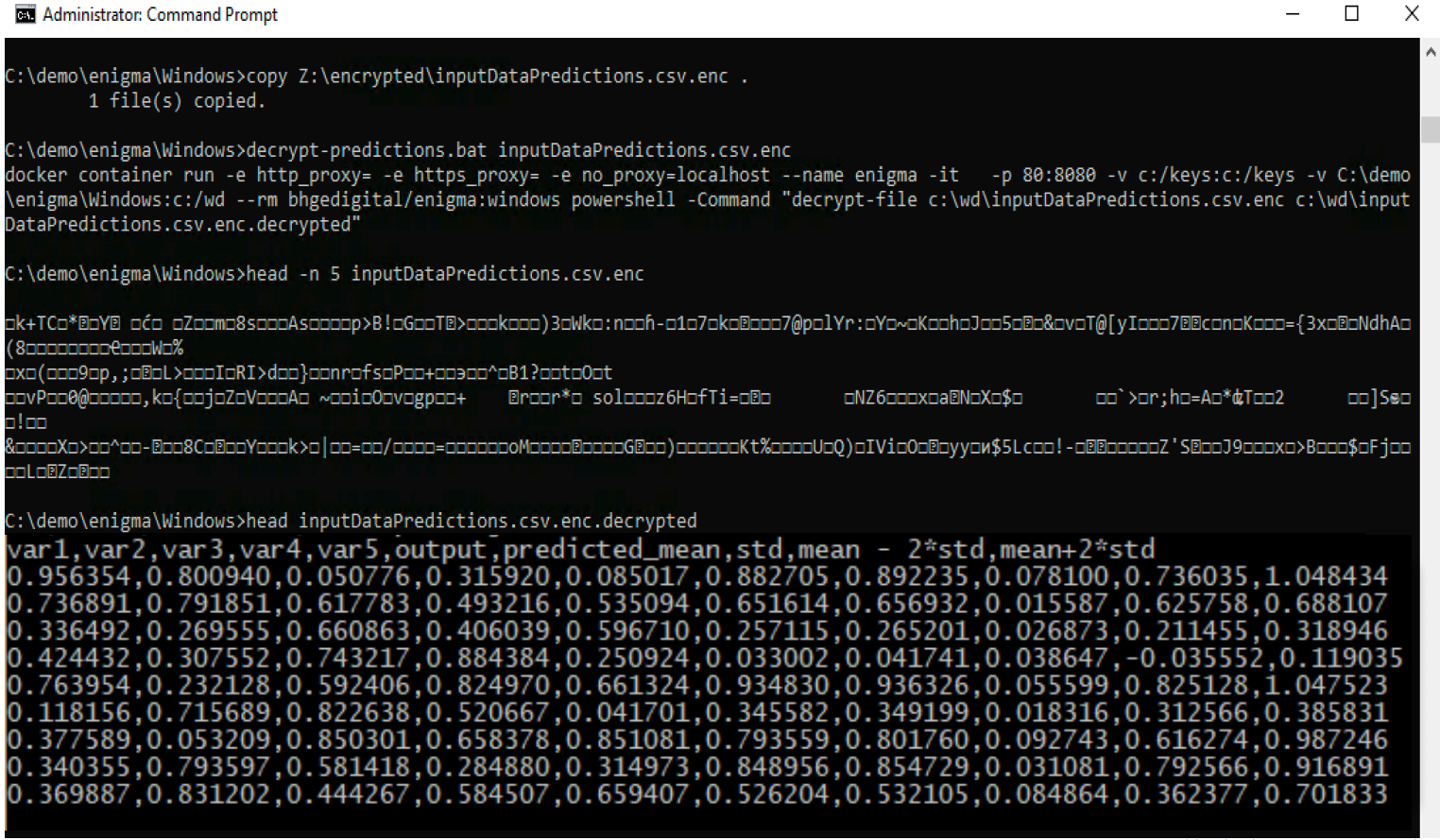
Demo 3: Step 2 – Decrypt predictions on Windows

copy encrypted file from shared folder →

decrypt using enigma windows container →

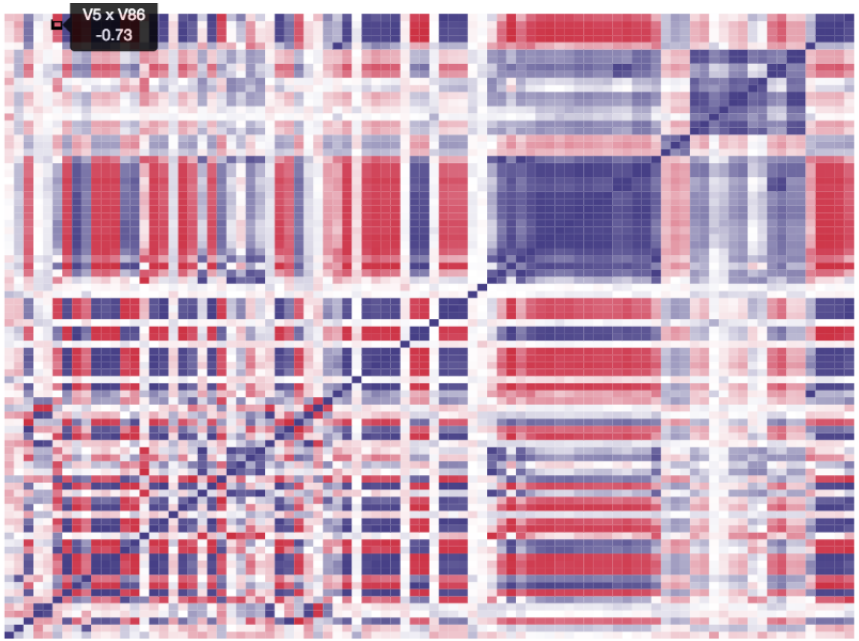
encrypted content →

decrypted content →

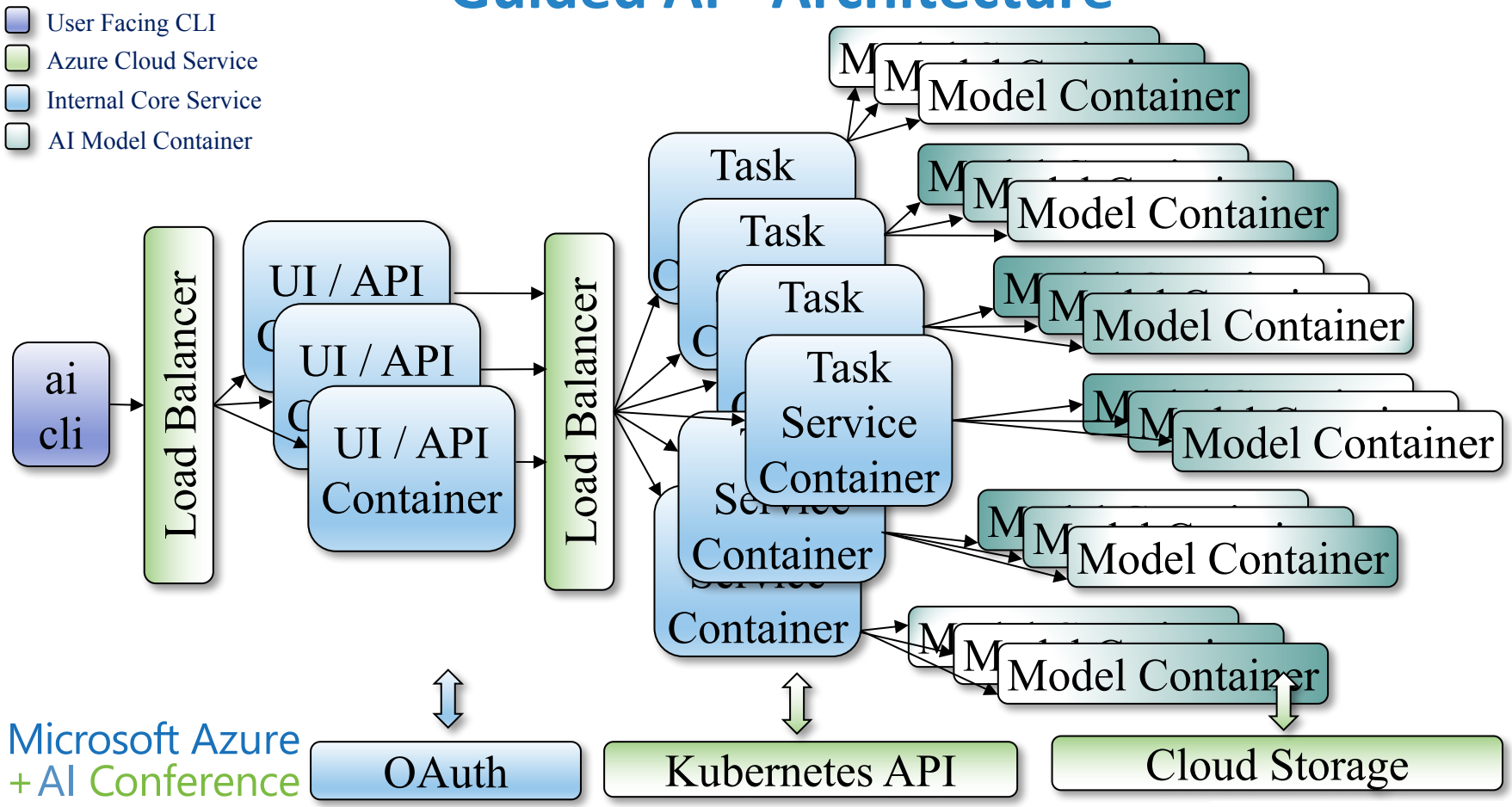


Microsoft Azure + AI Conference
CO-PRODUCED BY
Microsoft & DEVintersection

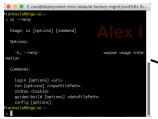
Guided AI: Automatically build models



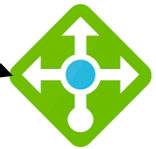
Guided AI - Architecture



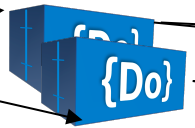
Demo 4 – Guided AI
Scale up on Azure



AI
Command Line
Interface



Load
Balancer



AI Core Services

Model Containers

Kubernetes
Dashboard

Azure
Kubernetes
Service

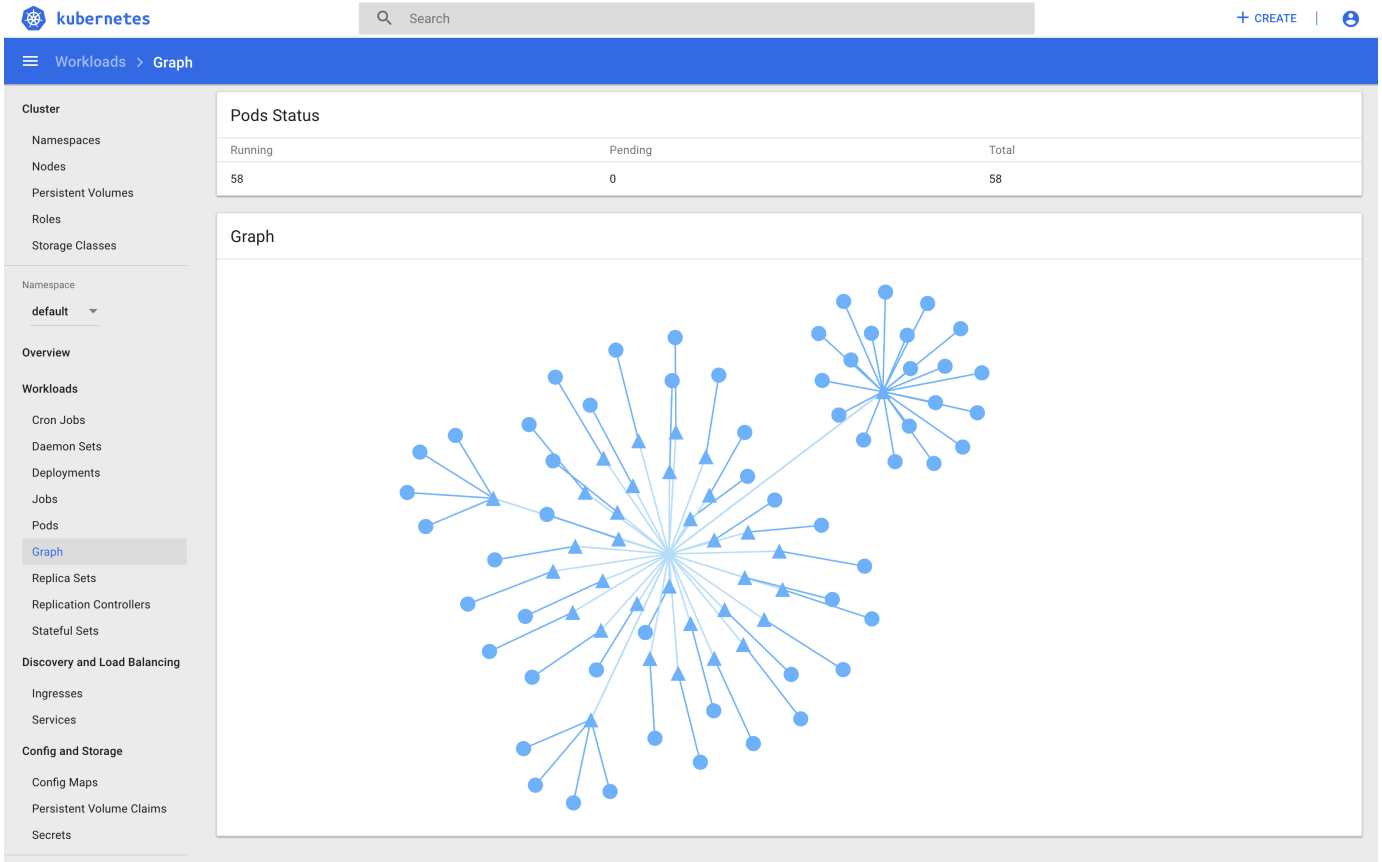
Azure Database
for PostgreSQL

Azure
Container
Registry

Storage
Account

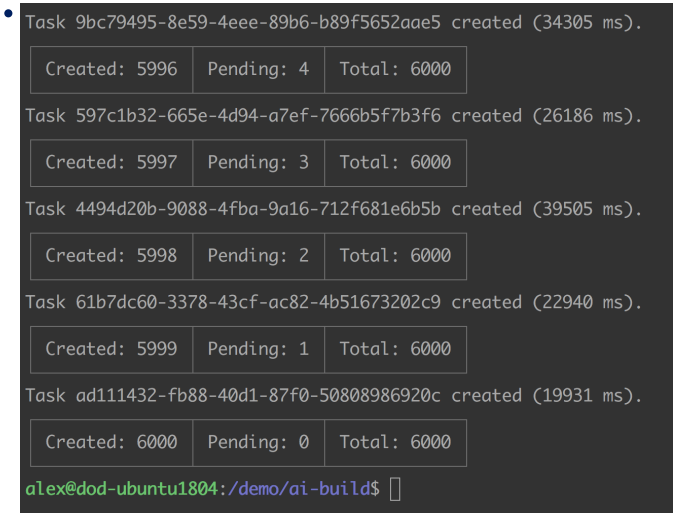
Demo 4: Step 1 – Logon to Kubernetes Dashboard

This customized version of the Kubernetes Dashboard has a Graph tab which allows us to visualize the workloads. The Task service is in the upper right part of the graph and is scaled to 20 pods.



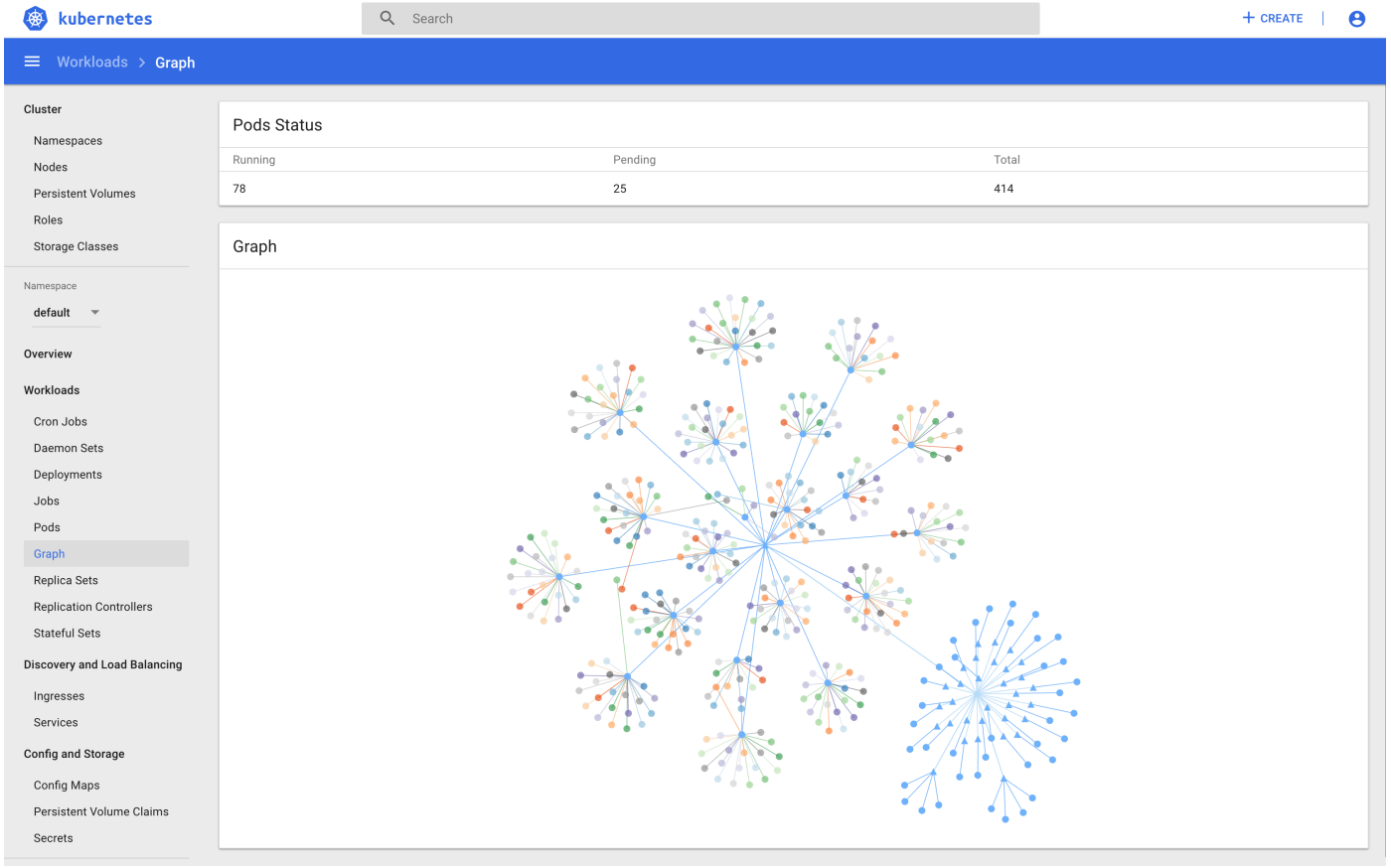
Demo 4: Step 2 – Submit dataset using AI CLI

```
alex@dod-ubuntu1804:/demo/ai-build$ ai guided-build -i 10 -e ./guided_build_1000.csv
{"uploaded":[{"filename":"guided_build_1000.csv","id":"8f700b0e201542819c79c12c053bd9a2"}]}

| Created: 0 | Pending: 6000 | Total: 6000 |

Task f69d2c0e-a469-4020-bc18-ff72e6498cc0 created (20009 ms).

| Created: 1 | Pending: 5999 | Total: 6000 |

Task 06d7bf3a-1376-4844-ad11-ff95e999910f created (20043 ms).

| Created: 2 | Pending: 5998 | Total: 6000 |

Task ff560320-6888-4c52-805c-787be0e15824 created (20156 ms).

| Created: 3 | Pending: 5997 | Total: 6000 |
```



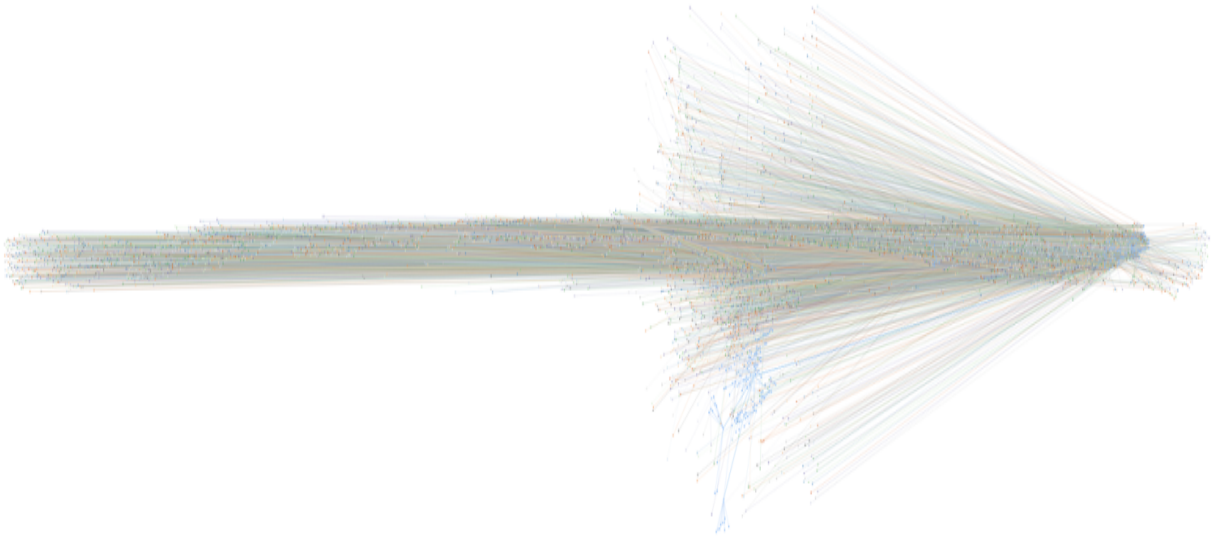
Demo 4: Step 3 – Observe new model containers

Each colorful dot is a model pod. Model pods are connected to their parent task service pod, which makes this workload visualization look like a fireworks explosion.



30,000 Models built concurrently

50 Nodes
4800 vCPUs
30 TB RAM



Places where you will find containers

- **CI / CD Systems** – Use Docker in Docker to build and ship container images
- **Software Test execution** – Create environment, execute test, remove container
- **Development** – IDE in a container, Cloud IDE, super easy developer onboarding
- **CLI** – e.g. Azure cli in Docker, SQL client in Docker
- **Diagnostics, Troubleshooting, Support** – Run Docker to collect diagnostic info
- **Patching** - Run Docker command to fix and patch installed software
- **Backups** – Run your backup job in a container
- **Firmware updates** via containers!
- **Initialization scripts** – run privileged to install software or do admin work on host

Depend on Docker Summary

- It is easy to build Windows and Linux containers
- Just run, instead of “install first, and then run”
- You choose whether to run your workload on CPU or GPU
- Docker enables us to scale workloads up and down at will

Docker makes IT possible

References

- GE Aircraft Engines https://www.ge.com/reports/every-two-seconds-aircraft-powered-ge-technology-takes-off-somewhere-world/
- GE Power Generation https://www.ge.com/reports/ultra-super-critical-badass-machines-help-make-30-percent-worlds-power/
- Container Hacks and Fun Images, Jessie Frazelle https://www.youtube.com/watch?v=cYsVvV1aVss
- GE SemTK Project - Paul Cuddihy, Jenny Weisenberg, Justin McHugh, et.al. https://github.com/ge-semtk/semtk
- Enigma DoD Project – Yekta Yazdani https://github.com/bhgedigital/enigma
- Google Images https://images.google.com/
- Depend-on-Docker Project https://github.com/bhgedigital/depend-on-docker
- Depend on Docker DockerCon'18: https://embed.vidyard.com/share/5hUnYnLyxkLJGSuK89woNn
- NVIDIA Driver Containers: https://github.com/NVIDIA/nvidia-docker/wiki/Driver-containers-(EXPERIMENTAL)
- Microsoft Azure: https://azure.microsoft.com/en-us/

Questions?

Please use EventsXD to fill out a session evaluation.

Thank you!

Microsoft Azure
+ AI Conference
CO-PRODUCED BY
Microsoft & DEVintersection